CHAPTER 11

VALID COMPONENT LIBRARIES

Creating and Maintaining Valid Libraries

11.1 INTRODUCTION

This document is a description of how to create and maintain libraries on the Valid SCALDsystem. It covers modifying existing libraries, creating new libraries, and adding parts to existing libraries.

QUALIFICATIONS OF THE LIBRARIAN

The person who maintains the libraries (the librarian) should know the SCALDsystem moderately well, be reasonably conversant with UNIX, and know how to use at least one text editor under UNIX. Maintaining libraries requires a certain amount of caution, since an error in a library can affect many users.

A QUICK OVERVIEW

Adding a part to an existing library is perhaps the simplest operation, since it does not involve determining conventions or syntax. What is required is one or more body drawings, a timing model, a Simulator model, and physical data (generally represented in a text file).

Modifying an existing library (perhaps to conform to local conventions) is more difficult, at least the first time. Technically, it is no harder, but it is important to get all conventions, syntax, and models correct at this stage. Once the library is in common use, changes become extremely difficult.

Most challenging is the creation of a library from scratch. This is difficult primarily because the conventions must be determined before any use is made of the library. This means there is no feedback from users at all, and no example of how someone else has treated the same problem.

11.2 CONVENTIONS

Conventions govern, to a large extent, how the bodies will look and how the models are made. Many of them will be determined by corporate policy; others are determined by the design style preferred by the users.

BIT ORDERING

There are two possible ways to number the bits in a multi-bit signal (a bus). They may be numbered from right to left, so the LSB (least significant bit) of a signal is bit 0 and the MSB (most significant bit) of the signal is bit N-1 (for an N bit signal). Alternatively, the bits can be numbered so that the MSB is bit 0, and the LSB is bit N-1.

Bit ordering is usually a company wide decision. If a company standard exists, you will almost surely want to use it. If no policy already exists, you may be influenced by the hardware you wish to connect to. DEC, for example uses right to left ordering, while IBM uses left to right ordering. Data books from IC manufacturers (such as TI) are almost invariably in right to left format. This format also makes more sense from a mathematical perspective, since bit X of a work has a value of 2X, independent of the length of a word.

VECTORED VS. NON-VECTORED PARTS

There are at least two design styles that may be used with the SCALDsystem. These are design using non-vectored parts and design using vectored parts.

Non-vectored parts are sections of packages. Each pin on each part in the schematic corresponds to a pin on a non-vectored part in the design. Any design can be drawn out fully using nothing but non-vectored parts.

The other design style uses vectored parts, each of which represents one or more sections of a chip. A 32 bit latch, for example, represents only one vectored part but 32 physical sections (most likely 4 physical parts). On the schematic, the data pin of such a latch represents 32 different physical pins in the design. The SCALDsystem will assign the physical sections corresponding to a vectored part automatically.

Both design styles have advantages. The vectored parts offer much faster compilation, verification, and simulation. Drawings are smaller and simpler, and usually easier to understand.

Non-vectored parts, however, offer a closer correspondence to PC board or wire wrap implementations. Pin numbers and locations can be added to the schematic, which is difficult with vectored parts (because each pin on the vectored part represents many physical pins, and a vectored part may be spread over several packages).

What implications does this have for libraries? Each part must have a body corresponding to how it will be used. A vectored part has fewer pins, with some of the pins representing vectors of signals (often of arbitrary width). The body of a non-vectored part, on the other hand, will have roughly as many pins as the physical part (except perhaps power and ground). Both versions can coexist. Of course, if only one or the other will be used then there is no need to create both.

SIZE WIDE PARTS

'Size wide' is a term only applicable to vectored parts. Many parts may be made to handle an arbitrary number of bits simply by providing one section per bit and paralleling the control signals. This is called a size wide part, since the width of the data path is determined by the 'size' parameter placed on the part. The Valid libraries, for example, contain size wide latchs and multiplexers.

However, not all parts can be made size wide. In an ALU, for example, carry in and carry out are connected neither in parallel nor on a per bit basis. Therefore the ALU is not a sizable part. Gates, on the other hand, require no control signals and can always be made size wide.

When a new part is added to a library, the librarian must decide if it can reasonably be made into a size wide part. If it is possible to make it size wide, the librarian must then decide which control signals should be driven in parallel and which should be provided on a per bit basis.

Suppose you wish to enter a D flip-flop that has a preset and clear for each section. Should these inputs be size wide (meaning that each bit can be cleared and set individually) or should these be single bit signals that set and clear all the flip-flops together? The answer depends on how you expect the part to be used. The decision does not rule out any particular design, but it does make some designs easier to enter than others.

If you make the preset and clear size wide, then a user who wishes to clear the entire register must sign extend the clear signal to the correct size to prevent a width mismatch. Since most size wide flip-flops are used as registers, this is a good argument for making PRESET and CLEAR single bit signals.

On the other hand, if the signals are made single bit, and the user really needs a register where each bit can asynchronously cleared and preset independently, the user must draw the register with one body per bit. This is

equivalent to the non-vectored design style for this particular register.

MULTIPLE VERSIONS OF PARTS

It is possible to have more than one version of a part. The only constraint is that the two versions have the same pin names. If one part is vectored and the other has the pins drawn out, then the drawn-out version must be equal to the vectored version with some specific value of 'size'.

The exception to this rule occurs in parts with asymmetrical sections. In this case the versions of the part representing the different sections must have no identical pin names. This is to allow the different sections to be distinguished. Additionally, there must be a property attached to each section identifying the section. The Valid convention is to name this property 'section' and to give the property a value identifying the section number of the part that the body corresponds to.

APPEARANCE OF BODIES

Appearance of bodies is a matter of standards and taste. For an example of the standards we use, see Valid Library Style and Standards.

11.3 SIGNAL SYNTAX

WHAT IS SIGNAL SYNTAX?

"Signal syntax" is description of how signals are represented. It consists of two major pieces of information. The first is the order of the name, subscripts, and assertions. The second is which characters are used to show high and low asserted signals, and to separate bit numbers in vectors.

Different installations may use different signal syntaxes. The signal syntax in use by any given site is determined by the file '/u0/lib/ged/config.dat'. If the analysis programs are to run on the host, then an identical file must reside on the host.

For more information on signal syntaxes and a description of allowable syntaxes, see SCALD Signal Name Syntax.

DECIDING ON A SIGNAL SYNTAX

Signal syntax is often set by company standards. If your company has no standard, we suggest you use the syntax Valid supplies, since that means you will not have to run the program to convert the libraries (which are supplied in Valid format) into your own syntax.

CONVERTING LIBRARIES TO YOUR OWN FORMAT

If your libraries are not kept in the Valid format, then you will have to convert the library as distributed by Valid into the format you want. This is accomplished by the program 'translate'. You run the translate program as follows:

```
%translate <SCALD directory name> <configuration file name>
```

The <SCALD directory name> is the name of the '.lib' file for the library. For the lsttl library, for example, it is '/u0/lib/lsttl/lsttl.lib' . The configuration file is normally '/u0/lib/ged/config.dat' . The conversion is done in place, and cannot be repeated since the program always assumes Valid format as input. If you control-c out of the program, or the system crashes, your library will be left in an unusable state. This means you should always have a copy of what the library looked like before you started. For Valid library distributions, this is no problem since you can always read them off the tape again. If you try to modify your own files you should first make a copy of the library either on tape or disk.

11.4 ESTIMATING DELAYS

The Valid Timing Verifier needs a model of each part. The model describes the timing characteristics of that part, including both the minimum and maximum delays possible for that part. For a few families, such as ECL, these delays are published in the data books, and are equal for rising and falling signals. These families are easy to model accurately.

For other families such as TTL, the data books are woefully incomplete. They do not list minimum delays at all. A 74LS74, for example, is specified to have a 5 ns hold time, but has no minimum delay on its output. This means you cannot hook the output of an LS74 to its input and formally show that it will work. The Timing Verifier would therefore report this as an error. Therefore in TTL, the minimum delays must be estimated to give 'reasonable' results using the normal parts.

Other values, such as minimum pulse widths on clocks, are sometimes not given. This can be estimated from maximum toggle frequency, if available,

11.5 ADDING PHYSICAL INFORMATION

Physical information is the information about a part that is not necessary to use the part, but is necessary to build an implementation using the part. For example, pin numbers are not needed in order to analyze the behavior of a circuit, but they must be assigned before a copy of the circuit can be built.

How is physical information kept in the system? The master copy of physical information is kept in the '.prt' file for that library. For example, if the LSTTL library is known as '/u0/lib/lsttl/lsttl.lib', then the physical information about LSTTL is kept in '/u0/lib/lsttl/lsttl.prt' . This is the file read by the Packager when generating net lists, checking loading, and so forth.

How does the information get in this file? It can get there one of two ways. It can be entered on the body drawings, or it can be entered in text form. If both are specified, then the text form overrides.

This file is created by compiling a library drawing (a drawing which contains one example of each part) for CHIPS. If new parts are added to a library, they must be added to the library drawing and the library drawing recompiled for the part to be known to the physical design programs. The name of the drawing does not matter, but by convention it is called 'X LIBRARY', where X is the logic family. The drawing for LSTTL is called 'LSTTL LIBRARY', for example. This drawing often comprises several pages. The order of the parts on pages does not matter.

SPECIFYING PHYSICAL INFORMATION ON BODIES

All the physical information is represented by properties. Information about the whole part, such as power and ground pins, and the logic family, is specified by body properties. Information associated with one of the pins, such as input and output loading, is specified as pin properties.

To enter this information on the body, attach the desired properties to the pins of the body and the origin of the body. The information will be included in the '.prt' file when you compile the library drawing for CHIPS.

Sometimes, however, it is more convenient to work with a text file representation for physical information. In this case the physical information is kept in a file with the name 'phys_dat' in the same subdirectory as the body and other drawings for the part.

This information is gathered and added to the '.prt' file with the command 'addphysinfo'. This command gathers the physical information together, and runs a program to add this to the '.prt' file.

FORMAT OF THE PHYSICAL DATA

The physical data in 'phys_dat'is represented as follows:

```
PART <name>
  <Body properties>
PIN
  <pin information>
END
```

PART, PIN and END must begin in the first character of each line. Body properties and pin properties must not begin in the first character of the line.

Body properties consist of the name followed by the value. Pin information is kept in the form:

<pin name> <pin numbers> <io> <loading>

The pin name must be in Valid standard format. This means the order must be a name, optionally followed by a bit subscript (only one bit), optionally followed by a '' indicating a low asserted signal.

The pin number must be in the form expected by the Packager (See the Packager Reference Manual.)

The <io> field contains one of the following: INPUT, OUTPUT, TS, OC, OE, or ANALOG. TS stands for Tri-State, OE for open-emitter, and OC for open collector. ANALOG is used for non-digital pins such as the RC pins on one shots.

ANALOG pins do not need any loading specification. All others need one loading specification except for tri-state pins which require both an input and an output loading specification (in that order). A loading specification (described in more detail in the Packager Reference Manual), consists of :

(<low state loading or drive>,<high state loading or drive>)

Here is a hypothetical example showing all of the above types of pins:

```
PART 74LS00
 FAMILY LSTTL
 POWER_PINS (VCC:14;GND:7)
PINS
 A<0>      (1,2,3)          INPUT    (-0.4,0.02)
 B<0>      (4,5,6)          INPUT    (-0.4,0.02)
 Y<0>*     (7,8,9)          OUTPUT   (8.0,-0.4)
 X<0>*     (19,11,12)       OC       (8.0,*)
 Z<0>      (13,14,15)       TS       (-0.02,0.02)     (24.0,-2.6)
 R/C<0>    (16,17,18)       ANALOG
END
```

11.6 ADDING PARTS TO EXISTING LIBRARIES

Adding a part to an existing library involves several steps. These are drawing the body, modelling the part, adding the physical data, and updating the library documentation. These operations are covered in this section.

MAKE THE PART

The first step is to use the Graphics Editor to make the drawings that define the part. You will need to make several drawings; at least one body drawing to define what the part looks like when used; a part drawing to tell the Compiler that this is indeed a physical part; and timing and simulation models if these are used.

MAKE THE PHYSICAL DATA FILE

The physical information may be entered on the part itself or in a separate text file. Putting the information on the part is conceptually simpler, but means that the information is read and written every time the part is used in any drawing. This increases the size of the data files and decreases the speed of the Graphics Editor. Since the physical information is not used by the Timing Verifier and Simulator, this is usually an unnecessary waste of resources.

Putting the physical information in a text file avoids this problem since the physical information is not carried around with the part, but is kept in a file until needed by the Packager. The drawback, of course, is that you need another file. However, this is the recommended way to include physical data. See the section on physical data for the format of this file.

ADD THE PART TO THE LIBRARY DRAWING

The next step is to add the part to the library drawing (X LIBRARY). Even if several different versions have been defined, only one should be added. If you have a vectored version, that should be the one you add. Parts which are size wide should be given a size of 1 bit. Parts having asymmetrical sections should have one of each of the sections added to the drawing. The asymmetrical sections can have no pin names in common.

At this stage, you probably want to add the part to the example drawing for the library. This drawing, which for library X is called 'EXAMPLE OF EACH X PART', is primarily for documentation purposes. It shows an example of each body with all of the versions of that body. It is also useful in testing the models for the library since when used in a compilation, it invokes all versions of all of the parts.

COMPILE THE LIBRARY DRAWING FOR CHIPS

The next step is to compile the library drawing for CHIPS. To do this, set the root drawing to 'X LIBRARY' (where X is the name of your library) and add the directive 'output CHIPS;'. The Compiler will produce a file called 'chips.dat'. This file should be moved to 'X.prt'. For LSTTL 'chips.dat' is moved to 'lsttl.prt' (in the same directory as 'lsttl.lib')

ADDING PHYSICAL INFO FROM TEXT FILES

If you put the physical data on the body drawings, you are now done. If you put the physical information in text files, however, you must change your directory to the directory containing 'x.lib' and type

```
%addphysinfo x.prt
```

This will run a program to add your physical data to the '.prt' file. The errors, such as misspelled pin names, will be shown in the file 'liblog.dat' and 'liborig.dat'.

COPYING THE LIBRARY INFORMATION TO THE HOST

If you run on a host machine, you will probably want to copy the new library information to the host. See the section on 'MAINTAINING LIBRARIES ON A FOREIGN HOST' for more information on this subject.

11.7 CREATING A NEW LIBRARY FROM SCRATCH

Creating a library from scratch is much like adding a part to a library, except the library drawing must be made up from scratch.

The hard part, however, is deciding on the standards and conventions to be used. This includes things like the physical appearance of the bodies and the delays in the cases where they are not specified.

11.8 MAINTAINING LIBRARIES ON A FOREIGN HOST

If you work on a foreign host such as a VAX or 370, you will probably want to keep a copy of the libraries on this machine. Otherwise each user must keep a private version of the libraries which is very wasteful of disk space. These libraries should be kept in a read only directory.

The libraries can be copied over with the Filecopy program. We recommend that you run Filecopy without a 'transfer.log' file when copying libraries, since this will ensure that all libraries are on the host even if some have somehow been deleted since the last filecopy. This, however, means that the transfer will take several minutes. See File Copy, Chapter 9. for more details.

The 'filecopy.cmd' file is: (supposing the library is X, the host is a VAX, and the destination directory is [SCALD.LIBRARIES])

```
report_files on;
copy_file 'x.prt';
directory 'x.lib'
host_kind VMS;
host_destination '/dev/vms/scald/libraries';
end.
```

This will copy all the necessary information to the VAX.

11.9 UNIX CONSIDERATIONS

There are two issues that must be considered for libraries; disk space and file protections.

DISK SPACE

The libraries are stored in /u0, along with user files. There must be enough room for the libraries plus enough left over for the users. For Valid supplied libraries, the space required by the library will be indicated in the documentation that comes with the library. For user created

libraries, the amount of space required can be determined by changing your directory to the library in question and typing 'du'.

The amount of free space on the disk can be determined with the 'df' command. This will show the number of free blocks on /u0. You should leave at least 1000 free blocks after the libraries are installed for users to work with. If installing a given library would result in fewer than 1000 free blocks, you should either remove some files from /u0, not install the library, or acquire more disk space.

PROTECTION

The libraries, and their UNIX directories, should be write protected for everyone except the librarian. The files in the library should be owned by 'lib'. This can be checked by typing 'ls -l'. If there are any files not owned by lib, you can fix this by logging on as root, then changing your directory to the library in question, and typing 'find . -exec chown lib {} \;'. This means 'change the owner to lib for all files in this directory'.

'ls -l' will also show the protection of any file in the library. These files should be '-rw-r--r--' for all files and 'drwxr-xr-x' for all UNIX directories. This protection allows the user (which should be lib) to read and write the files, and allows everyone else to read them but not write them. (UNIX directories must have execute permission set in order to look inside the directory.) If either the "group" or "other" write permissions or both are set (e.g., '-rw-rw-r--' or '-rw-r--rw-') then the write permission should be removed by logging in as lib (since 'lib' owns all the files), changing your directory to the library in question, and typing
'find . -exec chmod go-w {} \;'

11.10 MAKING NEW LIBRARIES BY MODIFYING OLD ONES

A good way to build a library is to follow the example of an already completed library. The standards that were used for constructing the Valid libraries are contained in this chapter, Valid Library Style and Standards.

For example, if all you need are different shaped bodies, you can do this by copying the Valid library, changing it to your local signal syntax, and then modifying the body drawings to correspond to the desired standards. If the pin names are preserved, the models will not need to be changed.

Similarly, if you want a TTL library that, for example, does not use worst case timing specs, you can take the Valid library and change only the timing models (and the Simulator models if you use the Simulator).

Valid Library Styles and Standards

11.11 INTRODUCTION AND MOTIVATION

This document is intended to describe how to make Valid libraries. Valid libraries are those SCALD directories Valid Logic Systems provides that contain standard components (such as Timing Verifier and Simulator primitives) or devices (such as LSTTL, STTL, ECL, etc). There are goals of this document:

1. Pass along some library lore accumulated during the construction and use of the present libraries.

2. Document the Valid design style so that ALL libraries look and behave the same.

3. Explain the Valid standards for library design and motivate the reasons for chosing them.

The rules contained herein are already being adhered to. If a situation arises that is not covered by these standards, the library manager should be informed so that any design decision can be codified.

Library construction will be broken into two parts: creation of bodies and design of models. It is assumed that the designer understands how to use SCALD Directories (see Chapter 5).

11.12 NOMENCLATURE

The following list of terms is included to make sure that there is some common ground for communication.

DEVICE
: A physical device such as an LS00, 10121, etc. The device definition is found in a .PART drawing, the timing model in a .TIME drawing, and the Simulator model in a .SIM drawing.

PRIMITIVE
: A component that is not defined in terms of other components. A device is a primitive when compiling for the Packager since they are not defined in terms of other components. Timing Verifier primitives are always primitive; they are predefined in the Timing Verifier and cannot be defined in terms of other components. Any drawing in a design can be made into a primitive

by giving it the .PRIM type with the Graphics Editor or using the PRIMITIVE directive in the Compiler.

TEXT SIZE
The size of text in a drawing is controllable. That is, the size can be changed from the nominal. In this document, text sizes will be given with respect to the nominal as, for instance, (0.6) which means that the text is 0.6 the size of the nominal text.

11.13 HOW TO CREATE A BODY

The following rules for creating bodies are separated into gates and others since there are different standards for each. They will be extended as needed. If there are bodies that do not conform, let the library manager know.

A SHORT NOTE ON GRIDS

The Graphics Editor sets the grid to 0.05 2 when editing a body. This grid should be used for everything with the exception of notes and conections to slanted lines. (These occur on the select inputs of multiplexors, for example.) In these cases, a grid of 0.01 10 should be used. Do not use any other grid when making bodies.

STANDARDS FOR THE CREATION OF GATE BODIES

1. The origin of the gate (at 0,0) should be as close to the center of the body as possible. It should be equidistant from the input pins and output pins and centered on the output pins.

2. Gates should be 0.3" wide and 0.6" from input pin to output pin.

3. Input pins should be on the left, output pins on the right. Enable pins should be on the bottom.

4. ALL pins should be on 0.1" centers.

5. Input pins should be symmetrical with output pins.

6. If last two rules are incompatible, move output pin to 0.05" center. In this case, make sure the origin body is still at 0,0 and make the output pin(s) 0.05" below the center (origin).

7. All pins should be connected to the body with either a 0.1" stub (wire) or a bubble (0.1" circle).

8. Open collector pins are marked with a "OC" (0.6) placed immediately above the bubble for the pin (or, if no bubble, where the bubble would appear.

9. TRI_STATE pins are not marked as such.

10. The shape of the gate should indicate what the pins are; no extra notes should be used.

11. Body names should be in the same place on all body versions.

12. Wherever possible, bubbling should be done with versions AND bubble groups. If this means there are more than 4 bodies, use bubble groups only.

13. Version 1 of the body should be the simplest: no vectored pins, default bubble state.

14. Notes on bodies should be placed as follows:

    (a) Size should never be less than (0.6). This is the smallest legible font on the electrostatic plotter.

    (b) Device body names should be (0.6) and lower left justified on the body.

    (c) Primitive body names should be (0.75) and centered in the body.

    (d) If body name (0.75) will not fit in the body, make it smaller; but never smaller than (0.6).

15. ALL default properties should be placed on 0.1" centers.

16. NO invisible properties should be attached, except for a section identifier on each section of an asymetrical part.

17. When invisible properties are attached, they should be located 0.05" from the origin (no more, no less) and must not be co-located with any visible object.

18. Timing Verifier primitives have default SIZE property; no TIMES property.

19. Device bodies (parts) have no default SIZE or TIMES property.

20. SIZE property is placed immediately above the body.

21. Default properties should be placed in the same place on all body versions.

22. All body versions should have the same default properties (there are a few exeptions - but this happens VERY RARELY).

23. All default properties should have the same values on all body versions (with a few RARE exceptions).

STANDARDS FOR THE CREATION OF OTHER BODIES

1. The origin of the body (at 0,0) should be as close to the center of the body as possible.

2. Flip Flops should be 0.4" X 0.8".

3. Bodies should be made as small as possible but not crowded.

4. Input pins should be on the left and output pins on the right. Enable and select pins should be on the bottom.

5. ALL visible pins MUST be on 0.1" centers - there are no exceptions! Pass-through pins may, if absolutely necessary, be placed on 0.05 inch grid.

6. ALL pins should connect with the body with either a 0.1" stub (made with a wire) or a bubble (0.1" circle).

7. Bus through pins should be used wherever possible; especially on clocks, enable, and select lines. Bus through pins should NOT have stubs or wires and should not be labeled.

8. Open collector pins are marked with a "OC" (0.6) placed immediately above the bubble for the pin (or, if no bubble, where the bubble would appear.

9. TRI-STATE pins are not marked as such.

10. The shape of the body should reflect (wherever possible) the function of the body.

11. Edge-triggered clock pins are marked with a clock wedge 0.1" at the base and 0.1" tall. DC clocks are labeled with a note.

12. All pins (except for clocks; see above) are labeled with notes. The text size should be (0.75). The note should be centered on the pin and as close to the edge of the body as possible.

13. Pin labels should clearly identify the pin and use a name that a hardware designer will quickly recognize from the manufacturers data sheet for the device. It is more important to be consistent across a logic family than to be identical with the data sheet. For example, the enable pin on a TTL multiplexer would be called ENABLE, not STROBE, since all the other enables in the family are called ENABLE.

14. Versions of the body are not, typically, used for bubbling. Use bubble groups instead.

15. Versions 1 and greater of the body should be used to create bodies with vectored input and output pins.

16. Version 2 of the body should have all pins explicitly marked; that is, an octal latch must have eight individual input and output pins.

17. The name of the body should appear on the body as a note. The size should be as large as possible (0.8, 0.9, or N).

18. ALL default properties should be placed on 0.1" centers.

19. Invisible properties should not be used, except for section identifiers on asymetrical bodies, and the NEEDS_NO_SIZE and HAS_FIXED_SIZE properties.

20. When invisible properties are attached they should be located 0.05" from the origin (no more, no less) and must be co-located with any visible object.

21. Timing Verifier primitives have default SIZE property; no TIMES property.

22. Device bodies (parts) have no default SIZE or TIMES property.

23. SIZE property is placed immediately above the body.

24. Default properties should be placed in the same place on all body versions.

25. All body versions should have the same default properties (there are a few exeptions - but this happens VERY RARELY).

26. All default properties should have the same values on all body versions (there are a few_exceptions).

11.14 HOW TO DESIGN TIMING VERIFIER AND SIMULATOR MODELS

For each device in the library, there is a Timing Verifier (.TIME) and Simulator (.SIM) drawing defining the models for the device. There are several important goals to keep in mind when designing a model:

1. The user will not understand the internal structure of the models and does not want to have to look at them. Any errors during timing analysis, for instance, must be referred to signal names the user understands. This will not be the case if the model has a lot of unnamed signals. For this reason, the models should always be designed so that error messages will be reported with signal names that mean something to the user. All checker bodies -- Setup and hold checkers, min pulse width checkers -- should have their inputs connected to interface signals. When this is not possible signals internal to a model (local signals) should be given names that describe the signal. Try to make it possible to understand all error messages without having to refer to the model - reference everything back to the device itself (the PART being modeled). If, to achieve the above goals, the model has to be made larger or more complex than neccessary, that is fine; ease of understanding by the user is better than incremental execution improvements. Also, checker bodies have negligible inpact on verification time.

2. Do not connect sign-extenders or mergers to the interface signals of a body. (This will cause confusing synonyms to be generated.) Place a zero-delay, non-inverting buffer of the appropriate SIZE between such structures and the interface signal. (When applying this rule remember that for all practical purposes, NOT bodies are wires.)

3. Many parts have both a true and complement output. If one of the outputs is dotted with some other signal, the other output should not be affected. Timing Verifier and Simulator primitives have only a true output. To generate both outputs, an inverting and non inverting buffer should be used, one buffer driving the complemented output and the other buffer driving the un-complemented output.

4. Try to make the layout of the model follow the layout of the body. That is, the interface signals in the model should appear in approximately the same physical relationship as on the body.

5. Keep the model simple. Timing models do not need to reflect the complete logical behavior of the part in order to provide accurate timing information. Simple models are easier to design, easier to understand, easier to test, and execute faster.

STANDARDS FOR THE CREATION OF MODELS

1. Every model must have a DEFINE body (the default) and a DRAWING body (with TITLE and ABBREV properties attached).

2. A B SIZE PAGE must be used as a border. The name of the drawing and the initials of the creator must be placed, with notes (2.0), in the appropriate boxes in the lower right hand corner. The note (1.5) "1 of 1" should appear in the PAGE block.

3. The drawing should be centered on the page.

4. A note block (notes enclosed with wires to form a a block) should be included to document any non-obvious or critical design decisions. Any assumptions that were made that are not obvious

should be stated.

5. PATH properties should be attached with the AUTO_PATH.

6. All properties on bodies within the model should be above the body or to the right. The properties should be placed one above the other left aligned. Property values and names should both be displayed execpt for PATH which should be value only.

7. All signals must be in SCALD standard form ('' = low asserted, '..' = subrange, bit ordering = right to left).

8. NO bit lists should be used (within bit subscripts). Steps are handled correctly.

9. All interface signals (those with the 'I' property) should have an explicit width specified unless the signal is a scalar.

11.15 PHYSICAL PART ANNOTATION

Physical information must be added to the physical parts. These properties may be added directly to the drawings, or may be specified in a text file and merged with the drawing information with the librarian program. The text file method is preferred since it makes the body files smaller, is easier to change, and is easier to check.

The standard physical properties on pins are:

PIN_NUMBER
OUTPUT_LOAD
INPUT_LOAD
BIDIRECTIONAL
OUTPUT_TYPE

The properties above are associated with the pins of the body. Every pin on a body (on all versions of the body) must have the PIN_NUMBER property. It must have one of (or both) the OUTPUT_LOAD and INPUT_LOAD properties (if both, it must also have the BIDIRECTIONAL property). If the pin is an output (has the OUTPUT_LOAD property) and can be wire-tied to another output, it must be given the OUTPUT_TYPE property which specifies the type of pin (OC = open collector, OE = open emitter, or TS = TRI-STATE) and the logic function created by tying the outputs together (AND or OR).

The standard physical properties placed in the PART drawing are:

FAMILY
POWER_PINS
PART_NUMBER

These properties are attached to the DRAWING body within the PART drawing. The FAMILY property specifies the logic family of the part and can have any value. The standard values (the ones used in the Valid libraries) are:

TTL
STTL
LSTTL
ECL10K
ECL100K

The POWER_PINS property is used to specify the pins of the part that are connected to the power supplies. See the Packager documentation for a complete description of the form this property takes. The PART_NUMBER property is used to assign an internal part number for the part and is NOT used within the Valid libraries.

11.16 HOW TO BUILD A LIBRARY COMPONENT

This section describes a step by step procedure for the construction of a library component. It is intended to be complete but, since there is much that can be taken for granted, some important details may be left unsaid. As these are identified, they will be added to this document.

A component consists of a number of drawings with the same name. Each drawing describes a different aspect of the component and has a unique extension (name describing its function). The drawing defining the shape (which forms a symbolic representation for the component) has the BODY extension, the drawing describing the physical part information has the PART extension, the drawing defining the Timing Verifier model for the component has the TIME extension, and the drawing defining the Simulator model for the component has the SIM extension. Each of these drawings must be created to complete a library entry for the component.

The libraries are normally designed, tested, documented, and maintained by the librarian. It is assumed that the librarian is very familiar with the SCALDsystem, the SCALD design language, logic design, and UNIX. The rest of this document continues with this basis.

THE LIBRARY DIRECTORY

The first step in building a library is the creation (within the UNIX system) of the appropriate directories. The Valid standard libraries are kept in /u0/lib with sub directories holding components of various logic families. For instance, the sub directory /u0/lib/lsttl contains the LSTTL library. All components of the library should be kept in the same UNIX directory. The directory should be created with read-only access to everyone, with write/execute reserved for the librarian. The lib user owns all the files in the Valid libraries and should own any other libraries created.

A file, called the SCALD directory, resides in the UNIX library directory. The SCALD directory is a file created by the graphics editor (the user specifies its name) and is used to map SCALD component names to UNIX file names. Each component is stored in its own UNIX directory. The SCALD directory gives the name of this UNIX directory. The component directory contains all the files that are part of the component. These include .BODY, .LOGIC, .SIM, .TIME, .PART, and .PRIM drawings. Editor logs, revision histories, version control, compilation directives, etc. are also stored here.

LIBRARY COMPONENT INFORMATION

The second step is to gather the information needed to enter the library components. This consists of functional descriptions, pinouts, loading specifications, and timing behavior. The manufacturers' data books provide all of this information as well as suggesting a shape to be used for the body.

Some decisions must be made about how to assign values that are not specified in the data sheets. For example, minimum propagation delays are seldom specified for TTL. The librarian must decide what values must be chosen and should be consistent for all components in the library. Such descisions should be documented in a file placed in the directory so that users of the library can read them.

BUILDING THE BODY

The next step is to build the body. A body is the symbolic representation for the component. There are several crucial points to keep in mind:

1. The sizes of all bodies with similar complexities should be the same. All gates, for instance, should be the same size so that one may be replaced by another without changing the wires connected to it.

2. Flow through the bodies should be consistent. For instance, all inputs should be on the left, outputs on the right, and enables on the bottom. Such conventions make it easier for the user of the library.

3. Bodies should be made as small as possible so that a complex logic circuit can be placed on a B size print. The bodies should be large enough to prevent crowding of the notes within them and permit notes that are legible on a hardcopy.

4. The body should represent the logical function of the part where ever possible. Since the Graphics Editor makes it easy to build complex bodies, the librarian has considerable freedom which should be exploited.

5. The name of the component should appear on the body so that the body is easy to identify when found in a drawing. The smallest practical text size for this purpose is 0.6 of normal.

6. The names given to the pins of the part should correspond to some standard. This is very important since the Compiler reports many errors by pin name. The names should be chosen as shown in the manufacturers' data books, or some local standard must be created. Some bodies do not have their pins annotated (as is the case with gates) and a convention becomes very important. The Valid library convention is to letter the input pins as found in the data books, or alphabetically. The output pins are as found in the data books, or Y is used.

Pin Names

After the body has been defined, the pins must be named. Pin names should be chosen to be obvious to the user of the component since error messages often refer to these names. If names are used in the data books, it is a good idea to use them.

The SCALDsystem understands the notion of vectored components. A component can be a vectored component only if it is so defined. To make a vectored component, the pins (or some of the pins) of the component must be a function of the parameter SIZE. SIZE is used to specify the number of bits the component is to represent (usually). The pin names that are to be vectored are given a bit subscript that depends on the value of SIZE. A SIZE wide signal should be specified as <SIZE-1..0> or <0..SIZE-1> as appropriate for the signal syntax in use. (See Creating and Maintaining Valid Libraries in the previous section).

Notes

The body should be annotated with notes that describe the body's pins, its component name, and any other information that is important. The notes should be easily readable and should not be crowded. The most important piece of information to be placed on the body is its component name.

Origin

The origin body (the little X that appears in the center of the screen when the body is first edited) is used to specify the origin of the body. It should not be moved. If it is, the editor produces an error message (when the body is written) and moves it back to the center. The body should be symmetrical about the origin (the origin should be at the center of the body). All body properties are attached to the origin body.

Physical Information

Each pin of the body should be annotated with physical information. This information is specified in a text file associated with the library. The two properties that must always be specified are: PIN_NUMBER and INPUT_LOAD (or OUTPUT_LOAD). See the Packager documentation (Chapter 8) for a detailed description of the use of each of these properties.

The other properties that may be needed are:

OUTPUT_TYPE	if pin is open collector, TRI_STATE, open emitter, etc.
BIDIRECTIONAL	if pin is both an input and an output.

THE DEFINITION OF THE PART DRAWING

The PART drawing serves to specify additional physical information about the component, as well as the abbreviation to be used when constructing path elements. There are only two bodies in a PART drawing; DEFINE and DRAWING.

The DEFINE body is used to specify text macros and has two default text macros: X-FIRST and X_STEP. These values should not be changed (they are 0 and SIZE respectively). No other text macros should be defined.

The DRAWING body is used to specify properties of the entire DRAWING (in this case, the entire part). The properties that should be attached here are TITLE (which is the component name) and ABBREV (which is the abbreviation for the component name to be used when constrcuting path elements). Two physical properties should be added: FAMILY (which specifies the logic family of the part) and POWER_PINS (which specifies which pins are connected to power supplies). The Packager documentation describes these properties in detail. These properties are usually added in the text file. If the physical part name is different than the logical part name, the PART_NAME property should be attached. For example, the logical part name LS00 may correspond to the physical part name 74LS00.

THE TIMING MODEL

Each component must have a timing model. The creation of a timing model is very similar to the creation of any logic drawing except that the parts used are Timing Verifier components. See the Timing Verifier documentation for a description of timing models.

THE SIMULATION MODEL

Each component must have a simulation model. The creation of a simulation model is very similar to the creation of any logic drawing except that the parts used are Logic Simulator components. See the Logic Simulator documentation for a description of simulation models.

THE LIBRARY CHIPS FILE

See Creating and Maintaining Valid Libraries, for a description of the creation of CHIPS (part description) files.

TESTING THE LIBRARY

After a component has been entered, it should be tested. The minimal testing involves compiling the bodies and the models. Any syntax errors are discovered and can be corrected. The SIZE parameter should be set to 1 (the default) and to some other value to test the vector part implementation.

Functional testing of the models is more difficult. Timing models must be exercized to make sure that the model behaves correctly and that the DELAY, RISE, and FALL property values have been correctly assigned. Errors in setup and pulse width should be generated to make sure that the signals reported by the Timing Verifier have names that easily understood without looking at the model; all errors should be reported in terms of the pins of the part.

Simulator models should be tested on the simlulator to verify the functional behavior of the part. Complex parts may take a long time to verify, so budget a significant portion of the library development effort to testing.

11.17 SPECIAL LIBRARY COMPONENTS

There are several special library components that need to be discussed independently. These are not physical components, but rather are used because they have useful side effects.

THE NOT BODY

This body is used to convert a signal from one assertion to the other without a logical inversion taking place. It is used where a signal's assertion does not match the bubble state of the pin it is connected to. The NOT body provides an explicit change of assertion and should be used whenever bubble conventions are being followed.

The NOT body has two functions. First, it makes sure that the signals connected to it have the proper assertions (one of them must be low asserted and the other high asserted). Second, it synonyms the two signals. The NOT body definition is found in NOT.LOGIC. It consists of nothing more than a SYNONYM body to which the two NOT body signals are connected.

The NOT body does not expand into any physical components or primitives. Its only purpose is the side effect of synonyming two signals that have differing assertions.

THE SYNONYM BODY

Signals can be synonymed together making them aliases for each other. Both refer to the same physical signal (net). The synonym function is implemented by the Compiler. Two signals are synonymed if they are connected to the same pin of some body. If a wire is given two names and connected to a pin of come body, the Graphics Editor outputs the pin twice; once with each of the signals assigned to the wire. The Compiler will synonym the two signals because they are attached to the same pin. Bus through pins are implemented using the same function. In this case, there are two pins on the body with the same name and any signals connected to them are synonymed together.

The SYNONYM body is simply a body with two pins of the same name. The signals connected to it will be synonymed together. The assertions of the two signals must match. The synonym component has a definition found in SYNONYM.LOGIC. It is empty. There is a property on the DRAWING body which allows the drawing to be empty (this is normally not permitted). The property, ALLOW_PRIMITIVE, gives permission to the drawing to be a primitive.

THE MERGE BODIES

There are several merge bodies whose function is to combine a number of separate signals into a signal signal. This is performed by synonyming the single signal with the concatenation of the other signals. There are several mergers provided. Each accepts a different number of input signals to be concatenated together. The mergers provided in the Valid libraries are:

2 MERGE
3 MERGE
4 MERGE
5 MERGE
6 MERGE
7 MERGE
8 MERGE

Other mergers can be defined. The definition of the merger is found in a .LOGIC drawing which contains only a SYNONYM body.

THE PHANTOM GATE

Phantom gates are used when designing with a logic family which permits the tying of several outputs together. OPEN COLLECTOR outputs in TTL is one example. The phantom gate is used to tie outputs together with a gate that

describes the logic function (in the case of OPEN COLLECTOR, the function is AND) making the logic function much easier to understand. The phantom gates appear as normal logic gates. They are given an additional property WIRE_GATE which informs the Packager that they are to be removed before creating a physical net list. The Packager documentation describes the use of phantom gates as does the SCALD III Language documentation.

The WIRE_GATE property is attached to the DRAWING body of the .PART drawing for the phantom gate.

11.18 VALID COMPONENT LIBRARIES

The Valid Component Libraries are described on the following pages.

ECL 100K Library

There have been a few changes in the ECL 100K Library for Release 5.1.

The ECL 100K Library requires approximately 1.2 MBy (2310 blocks) of disk storage on the S-32. It contains bodies, physical, timing, and simulation models for the following components:

100101	triple 5-input or/nor
100102	quint 2-input or/nor w/ enable
100107	quint exclusive-or/nor
100112	quad driver w/ enable
100113	quad driver
100114	differential line receiver
100117	triple 2-wide or-and/or-and-invert
100118	5-wide 5-4-4-4-2 or-and/or-and-invert
100122	9 bit buffer
100123	hex bus driver
100124	hex TTL-to-ECL translator
100125	hex ECL-to-TTL translator
100126	9 bit backplane driver
100130	triple D-type latch
100131	triple D-type flip-flop
100136	4-stage counter/shift register
100141	8-bit shift register
100142	4 X 4 content addressable memory
100145	16 X 4 read/write register file
100150	hex D-type latch
100151	hex D-type flip-flop
100155	quad multiplexer/latch
100156	mask-merge
100158	8-bit shift matrix
100160	dual parity generator/checker
100163	dual 8-input multiplexer
100164	16-input multiplexer
100165	universal priority encoder
100166	9-bit comparator
100170	universal demultiplexer/decoder
100171	triple 4-input multiplexer w/ enable
100179	carry look-ahead
100180	fast 6-bit adder
100181	4-bit binary/BCD ALU
100182	9-bit wallace tree adder
100183	2x8-bit recode multiplier
100255	ECL 100K to TTL converter
100422	256x4-bit static random access memory
100474	1024x4-bit static random access memory

APPLICATION NOTE

This section describes the usage of some special parts in ECL 100K Library.

1. 100114 - hex TTL-to-ECL translator

   This part can be used in three ways: differential line receiver, noninverting translator, or inverting translator. Version one of the body is used for differential line receiver, version two is for noninverting translator, and version three is for inverting translator.

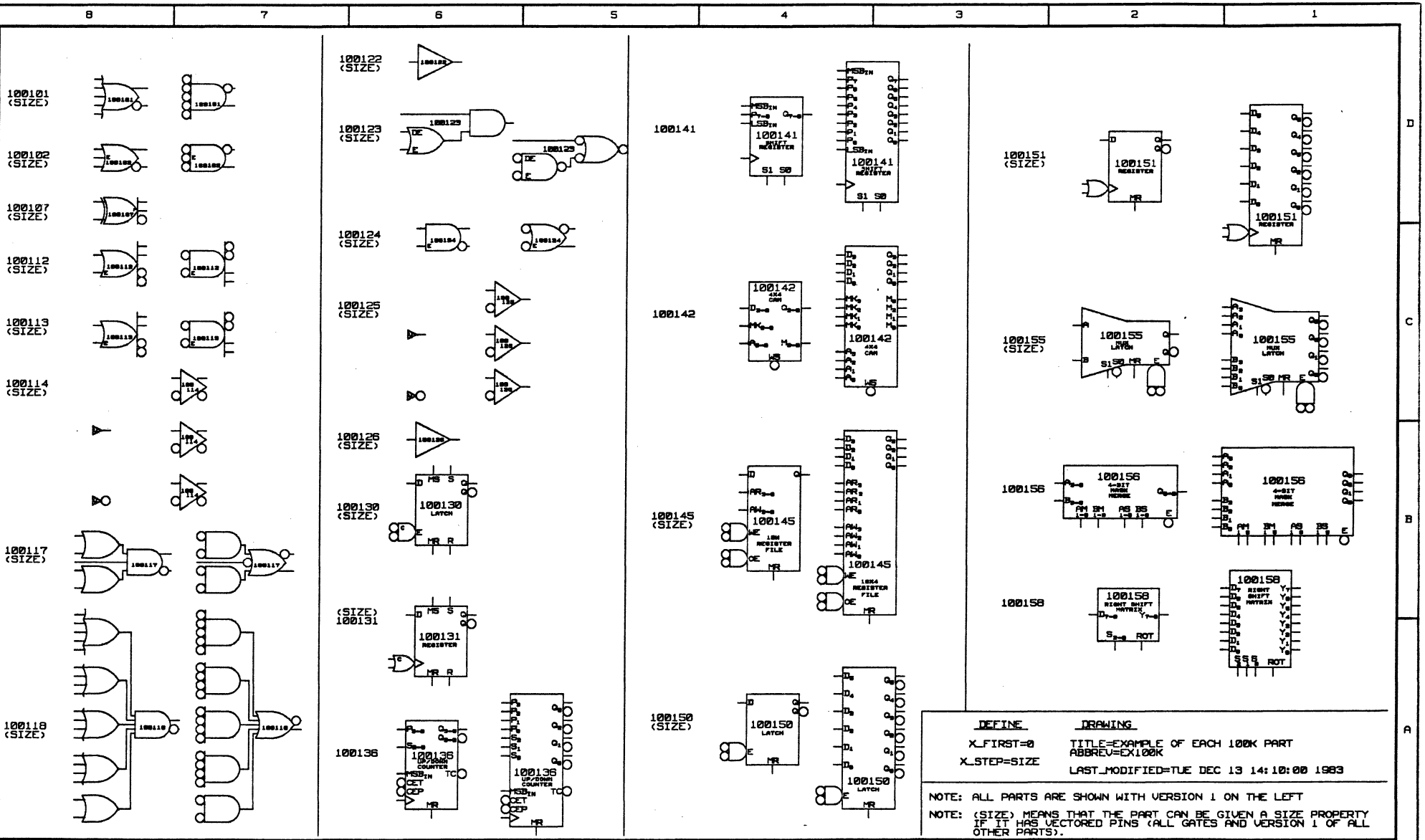
   When the part is used as inverting/noninverting translator, one of the input pin has to be tied to the bias pin (pin "V"). If the part is sized, then the user has to put a replicate body between the "V" pin and the input pin. This is because the "V" pin is a scalar pin and the input pin is a vectored pin. There is a restriction in the current implementation of the model in that the user cannot put a size greater than 5B to the part (there are 5 sections in a package), because otherwise the post processor will not be able to package the part correctly.

   Body version two should be used for inverting translator. The "V" pin is connected to the input of a replicate body (for size > 1B) and the output of the replicate body is connected to pin "A<SIZE-1..0>" of the part (the pin directly opposite the "V" pin). No replicate body is needed for size=1B.

   If noninverting translator is desired, then body version three should be used. In the same way as above, the "V" pin is connected to pin "B<SIZE-1..0>" (the pin directly opposite the "V" pin) through a replicate body. No replicate body is needed for size=1B.

2. 100125 - hex ECL-to-TTL translator

   Same as 100114 except that the maximum size is 6B.

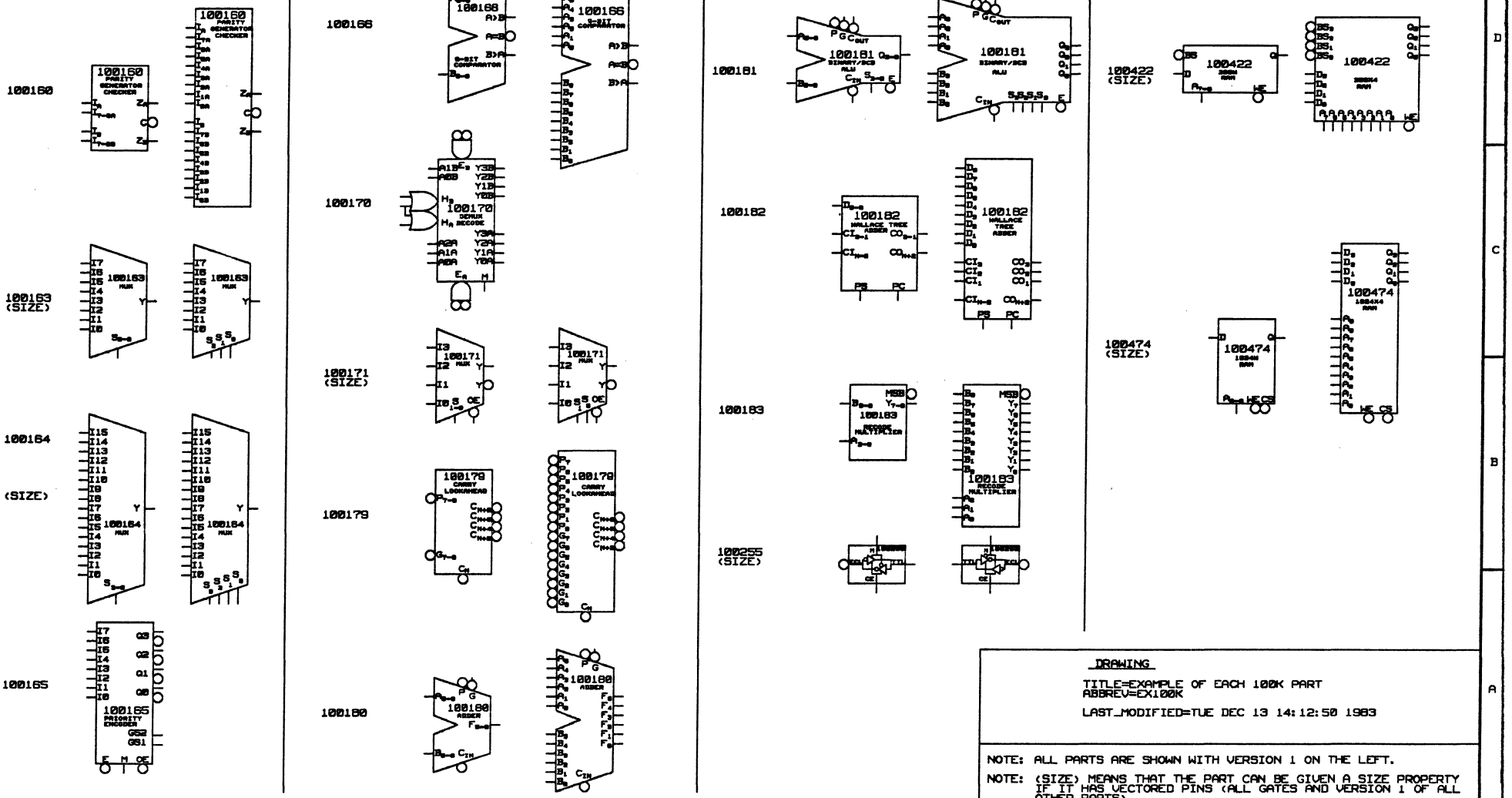


VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH 100K PART

DATE:

ENGINEER:



DRAWING

TITLE=EXAMPLE OF EACH 100K PART
ABBREV=EX100K

LAST_MODIFIED=TUE DEC 13 14:12:50 1983

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT.

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS (ALL GATES AND VERSION 1 OF ALL OTHER PARTS).

VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH 100K PART

DATE:

ENGINEER:

ECL 10K Library

A couple of new parts have been added to the ECL 10K Library for Release 5.0.

The ECL 10K Library requires approximately 0.8 MBy (1580) of disk storage on the S-32. It contains bodies, physical, timing, and simulation models for the following components:

10016	counter
10100	quad 2-input nor w/ strobe
10101	quad or/nor
10102	quad 2-input nor
10103	quad 2-input or
10104	quad 2-input and
10105	triple 2-3-2-input or/nor
10107	triple 2-input exclusive-or/exclusive-nor
10109	dual 4-5-input or/nor
10110	dual 3-input 3-output or
10111	dual 3-input 3-output nor
10113	quad exclusive-or
10115	quad line receiver
10117	dual 2-wide 2-3-input or-and/or-and-invert
10118	dual 2-wide 3-input or-and
10119	4-wide 4-3-3-3-input or-and
10121	4-wide or-and/or-and-invert
10123	triple 4-3-3 input bus driver
10124	quad MTTL to MECL translator
10125	quad MECL to MTTL translator
10130	dual latch
10131	dual D-type master-slave flip-flop
10132	dual multiplexer w/ latch and common reset
10133	quad latch
10134	dual multiplexer w/ latch
10135	dual JK master-slave flip-flop
10136	universal hexadecimal counter
10137	universal decade counter
10141	4-bit universal shift register
10145	16 X 4 register file
10153	quad latch
10158	quad 2-input multiplexer
10159	quad 2-input inverting multiplexer
10160	12-bit parity generator/checker
10161	binary to 1-of-8 decoder (low)
10162	binary to 1-of-8 decoder (high)
10163	error detection/correction circuit
10164	8-line multiplexer
10165	8-input priority encoder
10166	5-bit magnitude comparator

10170	9 + 2-bit parity generator/checker
10171	dual binary to 1-of-4 decoder (low)
10172	dual binary to 1-of-4 decoder (high)
10173	quad 2-input multiplexer/latch
10174	dual 4-to-1 multiplexer
10175	quint latch
10176	hex D-type master-slave flip-flop
10179	look-ahead carry block
10180	dual 2-bit adder/subtractor
10181	4-bit arithmetic logic unit and function generator
10186	hex D-type master-slave flip-flop w/ reset
10188	hex buffer with enable
10189	hex inverter w/ enable
10195	hex inverter/buffer
10197	hex and
10210	high-speed dual 3-input 3-output or gate
10211	high-speed dual 3-input 3-output nor gate
10216	high-speed triple line receiver
10415	1Kx1 RAM

APPLICATION NOTE

This section describes the usage of some special parts in ECL 10K Library.

1. 10115 - quad line receiver

This part can be used in three ways: differential line receiver, noninverting translator, or inverting translator. Version one of the body is used for differential line receiver, version two is for noninverting translator, and version three is for inverting translator.

When the part is used as inverting/noninverting translator, one of the input pin has to be tied to the bias pin (pin "V"). If the part is sized, then the user has to put a replicate body between the "V" pin and the input pin. This is because the "V" pin is a scalar pin and the input pin is a vectored pin. There is a restriction in the current implementation of the model in that the user cannot put a size greater than 4B to the part (there are 4 sections in a package), because otherwise the packager will not be able to package the part correctly.

Body version two should be used for inverting translator. The "V" pin is connected to the input of a replicate body (for size > 1B) and the output of the replicate body is connected to pin "A<SIZE-1..0>" of the

part (the pin directly opposite the "V" pin). No replicate body is needed for size=1B.

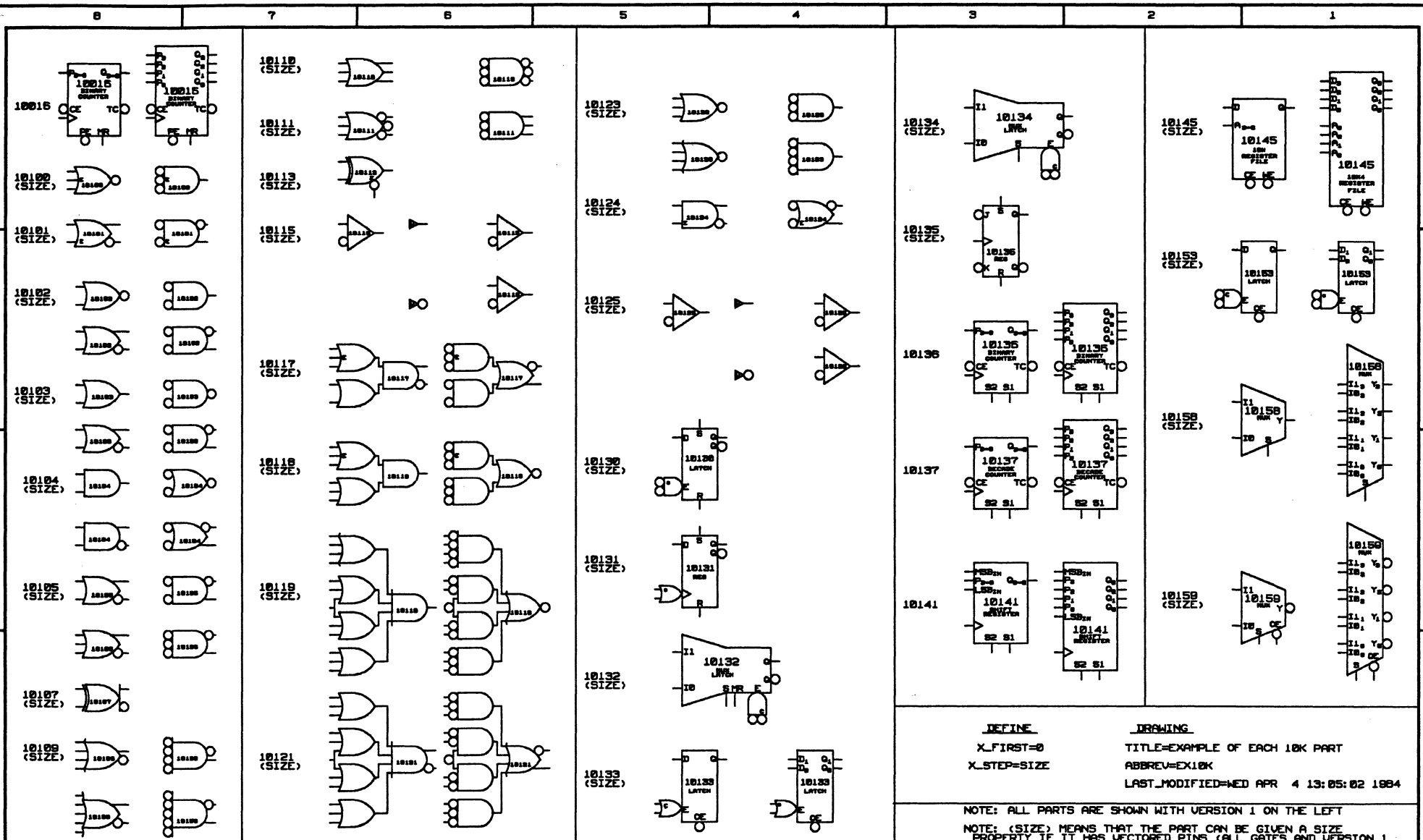
If noninverting translator is desired, then body version three should be used. In the same way as above, the "V" pin is connected to pin "B<SIZE-1..0>" (the pin directly opposite the "V" pin) through a replicate body. No replicate body is needed for size=1B.

2. 10125 - quad MECL to MTTL translator

Same as 10115

3. 10216 - high-speed triple line receiver

Same as 10115 except that the maximum size is 3B



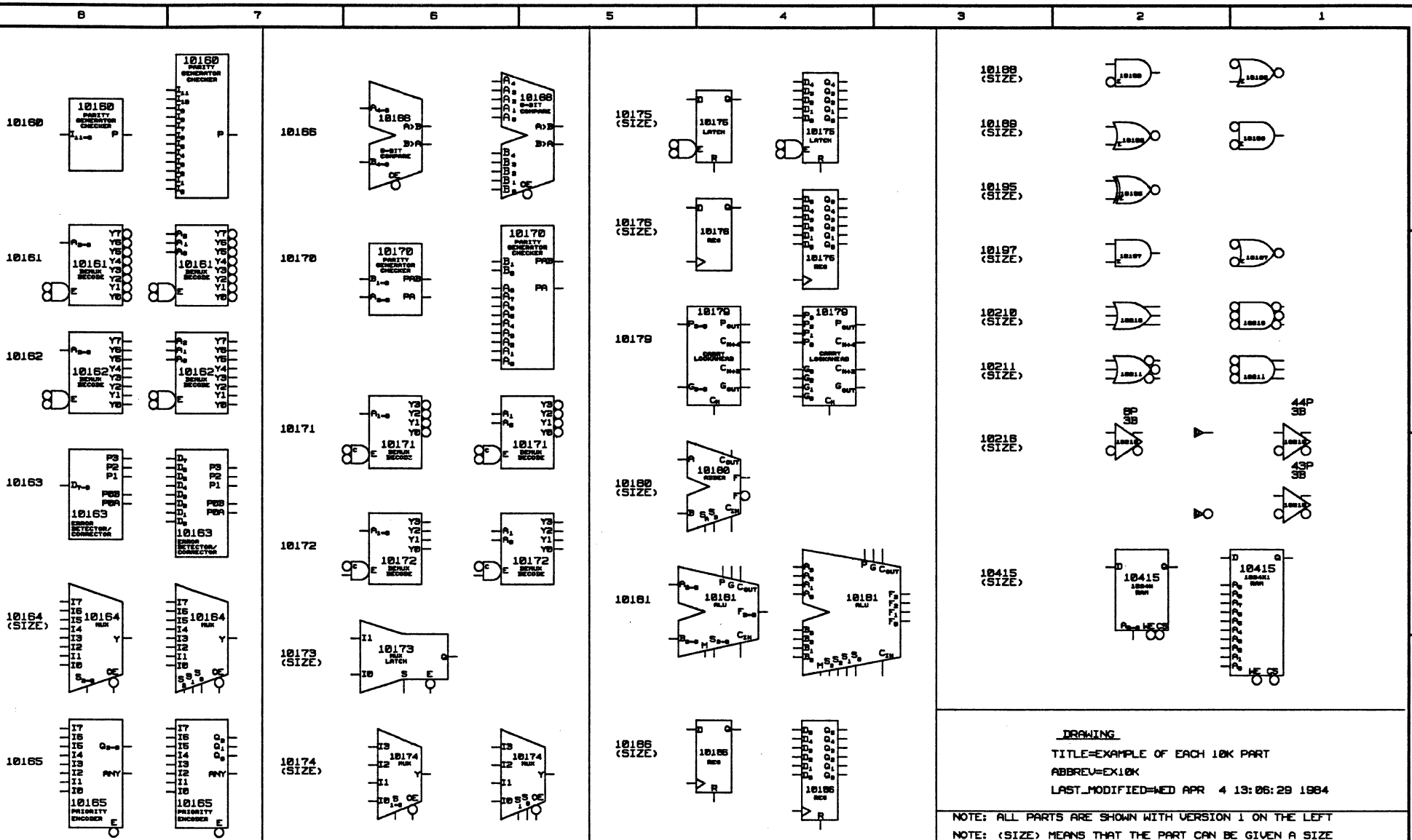
DEFINE	DRAWING
X_FIRST=0	TITLE=EXAMPLE OF EACH 10K PART
X_STEP=SIZE	ABBREV=EX10K
	LAST_MODIFIED=WED APR 4 13:05:02 1984

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS (ALL GATES AND VERSION 1 OF ALL OTHER PARTS)

VALID
LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH 10K PART	DATE:
ENGINEER: RKM	PAGE: 1 OF 2



VALID LOGIC SYSTEMS INCORPORATED

TITLE:	DATE:
EXAMPLE OF EACH 10K PART	
ENGINEER: RKM	PAGE: 2 OF 2

LSTTL Library

There have been a few changes in the LSTTL Library for Release 5.0.

The LSTTL Library requires approximately 3.0 MBy (5726 blocks) of disk storage on the S-32. It contains bodies, physical, timing, and simulation models for the following 110 components:

74LS00	quad 2-input nand
74LS02	quad 2-input nor
74LS03	quad 2-input open-collector nand
74LS04	hex inverter
74LS05	hex open-collector inverter
74LS08	quad 2-input and
74LS09	quad 2-input open-collector and
74LS10	triple 3-input nand
74LS11	triple 3-input and
74LS12	triple 3-input open-collector nand
74LS13	dual 4-input nand schmitt trigger
74LS14	hex schmitt-trigger inverter
74LS15	triple 3-input open-collector and
74LS20	dual 4-input nand
74LS21	dual 4-input and
74LS22	dual 4-input open-collector nand
74LS24	quad 2-input schmitt-trigger nand
74LS27	triple 3-input nor
74LS28	quad 2-input nor
74LS30	8-input nand
74LS32	quad 2-input or
74LS33	quad 2-input nor
74LS37	quad 2-input nand buffer
74LS38	quad 2-input open-collector nand buffer
74LS40	dual 4-input nand
74LS42	4-to-10-line decoder
74LS51	2-wide 3-input, 2-wide 2-input and-or-invert
74LS54	4-wide and-or-invert
74LS73	dual JK flip-flops w/ clear
74LS74	dual positive-edge-triggered D flip-flop
74LS75	4-bit bistable latch
74LS76	dual JK flip-flop w/ preset & clear
74LS83	4-bit binary full adders w/ fast carry
74LS85	4-bit magnitude comparator
74LS86	quad 2-input exclusive-or
74LS93	4-bit binary counters
74LS95	4-bit shift register
74LS107	dual JK flip-flops w/ clear
74LS109	dual JKbar positive-edge-triggered flip-flop

74LS112	dual JK negative-edge-triggered flip-flop
74LS123	dual retriggerable monostable multivibrators with clear
74LS125	quad bus buffers with three-state outputs
74LS132	quad 2-input positive-nand Schmitt triggers
74LS136	quad 2-input exclusive-or
74LS138	3-to-8 line decoders/multiplexers
74LS139	dual 2-to-4 line decoders/multiplexers
74LS148	8-line to 3-line octal priority encoder
74LS151	1-of-8 data selectors/multiplexers
74LS153	dual 4-line to 1-line data multiplexer
74LS155	decoders/demultiplexers
74LS157	quad 2-to-1-line non-inverting multiplexer
74LS158	quad 2-to-1-line inverting data multiplexer
74LS160	4-bit synchronous decade counters with direct clear
74LS161	4-bit synchronous binary counters with direct clear
74LS162	4-bit synchronous decade counters with synch clear
74LS163	4-bit synchronous binary counters with synch clear
74LS164	8-bit parallel output serial shift register
74LS165	parallel-load 8-bit shift registers
74LS166	8-bit shift registers
74LS169	4-bit synchronous binary up/down counters
74LS173	4-bit D-type registers w/ 3-state outputs
74LS174	hex D-type flip-flops
74LS175	quad D-type flip-flops
74LS181	arithmetic logic units/function generators
74LS190	synchronous BCD up/down counter
74LS191	synchronous binary up/down counter
74LS192	synchronous BCD up/down dual clock counters
74LS193	synchronous binary up/down dual clock counters
74LS194A	4-bit bidirectional shift register
74LS195	4-bit parallel-access shift registers
74LS197	presetable binary counters/latchs
74LS219	64-bit random access memory
74LS221	dual monostable multivibrators
74LS240	octal inverting 3-state bus transceiver
74LS241	octal non-inverting 3-state bus transceiver
74LS244	octal non-inverting 3-state bus transceiver
74LS245	octal non-inverting 3-state bus transceiver
74LS251	3-state data multiplexer
74LS253	dual data selectors/multiplexers
74LS257	quad 3-state non-inverting data multiplexer
74LS258	quad 3-state inverting data multiplexer
74LS259	8-bit addressable latches
74LS266	quad 2-input exclusive-nor gates w/ open collector
74LS273	octal D-type flip-flops
74LS279	quad S-R latches

74LS280	9-bit odd/even parity generators/checkers
74LS283	4-bit binary full adders
74LS298	quad 2-input multiplexers w/ storage
74LS299	8-bit bidirectional 3-state shift/storage register
74LS323	8-bit bidirectional universal shift/storage registers w/ 3-state outputs
74LS367	hex bus drivers
74LS368	hex bus drivers
74LS373	octal 3-state D-latch w/ common enable
74LS374	octal 3-state positive-edge-triggered D register
74LS377	octal D-type flip-flops with enable
74LS378	hex D-type flip-flops
74LS379	quad D-type flip-flops with enable
74LS381	arithmetic logic unit/function generator
74LS393	dual 4-bit binary counters
74LS540	octal buffers and line drivers w/ 3-state outputs
74LS541	octal buffers and line drivers w/ 3-state outputs
74LS590	8-bit binary counters w/ output registers
74LS593	8-bit binary counters w/ input registers
74LS640	octal 3-state inverting bus transceiver
74LS641	octal open-collector non-inverting bus transceiver
74LS642	octal open collector inverting bus transceiver
74LS645	octal 3-state non-inverting bus transceiver
74LS669	synchronous 4-bit up/down counters
74LS670	4 x 4 register files w/ 3-state outputs
74LS674	16-bit shift registers

LS00 (SIZE)

LS02 (SIZE)

LS03 (SIZE)

MAX_DELAY=10000NS

LS04 (SIZE)

LS05 (SIZE)

MAX_DELAY=10000NS

LS08 (SIZE)

LS09 (SIZE)

MAX_DELAY=10000NS

LS10 (SIZE)

LS11 (SIZE)

LS12 (SIZE)

MAX_DELAY=10000NS

LS13 (SIZE)

LS14 (SIZE)

LS15 (SIZE)

MAX_DELAY=10000NS

LS20 (SIZE)

LS21 (SIZE)

LS22 (SIZE)

MAX_DELAY=10000NS

LS24 (SIZE)

LS27 (SIZE)

LS28 (SIZE)

LS30 (SIZE)

LS32 (SIZE)

LS33 (SIZE)

MAX_DELAY=10000NS

LS37 (SIZE)

LS38 (SIZE)

MAX_DELAY=10000NS

LS40 (SIZE)

LS42

LS51 (SIZE)

LS54 (SIZE)

DEFINE

X_FIRST=0

X_STEP=SIZE

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

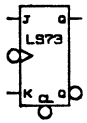
DRAWING:
TITLE=EXAMPLE OF EACH LSTTL PART
ABBREV=EXLS

ENGINEER: JIMMY

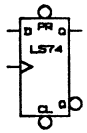
DATE: Wed Mar 14 01:29:31 1984

PAGE: 1 OF 7

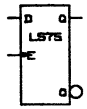
LS73 (SIZE)



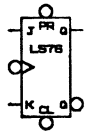
LS74 (SIZE)



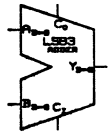
LS75 (SIZE)

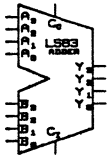


LS76 (SIZE)

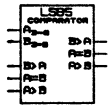


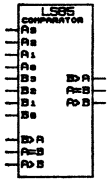
LS83





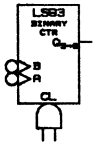
LS85

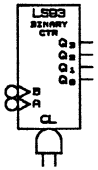




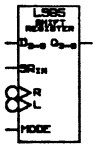
LS86 (SIZE)

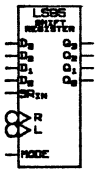
LS93



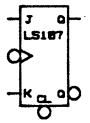


LS95

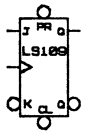




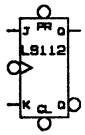
LS107 (SIZE)



LS109 (SIZE)

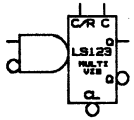


LS112 (SIZE)



PULSE_WIDTH=10000NS

LS123

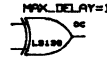


LS125 (SIZE)

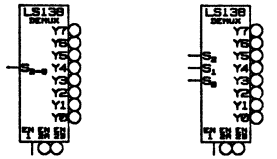
LS132 (SIZE)

MAX_DELAY=10000NS

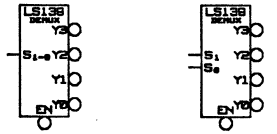
LS136 (SIZE)



LS138



LS139



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

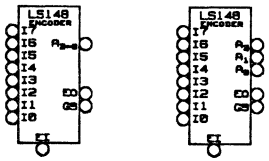
DRAWING:
TITLE=EXAMPLE OF EACH LSTTL PART
ABBREV=EXLS

ENGINEER:
JIMMY

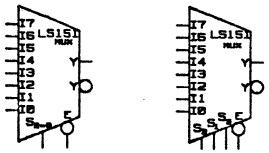
DATE:
Wed Mar 14 01:30:23 1984

PAGE:
2 OF 7

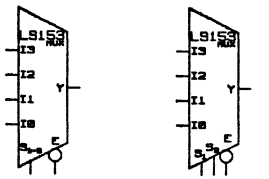
LS148



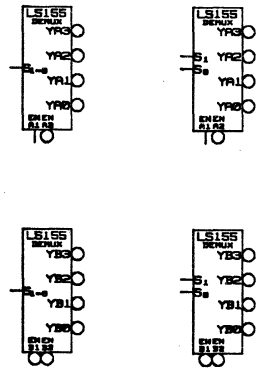
LS151 (SIZE)



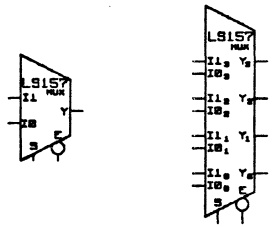
LS153 (SIZE)



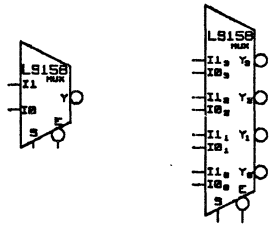
LS155



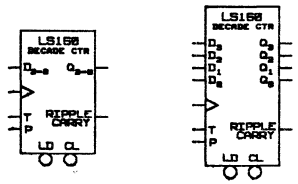
LS157 (SIZE)



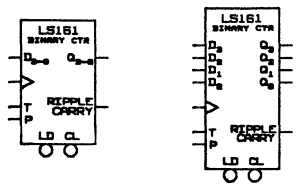
LS158 (SIZE)



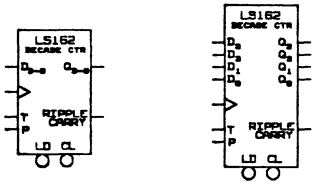
LS160



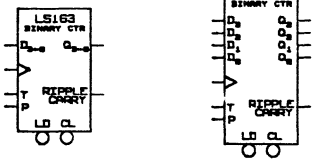
LS161



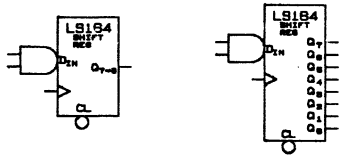
LS162



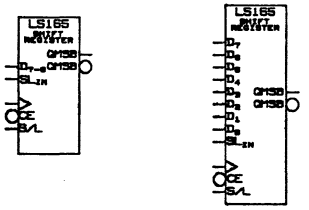
LS163



LS164



LS165



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

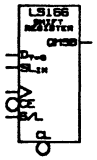
VALID LOGIC SYSTEMS INCORPORATED

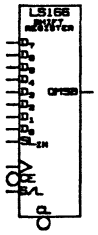
DRAWING:
TITLE=EXAMPLE OF EACH LSTTL PART
ABBREV=EXLS

ENGINEER: JIMMY

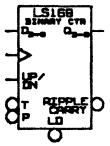
DATE: Wed Mar 14 01:31:32 1984

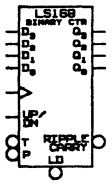
LS166



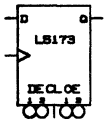


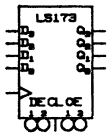
LS169



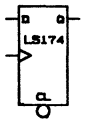


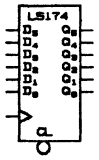
LS173



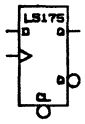


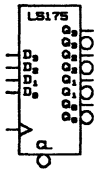
LS174
(SIZE)



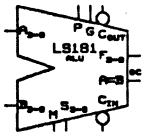


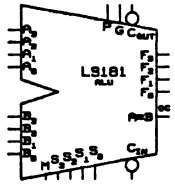
LS175
(SIZE)



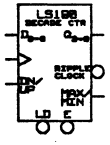


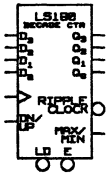
LS181



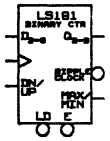


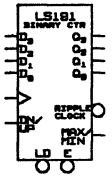
LS190



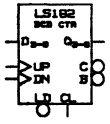


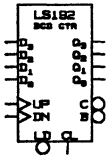
LS191



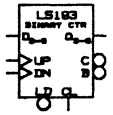


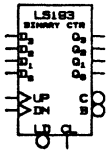
LS192



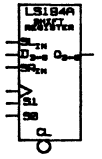


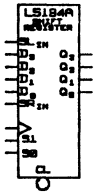
LS193



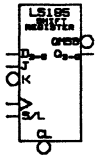


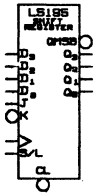
LS194A



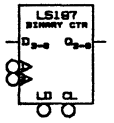


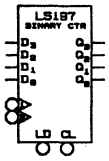
LS195





LS197





NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

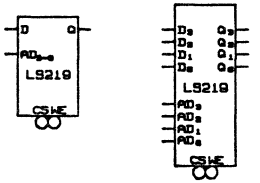
VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH LSTTL PART
ABBREV=EXLS

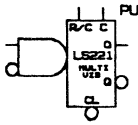
ENGINEER:
JIMMY

DATE:
Wed Mar 14 01:32:51 1984

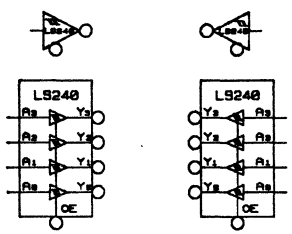
LS219 (SIZE)



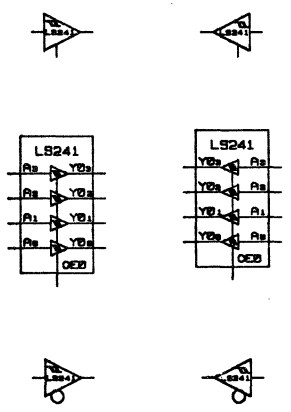
LS221



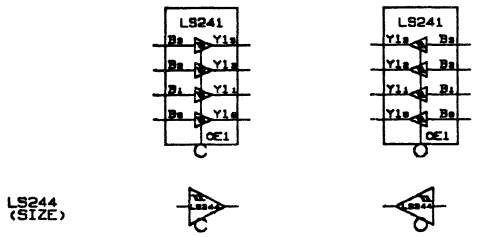
LS240 (SIZE)



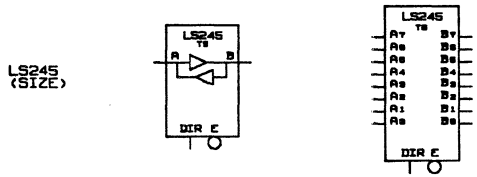
LS241 (SIZE)



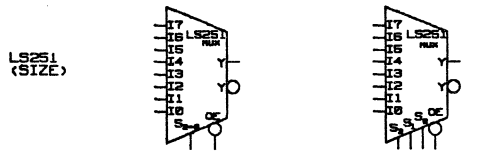
LS244 (SIZE)



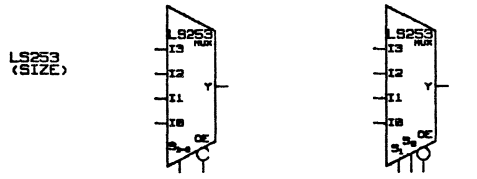
LS245 (SIZE)



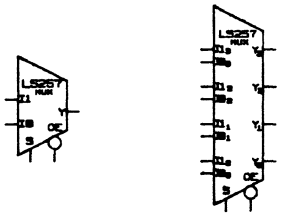
LS251 (SIZE)



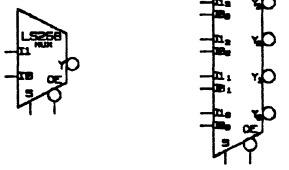
LS253 (SIZE)



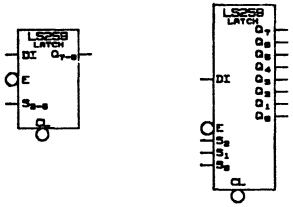
LS257 (SIZE)



LS258 (SIZE)

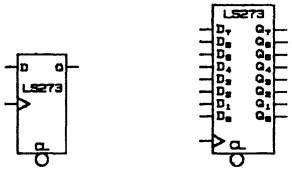


LS259



LS266 (SIZE)

LS273 (SIZE)



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

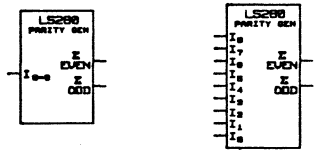
DRAWING:
TITLE=EXAMPLE OF EACH LSTTL PART
ABBREV=EXLS

ENGINEER: JIMMY

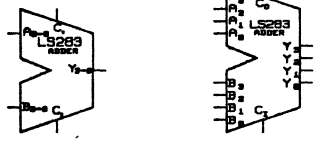
DATE: Wed Mar 14 01:34:33 1984

LS279 (SIZE)

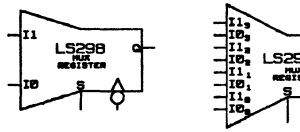
LS280



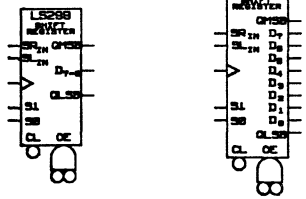
LS283



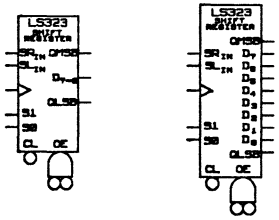
LS298 (SIZE)



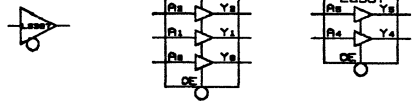
LS299



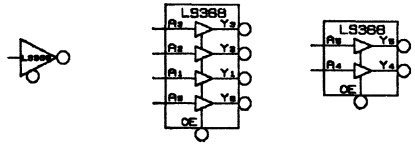
LS323



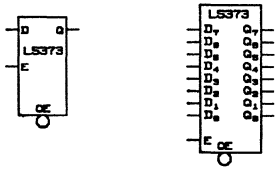
LS367 (SIZE)



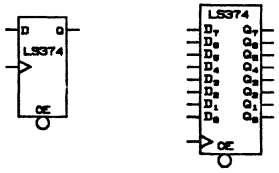
LS368 (SIZE)



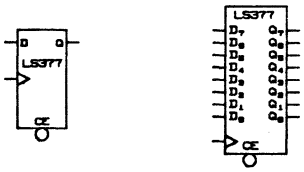
LS373 (SIZE)



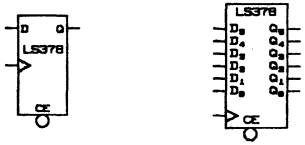
LS374 (SIZE)



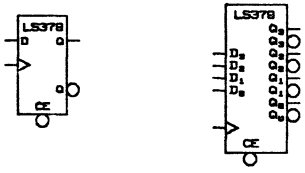
LS377 (SIZE)



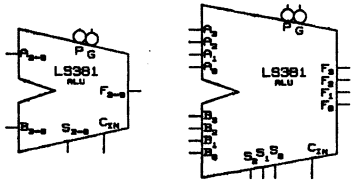
LS378 (SIZE)



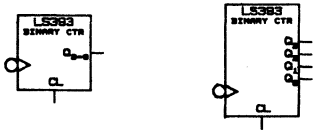
LS379 (SIZE)



LS381



LS393



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

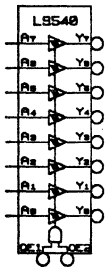
DRAWING:
TITLE=EXAMPLE OF EACH LSTTL PART
ABBREV=EXLS

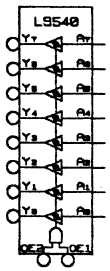
ENGINEER: JIMMY

DATE: Wed Mar 14 01:35:50 1984

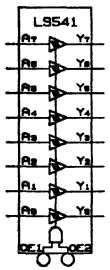
PAGE: 6 OF 7

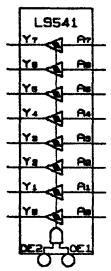
LS540
(SIZE)



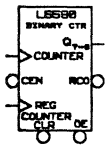


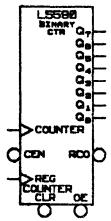
LS541
(SIZE)



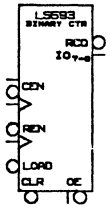


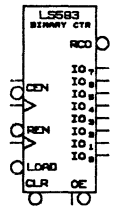
LS590



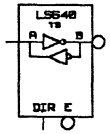


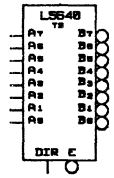
LS593





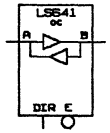
LS640
(SIZE)



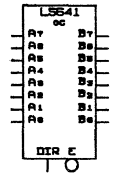


MAX_DELAY=10000NS

LS641
(SIZE)

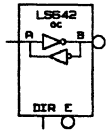


MAX_DELAY=10000NS

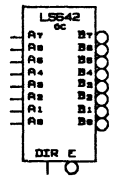


MAX_DELAY=10000NS

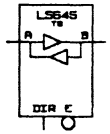
LS642
(SIZE)

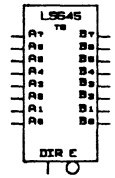


MAX_DELAY=10000NS

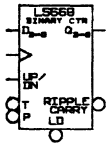


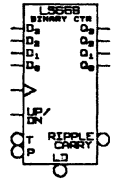
LS645
(SIZE)



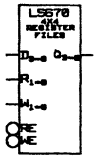


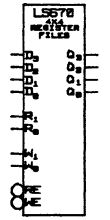
LS669



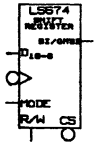


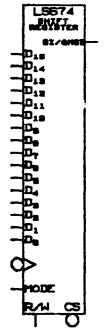
LS670





LS674





NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID
LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH LSTTL PART
ABBREV=EXLS

ENGINEER:
JIMMY

DATE:
Wed Mar 14 01:37:02 1984

PAGE:
7 OF 7

LSTTL Library Description

INTRODUCTION

This document describes the VALID LSTTL Library. This Library includes commonly used LSTTL parts, as manufactured by many suppliers. Each part has one or more bodies (the picture of the part), the physical information such as pin numbers and loading, a timing model, and a simulation model.

THE DRAWINGS (BODIES)

Each part has one or more drawings to represent it. For gates, the two versions represent different logical functions performed depending on the polarity of the input signal. An LS08, for example, performs an AND of high asserted signals, but an OR of low asserted signals. The versions of the parts allow it to be drawn either way.

Parts that are not gates are handled somewhat differently. If more than one version exists, version 1 is the vectored part (multiple bits signals are drawn as buses) and version 2 has the bits shown explicitly. For example, version 1 of the LS161 (a four bit counter) shows the output as a single pin with a width of four bits. The second version has all four bits explicitly shown, with one bit on each pin.

If a part has sections that are not interchangeable (such as the LS51) then there are additional versions describing the additional sections.

PIN NAMES

The following conventions were used for pin names in this Library.

1. All input to gates are labelled alphabetically starting with A.

2. All outputs of gates are called Y with a numeric suffix to distinguish between them.

3. All other pin names match the notes on the bodies.

4. If ambiguous, the note will be prefixed with ' IN' or ' OUT' to disambiguate it.

5. Clock pins are called 'CLOCK'.

6. Where the up and down clock pins are separate, they are called 'CLOCKUP' and 'CLOCKDN'.

7. Parts with sections that are different have the pin names suffixed with a number to distinguish between the sections.

INFORMATION SOURCE

The information used to construct these models came from the data books provided by the manufacturers. Specifications were taken from the TTL DATA BOOK (Second edition), the SUPPLEMENT TO THE TTL DATA BOOK (second edition) and the 1981 SUPPLEMENT TO THE TTL DATA BOOK, all by TI.

CALCULATION OF DELAYS

Each part has both minimum and maximum delays. The maximum delays are normally found in the catalog, and in these cases they have been used.

The minimum delay is not normally found in the catalog. We have used the lesser of 1/2 of the typical or 1/3 of the maximum as our minimum delay where it was not specified.

Minimum pulse widths are not found in the catalog for some of the parts. In this case an estimated time derived from the maximum toggle frequency was used.

DATA DEPENDENT DELAYS

Most TTL delays are functions of the value of the data. Rising delays are different from falling delays. In general, this is modelled correctly. However, there are some cases where this would have added greatly to the complexity of the model. In particular, tri-state parts are modeled with a single delay from enable to output, without consideration of whether the output is going to a 1 or a 0 state.

OPEN COLLECTOR GATES

Open collector gates have no fixed, maximum time delay. It is not possible to compute the delay from the schematic, since nothing is known about parasitics, tolerances, and so forth. The designer must assign an explicit maximum rising delay to each open collector gate. This delay is called MAX_DELAY and is arbitrarily set to 10000ns for any open collector part.

ONE-SHOTS

The same considerations hold true for one-shots. Here the user must calculate the pulse widths, taking into account all the one-shot tolerances, external component tolerances, temperature variations, and drift over the life of the circuit. This value must replace the value of PULSE_WIDTH that comes with the one-shot. The default value for PULSE_WIDTH is 10000ns.

FAST Library

There have been a few changes in the FAST Library for Release 5.0.

The FAST Library requires approximately 1.0 MBy (1896 blocks) of disk storage on the S-32. It contains bodies, physical, timing, and simulation models for the following 58 components:

74F00	quad 2-input nand
74F02	quad 2-input nor
74F04	hex inverter
74F08	quad 2-input and
74F10	triple 3-input nand
74F11	triple 3-input and
74F20	dual 4-input nand
74F32	quad 2-input or
74F64	4-2-3-2-input and-or-invert gate
74F74	dual positive-edge-triggered D flip-flop
74F86	quad 2-input exclusive-or
74F109	dual JKbar positive-edge-triggered flip-flop
74F112	dual JK negative-edge-triggered flip-flop
74F138	3-to-8 line decoders/multiplexers
74F139	dual 2-to-4 line decoders/multiplexers
74F148	8-line to 3-line octal priority encoder
74F151	1-of-8 data selectors/multiplexers
74F153	dual 4-line to 1-line data multiplexer
74F157	quad 2-to-1-line non-inverting multiplexer
74F158	quad 2-to-1-line inverting data multiplexer
74F160	4-bit synchronous decade counters with direct clear
74F161	4-bit synchronous binary counters with direct clear
74F162	4-bit synchronous decade counters with synch clear
74F163	4-bit synchronous binary counters with synch clear
74F164	8-bit parallel output serial shift register
74F168	4-bit synchronous decade up/down counters
74F169	4-bit synchronous binary up/down counters
74F174	hex D-type flip-flops
74F175	quad D-type flip-flops
74F181	arithmetic logic units/function generators
74F182	look-ahead carry generators
74F189	64-bit random access memory
74F190	synchronous BCD up/down counter
74F191	synchronous binary up/down counter
74F193	synchronous binary up/down dual clock counters
74F194	4-bit bidirectional shift register
74F219	64-bit random access memory
74F240	octal inverting 3-state bus transceiver
74F241	octal non-inverting 3-state bus transceiver
74F244	octal non-inverting 3-state bus transceiver

74F245	octal non-inverting 3-state bus transceiver
74F251	3-state data multiplexer
74F253	dual data selectors/multiplexers
74F257	quad 3-state non-inverting data multiplexer
74F258	quad 3-state inverting data multiplexer
74F280	9-bit odd/even parity generators/checkers
74F283	4-bit binary full adders
74F299	8-bit bidirectional 3-state shift/storage register
74F373	octal 3-state D-latch w/ common enable
74F374	octal 3-state positive-edge-triggered D register
74F379	quad D-type flip-flops with enable
74F381	arithmetic logic unit/function generator
74F399	quad 2-port register
74F521	8-bit identity comparator
74F524	8-bit registered comparator
74F533	octal transparent latch
74F538	1-of-8 decoder
74F547	octal decoder/demultiplexer

F00 (SIZE)

F02 (SIZE)

F04 (SIZE)

F08 (SIZE)

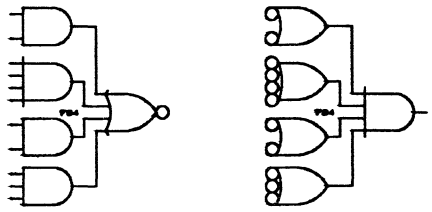
F10 (SIZE)

F11 (SIZE)

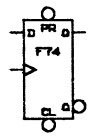
F20 (SIZE)

F32 (SIZE)

F84 (SIZE)



F74 (SIZE)



DEFINE	DRAWING
X_FIRST=0	TITLE=EXAMPLE OF EACH FAST PART
X_STEP=SIZE	ABBREV=EXF
	LAST_MODIFIED=MON JAN 23 09:44:47 1984

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

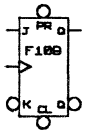
NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

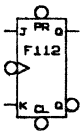
TITLE:	DATE:
EXAMPLE OF EACH FAST PART	7-6-83
ENGINEER: RKM	PAGE: 1 OF 6

F86
(SIZE)

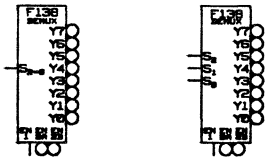
F109
(SIZE)



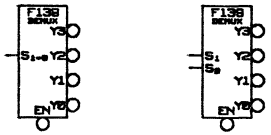
F112
(SIZE)



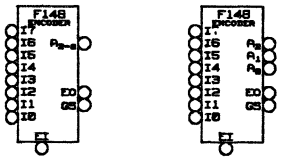
F138



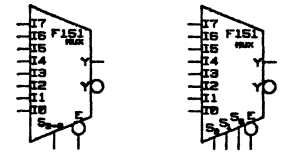
F139



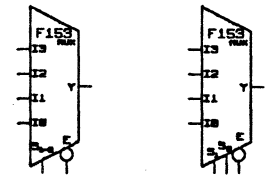
F148



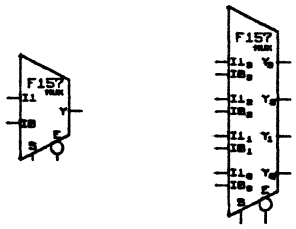
F151
(SIZE)



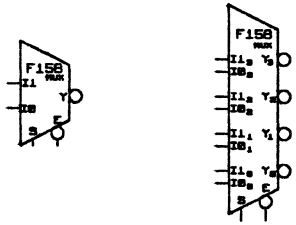
F153
(SIZE)



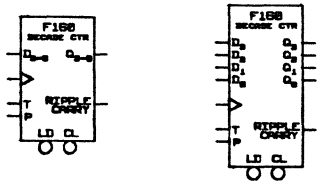
F157
(SIZE)



F158
(SIZE)



F160



DRAWING

TITLE=EXAMPLE OF EACH FAST PART

ABBREV=EXF

LAST_MODIFIED=MON JAN 23 09:45:07 1984

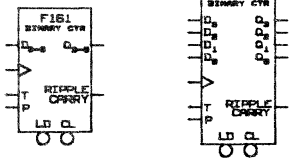
NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

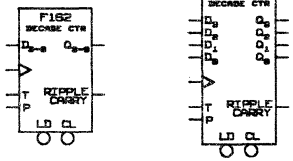
VALID LOGIC SYSTEMS INCORPORATED

TITLE:	DATE:
EXAMPLE OF EACH FAST PART	7-6-83
ENGINEER: RKM	PAGE: 2 OF 6

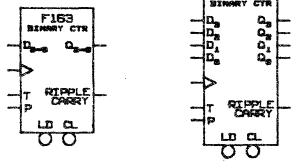
F161



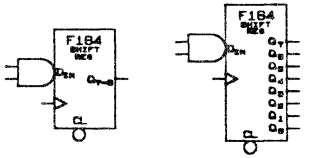
F162



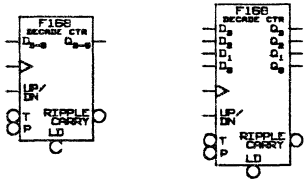
F163



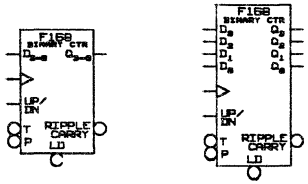
F164



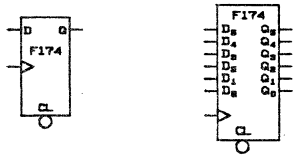
F168



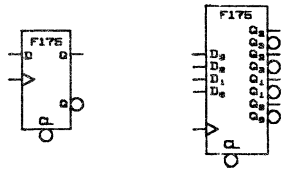
F169



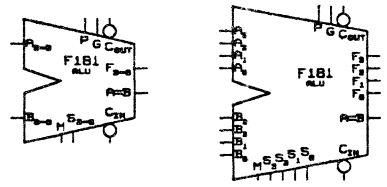
F174 (SIZE)



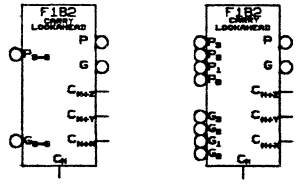
F175 (SIZE)



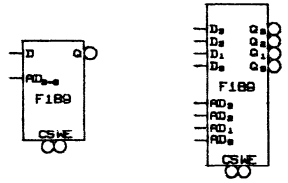
F181



F182



F189 (SIZE)



DRAWING

TITLE=EXAMPLE OF EACH FAST PART

ABBREV=EXF

LAST_MODIFIED=MON JAN 23 09:45:26 1984

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

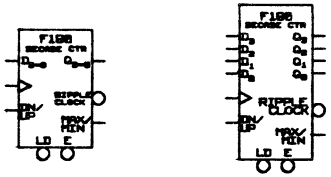
VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH FAST PART

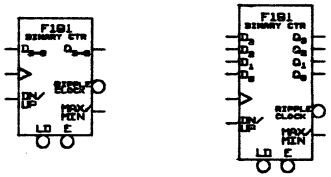
DATE: 7-6-83

ENGINEER: RKM

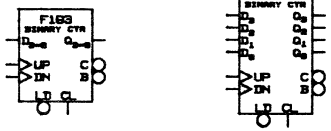
F190



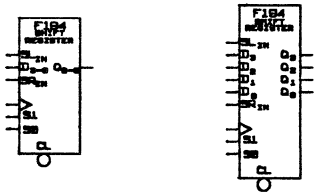
F191



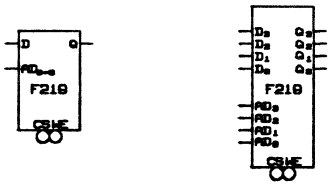
F193



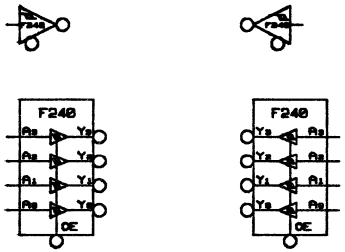
F194



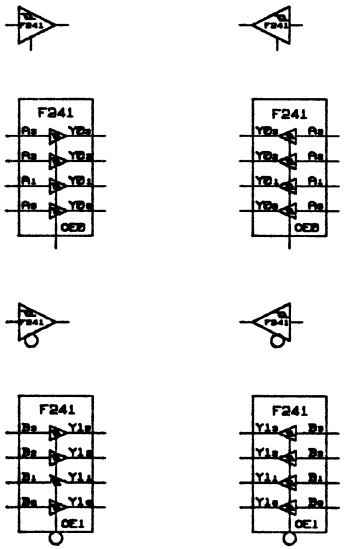
F219 (SIZE)



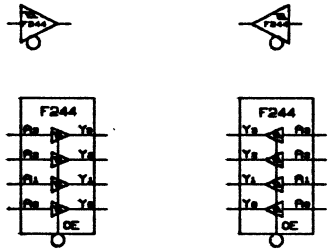
F240 (SIZE)



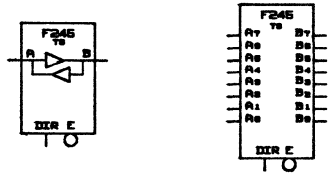
F241 (SIZE)



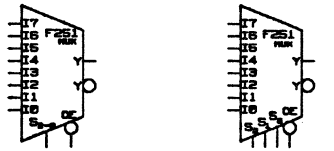
F244 (SIZE)



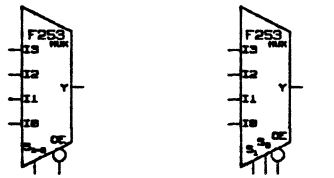
F245 (SIZE)



F251 (SIZE)



F253 (SIZE)



DRAWING

TITLE=EXAMPLE OF EACH FAST PART

ABBREV=EXF

LAST_MODIFIED=MON JAN 23 09:41:29 1984

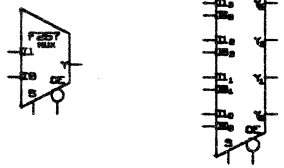
NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

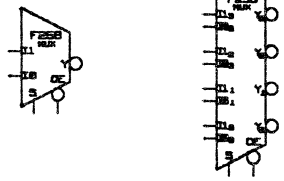
VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH FAST PART	DATE: 7-6-83
ENGINEER: RKM	PAGE: 4 OF 6

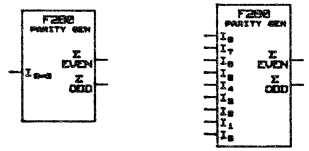
F257
(SIZE)



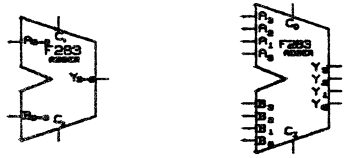
F258
(SIZE)



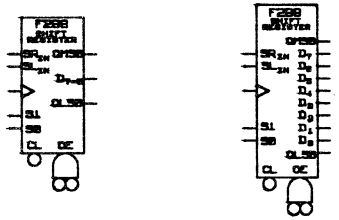
F280



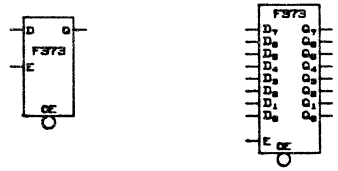
F283



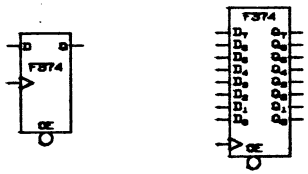
F299



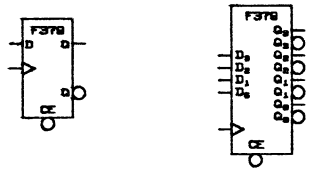
F373
(SIZE)



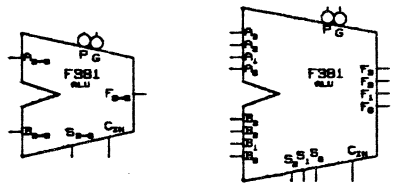
F374
(SIZE)



F379
(SIZE)



F381



DRAWING

TITLE=EXAMPLE OF EACH FAST PART

ABBREV=EXF

LAST_MODIFIED=Tue Mar 20 18:11:06 1984

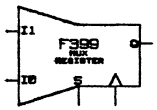
NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

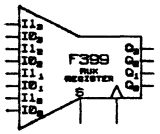
NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

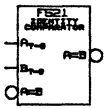
TITLE: EXAMPLE OF EACH FAST PART	DATE: 7-6-83
ENGINEER: RKM	PAGE: 5 OF 6

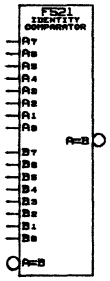
F399
(SIZE)





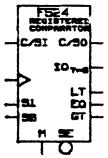
F521



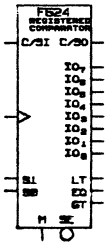


F524

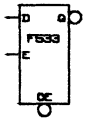
MAX_DELAY=10000NS

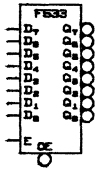


MAX_DELAY=10000NS

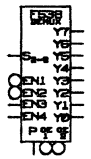


F539
(SIZE)



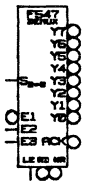


F536

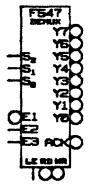


F547

MAX_DELAY=10000NS



MAX_DELAY=10000NS



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID
LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH FAST PART
ABBREV=EXF

ENGINEER:
RRJ

DATE:
Tue Mar 20 18:04:27 1984

PAGE: 6 OF 6

STTL Library

There have been a few changes in the STTL Library for Release 5.1.

The STTL Library requires approximately 1.5 MBy (2887 blocks) of disk storage on the S-32. It contains bodies, physical, timing, and simulation models for the following 59 components:

74S00	quad 2-input nand
74S02	quad 2-input nor
74S03	quad 2-input open-collector nand
74S04	hex inverter
74S05	hex open-collector inverter
74S08	quad 2-input and
74S09	quad 2-input open-collector and
74S10	triple 3-input nand
74S11	triple 3-input and
74S15	triple 3-input open-collector and
74S20	dual 4-input nand
74S22	dual 4-input open-collector nand
74S30	8-input nand
74S32	quad 2-input or
74S37	quad 2-input nand buffer
74S38	quad 2-input open-collector nand buffer
74S40	dual 4-input positive nand buffer
74S51	2-wide 3-input, 2-wide 2-input and-or-invert
74S64	4-2-3-2 input and-or-invert gates
74S74	dual positive-edge-triggered D flip-flop
74S85	4-bit magnitude comparator
74S86	quad 2-input exclusive-or
74S109	dual JKbar positive-edge-triggered flip-flop
74S112	dual JK negative-edge-triggered flip-flop
74S132	quad 2-input positive nand schmitt triggers
74S133	13-input positive nand gates
74S138	3-to-8 line decoders/multiplexers
74S139	dual 2-to-4 line decoders/multiplexers
74S140	dual 4-input positive nand 50-ohm line drivers
74S151	1-of-8 data selectors/multiplexers
74S153	dual 4-line to 1-line data multiplexer
74S157	quad 2-to-1-line non-inverting multiplexer
74S158	quad 2-to-1-line inverting data multiplexer
74S162	4-bit synchronous decade counters with synch clear
74S163	4-bit synchronous binary counters with synch clear
74S169	4-bit synchronous binary up/down counters
74S174	hex D-type flip-flops
74S175	quad D-type flip-flops
74S181	arithmetic logic units/function generators
74S182	look-ahead carry generators

74S189	64-bit random access memories
74S194	4-bit bidirectional shift register
74S201	256-bit random memories
74S225	asynchronous first in first out memories
74S240	octal inverting 3-state bus transceiver
74S241	octal non-inverting 3-state bus transceiver
74S244	octal non-inverting 3-state bus transceiver
74S251	3-state data multiplexer
74S253	dual data selectors/multiplexers
74S257	quad 3-state non-inverting data multiplexer
74S258	quad 3-state inverting data multiplexer
74S260	dual 5-input positive nor gates
74S280	9-bit odd/even parity generators/checkers
74S283	4-bit binary full adders
74S299	8-bit bidirectional 3-state shift/storage register
74S373	octal 3-state D-latch w/ common enable
74S374	octal 3-state positive-edge-triggered D register
74S381	aritmetic logic unit/function generator
74S471	programmable read only memories

S00 (SIZE)

S02 (SIZE)

S03 (SIZE) MAX_DELAY=10000NS MAX_DELAY=10000NS

S04 (SIZE)

S05 (SIZE) MAX_DELAY=10000NS

S08 (SIZE)

S09 (SIZE) MAX_DELAY=10000NS MAX_DELAY=10000NS

S10 (SIZE)

S11 (SIZE)

S15 (SIZE) MAX_DELAY=10000NS MAX_DELAY=10000NS

S20 (SIZE)

S22 (SIZE) MAX_DELAY=10000NS MAX_DELAY=10000NS

S30 (SIZE)

S32 (SIZE)

S37 (SIZE)

S38 (SIZE) MAX_DELAY=10000NS MAX_DELAY=10000NS

S40 (SIZE)

S51 (SIZE)

S64 (SIZE)

S74 (SIZE)

DEFINE	DRAWING
X_FIRST=0	TITLE=EXAMPLE OF EACH STTL PART
X_STEP=SIZE	ABBREV=EXS
	LAST_MODIFIED=Tue Apr 3 10:48:15 1984

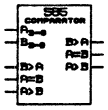
NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

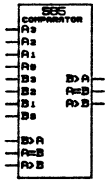
NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

TITLE:	DATE:
EXAMPLE OF EACH STTL PART	6/22/83
ENGINEER: EWB	PAGE: 1 OF 5

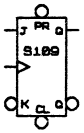
S85



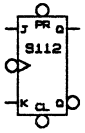


S86
(SIZE)

S109
(SIZE)

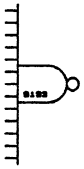


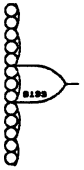
S112
(SIZE)



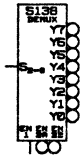
S132
(SIZE)

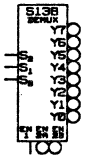
S133
(SIZE)





S138

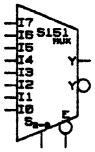


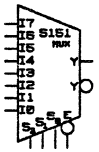


S139

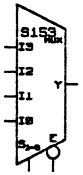
S140
(SIZE)

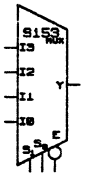
S151
(SIZE)



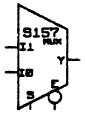


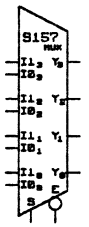
S153
(SIZE)



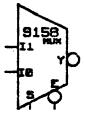


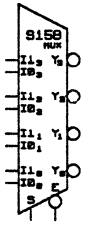
S157
(SIZE)





S158
(SIZE)





DRAWING

TITLE=EXAMPLE OF EACH STTL PART

ABBREV=EXS

LAST_MODIFIED=FRI FEB 3 11:08:20 1984

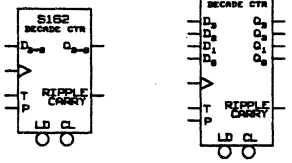
NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

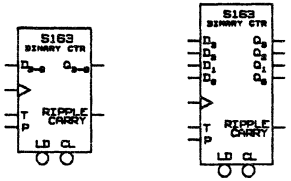
VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH STTL PART	DATE: 6/23/83
ENGINEER: EWB	PAGE: 2 OF 5

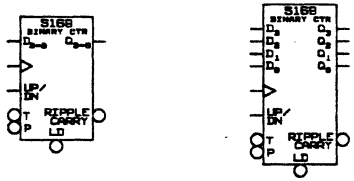
S162



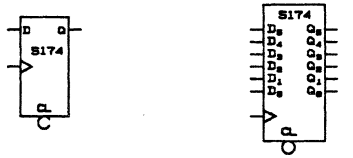
S163



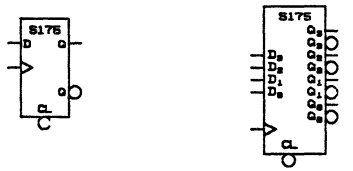
S169



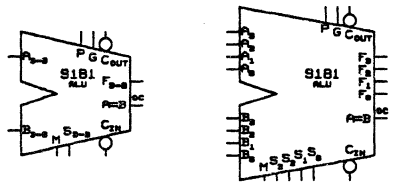
S174 (SIZE)



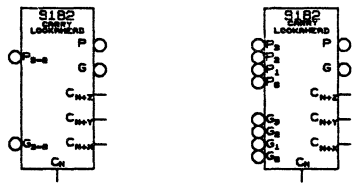
S175 (SIZE)



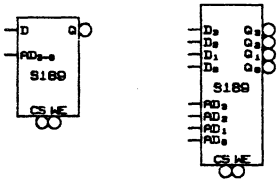
S181



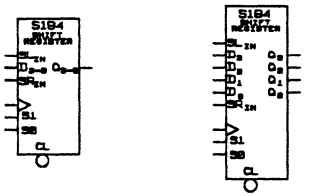
S182



S189



S194



DRAWING

TITLE=EXAMPLE OF EACH STTL PART

ABBREV=EXS

LAST_MODIFIED=FRI FEB 3 10:50:26 1984

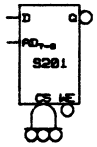
NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

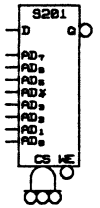
NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

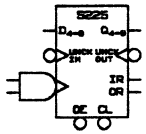
TITLE: EXAMPLE OF EACH STTL PART	DATE: 6/23/83
ENGINEER: EWB	PAGE: 3 OF 5

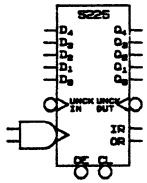
S201
(SIZE)





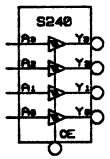
S225

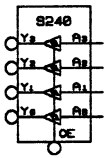




S240
(SIZE)

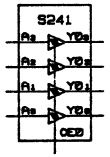
S240

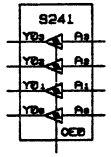




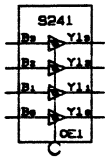
S241
(SIZE)

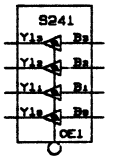
S241



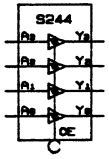


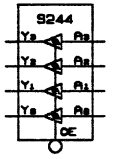
S244
(SIZE)



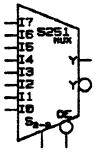


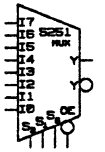
S244



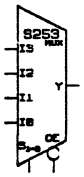


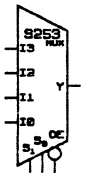
S251
(SIZE)



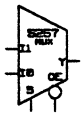


S253
(SIZE)

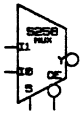




S257
(SIZE)

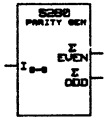


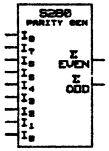
S258
(SIZE)



S260
(SIZE)

S280





DRAWING

TITLE=EXAMPLE OF EACH STTL PART

ABBREV=EXS

LAST_MODIFIED=FRI FEB 3 10:49:37 1984

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

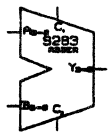
VALID LOGIC SYSTEMS INCORPORATED

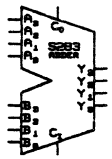
TITLE: EXAMPLE OF EACH STTL PART

DATE: 6/23/83

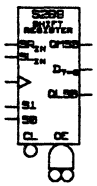
ENGINEER: EWB

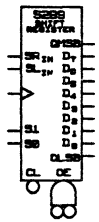
S283



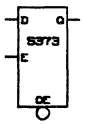


S299

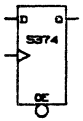


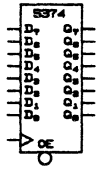


S373
(SIZE)

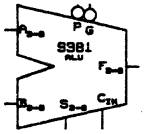


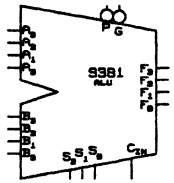
S374
(SIZE)



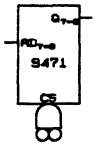


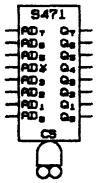
S381





S471





DRAWING

TITLE=EXAMPLE OF EACH STTL PART

ABBREV=EXS

LAST_MODIFIED=FRI FEB 3 10:49:05 1984

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

TITLE:	DATE:
EXAMPLE OF EACH STTL PART	6/23/83
ENGINEER: EWB	PAGE: 5 OF 5

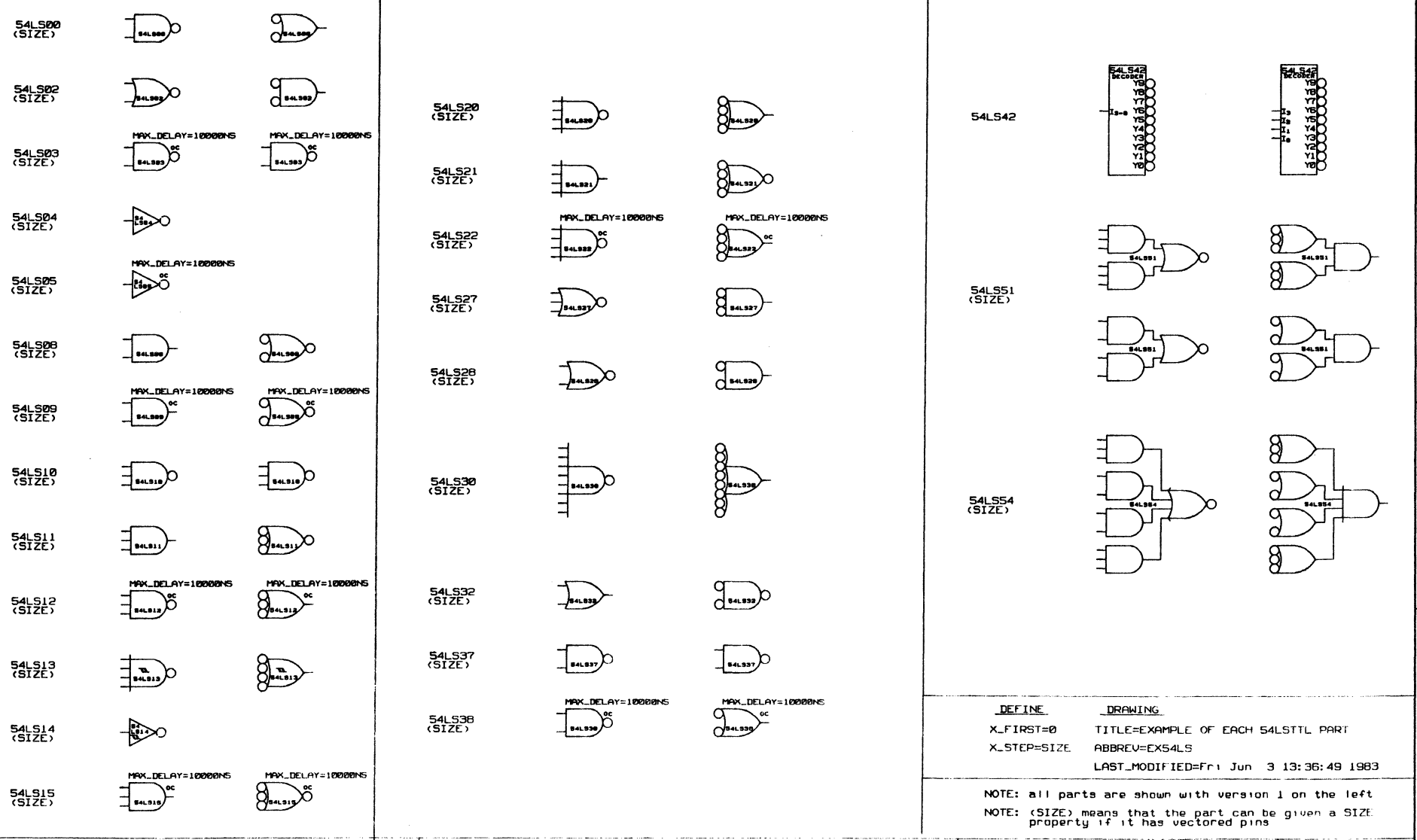
54LSTTL Library

There have been no changes in the 54LSTTL Library since Release 2.3.

The 54LSTTL Library requires approximately 2.0 MBy of disk storage on the S-32. It contains bodies, physical, timing, and simulation models for the following components:

54LS00	quad 2-input nand
54LS02	quad 2-input nor
54LS03	quad 2-input open-collector nand
54LS04	hex inverter
54LS05	hex open-collector inverter
54LS08	quad 2-input and
54LS09	quad 2-input open-collector and
54LS10	triple 3-input nand
54LS11	triple 3-input and
54LS12	triple 3-input open-collector nand
54LS13	dual 4-input nand schmitt trigger
54LS14	hex schmitt-trigger inverter
54LS15	triple 3-input open-collector and
54LS20	dual 4-input nand
54LS21	dual 4-input and
54LS22	dual 4-input open-collector nand
54LS27	triple 3-input nor
54LS28	quad 2-input nor
54LS30	8-input nand
54LS32	quad 2-input or
54LS37	quad 2-input nand buffer
54LS38	quad 2-input open-collector nand buffer
54LS42	4-to-10-line decoder
54LS51	2-wide 3-input, 2-wide 2-input and-or-invert
54LS54	4-wide and-or-invert
54LS74	dual positive-edge-triggered D flip-flop
54LS75	4-bit bistable latch
54LS76	dual JK flip-flop w/ preset & clear
54LS85	4-bit magnitude comparator
54LS86	quad 2-input exclusive-or
54LS109	dual JKbar positive-edge-triggered flip-flop
54LS112	dual JK negative-edge-triggered flip-flop
54LS123	dual retriggerable monostable multivibrators with clear
54LS125	quad bus buffers with three-state outputs
54LS136	quad 2-input exclusive-or
54LS138	3-to-8 line decoders/multiplexers
54LS139	dual 2-to-4 line decoders/multiplexers
54LS148	8-line to 3-line octal priority encoder
54LS151	1-of-8 data selectors/multiplexers
54LS153	dual 4-line to 1-line data multiplexer
54LS157	quad 2-to-1-line non-inverting multiplexer

54LS158	quad 2-to-1-line inverting data multiplexer
54LS160	4-bit synchronous decade counters with direct clear
54LS161	4-bit synchronous binary counters with direct clear
54LS163	4-bit synchronous binary counters with synch clear
54LS164	8-bit parallel output serial shift register
54LS169	4-bit synchronous binary up/down counters
54LS174	hex D-type flip-flops
54LS175	quad D-type flip-flops
54LS190	synchronous BCD up/down counter
54LS191	synchronous binary up/down counter
54LS193	synchronous binary up/down dual clock counters
54LS219	64-bit random access memory
54LS221	dual monostable multivibrators
54LS240	octal inverting 3-state bus transceiver
54LS241	octal non-inverting 3-state bus transceiver
54LS244	octal non-inverting 3-state bus transceiver
54LS251	3-state data multiplexer
54LS253	dual data selectors/multiplexers
54LS257	quad 3-state non-inverting data multiplexer
54LS258	quad 3-state inverting data multiplexer
54LS260	dual 5-input positive nor gates
54LS280	9-bit odd/even parity generators/checkers
54LS283	4-bit binary full adders
54LS299	8-bit bidirectional 3-state shift/storage register
54LS367	hex bus drivers
54LS368	hex bus drivers
54LS373	octal 3-state D-latch w/ common enable
54LS374	octal 3-state positive-edge-triggered D register
54LS640	octal 3-state inverting bus transceiver
54LS641	octal open-collector non-inverting bus transceiver
54LS642	octal open collector inverting bus transceiver
54LS645	octal 3-state non-inverting bus transceiver



DEFINE	DRAWING
X_FIRST=0	TITLE=EXAMPLE OF EACH 54LSTTL PART
X_STEP=SIZE	ABBREV=EX54LS
	LAST_MODIFIED=Fri Jun 3 13:36:49 1983

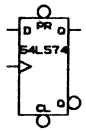
NOTE: all parts are shown with version 1 on the left

NOTE: (SIZE) means that the part can be given a SIZE property if it has vectored pins

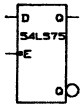
VALID LOGIC SYSTEMS INCORPORATED

TITLE:	DATE:
EXAMPLE OF EACH 54LSTTL PART	04/21/83
ENGINEER: RKM	PAGE: 1 of 5

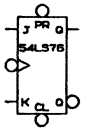
54LS74
(SIZE)



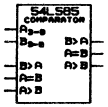
54LS75
(SIZE)

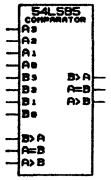


54LS76
(SIZE)



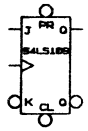
54LS85



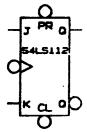


54LS86
(SIZE)

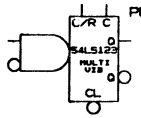
54LS109
(SIZE)



54LS112
(SIZE)

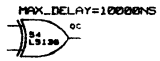


54LS123

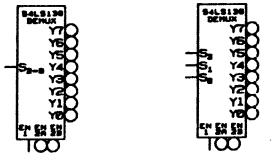


54LS125
(SIZE)

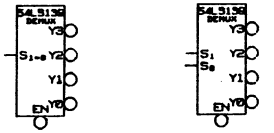
54LS136
(SIZE)



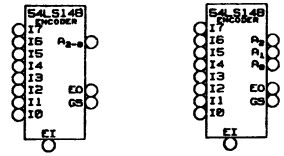
54LS138



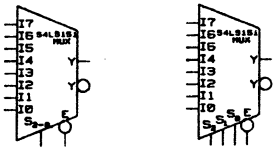
54LS139



54LS148



54LS151
(SIZE)



DRAWING

TITLE=EXAMPLE OF EACH 54LSTTL PART

ABBREV=EX54LS

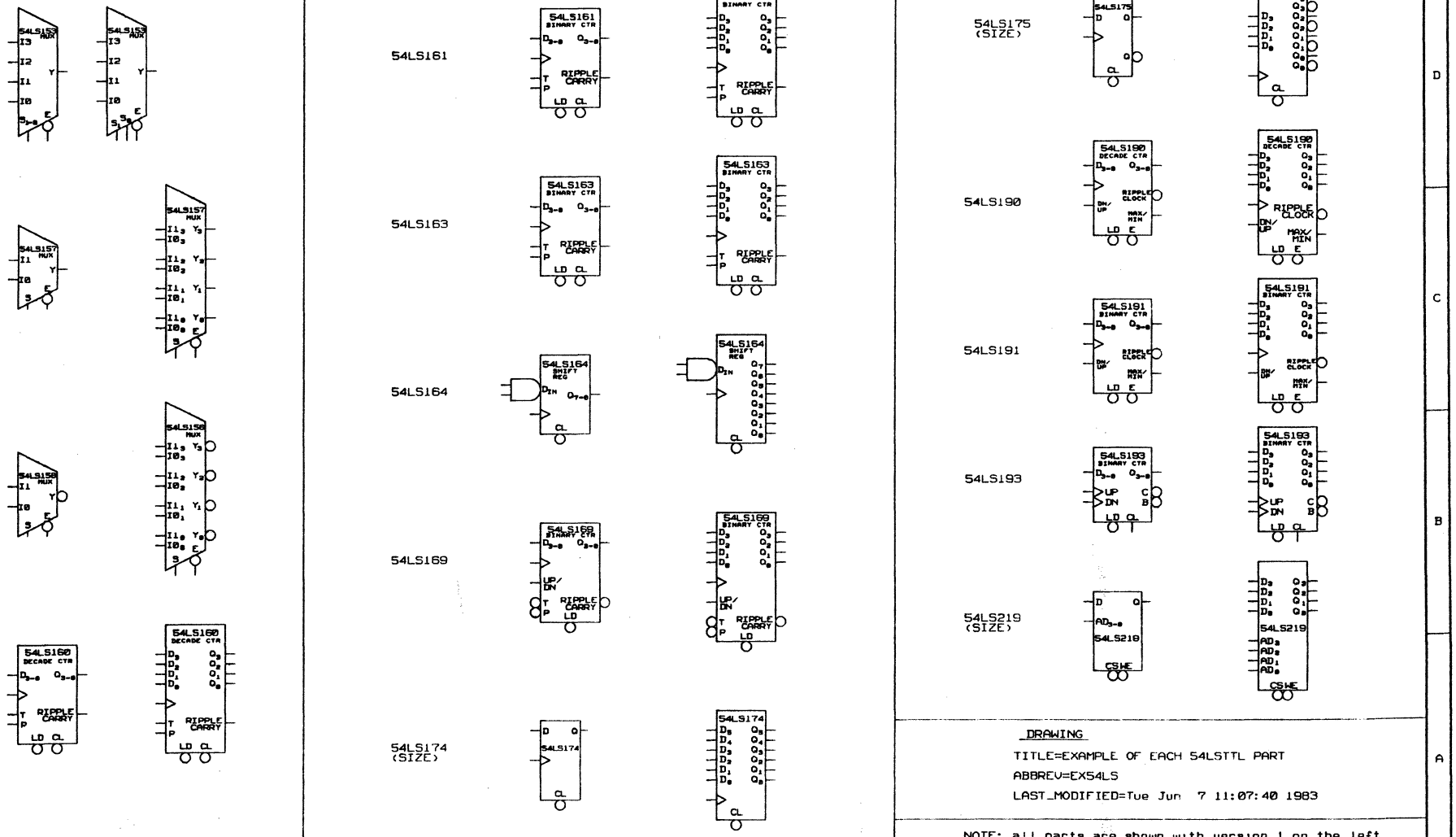
LAST_MODIFIED=Tue Jun 7 11:42:05 1983

NOTE: all parts are shown with version 1 on the left

NOTE: (SIZE) means that the part can be given a SIZE property if it has vectored pins

VALID
LOGIC SYSTEMS INCORPORATED

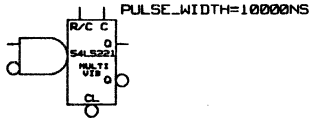
TITLE:	DATE:
EXAMPLE OF EACH 54LSTTL PART	04/21/83
ENGINEER: RKM	PAGE: 2 of 5



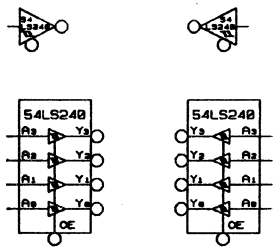
VALID LOGIC SYSTEMS INCORPORATED

TITLE:	DATE:
EXAMPLE OF EACH 54LSTTL PART	04/21/83
ENGINEER: RKM	PAGE: 3 of 5

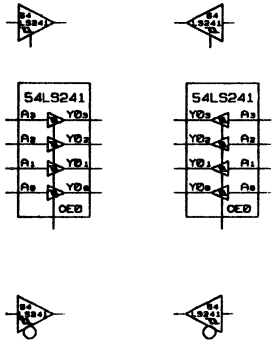
54LS221



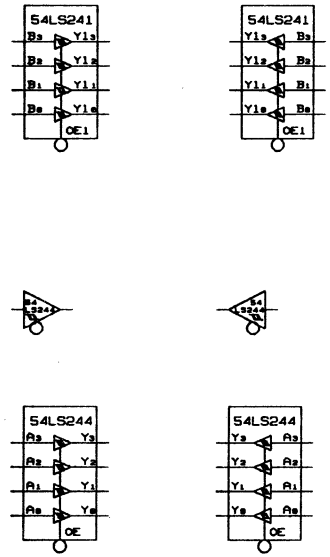
54LS240 (SIZE)



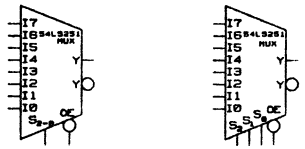
54LS241 (SIZE)



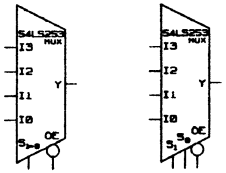
54LS244 (SIZE)



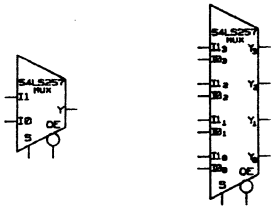
54LS251 (SIZE)



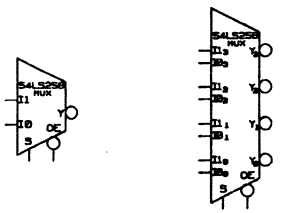
54LS253 (SIZE)



54LS257 (SIZE)



54LS258 (SIZE)



DRAWING

TITLE=EXAMPLE OF EACH 54LSTTL PART

ABBREV=EX54LS

LAST_MODIFIED=Wed Jun 8 15:30:09 1983

NOTE: all parts are shown with version 1 on the left

NOTE: (SIZE) means that the part can be given a SIZE property if it has vectored pins

VALID LOGIC SYSTEMS INCORPORATED

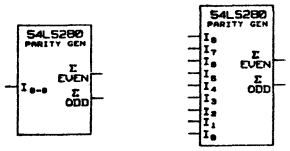
TITLE: EXAMPLE OF EACH 54LSTTL PART

DATE: 04/21/83

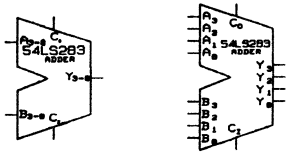
ENGINEER: RKM

54LS260
(SIZE)

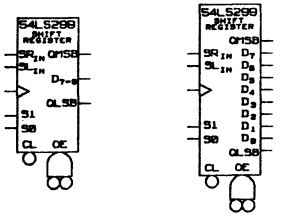
54LS280



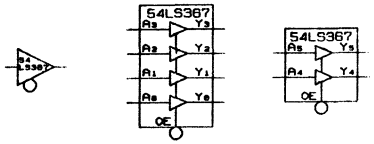
54LS283



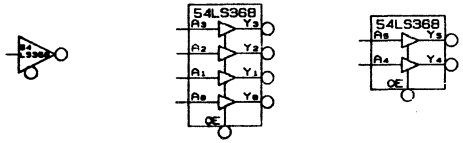
54LS299



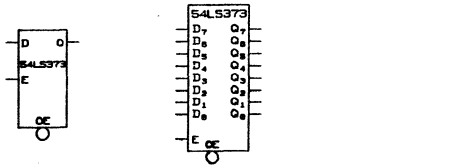
54LS367
(SIZE)



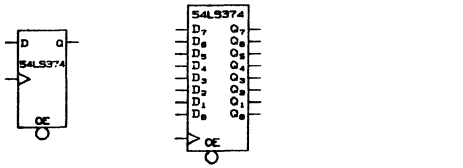
54LS368
(SIZE)



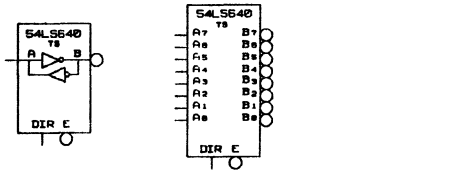
54LS373
(SIZE)



54LS374
(SIZE)

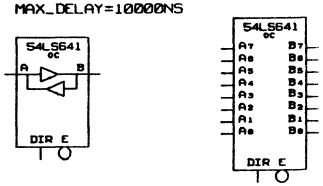


54LS640
(SIZE)



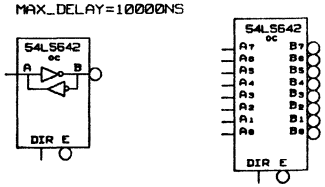
MAX_DELAY=10000NS

54LS641
(SIZE)



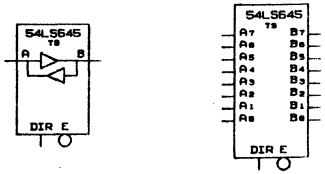
MAX_DELAY=10000NS

54LS642



MAX_DELAY=10000NS

54LS645
(SIZE)



DRAWING

TITLE=EXAMPLE OF EACH 54LSTTL PART

ABBREV=EXLS

LAST_MODIFIED=Fri Jun 3 13:55:01 1983

NOTE: all parts are shown with version 1 on the left

NOTE: (SIZE) means that the part can be given a SIZE property if it has vectored pins

VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH 54LSTTL PART	DATE: 04/21/83
ENGINEER: RKM	PAGE: 5 of 5

LSI5KC Library

There has been no change in the LSI5KC Library since Release 4.0.

The LSI5KC Library requires approximately 0.6 MBy of disk storage on the S-32. It contains bodies and physical for the following 111 components from the LSI Logic 5000 Series CMOS Gate Array Macrocells::

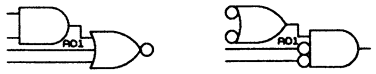
A01	2and into 3nor
A02	2 2and into 2nor
A03	2or into 3nand
A04	2 2or into 2nand
A05	inverting 2 of 3 majority
BTS1	3state output buffer
BTS14	3state output buffer
BTS18	3state output buffer
BTS2	3state output buffer
BTS3	3state output buffer
BTS4	3state internal buffer
BTS5	inverting 3state internal buffer
BTS6	3state output buffer to CMOS
BTS7	3state I/O buffer
BTS7D	3state I/O buffer with pull down
BTS7L	3state I/O buffer with low power
BTS7LO	3state I/O buffer with low power, open drain
BTS7OD	3state I/O buffer open drain
BTS7U	3state I/O buffer with pull up
BTS78	3state I/O buffer
BTS8	3state I/O buffer
BTS8U	3state I/O buffer with pull up
BTS9	3state I/O buffer
BTS9D	3state I/O buffer with pull down
BTS9U	3state I/O buffer with pull up
B1	output buffer
B1I	internal buffer (like B1)
B1OD	output buffer with open drain
B14	output buffer
B18	output buffer
B2	output buffer
B2I	INV into 3//INV
B2OD	output buffer with open drain
B3	output buffer
B3I	2//INV into 2//INV
B3OD	output buffer with open drain
B4I	4//INV
B5I	3//INV
EN	exclusive 2nor
EO	exclusive 2or

EON1	2or, 2nand into 2nand
EO1	2and, 2nor into 2nor
FDA	DFF CDM CDS SDM SDS CPM CPS
FD1	DFF
FD1S	DFF SCAN
FD2	DFF CD
FD2S	DFF CD SCAN
FD2TS	DFF CD tristate output
FD3	DFF CD SD
FD3S	DFF CD SD SCAN
FD4	DFF SD
FD4S	DFF SD SCAN
FD5	DFF NO CPBUFS
FD5S	DFF SCAN NO CPBUFS
FD6	DFF CD NO CPBUFS
FD6S	DFF CD SCAN NO CPBUFS
FD7	DFF CD SD NO CPBUFS
FD7S	DFF CD SD SCAN NO CPBUFS
FD8	DFF SD NO CPBUFS
FD8S	DFF SD SCAN NO CPBUFS
FJKA	JKFF CDM CDS SDM SDS CPM CPS
FJK1	JKFF
FJK1S	JKFF SCAN
FJK2	JKFF CD
FJK2S	JKFF CD SCAN
FJK3	JKFF CD SD
FJK3S	JKFF CD SD SCAN
FT2	TFF CD NO CPBUFS
FT3	TFF CD SD NO CPBUFS
FT4	TFF SD NO CPBUFS
IBUF	input pad with buffer for CMOS input
IBUFD	input pad with pulldown and buffer for CMOS input
IBUFI	buffer for bidirectional CMOS input
IBUFU	input pad with pullup and buffer for CMOS input
IV	single inverter
IVA	single inverter with parallel p transistor
IVDA	tandem inverter pair
IVP	power inverter (2 // INV)
LD1	D-latch gated
LD2	D-latch gated low
LSR1	SR-latch separate gate
LSR2	SR-latch common gate
LS1	D-latch LSSD
LS2	D-latch into D-latch LSSD
MUX8	8 to 1 multiplexer
ND2	2nand
ND3	3nand
ND4	4nand
ND6	6nand
ND8	8nand

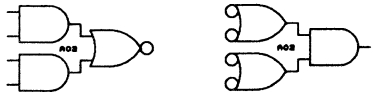
NR2	2nor
NR3	3nor
NR4	4nor
NR6	6nor
NR8	8nor
OSC2	complete oscillator with X-tal connections
RAM1	latch with tristate output
SCHMDT1	input pad with schmitt trigger
SCHMDT2	input pad with inverting schmitt trigger
ST	inverting schmitt trigger for intra chip wave shaping
ST1	schmitt trigger for intra chip wave shaping
TLCHT	input pad with buffer for TTL input
TLCHTI	buffer for bidirectional TTL input

GATES

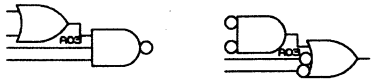
AO1 (SIZE)



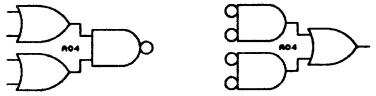
AO2 (SIZE)



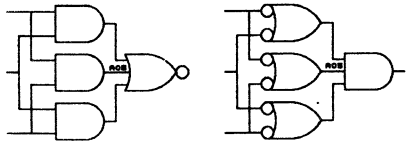
AO3 (SIZE)



AO4 (SIZE)



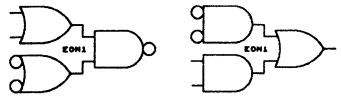
AO5 (SIZE)



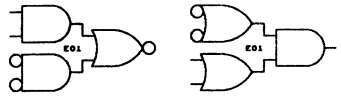
EN (SIZE)

EO (SIZE)

EON1 (SIZE)



EO1 (SIZE)



IV (SIZE)

ND2 (SIZE)

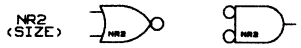
ND3 (SIZE)

ND4 (SIZE)

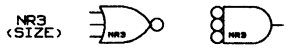
ND6 (SIZE)

ND8 (SIZE)

NR2 (SIZE)

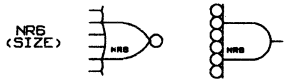


NR3 (SIZE)

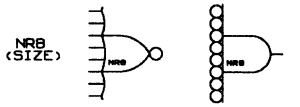


NR4 (SIZE)

NR6 (SIZE)

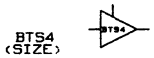


NR8 (SIZE)

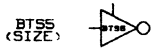


INTERNAL BUFFERS

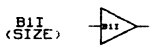
BTS4 (SIZE)



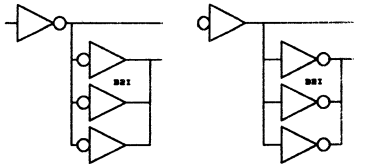
BTS5 (SIZE)



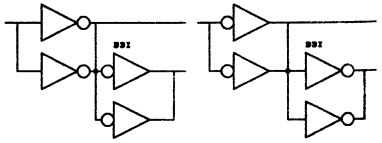
B1I (SIZE)



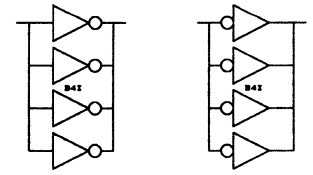
B2I (SIZE)



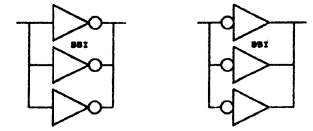
B3I (SIZE)



B4I (SIZE)

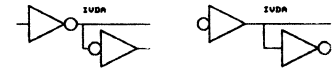


B5I (SIZE)



IVA (SIZE)

IVDA (SIZE)



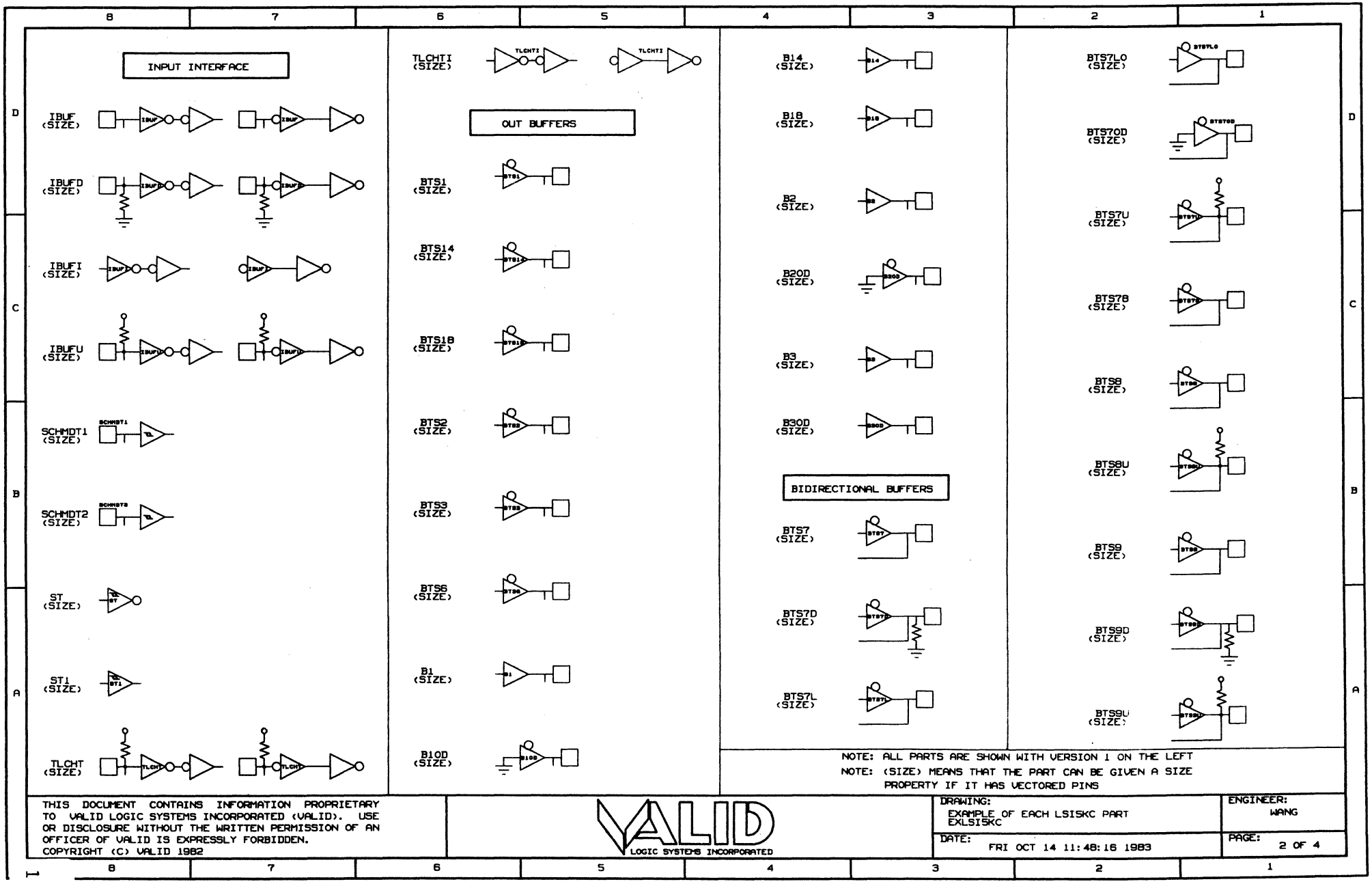
IVP (SIZE)

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT
NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

DEFINE
X_FIRST=0
X_STEP=SIZE

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH LSI5KC PART EXLSI5KC
ENGINEER: WANG
DATE: FRI OCT 14 11:43:04 1983
PAGE: 1 OF 4



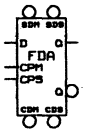
NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT
NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

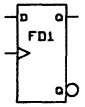
DRAWING: EXAMPLE OF EACH LSISKC PART EXLSI5KC
ENGINEER: WANG
DATE: FRI OCT 14 11:48:16 1983

FLIP FLOPS

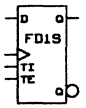
FDA (SIZE)



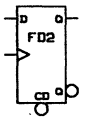
FD1 (SIZE)



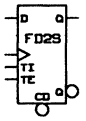
FD1S (SIZE)



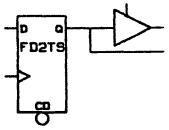
FD2 (SIZE)



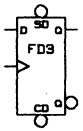
FD2S (SIZE)



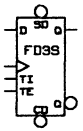
FD2TS (SIZE)



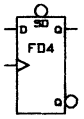
FD3 (SIZE)



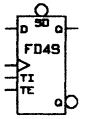
FD3S (SIZE)



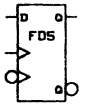
FD4 (SIZE)



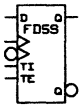
FD4S (SIZE)



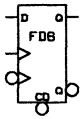
FD5 (SIZE)



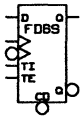
FD5S (SIZE)



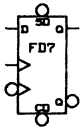
FD6 (SIZE)



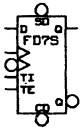
FD6S (SIZE)



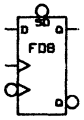
FD7 (SIZE)



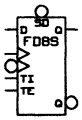
FD7S (SIZE)



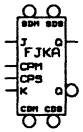
FD8 (SIZE)



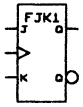
FD8S (SIZE)



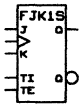
FJKA (SIZE)



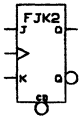
FJK1 (SIZE)



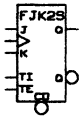
FJK1S (SIZE)



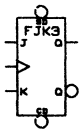
FJK2 (SIZE)



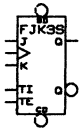
FJK2S (SIZE)



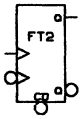
FJK3 (SIZE)



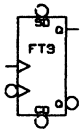
FJK3S (SIZE)



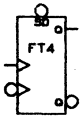
FT2 (SIZE)



FT3 (SIZE)



FT4 (SIZE)



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH LSI5KC PART EXLSI5KC

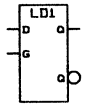
ENGINEER: WANG

DATE: FRI OCT 14 11:52:34 1983

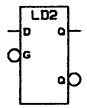
PAGE: 3 OF 4

LATCHES

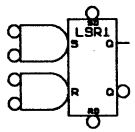
LD1 (SIZE)



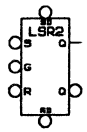
LD2 (SIZE)



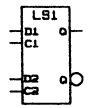
LSR1 (SIZE)



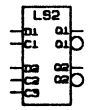
LSR2 (SIZE)



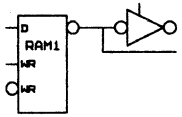
LS1 (SIZE)



LS2 (SIZE)

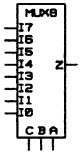


RAM1 (SIZE)



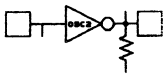
MULTIPLEXERS

MUX8 (SIZE)



OSCILATORS

OSC2



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT
NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH LSI5KC PART EXLSI5KC

ENGINEER: WANG

DATE: FRI OCT 14 11:54:08 1983

LSI5KF Library

There has been no change in the LSI5KF Library since Release 4.0.

The LSI5KF Library requires approximately 1.5 MBy of disk storage on the S-32. It contains bodies and physical for the following 178 components from the LSI Logic 5000 Series CMOS Gate Array Macrofunctions:

CB4C	4 bit binary counter, fast, sync clr
CB4F	4 bit binary counter, fast, indiv CD SD
CB5C	5 bit binary up counter, sync clr
CB5F	5 bit binary up counter, fast, indiv CD SD
CB6C	6 bit MOD 64 fast binary counter, sync clr
CB6F	6 bit MOD 64 fast binary counter, indiv CD SD
CB7C	7 bit MOD 128 fast binary counter, sync clr
CB7F	7 bit MOD 128 fast binary counter, indiv CD SD
CB8C	8 bit MOD 256 fast binary counter, sync clr
CB8F	8 bit MOD 256 fast binary counter, indiv CD SD
CB9C	9 bit MOD 512 fast binary counter, sync clr
CB9F	9 bit MOD 512 fast binary counter, indiv CD SD
CB10C	10 bit MOD 1024 fast binary counter, sync clr
CB10F	10 bit MOD 1024 fast binary counter, indiv CD SD
CB41	4 bit binary counter, expandable enable async clr
CB42	4 bit binary counter, expandable enable sync clr
CMP4	4 bit equality comparator
CMP8	8 bit equality comparator
CM3B	2 bit binary counter
CM4B	2 bit binary counter
CM4J	2 bit Johnson counter (same as C2G)
CM5B	3 bit binary counter
CM5SR	3 bit shift counter
CM6B	3 bit binary counter
CM6J	3 bit Johnson counter
CM7B	3 bit binary counter
CM8B	3 bit binary counter
CM8BR	3 bit binary ripple counter
CM8J	4 bit Johnson counter
CM8SR	3 bit shift counter
CM9B	4 bit binary counter
CM9BR	3 bit binary ripple counter
CM9SR	4 bit shift counter
CM10B	4 bit binary counter
CM10BR	4 bit binary ripple counter
CM10J	5 bit Johnson counter
CM10SR	4 bit shift counter
CM11B	4 bit binary counter
CM11BR	4 bit binary ripple counter
CM12B	4 bit binary counter

CM12BR	4 bit binary ripple counter
CM12J	6 bit Johnson counter
CM12SR	4 bit shift counter
CM13B	4 bit binary counter
CM13BR	4 bit binary ripple counter
CM14B	4 bit binary counter
CM14BR	4 bit binary ripple counter
CM14J	7 bit Johnson counter
CM15B	4 bit binary counter
CM15BR	4 bit binary ripple counter
CM16B	4 bit binary counter
CM16BR	4 bit binary ripple counter
CM16J	8 bit Johnson counter
CM17B	5 bit binary counter
CM17BR	5 bit binary ripple counter
CM18BR	5 bit binary ripple counter
CM19BR	5 bit binary ripple counter
CM20BR	5 bit binary ripple counter
CM21BR	5 bit binary ripple counter
CM22BR	5 bit binary ripple counter
CM23BR	5 bit binary ripple counter
CM24BR	5 bit binary ripple counter
CM25BR	5 bit binary ripple counter
CM26BR	5 bit binary ripple counter
CM27BR	5 bit binary ripple counter
CM28BR	5 bit binary ripple counter
CM29BR	5 bit binary ripple counter
CM30BR	5 bit binary ripple counter
CM31BR	5 bit binary ripple counter
CM32BR	5 bit binary ripple counter
CPG1	two phase clock gen, unbuffered, hi underlap lo drive
CPG2	two phase clock gen, buffered, lo underlap lo drive
CPG3	two phase clock gen, unbuffered, hi underlap hi drive
CPG4	two phase clock gen, buffered, lo underlap hi drive
CUD41	4 bit U/D counter, expandable, async clr
CUD42	4 bit U/D counter, expandable, async load & clr
C2G	2 bit Gray counter (same as CM4J)
C3G	3 bit Gray counter
C3LSR	3 bit linear feedback shift register
C4LSR	4 bit linear feedback shift register
C5LSR	5 bit linear feedback shift register
C6LSR	6 bit MOD 63 linear feedback shift register
C7LSR	7 bit MOD 127 linear feedback shift register
C8LSR	8 bit MOD 255 linear feedback shift register
C9LSR	9 bit MOD 511 linear feedback shift register
C10LSR	10 bit MOD 1023 linear feedback shift register
C11LSR	11 bit MOD 2047 linear feedback shift register
C12LSR	12 bit MOD 4095 linear feedback shift register

C13LSR	13 bit MOD 8191 linear feedback shift register
C14LSR	14 bit MOD 16383 linear feedback shift register
C15LSR	15 bit MOD 32767 linear feedback shift register
C16LSR	16 bit MOD 65535 linear feedback shift register
C17LSR	17 bit MOD 131071 linear feedback shift register
C18LSR	18 bit MOD 262143 linear feedback shift register
C19LSR	19 bit MOD 524287 linear feedback shift register
C20LSR	20 bit MOD 1048575 linear feedback shift register
DM6JH	spike free decoder for MOD 6 Johnson counter, active hi
DM6JL	spike free decoder for MOD 6 Johnson counter, active lo
DM8JH	spike free decoder for MOD 8 Johnson counter, active hi
DM8JL	spike free decoder for MOD 8 Johnson counter, active lo
D24GH	2 to 4 decoder, gated outputs active hi
D24GL	2 to 4 decoder, gated outputs active lo
D24H	2 to 4 decoder, outputs active hi
D24L	2 to 4 decoder, outputs active lo
D38GH	3 to 8 decoder, gated outputs active hi
D38GL	3 to 8 decoder, gated outputs active lo
D38H	3 to 8 decoder, outputs active hi
D38L	3 to 8 decoder, outputs active lo
D410H	4 to 10 decoder, outputs active hi
D410L	4 to 10 decoder, outputs active lo
FA1	full adder
FA2	2 bit binary adder (7482)
FAS2	2 bit binary adder subtracter, A+B, A-B
HA1	half adder
L4	4 bit data latch
L8	8 bit data latch
MAG2	2 bit magnitude comparator, expandable (1/2 7485)
MAG2H	2 bit magnitude comparator
MAG4	4 bit magnitude comparator, exapndable (7485)
MR41	4 bit register with 2 bit muxed inputs
MR42	4 bit register with 2 bit muxed inputs
MR43	4 bit register with 2 bit muxed inputs, sync clr
MR44	4 bit register with 2 bit muxed inputs, sync clr
MUX22H	dual 2 bit non inverting mux
MUX24H	quad 2 bit non inverting mux
MUX24L	quad 2 bit inverting mux
MUX31H	3 bit non inverting mux
MUX31L	3 bit inverting mux
MUX32H	dual 3 bit non inverting mux
MUX41GH	4 bit non inverting mux, gated
MUX41H	4 bit non inverting mux
MUX41L	4 bit inverting mux
MUX42H	dual 4 bit non inverting mux
MUX51H	5 bit non inverting mux
MUX51L	5 bit inverting mux
MUX61H	6 bit non inverting mux

MUX61L	6 bit inverting mux
MUX71H	7 bit non inverting mux
MUX71L	7 bit inverting mux
M42C	bcd to decimal decoder (7442)
M82C	2 bit binary full adder (7482)
M85C	4 bit magnitude comparator (7485)
M138C	gated 3 to 8 binary decoder (74138)
M138D	3 to 8 decoder, gated outputs, active lo (74138)
M150C	gated 16 input mux (74150)
M151C	gated 8 input mux (74LS151)
M152C	8 input mux (74LS152)
M153C	gated dual 4 input mux (74LS153)
M157C	4 x 2 mux (74LS157)
M158C	4 x 2 mux, outputs active lo (74LS158)
M160C	4 bit bcd counter (74LS160)
M160D	4 bit bcd counter (74LS160)
M161C	4 bit counter (74LS161)
M161D	4 bit binary counter (74LS161)
M162C	4 bit bcd counter (74LS162)
M162D	4 bit bcd counter (74LS162)
M163C	4 bit counter (74LS163)
M163D	4 bit binary counter (74LS163)
M163F	4 bit counter fast (74LS163)
M169C	4 bit U/D counter (74LS169)
M244C	octal tristate buffer on chip (SN74244)
M244XC	octal tristate buffer off chip (SN74244)
PAR8	8 bit odd parity detector
PAR9	9 bit odd parity detector
PS2	2 bit external clock prescaler
PS3	3 bit external clock prescaler
PS4	4 bit external clock prescaler
R41	4 bit data register
R42	4 bit data register
R81	8 bit data register
R82	8 bit data register
SR41	4 bit shift register
SR42	4 bit shift register
SR43	4 bit shift register
SR44	4 bit shift register, sync parallel load
SR45	4 bit shift register, sync parallel load and clear
SR46	4 bit shift register, async parallel load
SR47	4 bit shift register, sync clear
SYNC01	synchronizer for async 0 to 1 event
SYNC10	synchronizer for async 1 to 0 event

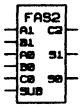
ADDERS

HA1 (SIZE)

FA1 (SIZE)

FA2 (SIZE)

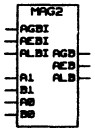
FAS2 (SIZE)



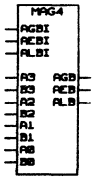
COMPARATORS

MAG2H

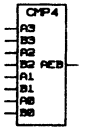
MAG2



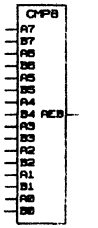
MAG4



CMP4

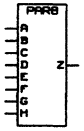


CMP8

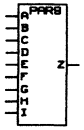


PARITY GENERATORS

PAR8

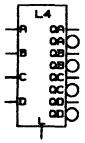


PAR9



LATCH REGISTERS

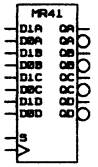
L4 (SIZE)



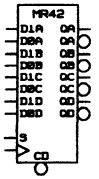
LB (SIZE)

MULTIPLEX REGISTERS

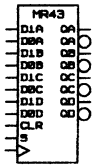
MR41 (SIZE)



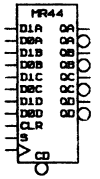
MR42 (SIZE)



MR43 (SIZE)

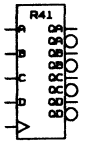


MR44 (SIZE)

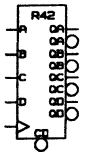


FLIPFLOP REGISTERS

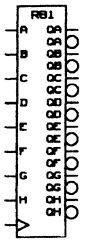
R41 (SIZE)



R42 (SIZE)



RB1 (SIZE)



DEFINE

X_FIRST=0

X_STEP=SIZE

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

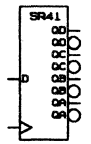
DRAWING: EXAMPLE OF EACH LSI5KF PART EXL55KF

ENGINEER: WANG

DATE: Sat Oct 15 16:09:40 1983

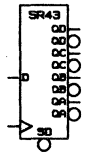
SHIFT REGISTERS

SR41

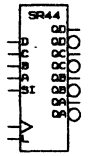


SR42

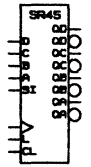
SR43



SR44



SR45



SR46

MODULO 2 COUNTERS

PS2 (SIZE)

MODULO 3 COUNTERS

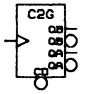
CM3B

PS3 (SIZE)

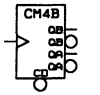
MODULO 4 COUNTERS

CM4J

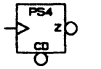
C2G



CM4B

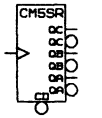


PS4 (SIZE)

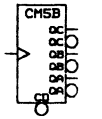


MODULO 5 COUNTERS

CM5SR

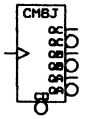


CM5B

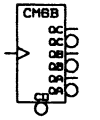


MODULO 6 COUNTERS

CM6J

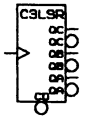


CM6B

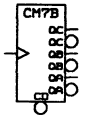


MODULO 7 COUNTERS

C3LSR

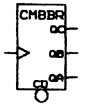


CM7B

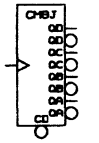


MODULO 8 COUNTERS

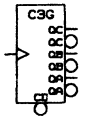
CM8BR



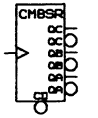
CM8J



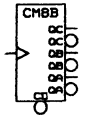
C3G



CM8SR



CM8B



MODULO 9 COUNTERS

CM9BR

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

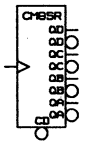
DRAWING: EXAMPLE OF EACH LSISKF PART EXLSISKF

ENGINEER: WANG

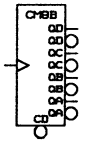
DATE: Sat Oct 15 16:11:17 1983

MODULO 9 COUNTERS (CONT'D)

CM9SR



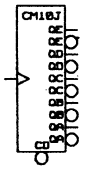
CM9B



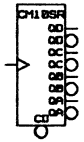
MODULO 10 COUNTERS

CM10BR

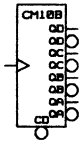
CM10J



CM10SR

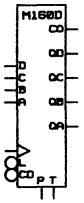


CM10B



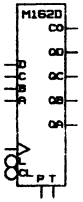
M160C

M160D



M162C

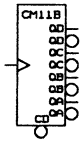
M162D



MODULO 11 COUNTERS

CM11BR

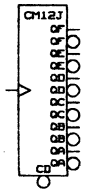
CM11B



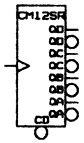
MODULO 12 COUNTERS

CM12BR

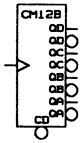
CM12J



CM12SR



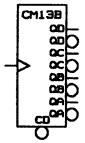
CM12B



MODULO 13 COUNTERS

CM13BR

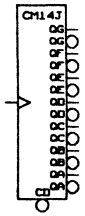
CM13B



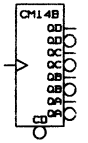
MODULO 14 COUNTERS

CM14BR

CM14J



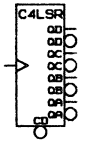
CM14B



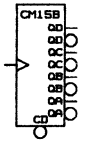
MODULO 15 COUNTERS

CM15BR

C4LSR

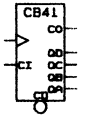


CM15B



MODULO 16 COUNTERS

CB41



NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

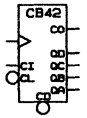
DRAWING: EXAMPLE OF EACH LSI5KF PART EXLSI5KF

ENGINEER: WANG

DATE: Sat Oct 15 16:12:45 1983

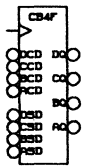
MODULO 16 COUNTERS (CONT'D)

CB42



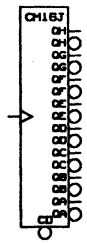
CB4C

CB4F

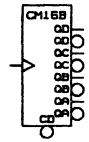


CM16BR

CM16J

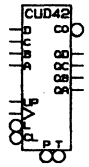


CM16B

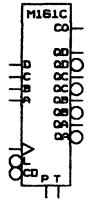


CUD41

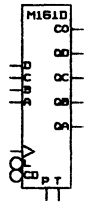
CUD42



M161C

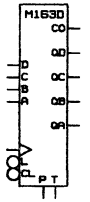


M161D

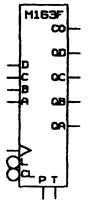


M163C

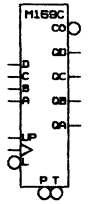
M163D



M163F

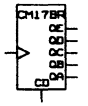


M169C

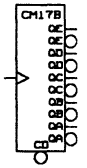


MODULO 17 COUNTERS

CM17BR

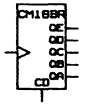


CM17B



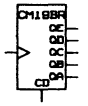
MODULO 18 COUNTERS

CM18BR



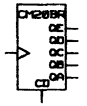
MODULO 19 COUNTERS

CM19BR



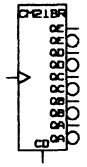
MODULO 20 COUNTERS

CM20BR



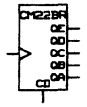
MODULO 21 COUNTERS

CM21BR



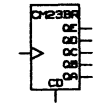
MODULO 22 COUNTERS

CM22BR



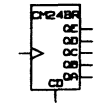
MODULO 23 COUNTERS

CM23BR



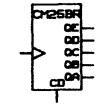
MODULO 24 COUNTERS

CM24BR



MODULO 25 COUNTERS

CM25BR



NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH LSISKF PART EXLSISKF

ENGINEER: WANG

DATE: Sat Oct 15 16:14:29 1983

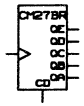
PAGE: 4 OF 9

MODULO 26 COUNTERS

CM26BR

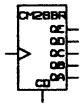
MODULO 27 COUNTERS

CM27BR



MODULO 28 COUNTERS

CM28BR

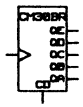


MODULO 29 COUNTERS

CM29BR

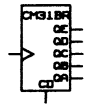
MODULO 30 COUNTERS

CM30BR

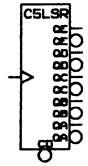


MODULO 31 COUNTERS

CM31BR

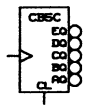


C5LSR

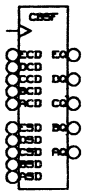


MODULO 32 COUNTERS

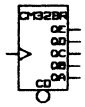
CB5C



CB5F

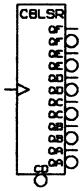


CM32BR

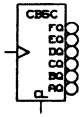


LARGE MODULO COUNTERS

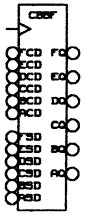
C8LSR



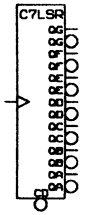
CB8C



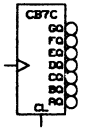
CB8F



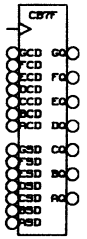
C7LSR



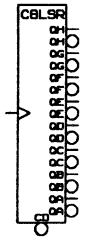
CB7C



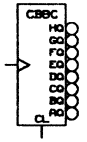
CB7F



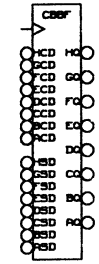
C8LSR



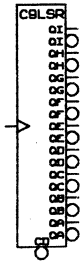
CB8C



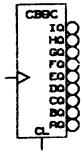
CB8F



C9LSR



CB9C



NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH LSI5KF PART EXLSI5KF

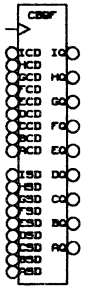
ENGINEER: WANG

DATE: Sat Oct 15 16:16:22 1983

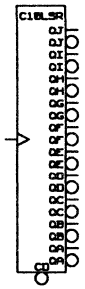
PAGE: 5 OF 9

LARGE MODULO COUNTERS (CONT'D)

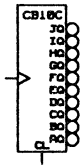
CB9F



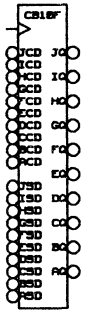
C10LSR



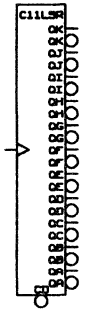
CB10C



CB10F

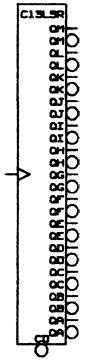


C11LSR

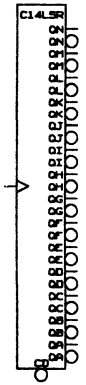


C12LSR

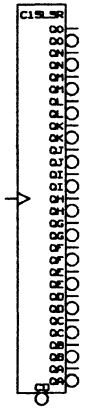
C13LSR



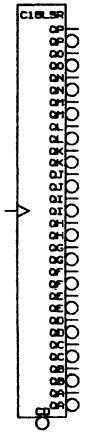
C14LSR



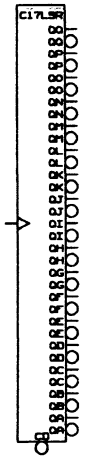
C15LSR



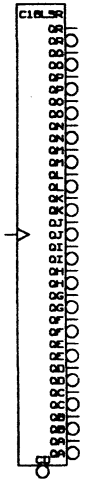
C16LSR



C17LSR



C18LSR



NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

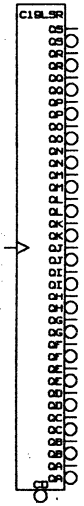
DRAWING: EXAMPLE OF EACH LSISKF PART EXLSISKF

ENGINEER: WANG

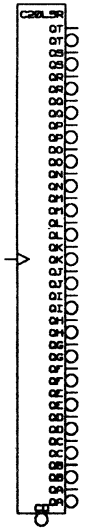
DATE: Sat Oct 15 16:17:34 1983

LARGE MODULO COUNTERS (CONT'D)

C19LSR



C20LSR



NON OVERLAPPING CLOCK GENERATORS

CPG1 (SIZE)

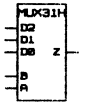
CPG2 (SIZE)

CPG3 (SIZE)

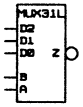
CPG4 (SIZE)

MULTIPLEXERS

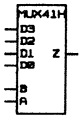
MUX31H (SIZE)



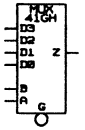
MUX31L (SIZE)



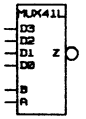
MUX41H (SIZE)



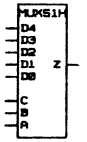
MUX41GH (SIZE)



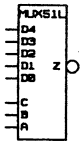
MUX41L (SIZE)



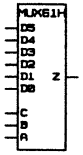
MUX51H (SIZE)



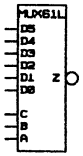
MUX51L (SIZE)



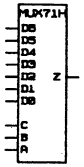
MUX61H (SIZE)



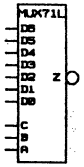
MUX61L (SIZE)



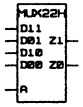
MUX71H (SIZE)



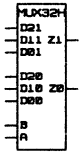
MUX71L (SIZE)



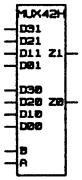
MUX22H (SIZE)



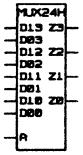
MUX32H (SIZE)



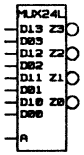
MUX42H (SIZE)



MUX24H (SIZE)



MUX24L (SIZE)



NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH LSISKF PART EXLSISKF

ENGINEER: WANG

DATE: Sat Oct 15 18:19:13 1983

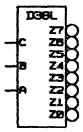
DECODERS

D24H

D24L

D38H

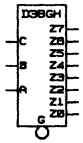
D38L



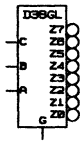
D24GH

D24GL

D38GH



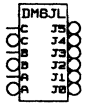
D38GL



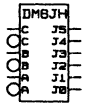
D410L

D410H

DM8JL



DM8JH



DM8JL

DM8JH

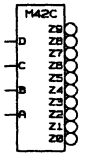
SYNCHRONIZERS

SYNC01 (SIZE)

SYNC10 (SIZE)

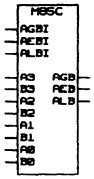
74XX TTL MSI

M42C



M82C (SIZE)

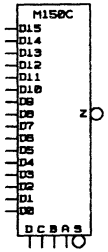
M85C



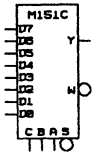
M138C

M138D

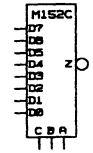
M150C (SIZE)



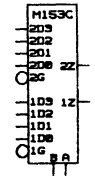
M151C (SIZE)



M152C (SIZE)



M153C (SIZE)



NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

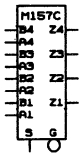
DRAWING: EXAMPLE OF EACH LSI5KF PART EXLSI5KF

ENGINEER: WANG

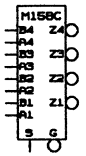
DATE: Sat Oct 15 16:22:26 1983

74XX TTL MSI (CONT'D)

M157C (SIZE)



M158C (SIZE)



M244C (SIZE)

M244XC (SIZE)

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH LSISKF PART EXLSISKF	ENGINEER: WANG
DATE: SAT OCT 15 16:23:53 1983	PAGE: 9 OF 9

MEMORY Library

There have been a few changes in the MEMORY Library for Release 5.0.

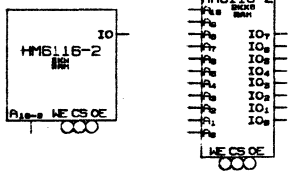
The MEMORY Library requires approximately 1.6 MBy (3000 blocks) of disk storage on the S-32. It contains complete bodies, physical, timing, and simulation models for the following 49 components:

2114AL-1	1024 x 4 static RAM
2147H-1	4096 x 1 static RAM
2148H-3	1024 x 4 static RAM
2164A-15	65536 x 1 dynamic RAM
2716-1	2048 x 8 EPROM
2732A-2	4096 x 8 EPROM
2764-2	8192 x 8 EPROM
27LS18	32 x 8 bipolar PROM
27LS19	32 x 8 bipolar PROM
27S13	512 x 4 bipolar PROM
27S13A	512 x 4 bipolar PROM
27S181	1024 x 8 bipolar PROM
27S181A	1024 x 8 bipolar PROM
27S185	2048 x 4 bipolar PROM
27S185A	2048 x 4 bipolar PROM
27S19	32 x 8 bipolar PROM
27S19A	32 x 8 bipolar PROM
27S21	256 x 4 bipolar PROM
27S21A	256 x 4 bipolar PROM
27S25	512 x 8 registered PROM
27S25A	512 x 8 registered PROM
27S27	512 x 8 registered PROM
27S28	512 x 8 bipolar PROM
27S28A	512 x 8 bipolar PROM
27S29	512 x 8 bipolar PROM
27S291	2048 x 8 bipolar PROM
27S291A	2048 x 8 bipolar PROM
27S29A	512 x 8 bipolar PROM
27S33	1024 x 4 bipolar PROM
27S33A	1024 x 4 bipolar PROM
27S41	4096 x 4 bipolar PROM
27S41A	4096 x 4 bipolar PROM
27S43	4096 x 8 bipolar PROM
27S43A	4096 x 8 bipolar PROM
28L22	256 x 8 low-power PROM
28L42	512 x 8 low-power PROM
28R85	512 x 8 registered PROM
AM9128-70	2048 x 8 static R/W RAM
HM4864-2	65536 x 1 dynamic RAM
HM6116-2	2048 x 8 CMOS RAM

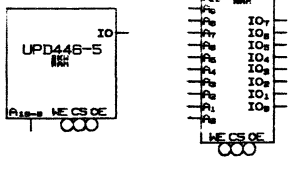
HM6167	16384 x 1 CMOS RAM
IMS1420-45	4096 x 4 static RAM
MCM6664-15	65536 x 1 dynamic RAM
NMC2142A	1024 x 4 static RAM
TC5517B	2048 x 8 CMOS RAM
TMS2149-3	1024 x 4 static RAM
TMS2167-4	16384 x 1 static RAM
TMS4164-12	65536 x 1 dynamic RAM
UPD446-5	2048 x 8 CMOS RAM

2K X 8 STATIC RAMS

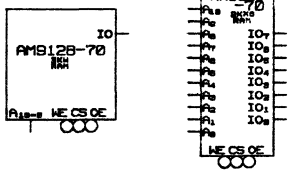
HM6116-2
(SIZE)



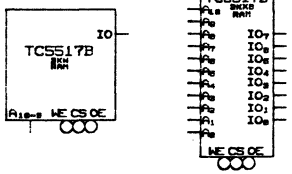
UPD446-5
(SIZE)



AM9128-70
(SIZE)

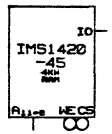


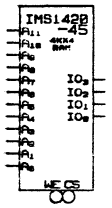
TC5517B
(SIZE)



4K X 4 STATIC RAMS

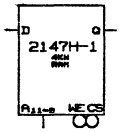
IMS1420-45
(SIZE)

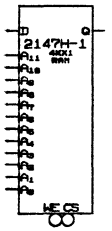




4K X 1 STATIC RAMS

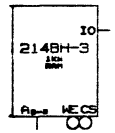
2147H-1
(SIZE)

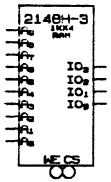




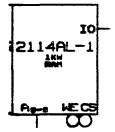
1K X 4 STATIC RAMS

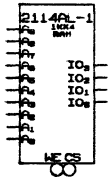
2148H-3
(SIZE)



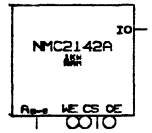


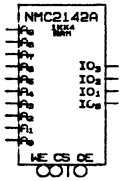
2114AL-1
(SIZE)



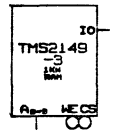


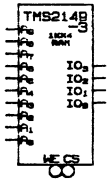
NMC2142A
(SIZE)





TMS2149-3
(SIZE)





DEFINE
X_FIRST=0
X_STEP=SIZE

VALID
LOGIC SYSTEMS INCORPORATED

DRAWING:
EXAMPLE OF EACH MEMORY PART
EXMEM

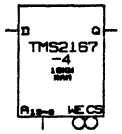
ENGINEER:
EWB

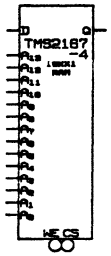
DATE:
THU MAR 15 17:54:50 1984

PAGE:
1 OF 4

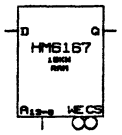
16K X 1 STATIC RAMS

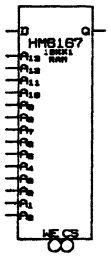
TMS2167-4 (SIZE)





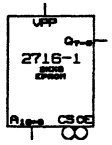
HM6167 (SIZE)

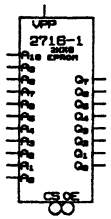




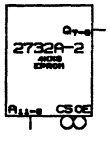
EPROMS

2716-1

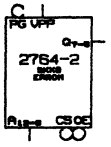


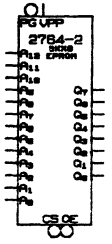


2732A-2



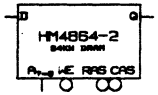
2764-2

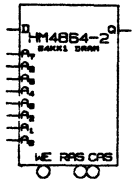




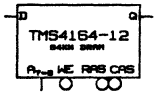
64K X 1 DYNAMIC RAMS

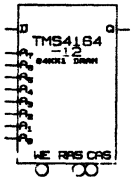
HM4864-2 (SIZE)



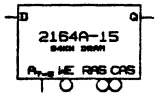


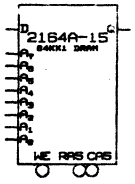
TMS4164-12 (SIZE)



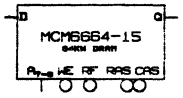


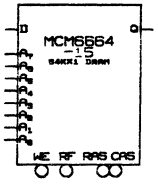
2164A-15 (SIZE)





MCM6664-15 (SIZE)





VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH MEMORY PART EXMEM

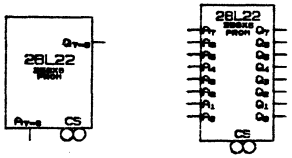
ENGINEER: EWB

DATE: THU MAR 15 17:56:56 1984

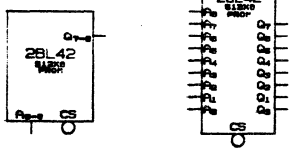
11-98

BIPOLAR PROM

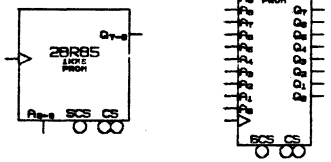
28L22



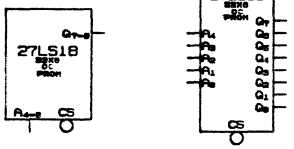
28L42



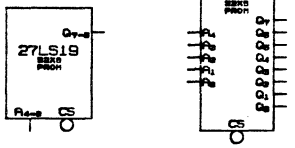
28R85



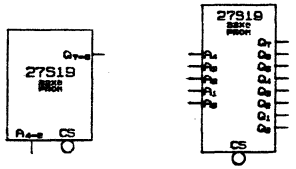
27LS18



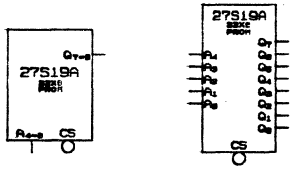
27LS19



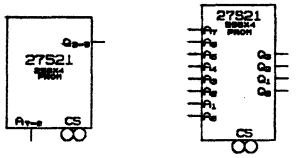
27S19



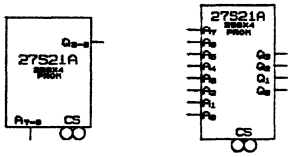
27S19A



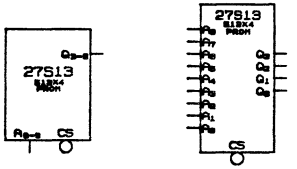
27S21



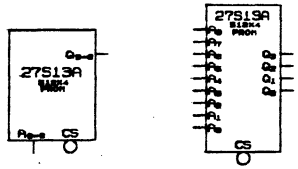
27S21A



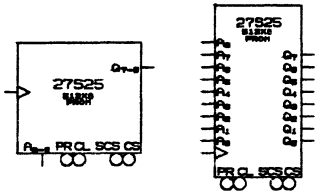
27S13



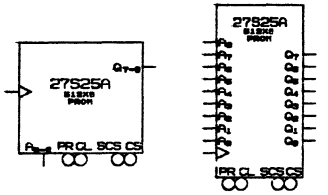
27S13A



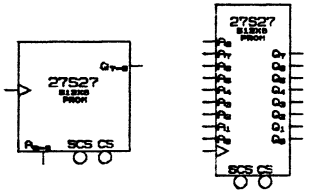
27S25



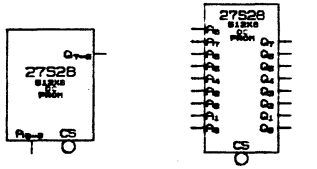
27S25A



27S27



27S28



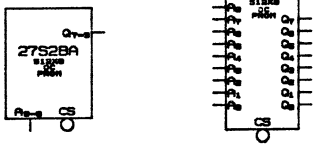
VALID
LOGIC SYSTEMS INCORPORATED

DRAWING:
EXAMPLE OF EACH MEMORY PART
EXMEM

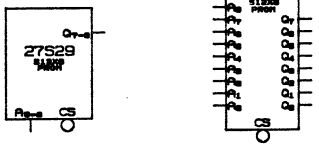
ENGINEER:
EWB

DATE:
THU JAN 5 11:02:16 1984

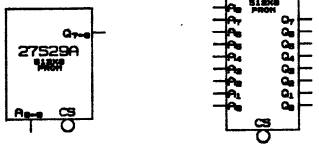
27528A



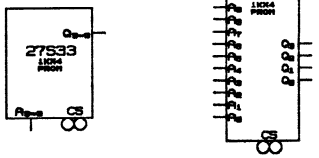
27529



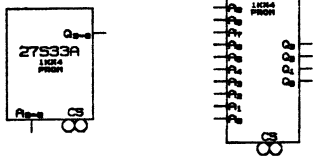
27529A



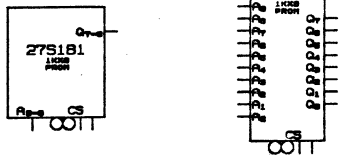
27533



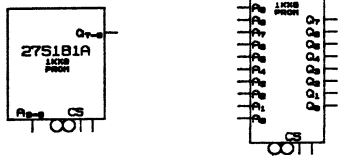
27533A



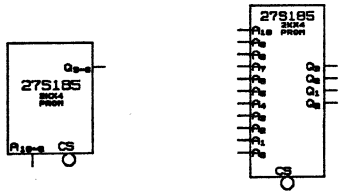
27S181



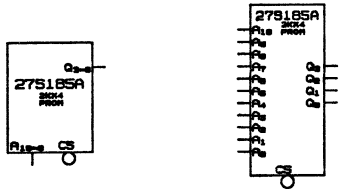
27S181A



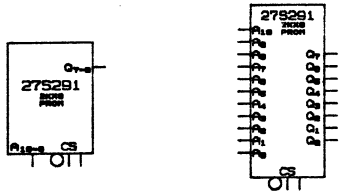
27S185



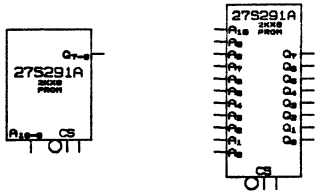
27S185A



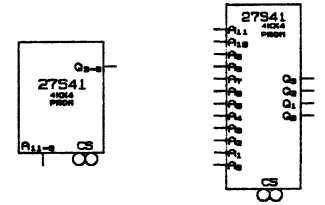
27S291



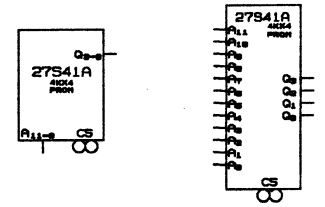
27S291A



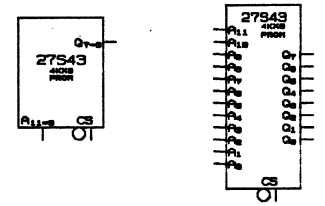
27S41



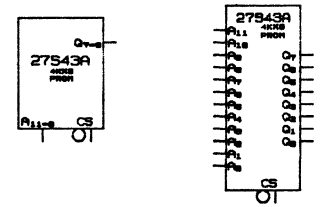
27S41A



27S43



27S43A



VALID
LOGIC SYSTEMS INCORPORATED

DRAWING:
EXAMPLE OF EACH MEMORY PART
EXMEM

ENGINEER:
EWB

DATE:
Mon Mar 26 14:26:56 1984

PAGE:
4 OF 4

STANDARD Library

There have been some changes in the Standard Library for Release 5.0.

The STANDARD Library requires approximately 0.2 MBy (363 blocks) of disk storage on the S-32. There are 29 parts.

The STANDARD Library contains standard shapes used in many drawings. These parts have no physical meaning but are used to convey design information to the Compiler, Timing Verifier, Simulator, and Packager. Some of the bodies are used to more concisely represent schematics. The STANDARD Library contains the following bodies:

A SIZE PAGE
: Used as an A-size border around drawings. It has no other significance. It is not required.

B SIZE PAGE
: Used as a B-size border around drawings. It has no other significance. It is not required.

DEFINE
: Used to define text macros which are specified as properties of this body. The property name is the text macro name and the value is its definition.

DRAWING
: Used to attach properties to the entire drawing.

NOT
: Used to change the logical assertion level of a signal without a physical inversion. The pins of the body are bubbleable. There are 4 different versions in 4 different orientations.

MERGE
: Used to merge several signals into one. There are 4 versions of each body. Versions 1 and 2 have inputs on 0.2 inch centers and versions 3 and 4 have inputs on 0.1 inch centers. One version of each can be used as a merge (multiple inputs on the left, single output on the right) and the other as a demerge (single input on the left, multiple outputs on the right). Mergers available are:

    2 MERGE
    3 MERGE
    4 MERGE

5 MERGE
6 MERGE
7 MERGE
8 MERGE
9 MERGE
10 MERGE

The versions of the 2, 4, 6, 8, and 10 mergers having inputs on 0.1 inch centers have outputs off grid. To connect a wire to these points, use the BLUE cursor button. For additional information, see the SCALD Graphics Editor User's manual.

MSB TAP and LSB TAP
Used to extract the most or least significant bits of a signal. The width of the signal to be extracted is specified by the SIZE property.

REPLICATE
Used to extend a one-bit signal to a SIZE-bit signal.

SIGN EXTEND
Used to extend an n-bit signal to a SIZE-bit signal by replicating the sign bit. The SIZE property is attached to the body It always extends the MSB (the left-most bit of the signal).

SLASH
Used to check the width of the signal to which it is attached. The SIZE property attached to the body is checked by the Compiler against the width of the signal. If the two do not match, an error is output.

SYNONYM
Used to synonym signals together. The signals must have the same assertion and be of the same width.

TIME_DIRECTIVES
Used to pass directives to the Timing Verifier. Properties attached to this body are Timing Verifier directives.

SIM_DIRECTIVES
Used to pass directives to the Simulator. Properties attached to this body are Simulator directives.

VALID B SIZE PAGE and VALID A SIZE PAGE
Used for Valid supplied models and drawings.

ORIGIN
The Graphics Editor automatically uses this body to indicate the origin of any body.

PIN NAMES

The Graphics Editor automatically annotates this body with the names of the pins of the drawing.

FLAG

Flag bodies are graphical indicators of the interface signals of a drawing. For some forms of Packager output, flag bodies are required to indicate the physical pins of a design.

SIGN EXTEND (SIZE)

SYNONYM

ORIGIN

DEFINE

DEFINE
X_FIRST=0
X_STEP=SIZE

DRAWING

DRAWING

LAST_MODIFIED=TUE MAR 27 18:30:44 1984

REPLICATE (SIZE)

PIN NAMES

PIN NAMES

VERSION 1 2 3 4

NOT

2 MERGE (OFF GRID)

3 MERGE (OFF GRID)

4 MERGE (OFF GRID)

5 MERGE

6 MERGE (OFF GRID)

7 MERGE

8 MERGE (OFF GRID)

VERSION 1 2 3 4

9 MERGE (OFF GRID)

10 MERGE

SLASH (SIZE)

FLAG

MSB TAP (VERSIONS 1 TO 4) (SIZE)

MSB TAP (VERSIONS 5 TO 8) (SIZE)

LSB TAP (VERSIONS 1 TO 4) (SIZE)

LSB TAP (VERSIONS 5 TO 8) (SIZE)

OBSOLETE : SEE RELEASE 7.0 CHANGES DOCUMENT MENU

If this expression is TRUE	then this version is used
FALSE	2
FALSE	3

DRAWING

TITLE=EXAMPLE OF EACH STANDARD BODY
ABBREV=EXSTAN

LAST_MODIFIED=FRI FEB 3 17:01:20 1984

OTHER BODIES IN THE STANDARD LIBRARY

A SIZE PAGE - 'A' SIZE BORDER FOR A DRAWING
B SIZE PAGE - 'B' SIZE BORDER FOR A DRAWING

VALID A SIZE PAGE - VALID'S STANDARD A SIZE PAGE
VALID B SIZE PAGE - VALID'S STANDARD B SIZE PAGE

NOTE: (SIZE) MEANS THAT THE BODY CAN BE GIVEN A SIZE PROPERTY

NOTE: (OFF GRID) MEANS THAT THE OUTPUT PIN IS OFF GRID (0.1). THE OUTPUT PIN SHOULD BE WIRED WITH THE BLUE CURSOR BUTTON (WHICH SNAPS TO THE PIN). THE WIRE CAN THEN BE PLACED DOWN WITH THE WHITE CURSOR BUTTON.

VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH STANDARD BODY

DATE:

ENGINEER: MJP

TIME_DIRECTIVES

TIMING VERIFIER DIRECTIVES

```
CLOCK_PERIOD=100.0
CLOCK_INTERVALS=8
CLOCK_SKEW=0.0
PREC_CLOCK_SKEW=0.0
DOT_TYPE=DOT_OR
LIST=NOUNNAMED
MAX_ERRORS=10
MAX_EXP_ERRORS=0
MAX_EVAL_PASSES=200
PRINT_WIDTH=132
RISE_FALL_ANAL=ON
RISE_FALL_MODELS=ON
TIMING_DIAGRAMS=OFF
TS_BUS_TYPE=DOT_TS
WIRE_DELAY=0.0-0.0
```

SIM_DIRECTIVES

SIMULATOR DIRECTIVES

```
CLOCK_PERIOD=100.0
CLOCK_INTERVALS=8
COMPILER_OUTPUT='CMPEXP.DAT'
COMMAND_FILE='TEST.CMD'
MEM_STATE=4
OUTPUT=LIST,COMMAND_LOG
SESSION_LOG=ON
SYNONYM_FILE='CMPSYN.DAT'
TERMINAL=VT100
```

DRAWING

TITLE=EXAMPLE OF EACH STANDARD BODY
ABBREV=EXSTAN

LAST_MODIFIED=Tue Mar 27 16:04:07 1984

OTHER BODIES IN THE STANDARD LIBRARY

A SIZE PAGE - 'A' SIZE BORDER FOR A DRAWING
B SIZE PAGE - 'B' SIZE BORDER FOR A DRAWING

VALID A SIZE PAGE - VALID'S STANDARD A SIZE PAGE
VALID B SIZE PAGE - VALID'S STANDARD B SIZE PAGE

NOTE: (SIZE) MEANS THAT THE BODY CAN BE GIVEN A SIZE PROPERTY

NOTE: (OFF GRID) MEANS THAT THE OUTPUT PIN IS OFF GRID (0.1). THE OUTPUT PIN SHOULD BE WIRED WITH THE BLUE CURSOR BUTTON (WHICH SNAPS TO THE PIN). THE WIRE CAN THEN BE PLACED DOWN WITH THE WHITE CURSOR BUTTON.

VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH STANDARD BODY

DATE:

ENGINEER: MJP

PHANTOM Library

There have been a few changes in the PHANTOM Library for Release 5.0.

The PHANTOM Library requires approximately 0.2 MBy (351 blocks) of disk storage on the S-32. There are 14 parts.

When connecting outputs together, PHANTOM gates are suggested to make the logic function of such a connection obvious and to make the drawings more closely approximate those that use logic without connectable outputs. The PHANTOM body is removed by the Packager - it is just a notational convenience. The bodies are used for OPEN COLLECTOR and OPEN EMITTER connections but not for TRI-STATE.

PHAN AND

The Phantom AND body is used where connecting outputs together forms an AND function. This is the case for OPEN COLLECTOR. There are two versions of each body. The bodies are:

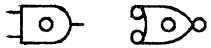
```
PHAN 2 AND
PHAN 3 AND
PHAN 4 AND
PHAN 5 AND
PHAN 6 AND
PHAN 7 AND
PHAN 8 AND
```

PHAN OR

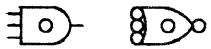
The Phantom OR body is used where connecting outputs together forms an OR function. This is the case for OPEN EMITTER. There are two versions of each body. The bodies are:

```
PHAN 2 OR
PHAN 3 OR
PHAN 4 OR
PHAN 5 OR
PHAN 6 OR
PHAN 7 OR
PHAN 8 OR
```

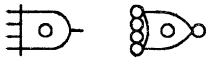
PHAN 2 AND (SIZE)



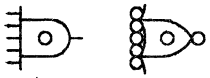
PHAN 3 AND (SIZE)



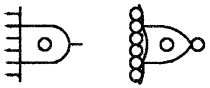
PHAN 4 AND (SIZE) (OFF GRID)



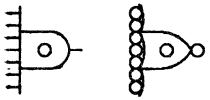
PHAN 5 AND (SIZE)



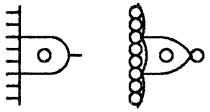
PHAN 6 AND (SIZE) (OFF GRID)



PHAN 7 AND (SIZE)



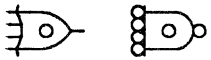
PHAN 8 AND (SIZE) (OFF GRID)



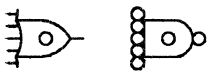
PHAN 2 OR (SIZE)

PHAN 3 OR (SIZE)

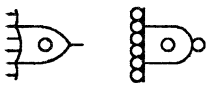
PHAN 4 OR (SIZE) (OFF GRID)



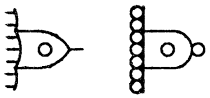
PHAN 5 OR (SIZE)



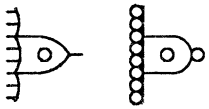
PHAN 6 OR (SIZE) (OFF GRID)



PHAN 7 OR (SIZE)



PHAN 8 OR (SIZE) (OFF GRID)



THE PHANTOM BODIES

PHANTOM BODIES ARE USED TO CONNECT OUTPUTS TOGETHER. THE BODY SERVES TWO PURPOSES. FIRST, IT MAKES THE LOGIC FUNCTION FORMED BY CONNECTING OUTPUTS EXPLICIT. THIS MAKES DIAGRAMS EASIER TO READ AND UNDERSTAND. THE SECOND PURPOSE IS TO SEPARATE THE INPUTS OF THE GATE FROM THE OUTPUT SO THAT THEY CAN BE TREATED AS LOGICALLY SEPARATE SIGNALS. THIS IS USEFUL WHEN THE BEHAVIOR OF ONE OF THE SIGNALS BEING TIED TOGETHER IS TO BE EXAMINED. THE POST PROCESSOR ELIMINATES ALL PHANTOM GATES, CONNECTING THE INPUTS TO THE GATE TO EACH OTHER (WIRE TIE) AND CONNECTING THEM TO THE OUTPUT SIGNAL. WIRE GATES SHOULD ALWAYS BE USED WHEN OUTPUTS ARE CONNECTED TOGETHER, BUT DIRECT WIRE-TYING IS PERMITTED. FOR A FURTHER DESCRIPTION OF PHANTOM GATES, WIRE-TIED NETS, OPEN-COLLECTOR, ETC. SEE THE POST PROCESSOR DOCUMENTATION.

DEFINE

X_FIRST=0

X_STEP=SIZE

DRAWING

TITLE=EXAMPLE OF EACH PHANTOM BODY
ABBREV=EXPHAN

LAST_MODIFIED=TUE OCT 25 12:17:22 1983

NOTE: ALL BODIES ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE BODY CAN BE GIVEN A SIZE PROPERTY

NOTE: (OFF GRID) MEANS THAT THE OUTPUT PIN IS OFF GRID (0.1). THE OUTPUT PIN SHOULD BE WIRED WITH THE BLUE CURSOR BUTTON (WHICH SNAPS TO THE PIN). THE WIRE CAN THEN BE PLACED DOWN WITH THE WHITE CURSOR BUTTON.

VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH PHANTOM BODY

DATE:

ENGINEER: MJP

PAGE: 1 OF 1

TIME Library

There have been no changes in the TIME Library for Release 5.0.

The TIME Library requires approximately 0.2 MBy (345 blocks) of disk storage on the S-32. There are 42 components.

The TIME Library contains all the Timing Verifier components which are used in the construction of models for the Timing Verifier. Each component consists of a body and a .PRIM drawing. Every pin of each body is individually bubbleable. This is quite different from the bubbling capability of physical parts. See the Timing Verifier documentation for a description of the use of this feature. There is only one version of each of the Timing Verifier primitives. This is a consequence of the ability to bubble each primitive pin individually.

The Library consists of the following parts:

AND

There are several different AND primitives, each with a different number of inputs. The AND primitives are:

AND - SIZE number of inputs
2 AND
3 AND
4 AND
5 AND
6 AND
7 AND
8 AND

OR

There are several different OR primitives, each with a different number of inputs. The OR primitives are:

OR - SIZE number of inputs
2 OR
3 OR
4 OR
5 OR
6 OR
7 OR
8 OR

CHG

There are several different CHG primitives, each with a different number of inputs. The CHG primitives are:

```
CHG     - SIZE number of inputs
2 CHG
3 CHG
4 CHG
5 CHG
6 CHG
7 CHG
8 CHG
```

XOR

The exclusive-OR primitive.

BUF

A buffer primitive.

TS BUF

A TRI-STATE buffer primitive.

THRESHOLD

The threshold gate outputs a "1" if its input is "1" and otherwise outputs "CHANGING".

MUX

Several multiplexers are provided. They are:

```
2 MUX
4 MUX
8 MUX
```

REG and REG RS

The two register primitives. Each is provided in two versions: The GLITCHY version always changes at clock time and the SMOOTH version stays stable at clock time if the input has not changed since the previous clock.

LATCH and LATCH RS

The two latch primitives.

SETUP HOLD and SETUP RISE HOLD FALL

The two setup and hold checkers. There are two versions of these checkers. An enable pin is provided in the second version.

MIN PULSE WIDTH

The minimum pulse width checker. There are two versions of this checker. An enable pin is provided in the second version.

EDGE TO EDGE
: The edge to edge checker. There are two versions of this checker. An enable pin is provided in the second version.

RES
: The resistor primitive.

IDENTITY
: An identity primitive. This primitive is similar to the BUF except that it will output 'Z' if the input is 'Z'. (The BUF output will be 'U' if the input is 'Z'.)

TRANSMISSION GATE
: The TRANSMISSION GATE has an enable EN and two bi-directional pins T1 and T2. When the enable is true, then the signals connected to T1 and T2 are connected together. When the enable is false, then the signals connected to T1 and T2 are not driven by the TRANSMISSION GATE. (This primitive is intended to be used to model both uni-directional and bi-directional MOS transistors.)

AND (SIZE)

2 AND (SIZE)

3 AND (SIZE)

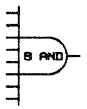
4 AND (SIZE) (OFF GRID)

5 AND (SIZE)

6 AND (SIZE) (OFF GRID)

7 AND (SIZE)

8 AND (SIZE) (OFF GRID)



OR (SIZE)

2 OR (SIZE)

3 OR (SIZE)

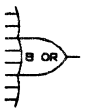
4 OR (SIZE) (OFF GRID)

5 OR (SIZE)

6 OR (SIZE) (OFF GRID)

7 OR (SIZE)

8 OR (SIZE) (OFF GRID)



CHG (SIZE)

2 CHG (SIZE)

3 CHG (SIZE)

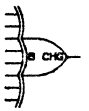
4 CHG (SIZE) (OFF GRID)

5 CHG (SIZE)

6 CHG (SIZE) (OFF GRID)

7 CHG (SIZE)

8 CHG (SIZE) (OFF GRID)



XOR (SIZE)

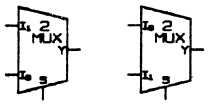
BUF (SIZE)

IDENTITY (SIZE)

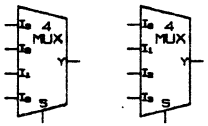
TS BUF (SIZE)

THRESHOLD (SIZE)

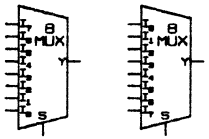
2 MUX (SIZE)



4 MUX (SIZE)



8 MUX (SIZE)



RES (SIZE)

NOTE: EVERY PIN OF EACH PART IN THIS LIBRARY IS INDIVIDUALLY BUBBLEABLE. THIS IS QUITE DIFFERENT FROM LOGICAL BUBBLING OF REAL PARTS. THE PURPOSE OF ALLOWING ARBITRARY BUBBLING IS TO REDUCE THE COMPLEXITY OF THE TIMING MODELS AND TO INCREASE THE SPEED OF THE TIMING VERIFIER. ELIMINATION OF SOME BUFS THROUGH THE USE OF BUBBLING REDUCES THE NUMBER OF LOCAL SIGNALS THAT APPEAR IN THE MODELS MAKING IT MORE LIKELY THAT TIMING ERRORS WILL BE REPORTED FOR SIGNALS THAT ARE UNDERSTOOD BY THE USER OF THE MODEL (WHO DOESN'T KNOW THE INTERNALS OF EACH TIMING MODEL).

DEFINE

X_FIRST=0

X_STEP=SIZE

DRAWING

TITLE=EXAMPLE OF EACH TIME PRIMITIVE
ABBREV=ETP

LAST_MODIFIED=WED OCT 19 09:05:21 1983

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

NOTE: (OFF GRID) MEANS THAT THE OUTPUT PIN IS OFF GRID (0.1). THE OUTPUT PIN SHOULD BE WIRED WITH THE BLUE CURSOR BUTTON (WHICH SNAPS TO THE PIN). THE WIRE CAN THEN BE PLACED DOWN WITH THE WHITE CURSOR BUTTON.

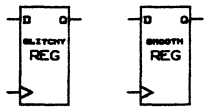
VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH TIME PRIMITIVE

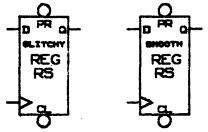
DATE:

ENGINEER: MJP

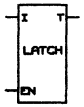
REG (SIZE)



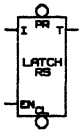
REG RS (SIZE)



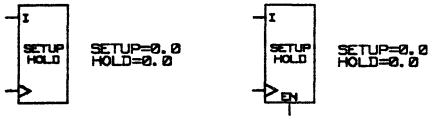
LATCH (SIZE)



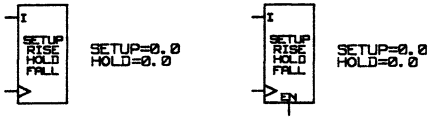
LATCH RS (SIZE)



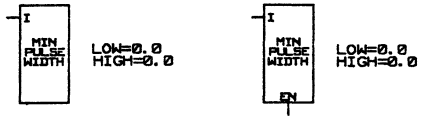
SETUP HOLD (SIZE)



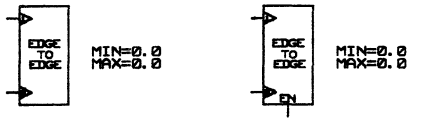
SETUP RISE HOLD FALL (SIZE)



MIN PULSE WIDTH (SIZE)



EDGE TO EDGE (SIZE)



TRANSMISSION GATE (SIZE)

NOTE: EVERY PIN OF EACH PART IN THIS LIBRARY IS INDIVIDUALLY BUBBLEABLE. THIS IS QUITE DIFFERENT FROM LOGICAL BUBBLING OF REAL PARTS. THE PURPOSE OF ALLOWING ARBITRARY BUBBLING IS TO REDUCE THE COMPLEXITY OF THE TIMING MODELS AND TO INCREASE THE SPEED OF THE TIMING VERIFIER. ELIMINATION OF SOME BUFS THROUGH THE USE OF BUBBLING REDUCES THE NUMBER OF LOCAL SIGNALS THAT APPEAR IN THE MODELS MAKING IT MORE LIKELY THAT TIMING ERRORS WILL BE REPORTED FOR SIGNALS THAT ARE UNDERSTOOD BY THE USER OF THE MODEL (WHO DOESN'T KNOW THE INTERNALS OF EACH TIMING MODEL).

DRAWING

TITLE=EXAMPLE OF EACH TIME PRIMITIVE
ABBREV=ETP

LAST_MODIFIED=WED OCT 19 09:02:40 1983

NOTE: EACH PART IS SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH TIME PRIMITIVE

DATE:

ENGINEER: MJP

PAGE: 2 OF 2

SIM Library

There have been no changes in the SIM Library for Release 5.0.

The SIM Library requires approximately 0.2 MBy (418 blocks) of disk storage on the S-32. There are 41 components.

The SIM Library contains all the Simulator components which are used in the construction of models for the Logic Simulator. Each component consists of a body and a .PRIM drawing. Every pin of each body is individually bubbleable. This is quite different from the bubbling capability of physical parts. See the logic Simulator documentation for a description of the use of this feature. There is only one version of each of the Simulator primitives. This is a consequence of the ability to bubble each primitive pin individually.

The Library consists of the following parts:

AND

There are several different AND primitives, each with a different number of inputs. The AND primitives are:

2 AND
3 AND
4 AND
5 AND
6 AND
7 AND
8 AND

OR

There are several different OR primitives, each with a different number of inputs. The OR primitives are:

2 OR
3 OR
4 OR
5 OR
6 OR
7 OR
8 OR

XOR

The exclusive-OR primitive.

BUF

A buffer primitive.

TS BUF
A TRI-STATE buffer primitive.

MUX
Several multiplexers are provided. They are:

- 2 MUX
- 4 MUX
- 8 MUX

REG, REG RS, REG RS COMP
The three register primitives.

LATCH, LATCH RS, LATCH RS COMP
The three latch primitives.

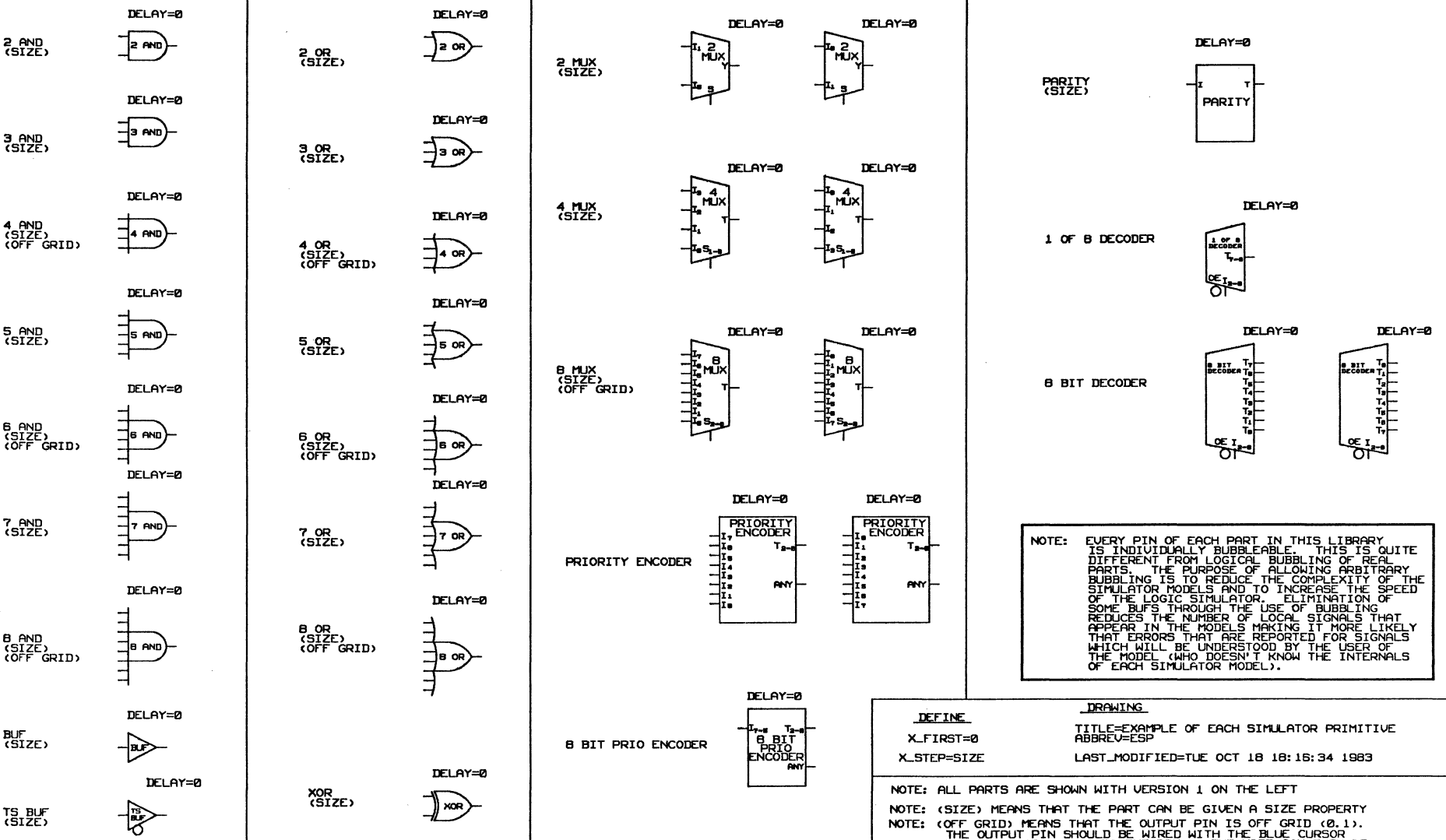
RES
The resistor primitive.

PASS TRANSISTOR
The pass transistor primitive.

COUNTER SHIFT REGISTER
The counter/shift register primitive. This is equivalent to the ECL 100136 part.

MISCELLANEOUS
The following primitives are also available:

- 1 OF 8 DECODER
- 8 BIT DECODER
- PRIORITY ENCODER
- 8 BIT PRIO ENCODER
- ALU
- ADDER
- CARRY SAVE ADDER
- COMPARATOR
- LOOKAHEAD
- MEMORY
- PARITY



NOTE: EVERY PIN OF EACH PART IN THIS LIBRARY IS INDIVIDUALLY BUBBLEABLE. THIS IS QUITE DIFFERENT FROM LOGICAL BUBBLING OF REAL PARTS. THE PURPOSE OF ALLOWING ARBITRARY BUBBLING IS TO REDUCE THE COMPLEXITY OF THE SIMULATOR MODELS AND TO INCREASE THE SPEED OF THE LOGIC SIMULATOR. ELIMINATION OF SOME BUFS THROUGH THE USE OF BUBBLING REDUCES THE NUMBER OF LOCAL SIGNALS THAT APPEAR IN THE MODELS MAKING IT MORE LIKELY THAT ERRORS THAT ARE REPORTED FOR SIGNALS WHICH WILL BE UNDERSTOOD BY THE USER OF THE MODEL (WHO DOESN'T KNOW THE INTERNALS OF EACH SIMULATOR MODEL).

DEFINE

X_FIRST=0

X_STEP=SIZE

DRAWING

TITLE=EXAMPLE OF EACH SIMULATOR PRIMITIVE
ABBREV=ESP

LAST_MODIFIED=TUE OCT 18 18:16:34 1983

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

NOTE: (OFF GRID) MEANS THAT THE OUTPUT PIN IS OFF GRID (0.1). THE OUTPUT PIN SHOULD BE WIRED WITH THE BLUE CURSOR BUTTON (WHICH SNAPS TO THE PIN). THE WIRE CAN THEN BE PLACED DOWN WITH THE WHITE CURSOR BUTTON.

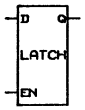
VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH SIMULATOR PRIMITIVE

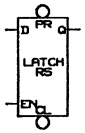
DATE:

ENGINEER: GSM

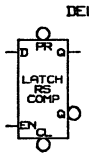
LATCH
(SIZE)



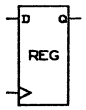
LATCH RS
(SIZE)



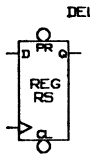
LATCH RS COMP
(SIZE)



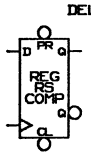
REG
(SIZE)



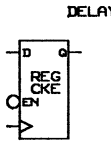
REG RS
(SIZE)



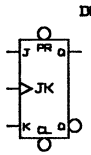
REG RS COMP
(SIZE)



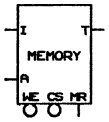
REG CKE
(SIZE)



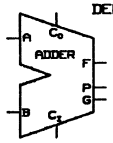
JK
(SIZE)



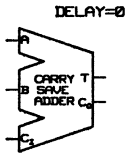
MEMORY
(SIZE)

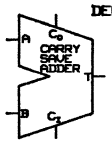


ADDER
(SIZE)

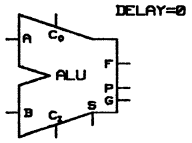


CARRY SAVE ADDER
(SIZE)

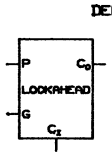




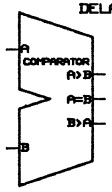
ALU
(SIZE)



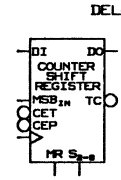
LOOKAHEAD
(SIZE)



COMPARATOR
(SIZE)

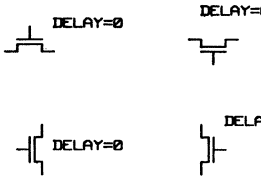


COUNTER SHIFT REGISTER
(SIZE)



RES
(SIZE)

PASS TRANSISTOR
(SIZE)



NOTE: EVERY PIN OF EACH PART IN THIS LIBRARY IS INDIVIDUALLY BUBBLEABLE. THIS IS QUITE DIFFERENT FROM LOGICAL BUBBLING OF REAL PARTS. THE PURPOSE OF ALLOWING ARBITRARY BUBBLING IS TO REDUCE THE COMPLEXITY OF THE SIMULATOR MODELS AND TO INCREASE THE SPEED OF THE LOGIC SIMULATOR. ELIMINATION OF SOME BUFS THROUGH THE USE OF BUBBLING REDUCES THE NUMBER OF LOCAL SIGNALS THAT APPEAR IN THE MODELS MAKING IT MORE LIKELY THAT ERRORS WILL BE REPORTED FOR SIGNALS THAT ARE UNDERSTOOD BY THE USER OF THE MODEL (WHO DOESN'T KNOW THE INTERNALS OF EACH SIMULATOR MODEL).

DRAWING

TITLE=EXAMPLE OF EACH SIMULATOR PRIMITIVE
ABBREV=ESP

LAST_MODIFIED=FRI FEB 3 16:43:13 1984

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

NOTE: (OFF GRID) MEANS THAT THE OUTPUT PIN IS OFF GRID (0.1). THE OUTPUT PIN SHOULD BE WIRED WITH THE BLUE CURSOR BUTTON (WHICH SNAPS TO THE PIN). THE WIRE CAN THEN BE PLACED DOWN WITH THE WHITE CURSOR BUTTON.

VALID
LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH SIMULATOR PRIMITIVE	DATE:
ENGINEER: GSM	PAGE: 2 OF 2

SPICE Library

There have been no changes in the SPICE Library since Release 3.1

The SPICE Library requires approximately 0.1 MBy of disk storage on the S-32. It contains bodies and physical for the following components:

CAPACITOR	capacitor
DIODE	diode
INDUCTOR	inductor
ISOURCE	current source
ISOURCE(I)	current-controlled current source
ISOURCE(V)	voltage-controlled current source
NJFET	n-channel JFET
NMOS	n-channel MOS transistor
NPN	NPN transistor
PJFET	p-channel JFET
PMOS	p-channel MOS transistor
PNP	PNP transistor
RESISTOR	resistor
TRANSMISSION	transmission line
VSOURCE	voltage source
VSOURCE(I)	current-controlled voltage source
VSOURCE(V)	voltage-controlled voltage source

RESISTOR

CAPACITOR

INDUCTOR

TRANSMISSION

V SOURCE(V)

I SOURCE(V)

V SOURCE(I)

I SOURCE(I)

V SOURCE

I SOURCE

DIODE

NPN

PNP

N JFET

P JFET

NMOS

PMOS

DEFINE

X_FIRST=0

X_STEP=SIZE

DRAWING

TITLE=EXAMPLE OF EACH SPICE PRIMITIVE
ABBREV=EXAMSPICE
LAST_MODIFIED=TUE MAR 17 16:00:23 1983

VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH SPICE PRIMITIVE	DATE: 5-MAR-83
ENGINEER: MJP	PAGE: 1 OF 1

TEGAS5 Library

There have been a few changes in the TEGAS5 Library for Release 4.2.

The TEGAS5 Library requires approximately 1.3 MBy of disk storage on the S-32. It contains bodies and physical for the following components:

1ADDER	1-bit full adder
1ALU	1-bit Arithmetic Logic Unit
2AND	2-input AND gate
3AND	3-input AND gate
4AND	4-input AND gate
5AND	5-input AND gate
6AND	6-input AND gate
7AND	7-input AND gate
8AND	8-input AND gate
2ANDNAND	2-input AND-NAND gate
3ANDNAND	3-input AND-NAND gate
4ANDNAND	4-input AND-NAND gate
5ANDNAND	5-input AND-NAND gate
6ANDNAND	6-input AND-NAND gate
7ANDNAND	7-input AND-NAND gate
8ANDNAND	8-input AND-NAND gate
4COMPARTR	4-bit magnitude comparator
4COUNTER	4-bit universal counter
2DECODER	1 out of 2 decoder
3DECODER	1 out of 3 decoder
4DECODER	1 out of 4 decoder
2DECODERE	1 out of 2 decoder (with enable)
3DECODERE	1 out of 3 decoder (with enable)
4DECODERE	1 out of 4 decoder (with enable)
8ENCODER	8-bit priority encoder
1FDETECT	1-bit fault detection gate
8FDETECT	8-bit fault detection gate
1FLTDUMMY	1-bit complex fault dummy delay element
8FLTDUMMY	8-bit complex fault dummy delay element
4MULTIPLY	4x4 multiplier
8MULTIPLY	8x8 multiplier
2MUX	2-input multiplexer
4MUX	4-input multiplexer
8MUX	8-input multiplexer
2MUXE	2-input multiplexer (with enable)
4MUXE	4-input multiplexer (with enable)
8MUXE	8-input multiplexer (with enable)

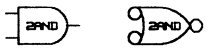
2NAND	2-input NAND gate
3NAND	3-input NAND gate
4NAND	4-input NAND gate
5NAND	5-input NAND gate
6NAND	6-input NAND gate
7NAND	7-input NAND gate
8NAND	8-input NAND gate
2NAND2OUT	2-input NAND gate with 2 outputs
3NAND2OUT	3-input NAND gate with 2 outputs
4NAND2OUT	4-input NAND gate with 2 outputs
5NAND2OUT	5-input NAND gate with 2 outputs
6NAND2OUT	6-input NAND gate with 2 outputs
7NAND2OUT	7-input NAND gate with 2 outputs
8NAND2OUT	8-input NAND gate with 2 outputs
2NOR	2-input NOR gate
3NOR	3-input NOR gate
4NOR	4-input NOR gate
5NOR	5-input NOR gate
6NOR	6-input NOR gate
7NOR	7-input NOR gate
8NOR	8-input NOR gate
2NXOR	2-input equivalence gate
2OR	2-input OR gate
3OR	3-input OR gate
4OR	4-input OR gate
5OR	5-input OR gate
6OR	6-input OR gate
7OR	7-input OR gate
8OR	8-input OR gate
2ORNOR	2-input OR-NOR gate
3ORNOR	3-input OR-NOR gate
4ORNOR	4-input OR-NOR gate
5ORNOR	5-input OR-NOR gate
6ORNOR	6-input OR-NOR gate
7ORNOR	7-input OR-NOR gate
8ORNOR	8-input OR-NOR gate
8PARITY	8-bit parity checker
4X4PLA	4x4 programmable logic array
16X4RAM	16x4 RAM
4X4REGFILE	4x4 register file
8X8REGFILE	8x8 register file
4REGISTER	4-bit register
8REGISTER	8-bit register
16X4ROM	16x4 ROM
4SERSHIFT	4-bit serial shift register
8SERSHIFT	8-bit serial shift register
1SHD01	1-bit timing analysis element, rising data edge
4SHD01	4-bit timing analysis element, rising data edge
1SHD10	1-bit timing analysis element, falling data edge
4SHD10	4-bit timing analysis element, falling data edge
1SHDALL	1-bit timing analysis element, any data change
4SHDALL	4-bit timing analysis element, any data change

4SHIFT	4-bit universal shift register
8SHIFT	8-bit universal shift register
4SUBTRACT	4-bit full subtractor
1TRANENE	1-bit transmission gate (low enable)
2TRANENE	2-bit transmission gate (low enable)
3TRANENE	3-bit transmission gate (low enable)
4TRANENE	4-bit transmission gate (low enable)
1TRANEPE	1-bit transmission gate (high enable)
2TRANEPE	2-bit transmission gate (high enable)
3TRANEPE	3-bit transmission gate (high enable)
4TRANEPE	4-bit transmission gate (high enable)
2TRIAND	2-input tristate AND gate
3TRIAND	3-input tristate AND gate
4TRIAND	4-input tristate AND gate
5TRIAND	5-input tristate AND gate
6TRIAND	6-input tristate AND gate
7TRIAND	7-input tristate AND gate
8TRIAND	8-input tristate AND gate
2TRINAND	2-input tristate NAND gate
3TRINAND	3-input tristate NAND gate
4TRINAND	4-input tristate NAND gate
5TRINAND	5-input tristate NAND gate
6TRINAND	6-input tristate NAND gate
7TRINAND	7-input tristate NAND gate
8TRINAND	8-input tristate NAND gate
2TRIOR	2-input tristate OR gate
3TRIOR	3-input tristate OR gate
4TRIOR	4-input tristate OR gate
5TRIOR	5-input tristate OR gate
6TRIOR	6-input tristate OR gate
7TRIOR	7-input tristate OR gate
8TRIOR	8-input tristate OR gate
1TRIREC0	1-input tristate receiver gate
2TRIREC0	2-input tristate receiver gate
3TRIREC0	3-input tristate receiver gate
4TRIREC0	4-input tristate receiver gate
5TRIREC0	5-input tristate receiver gate
6TRIREC0	6-input tristate receiver gate
7TRIREC0	7-input tristate receiver gate
8TRIREC0	8-input tristate receiver gate
1TRIREC1	1-input tristate receiver gate
2TRIREC1	2-input tristate receiver gate
3TRIREC1	3-input tristate receiver gate
4TRIREC1	4-input tristate receiver gate
5TRIREC1	5-input tristate receiver gate
6TRIREC1	6-input tristate receiver gate
7TRIREC1	7-input tristate receiver gate
8TRIREC1	8-input tristate receiver gate
1TRIRECX	1-input tristate receiver gate
2TRIRECX	2-input tristate receiver gate
3TRIRECX	3-input tristate receiver gate
4TRIRECX	4-input tristate receiver gate

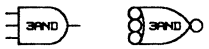
5TRIRECX	5-input tristate receiver gate
6TRIRECX	6-input tristate receiver gate
7TRIRECX	7-input tristate receiver gate
8TRIRECX	8-input tristate receiver gate
2WAND	2-input wired-AND gate
3WAND	3-input wired-AND gate
4WAND	4-input wired-AND gate
5WAND	5-input wired-AND gate
6WAND	6-input wired-AND gate
7WAND	7-input wired-AND gate
8WAND	8-input wired-AND gate
2WOR	2-input wired-OR gate
3WOR	3-input wired-OR gate
4WOR	4-input wired-OR gate
5WOR	5-input wired-OR gate
6WOR	6-input wired-OR gate
7WOR	7-input wired-OR gate
8WOR	8-input wired-OR gate
2XOR	2-input exclusive OR gate
BCD	BCD decoder (active low outputs)
BCDIN	BCD decoder (active high outputs)
BDSWITCH	bilateral transfer gate
CLOCK	clock element
D-DENE	negative edge triggered D flip-flop
D-DEPE	positive edge triggered D flip-flop
D-DMNE	master-slave D flip-flop
DELAY	delay gate w/o spike analysis
DELAYT	delay gate
DENE	negative edge triggered D flip-flop with preset and preclear
DEPE	positive edge triggered D flip-flop with preset and preclear
DMNE	master-slave D flip-flops with preset and preclear
GRND	ground
JK-JKENE	negative edge triggered JK flip-flop
JK-JKEPE	positive edge triggered JK flip-flop
JK-JKMNE	master-slave JK flip-flop
JKENE	negative edge triggered JK flip-flop with preset and preclear
JKEPE	positive edge triggered JK flip-flop with preset and preclear
JKMNE	master-slave JK flip-flop with preset and preclear
ONESHOT	one-shot (non-retriggerable)
PUSHPULL	push-pull gate
PWR	power
REONESHT	retriggerable one-shot
SR-SRENE	negative edge triggered SR flip-flop
SR-SREPE	positive edge triggered SR flip-flop
SR-SRMNE	master-slave SR flip-flop
SRENE	negative edge triggered SR flip-flop

	with preset and preclear
SREPE	positive edge triggered SR flip-flop
	with preset and preclear
SRMNE	master-slave SR flip-flop
	with preset and preclear
SRNANDL	NAND latch (unlocked)
SRNORL	NOR latch (unlocked)
T-TMNE	master-slave T flip-flop
TMNE	master-slave T flip-flop
	with preset and preclear
TNOT	inverter gate
TRANSP	special transmission gate

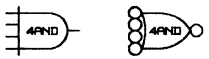
2AND (SIZE)



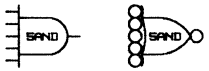
3AND (SIZE)



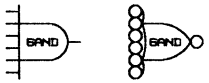
4AND (SIZE) (OFF GRID)



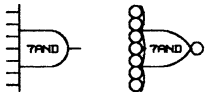
5AND (SIZE)



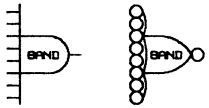
6AND (SIZE) (OFF GRID)



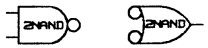
7AND (SIZE)



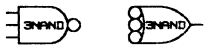
8AND (SIZE) (OFF GRID)



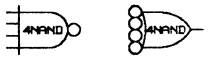
2NAND (SIZE)



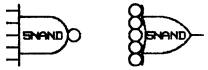
3NAND (SIZE)



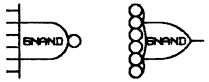
4NAND (SIZE) (OFF GRID)



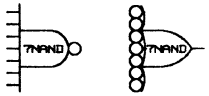
5NAND (SIZE)



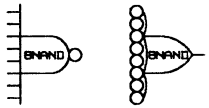
6NAND (SIZE) (OFF GRID)



7NAND (SIZE)

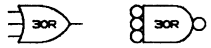


8NAND (SIZE) (OFF GRID)

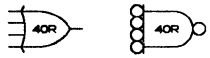


2OR (SIZE)

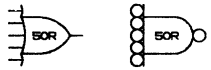
3OR (SIZE)



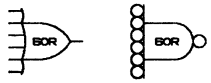
4OR (SIZE) (OFF GRID)



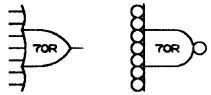
5OR (SIZE)



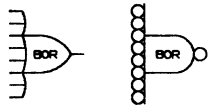
6OR (SIZE) (OFF GRID)



7OR (SIZE)



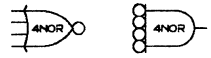
8OR (SIZE) (OFF GRID)



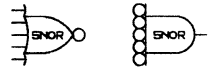
2NOR (SIZE)

3NOR (SIZE)

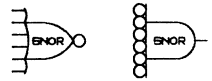
4NOR (SIZE) (OFF GRID)



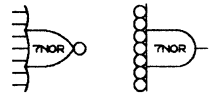
5NOR (SIZE)



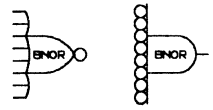
6NOR (SIZE) (OFF GRID)



7NOR (SIZE)



8NOR (SIZE) (OFF GRID)



2XOR (SIZE)

2NXOR (SIZE)

DELAY (SIZE)

DELAYT (SIZE)

TNOT (SIZE)

CLOCK (SIZE)

PWR (SIZE)

GRND (SIZE)

DEFINE

X_FIRST=0

X_STEP=SIZE

DRAWING

TITLE=EXAMPLE OF EACH TEGASS PRIMITIVE
ABBREV=EXTEG

LAST_MODIFIED=THU OCT 27 17:42:20 1983

NOTE: ALL PRIMITIVES ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PRIMITIVE CAN BE GIVEN A SIZE PROPERTY

NOTE: (OFF GRID) MEANS THAT THE OUTPUT PIN IS OFF GRID (0.1) THE OUTPUT PIN SHOULD BE WIRED WITH THE BLUE CURSOR BUTTON (WHICH SNAPS TO THE PIN). THE WIRE CAN THEN BE PLACED DOWN WITH THE WHITE CURSOR BUTTON.

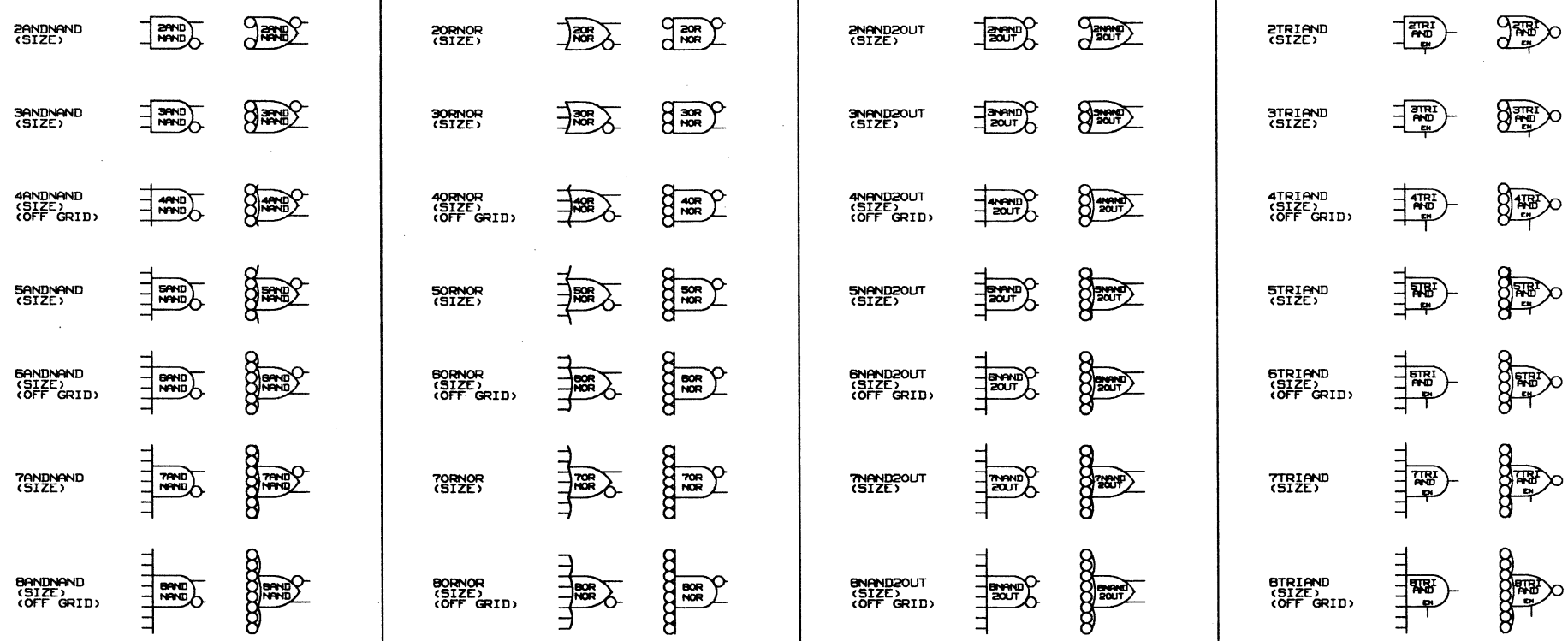
VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH TEGASS PRIMITIVE

DATE: 5-MAY-83

ENGINEER: PL

PAGE: 1 OF 7



BDSWITCH (SIZE)

PUSHPULL (SIZE)

ONESHOT (SIZE)

REONESHT (SIZE)

DRAWING

TITLE=EXAMPLE OF EACH TEGASS PRIMITIVE
ABBREV=EXTEG

LAST_MODIFIED=THU OCT 27 17:44:54 1983

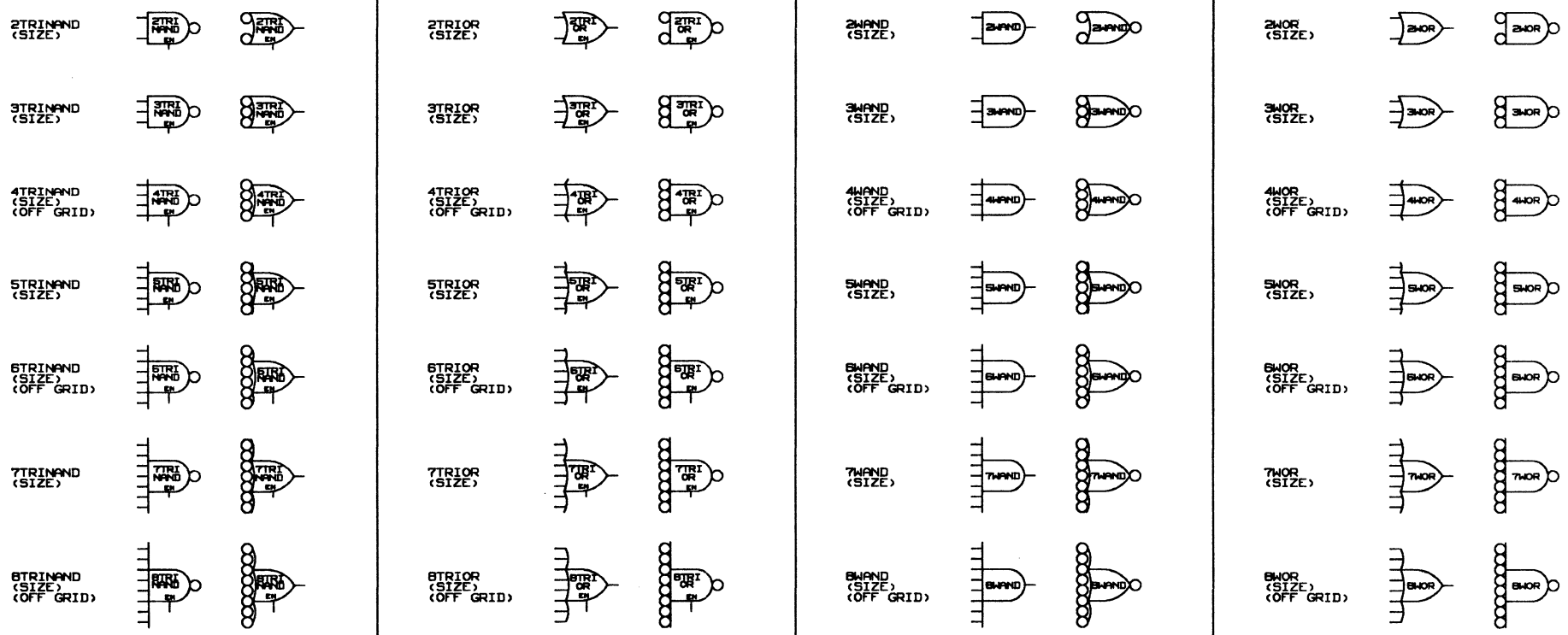
NOTE: ALL PRIMITIVES ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PRIMITIVE CAN BE GIVEN A SIZE PROPERTY

NOTE: (OFF GRID) MEANS THAT THE OUTPUT PIN IS OFF GRID (0.1) THE OUTPUT PIN SHOULD BE WIRED WITH THE BLUE CURSOR BUTTON (WHICH SNAPS TO THE PIN). THE WIRE CAN THEN BE PLACED DOWN WITH THE WHITE CURSOR BUTTON.

VALID LOGIC SYSTEMS INCORPORATED

TITLE:	DATE:
EXAMPLE OF EACH TEGASS PRIMITIVE	5-MAY-83
ENGINEER: PL	PAGE: 2 OF 7



DRAWING

TITLE=EXAMPLE OF EACH TEGASS PRIMITIVE
ABBREV=EXTEG

LAST_MODIFIED=THU OCT 27 17:46:58 1983

NOTE: ALL PRIMITIVES ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PRIMITIVE CAN BE GIVEN A SIZE PROPERTY

NOTE: (OFF GRID) MEANS THAT THE OUTPUT PIN IS OFF GRID (0.1) THE OUTPUT PIN SHOULD BE WIRED WITH THE BLUE CURSOR BUTTON (WHICH SNAPS TO THE PIN). THE WIRE CAN THEN BE PLACED DOWN WITH THE WHITE CURSOR BUTTON.

VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH TEGASS PRIMITIVE	DATE: 5-MAY-83
ENGINEER: PL	PAGE: 3 OF 7

1TRIRECX (SIZE)

2TRIRECX (SIZE)

3TRIRECX (SIZE)

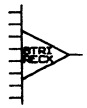
4TRIRECX (SIZE) (OFF GRID)

5TRIRECX (SIZE)

6TRIRECX (SIZE) (OFF GRID)

7TRIRECX (SIZE)

8TRIRECX (SIZE) (OFF GRID)



1TRIREC1 (SIZE)

2TRIREC1 (SIZE)

3TRIREC1 (SIZE)

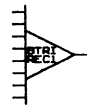
4TRIREC1 (SIZE) (OFF GRID)

5TRIREC1 (SIZE)

6TRIREC1 (SIZE) (OFF GRID)

7TRIREC1 (SIZE)

8TRIREC1 (SIZE) (OFF GRID)



1TRIREC0 (SIZE)

2TRIREC0 (SIZE)

3TRIREC0 (SIZE)

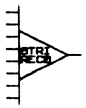
4TRIREC0 (SIZE) (OFF GRID)

5TRIREC0 (SIZE)

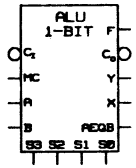
6TRIREC0 (SIZE) (OFF GRID)

7TRIREC0 (SIZE)

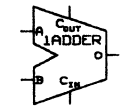
8TRIREC0 (SIZE) (OFF GRID)



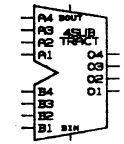
1ALU



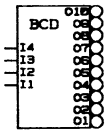
1ADDER (SIZE)



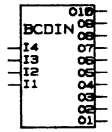
4SUBTRACT



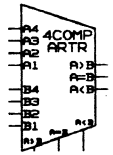
BCD



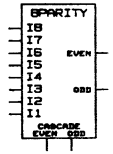
BCDIN



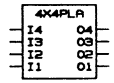
4COMPARTR



8PARITY



4X4PLA



DRAWING

TITLE=EXAMPLE OF EACH TEGASS PRIMITIVE
ABBREV=EXTEG

LAST_MODIFIED=THU OCT 27 17:48:04 1983

NOTE: ALL PRIMITIVES ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PRIMITIVE CAN BE GIVEN A SIZE PROPERTY

NOTE: (OFF GRID) MEANS THAT THE OUTPUT PIN IS OFF GRID (0.1) THE OUTPUT PIN SHOULD BE WIRED WITH THE BLUE CURSOR BUTTON (WHICH SNAPS TO THE PIN). THE WIRE CAN THEN BE PLACED DOWN WITH THE WHITE CURSOR BUTTON.

VALID LOGIC SYSTEMS INCORPORATED

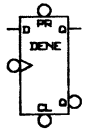
TITLE: EXAMPLE OF EACH TEGASS PRIMITIVE

DATE: 5-MAY-83

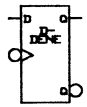
ENGINEER: PL

PAGE: 4 OF 7

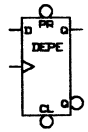
DENE (SIZE)



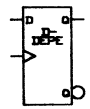
D-DENE (SIZE)



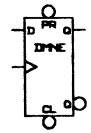
DEPE (SIZE)



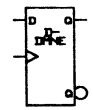
D-DEPE (SIZE)



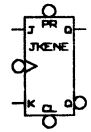
DMNE (SIZE)



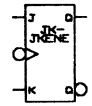
D-DMNE (SIZE)



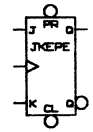
JKENE (SIZE)



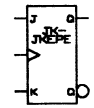
JK-JKENE (SIZE)



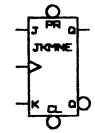
JKEPE (SIZE)



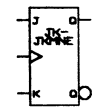
JK-JKEPE (SIZE)



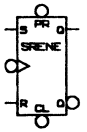
JKMNE (SIZE)



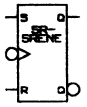
JK-JKMNE (SIZE)



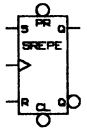
SRENE (SIZE)



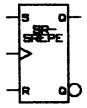
SR-SRENE (SIZE)



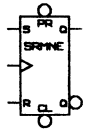
SREPE (SIZE)



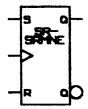
SR-SREPE (SIZE)



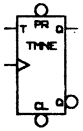
SRMNE (SIZE)



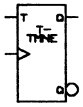
SR-SRMNE (SIZE)



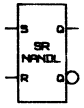
TMNE (SIZE)



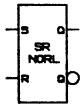
T-TMNE (SIZE)



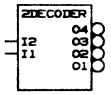
SRNANDL (SIZE)



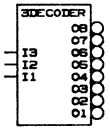
SRNORL (SIZE)



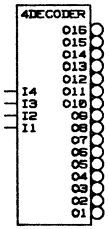
2DECODER



3DECODER



4DECODER



DRAWING

TITLE=EXAMPLE OF EACH TEGASS PRIMITIVE
ABBREV=EXTEG

LAST_MODIFIED=THU OCT 27 17:49:12 1983

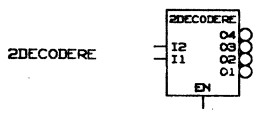
NOTE: ALL PRIMITIVES ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PRIMITIVE CAN BE GIVEN A SIZE PROPERTY

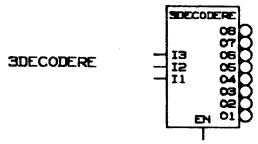
NOTE: (OFF GRID) MEANS THAT THE OUTPUT PIN IS OFF GRID (0.1) THE OUTPUT PIN SHOULD BE WIRED WITH THE BLUE CURSOR BUTTON (WHICH SNAPS TO THE PIN). THE WIRE CAN THEN BE PLACED DOWN WITH THE WHITE CURSOR BUTTON.

VALID LOGIC SYSTEMS INCORPORATED

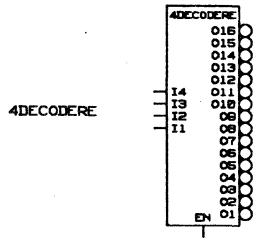
TITLE: EXAMPLE OF EACH TEGASS PRIMITIVE	DATE: 5-MAY-83
ENGINEER: PL	PAGE: 5 OF 7



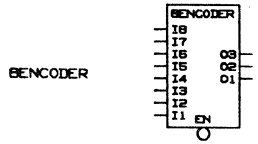
2DECODERE



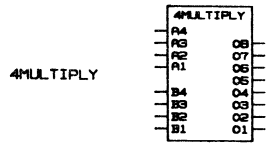
3DECODERE



4DECODERE

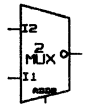


8ENCODER

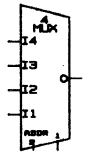


4MULTIPLY

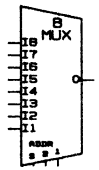
8MULTIPLY



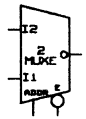
2MUX (SIZE)



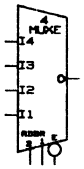
4MUX (SIZE)



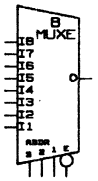
8MUX (SIZE)



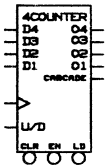
2MUXE (SIZE)



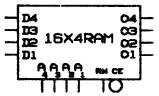
4MUXE (SIZE)



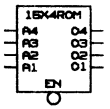
8MUXE (SIZE)



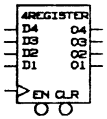
4COUNTER



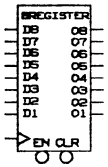
16X4RAM



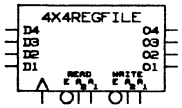
16X4ROM



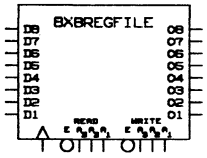
4REGISTER



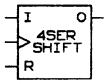
8REGISTER



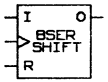
4X4REGFILE



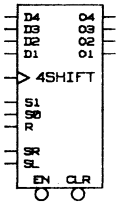
8XBREGFILE



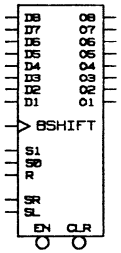
4SERSHIFT (SIZE)



8SERSHIFT (SIZE)



4SHIFT



8SHIFT

DRAWING

TITLE=EXAMPLE OF EACH TEGASS PRIMITIVE
ABBREV=EXTEG

LAST_MODIFIED=THU OCT 27 18:07:25 1983

NOTE: ALL PRIMITIVES ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PRIMITIVE CAN BE GIVEN A SIZE PROPERTY

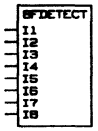
NOTE: (OFF GRID) MEANS THAT THE OUTPUT PIN IS OFF GRID (0.1) THE OUTPUT PIN SHOULD BE WIRED WITH THE BLUE CURSOR BUTTON (WHICH SNAPS TO THE PIN). THE WIRE CAN THEN BE PLACED DOWN WITH THE WHITE CURSOR BUTTON.

VALID LOGIC SYSTEMS INCORPORATED

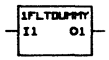
TITLE: EXAMPLE OF EACH TEGASS PRIMITIVE	DATE: 5-MAY-83
ENGINEER: PL	PAGE: 6 OF 7

1FDETECT

8FDETECT



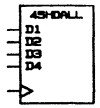
1FLTDUMMY



8FLTDUMMY

1SHDALL

4SHDALL

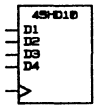


1SHD01

4SHD01

1SHD10

4SHD10

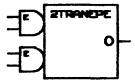


TRANSP (SIZE)

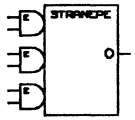
SIZE=4B

1TRANEPE (SIZE)

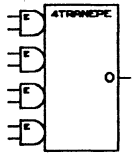
2TRANEPE (SIZE)



3TRANEPE (SIZE)

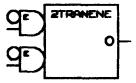


4TRANEPE (SIZE)

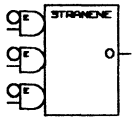


1TRANENE (SIZE)

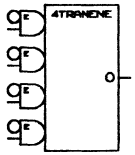
2TRANENE (SIZE)



3TRANENE (SIZE)



4TRANENE (SIZE)



DRAWING

TITLE=EXAMPLE OF EACH TEGASS PRIMITIVE
ABBREV=EXTEG

LAST_MODIFIED=THU OCT 27 17:51:24 1983

NOTE: ALL PRIMITIVES ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PRIMITIVE CAN BE GIVEN A SIZE PROPERTY

NOTE: (OFF GRID) MEANS THAT THE OUTPUT PIN IS OFF GRID (0.1) THE OUTPUT PIN SHOULD BE WIRED WITH THE BLUE CURSOR BUTTON (WHICH SNAPS TO THE PIN). THE WIRE CAN THEN BE PLACED DOWN WITH THE WHITE CURSOR BUTTON.

VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH TEGASS PRIMITIVE	DATE: 5-MAY-83
ENGINEER: PL	PAGE: 7 OF 7

C8000 Library

There have been no changes in the C8000 Library since Release 3.2.

The C8000 Library requires approximately 2.5 MBy of disk storage on the S-32. It contains bodies, physical, timing, and simulation models for the following components from the Fujitsu C8000VH Gate Array family:

A1N	1-bit full adder
A2N	2-bit full adder
A4H	4-bit full adder
D14	2-wide 3-and 4-input and-or-invert
D23	2-wide 2-and 3-input and-or-invert
D24	2-wide 2-and 4-input and-or-invert
D34	3-wide 2-and 4-input and-or-invert
D44	2-wide 2-or 2-and 4-input and-or-invert
EXTB	input/output pad
EXTBI	input interblock loading factor
EXTBO	output interblock loading factor
EXTI	input pad
EXTO	output pad
FD2	power D flip-flop
FD3	power D flip-flop with preset
FD6	D flip-flop
FDM	D flip-flop
FDN	D flip-flop with set
FDO	D flip-flop with reset
FDP	D flip-flop with set and reset
FDQ	4-bit D flip-flop
G14	2-wide 3-or 4-input or-and-invert
G23	2-wide 2-or 3-input or-and-invert
G24	2-wide 2-or 4-input or-and-invert
G34	3-wide 2-or 4-input or-and-invert
G44	2-wide 2-and 2-or 4-input or-and-invert
H6T	tri-state output and input buffer
I2B	input buffer
IKB	clocked input buffer (inverted)
ILB	clocked input buffer (noninverted)
K1B	clock buffer
K2B	power clock buffer
K3B	gated clock (and) buffer
K4B	gated clock (or) buffer
KCB	block clock buffer (non-inv)
LT1	set-reset latch with clear
LT2	1-bit data latch
LT4	4-bit data latch
N2B	2-input power nand
N2N	2-input nand

N2P	2-input power and
N3B	3-input power nand
N3N	3-input nand
N3P	3-input power and
N4B	4-input power nand
N4N	4-input nand
N4P	4-input power and
N6B	6-input power nand
N8B	8-input power nand
N9B	9-input power nand
NCB	12-input power nand
NGB	16-input power nand
O2B	output buffer
O4T	tri-state output buffer
R2B	2-input power nor
R2N	2-input nor
R2P	2-input power or
R3B	3-input power nor
R3N	3-input nor
R3P	3-input power or
R4B	4-input power nor
R4N	4-input nor
R4P	4-input power or
R6B	6-input power nor
R8B	8-input power nor
R9B	9-input power nor
RCB	12-input power nor
RGB	16-input power nor
T24	power 2-and 4-wide multiplexer
T26	power 2-and 6-wide multiplexer
T28	power 2-and 8-wide multiplexer
T2B	2-to-1 selector
T2C	dual 2-to-1 selector
T2D	2-to-1 selector
T32	power 3-and 2-wide multiplexer
T33	power 3-and 3-wide multiplexer
T34	power 3-and 4-wide multiplexer
T42	power 4-and 2-wide multiplexer
T43	power 4-and 3-wide multiplexer
T44	power 4-and 4-wide multiplexer
T4A	4-to-1 selector
U24	power 2-or 4-wide multiplexer
U26	power 2-or 6-wide multiplexer
U28	power 2-or 8-wide multiplexer
U32	power 3-or 2-wide multiplexer
U33	power 3-or 3-wide multiplexer
U34	power 3-or 4-wide multiplexer
U42	power 4-or 2-wide multiplexer
U43	power 4-or 3-wide multiplexer
U44	power 4-or 4-wide multiplexer
V1N	inverter
V2B	power inverter

V3A	1-to-2 selector
V3B	dual 1-to-2 selector
X1B	power exclusive-nor
X2B	power exclusive-or
Z00	0 clip
Z01	1 clip

COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
V1N	SIZE									
V2B	SIZE									
K1B	SIZE									
K2B	SIZE									
K3B	SIZE									
K4B	SIZE									
KCB	SIZE									

DEFINE
X_FIRST=0
X_STEP=SIZE

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH CB000 PART EXCB000
ENGINEER:
DATE: Mon Aug 22 13:12:55 1983
PAGE: 1 OF 16

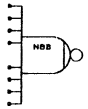
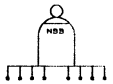
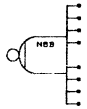
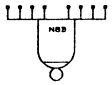
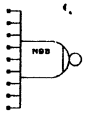
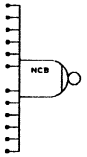
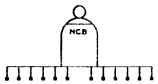
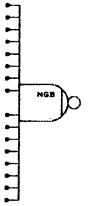
COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
N2P	SIZE									
N3P	SIZE									
N4P	SIZE									
N2N	SIZE									
N2B	SIZE									
N3N	SIZE									
N3B	SIZE									
N4N	SIZE									
N4B	SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH C8000 PART EXC8000

ENGINEER:

DATE: Mon Aug 22 11:46:01 1983

COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
N6B	SIZE									
N8B	SIZE									
N9B	SIZE									
NCB	SIZE									
NGB	SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH C8000 PART
EXC8000

ENGINEER:

DATE: Wed Aug 31 17:35:27 1983

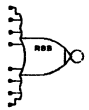
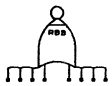
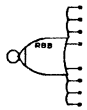
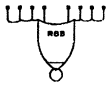
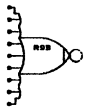
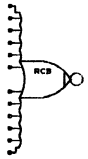
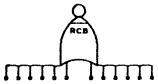
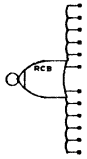
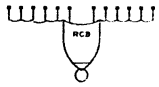
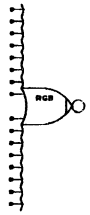
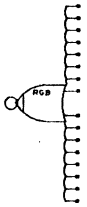
COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
R2P	SIZE									
R3P	SIZE									
R4P	SIZE									
R2N	SIZE									
R2B	SIZE									
R3N	SIZE									
R3B	SIZE									
R4N	SIZE									
R4B	SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH C8000 PART EXC8000

ENGINEER:

DATE: Wed Aug 31 17:42:18 1983

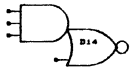
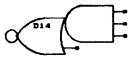
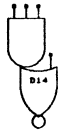
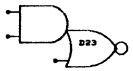
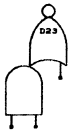
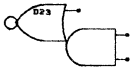
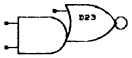
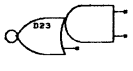
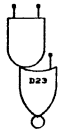
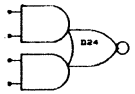
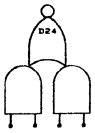
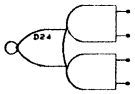
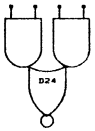
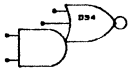
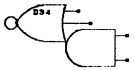
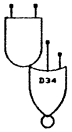
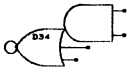
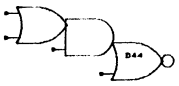
COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
R6B	SIZE									
R8B	SIZE									
R9B	SIZE									
RCB	SIZE									
RGB	SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH C8000 PART EXC8000

ENGINEER:

DATE: Mon Aug 22 11:58:14 1983

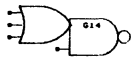
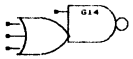
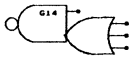
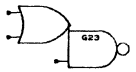
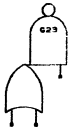
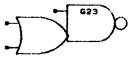
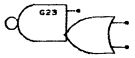
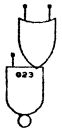
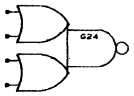
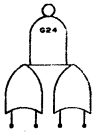
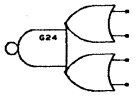
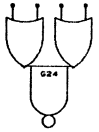
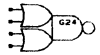
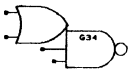
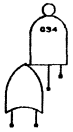
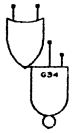
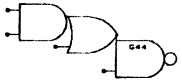
COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
X1B										
X2B	SIZE									
D14	SIZE									
D23	SIZE									
D24	SIZE									
D34	SIZE									
D44	SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH CA000 PART EXCA000

ENGINEER:

DATE: Mon Aug 22 12:05:35 1983

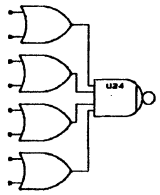
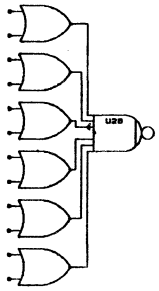
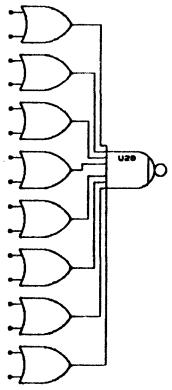
COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
G14	SIZE									
G23	SIZE									
G24	SIZE									
G34	SIZE									
G44	SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH C8000 PART EXC8000

ENGINEER:

DATE: Mon Aug 22 12:10:34 1983

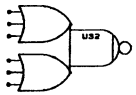
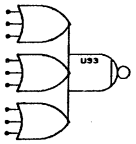
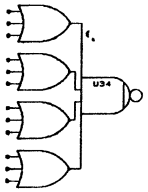
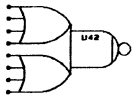
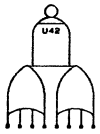
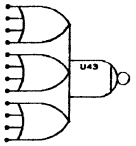
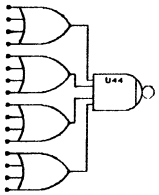
COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
U24	SIZE									
U26	SIZE									
U28	SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH C8000 PART EXC8000

ENGINEER:

DATE: Mon Aug 22 12:17:01 1983

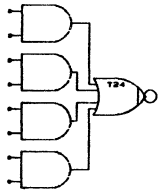
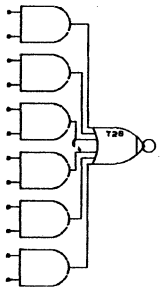
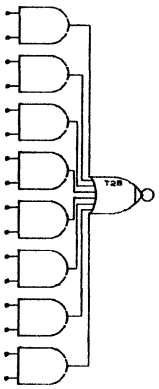
COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
U32	SIZE									
U33	SIZE									
U34	SIZE									
U42	SIZE									
U43	SIZE									
U44	SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH C8000 PART EXC8000

ENGINEER:

DATE: Mon Aug 22 12:22:09 1983

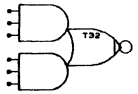
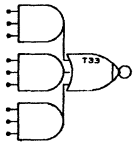
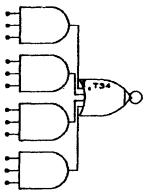
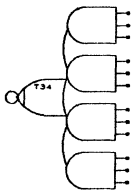
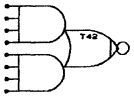
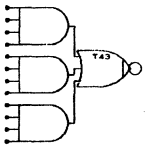
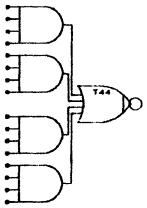
COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
T24	SIZE									
T26	SIZE									
T28	SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH C8000 PART EXC8000

ENGINEER:

DATE: Fri Aug 19 10:51:10 1983

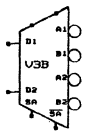
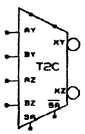
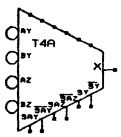
COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
T32	SIZE									
T33	SIZE									
T34	SIZE									
T42	SIZE									
T43	SIZE									
T44	SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH CB000 PART EXCB000

ENGINEER:

DATE: Mon Aug 22 12:27:31 1983

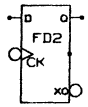
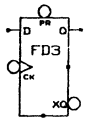
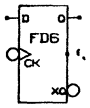
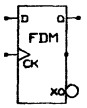
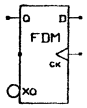
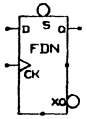
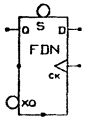
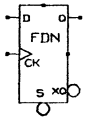
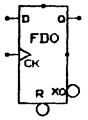
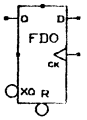
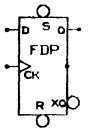
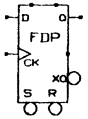
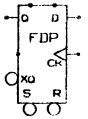
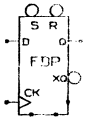
COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
V3A	SIZE									
V3B	SIZE									
T2D	SIZE									
T2C	SIZE									
T2C	SIZE									
T4A	SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH C8000 PART EXC8000

ENGINEER:

DATE: Fri Aug 19 10:48:11 1983

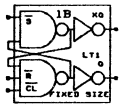
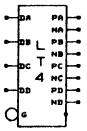
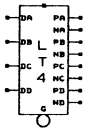
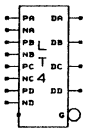
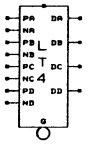
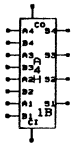
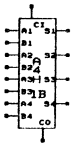
COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
FD2	SIZE									
FD3	SIZE									
FD6	SIZE									
FDM	SIZE									
FDN	SIZE									
FDO	SIZE									
FDP	SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH C8000 PART EXC8000

ENGINEER:

DATE: Fri Aug 19 10:47:07 1983

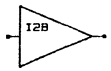
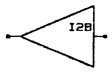
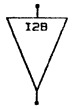
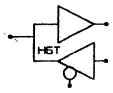
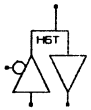
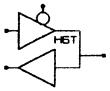
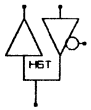
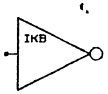
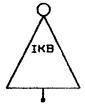
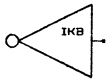
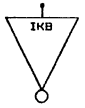
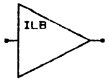
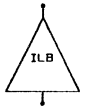
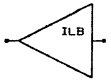
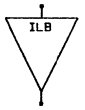
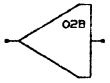
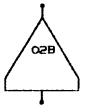
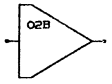
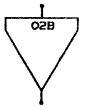
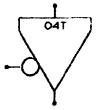
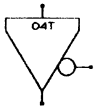
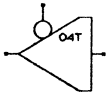
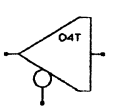
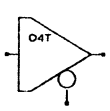
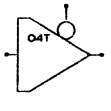
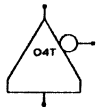
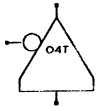
COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
FDQ	SIZE									
LT1	1 BIT									
LT2	SIZE									
LT4	SIZE									
A1N	SIZE									
A2N	SIZE									
A4H	FIXED SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
EXAMPLE OF EACH C8000 PART
EXC8000

ENGINEER:

DATE:
Thu Sep 1 19:29:55 1983

COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
I2B	SIZE									
H6T	SIZE									
IKB	SIZE									
ILB	SIZE									
O2B	SIZE									
O4T	SIZE									

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH CB000 PART EXCB000

ENGINEER:

DATE: Fri Aug 19 10:45:41 1983

COMPONENT	SIZE	NOTES	VERSION 1	VERSION 2	VERSION 3	VERSION 4	VERSION 5	VERSION 6	VERSION 7	VERSION 8
EXTBI	SIZE	BLOCK INPUT FLAG FOR FLDL								
EXTBO	SIZE	BLOCK OUTPUT FLAG FOR FLDL								
EXTI	SIZE	CHIP INPUT FLAG FOR FLDL								
EXTO	SIZE	CHIP OUTPUT FLAG FOR FLDL								
EXTB	SIZE	CHIP BIDIRECTIONAL FLAG FOR FLDL								
Z00	SIZE	LOGIC 0 CLIP								
Z01	SIZE	LOGIC 1 CLIP								

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH C8000 PART EXC8000

ENGINEER:

DATE: Fri Aug 19 10:45:09 1983

MCA12 Library

Release 4.2 is the first complete release of the MECL 10,000 Macrocell Gate Array Library.

The MCA12 Library requires approximately 3.1 MBy of disk storage on the S-32. It contains bodies, physical, timing, and simulation models for the following 167 components:

BOUT	output pad body (non-bidirectional)
G01	2-input or and gate
G02	2-input or and gate
G03	2-input or nand gate
G04	2-input or nand gate
G05	3-input or gate
G06	3-input nor gate
G07	3-input and gate
G08	3-input nand gate
G09	2-input (active low) and exclusive or gate
G10	2-input or exclusive or gate
G11	2-input exclusive or and gate
G12	2-to-1 mux
G13	2-to-1 mux with enable
G14	latch
G15	latch with dual enables
G16	transceiver--or gates
G17	transceiver nor driver, and receiver
G18	driver 2-input or gate
G19	driver 2-input nor gate
G20	4-input or gate
G21	or/and gate
G24	25 ohm 2-input or gate driver
G25	25 ohm 2-input nor gate driver
H01	3 & 4-input active low and/nand gates
H02	4-input or and/nand + 3-input and/nand
H03	2-wide 2-input or and/nand + 2-input or and/nand
H04	2-input active low and exclusive or + 2-input exclusive or gate
H05	7-input or/nor gate
H06	2-wide 2-input or 4-input and/nand gate
H07	6-input or and exclusive or gating structure
H08	6-input or and exclusive nor gating structure
H09	2-wide 2-input active low and exclusive or/nor
H10	exclusive or and/nand
H11	4-input exclusive or
H12	4-input exclusive nor
H15	2-wide 3-input active low or/nor
H16	6-input or and or/nor gating structure
H17	3-wide 3-2-2 and or/nor

H18	5-input gating structure
H27	3-wide 3-3-3 input and or/nor gating structure
H31	D flip-flop with reset and 2 clocks
H33	2-bit latch with 2 enable inputs
H34	latch with a 2-input data mux and 2 enable inputs
H35	dual latch
H40	dual 2-to-1 mux with column select
H41	dual 2-to-1 mux
H42	dual 2-to-1 mux with common enable and select inputs
H43	2-wide 2-input or 2-to-1 mux
H52	full adder
H54	half adder
H57	4-input gating structure
H58	4-input gating structure
H59	dual 4-input active low and/nand gates
H60	dual or and/nand gates
H61	dual 2-wide 2-input or and/nand gates
H62	dual 2-input active low and exclusive or/nor gates
H63	4-input active low and/nand gate
H64	2-input or 2-input active low and/nand gate
H65	2-wide 2-input or and/nand gate
H66	2-input active low or exclusive or/nor gate
H67	latch
H69	2-to-1 mux
H71	4-input exclusive or gate
H72	full adder
H73	half adder
H75	3-wide 3-2-2 active low and or/nor gate
H77	3-wide 3-input and or/nor gating structure
H78	D flip-flop with 2 clock inputs
H81	D flip-flop with differential clock and data
I01	dual 2-input or gates
I02	dual 2-input nor gates
I03	dual 2-input and gates
I04	dual 2-input nand gates
I05	dual 2-input nand gates
I06	dual 2-input (active low) nand gates with select
I07	dual or nand gates with select
I08	3-input (active low) and/nand gate
I09	or and/nand gate
I10	or and/nand gate
I11	2-input (active low) and exclusive or/nor gate
I12	2-to-1 mux with dual selects
I13	latch with dual enables
I14	latch with dual enables
I15	3-output and gate
I16	3-input and/nand gate
I20	dual differential line receiver
IO PAD	input/output pad body (bidirectional)
M13	12-input or/nor gate
M14	12-input or and/nand gating structure
M19	or exclusive or + and or exclusive nor gating

	structures
M20	4-wide 4-input active low and or/nor gating structure
M21	4-wide 3-input active low and or/nor gating structure
M22	4-wide 3-input and or/nor gating structure
M23	4-wide 3-3-2-1 and or/nor + 2-input exclusive or/nor gate
M24	6-wide 3-2-2-2-2-3 input and or/nor gating structure
M25	5-wide 2-2-3-3-3 input and or/nor gating structure
M26	5-wide 1-2-3-4-4 and or/nor gating structure
M28	3-wide 4-2-3 input and or/nor + 2 wide 2-3 and or/nor
M29	4-wide 5-3-4-2 input and or/nor gating structure
M30	4-wide 4-3-2-1 and or/nor + 5 input active low and/nand gate
M32	D flip-flop with a D-input mux and 2 clocks
M36	4-to-1 mux with enable
M37	4-to-1 mux with active low enable
M38	4-to-1 inverting mux with enable
M39	dual mux 4-to-1 and 2-to-1
M44	1-of-4 decoder (active high)
M45	1-of-4 decoder (active low)
M46	1-of-4 decoder (active high)
M47	1-of-4 decoder (active low)
M48	priority encoder
M49	priority expander
M50	full adder
M51	full adder
M53	full adder + half adder
M55	8-input gating structure
M56	9-input gating structure
M68	4-to-1 mux
M74	full adder
M76	6-wide 3-2-2-2-2-3 input and or/nor gating structure
M79	1-of-4 decoder (active low)
M80	4-to-1 mux with enable
001	2-input or and gate
002	2-input or and gate
003	2-input or nand gate
004	2-input or nand gate
005	3-input or gate
006	3-input nor gate
007	3-input and gate
008	3-input nand gate
009	2-input (active low) and exclusive or gate
010	2-input or exclusive or gate
011	2-input exclusive or and gate
012	2-to-1 mux
013	2-to-1 mux with enable
014	latch
015	latch with dual enables
016	transceiver--or gates
017	transceiver nor driver, and receiver
018	driver 2-input or gate

019	driver 2-input nor gate
020	4-input or gate
021	or/and gate
024	25 ohm 2-input or gate driver
025	25 ohm 2-input nor gate driver
T01	dual 2-input or gates
T02	dual 2-input nor gates
T03	dual 2-input and gates
T04	dual 2-input nand gates
T05	dual 2-input nand gates
T06	dual 2-input (active low) nand gates with select
T07	dual or nand gates with select
T08	3-input (active low) and/nand gate
T09	or and/nand gate
T10	or and/nand gate
T11	2-input (active low) and exclusive or/nor gate
T12	2-to-1 mux with dual selects
T13	latch with dual enables
T14	latch with dual enables
T20	dual differential line receiver
WIRED 2 OR	2-input wired-or body
WIRED 3 OR	3-input wired-or body
WIRED 4 OR	4-input wired-or body
WIRED 5 OR	5-input wired-or body
WIRED 6 OR	6-input wired-or body
WIRED 7 OR	7-input wired-or body
WIRED 8 OR	8-input wired-or body

APPLICATION NOTE

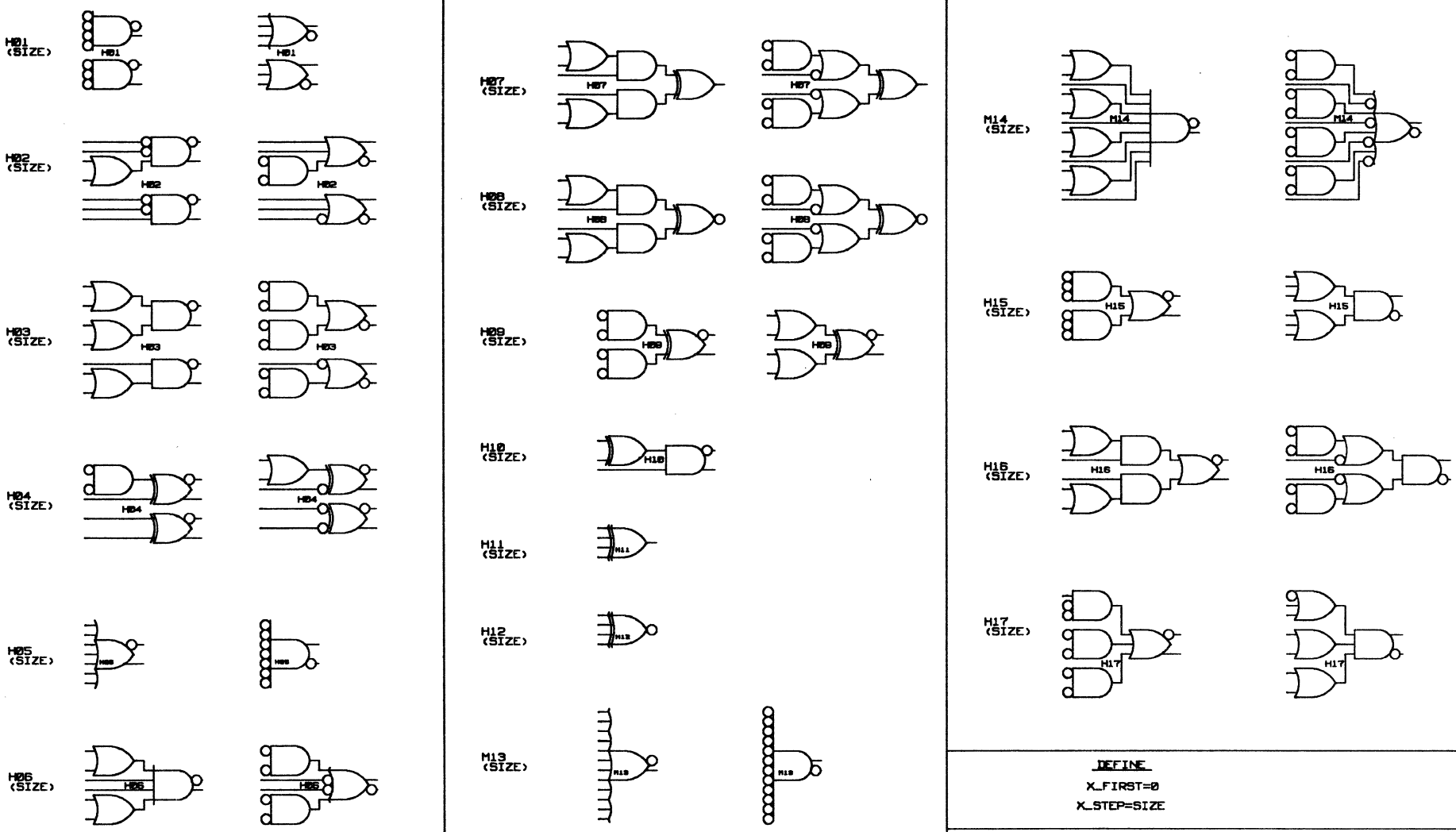
This section describes some important usage notes.

1. Special Bodies

Several special models have been created to generate the proper Motorola (LOGCAP) netlist. The WIRED OR bodies must be used to represent wire-tie connections. The IO PAD body should be used to connect bidirectional signals to the package pins. And, the BOUT body should be used to connect non-bidirectional signals to the package pins. These bodies are equivalent to Figure 4-2, page 79 of the Motorola Macrocell Array CAD Design Manual.

2. Packaging

All drawings must be packaged prior to executing the LOGCAP interface program (glogcap). GLOGCAP is a physical interface program and requires the Packager to package the pertinent cells together.



DEFINE

X_FIRST=0

X_STEP=SIZE

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

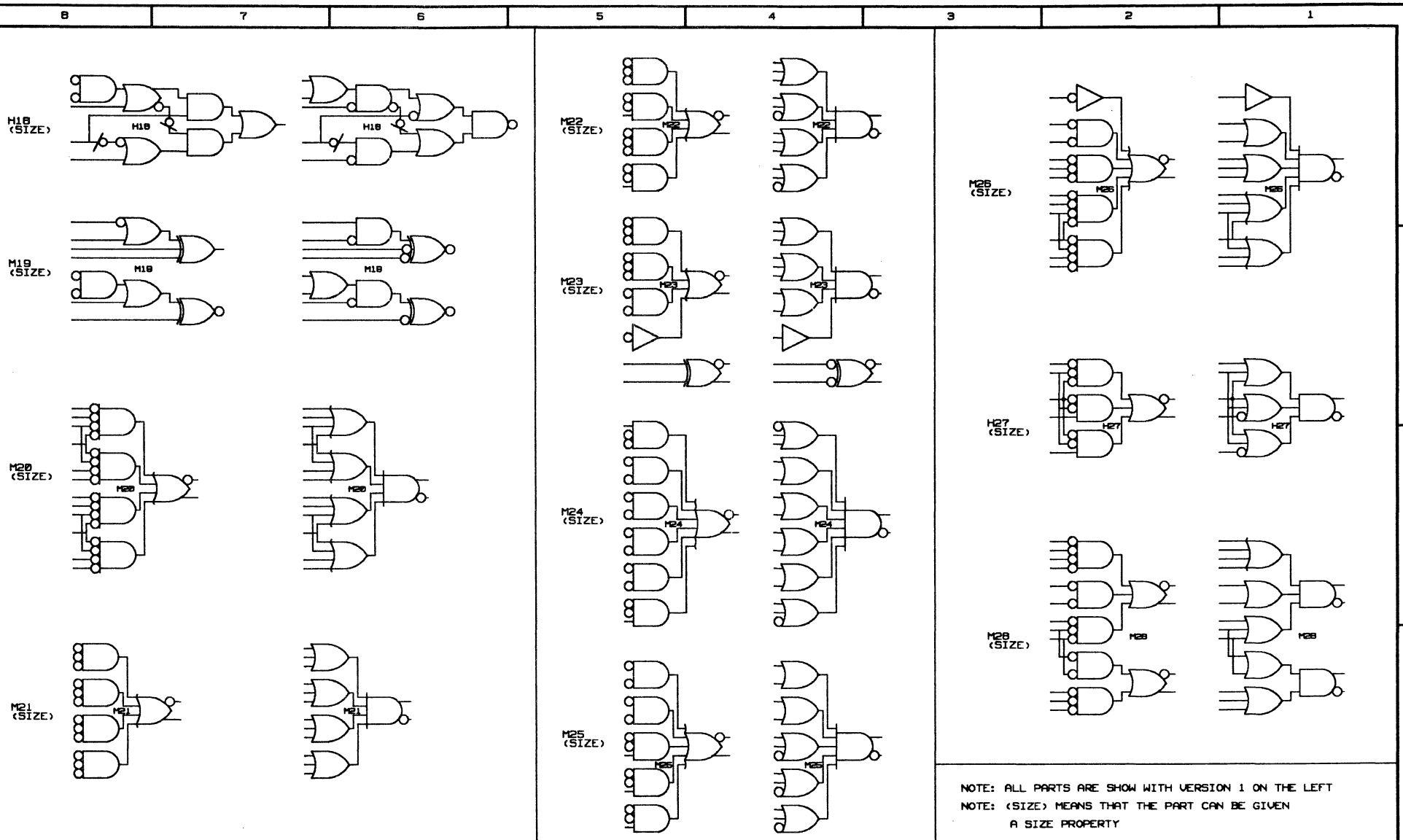
VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH MCA12 PART
ABBREV=EXMCA12

ENGINEER:
JIMMY

DATE:
THU SEP 22 17:20:12 1983

PAGE: 1 OF 10



NOTE: ALL PARTS ARE SHOW WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

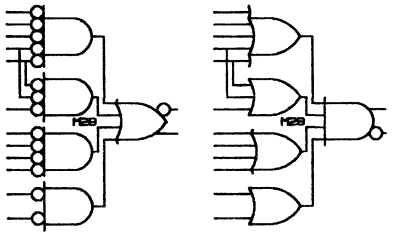
DRAWING:
TITLE=EXAMPLE OF EACH MCA12 PART
ABBREV=EXMCA12

ENGINEER:
JIMMY

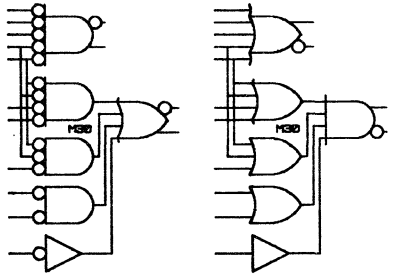
DATE:
THU SEP 1 21:54:51 1983

PAGE:
2 OF 10

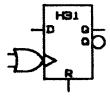
M29 (SIZE)



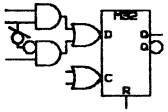
M30 (SIZE)



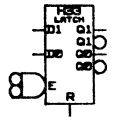
M31 (SIZE)



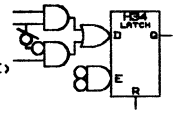
M32 (SIZE)



M33 (SIZE)

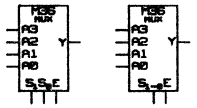


M34 (SIZE)

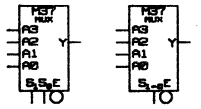


M35 (SIZE)

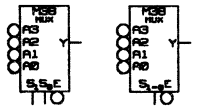
M36 (SIZE)



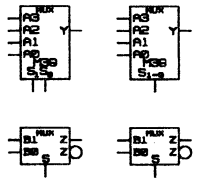
M37 (SIZE)



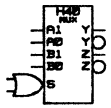
M38 (SIZE)



M39 (SIZE)

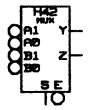


M40 (SIZE)

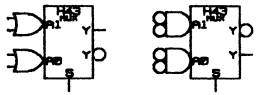


M41 (SIZE)

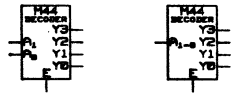
M42 (SIZE)



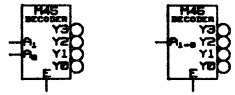
M43 (SIZE)



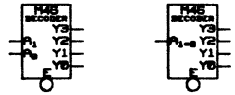
M44



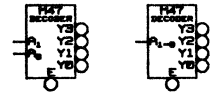
M45



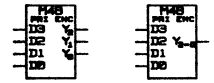
M46



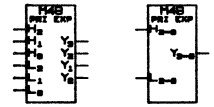
M47



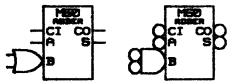
M48



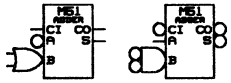
M49



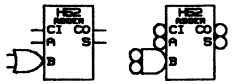
M50 (SIZE)



M51 (SIZE)



M52 (SIZE)



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH MCA12 PART
ABBREV=EXMCA12

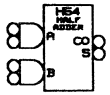
ENGINEER: JIMMY

DATE: Mon Dec 19 10:39:35 1983

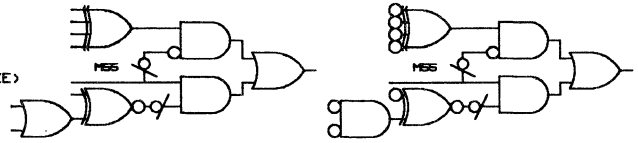
PAGE: 3 OF 10

M53 (SIZE)

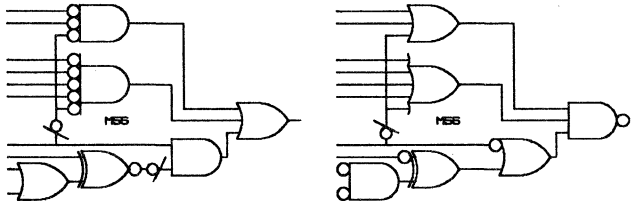
M54 (SIZE)



M55 (SIZE)

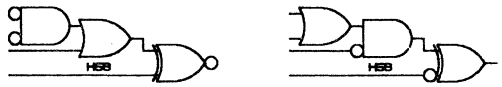


M56 (SIZE)



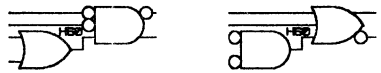
M57 (SIZE)

M58 (SIZE)

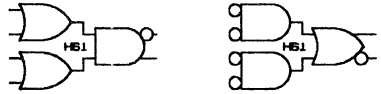


M59 (SIZE)

M60 (SIZE)



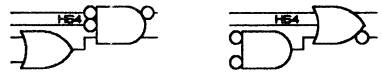
M61 (SIZE)



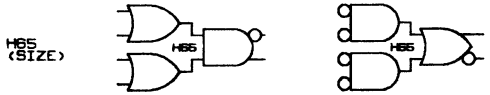
M62 (SIZE)

M63 (SIZE)

M64 (SIZE)



M65 (SIZE)



M66 (SIZE)

M67 (SIZE)

M68 (SIZE)

M69 (SIZE)

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH MCA12 PART
ABBREV=EXMCA12

ENGINEER:
JIMMY

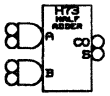
DATE:
Mon Dec 19 11:26:18 1983

PAGE: 4 OF 10

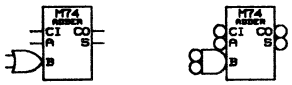
M71
(SIZE)

M72
(SIZE)

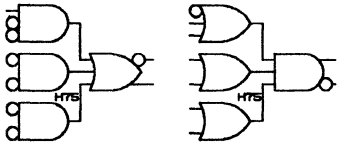
M73
(SIZE)



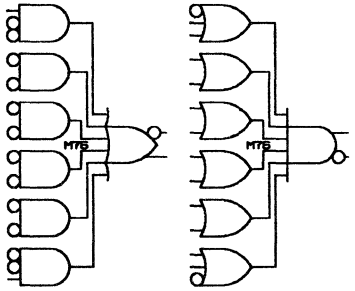
M74
(SIZE)



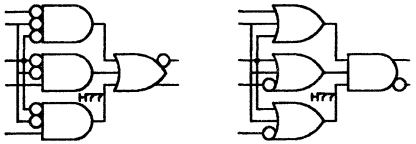
M75
(SIZE)



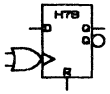
M76
(SIZE)



M77
(SIZE)



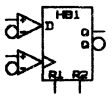
M78
(SIZE)



M79

M80
(SIZE)

M81
(SIZE)



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON LEFT
NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN
A SIZE PROPERTY

VALID
LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH MCA12 PART
ABBREV=EXMCA12

ENGINEER:
JIMMY

DATE:
THU SEP 1 19:16:18 1983

PAGE:
5 OF 10

I01 (SIZE)

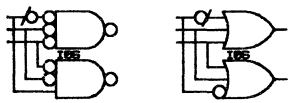
I02 (SIZE)

I03 (SIZE)

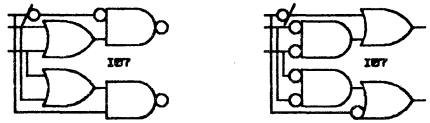
I04 (SIZE)

I05 (SIZE)

I06 (SIZE)



I07 (SIZE)



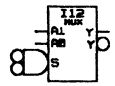
I08 (SIZE)

I09 (SIZE)

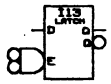
I10 (SIZE)

I11 (SIZE)

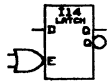
I12 (SIZE)



I13 (SIZE)



I14 (SIZE)



I15 (SIZE)

I16 (SIZE)

I20 (SIZE)

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH MCA12 PART
ABBREV=EXMCA12

ENGINEER:
JIMMY

DATE:
THU SEP 22 17:35:11 1983

PAGE:
6 OF 10

T01 (SIZE)

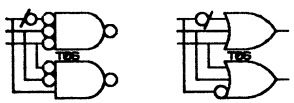
T02 (SIZE)

T03 (SIZE)

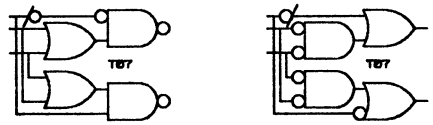
T04 (SIZE)

T05 (SIZE)

T06 (SIZE)



T07 (SIZE)



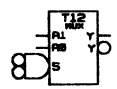
T08 (SIZE)

T09 (SIZE)

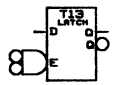
T10 (SIZE)

T11 (SIZE)

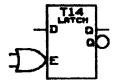
T12 (SIZE)



T13 (SIZE)



T14 (SIZE)



T20 (SIZE)

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH MCA12 PART
ABBREV=EXMCA12

ENGINEER:
JIMMY

DATE:
THU SEP 22 17:39:04 1983

PAGE: 7 OF 10

001 (SIZE)

002 (SIZE)

003 (SIZE)

004 (SIZE)

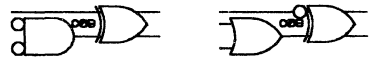
005 (SIZE)

006 (SIZE)

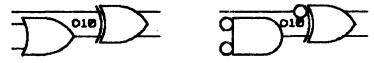
007 (SIZE)

008 (SIZE)

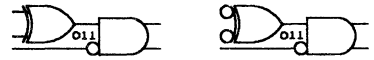
009 (SIZE)



010 (SIZE)



011 (SIZE)



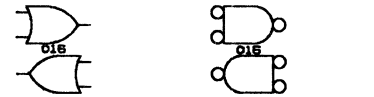
012 (SIZE)

013 (SIZE)

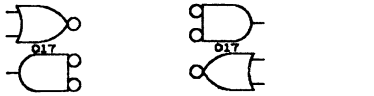
014 (SIZE)

015 (SIZE)

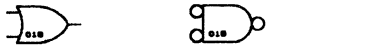
016 (SIZE)



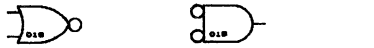
017 (SIZE)



018 (SIZE)

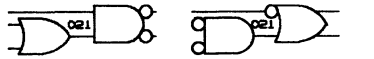


019 (SIZE)

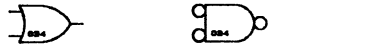


020 (SIZE)

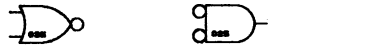
021 (SIZE)



024 (SIZE)



025 (SIZE)



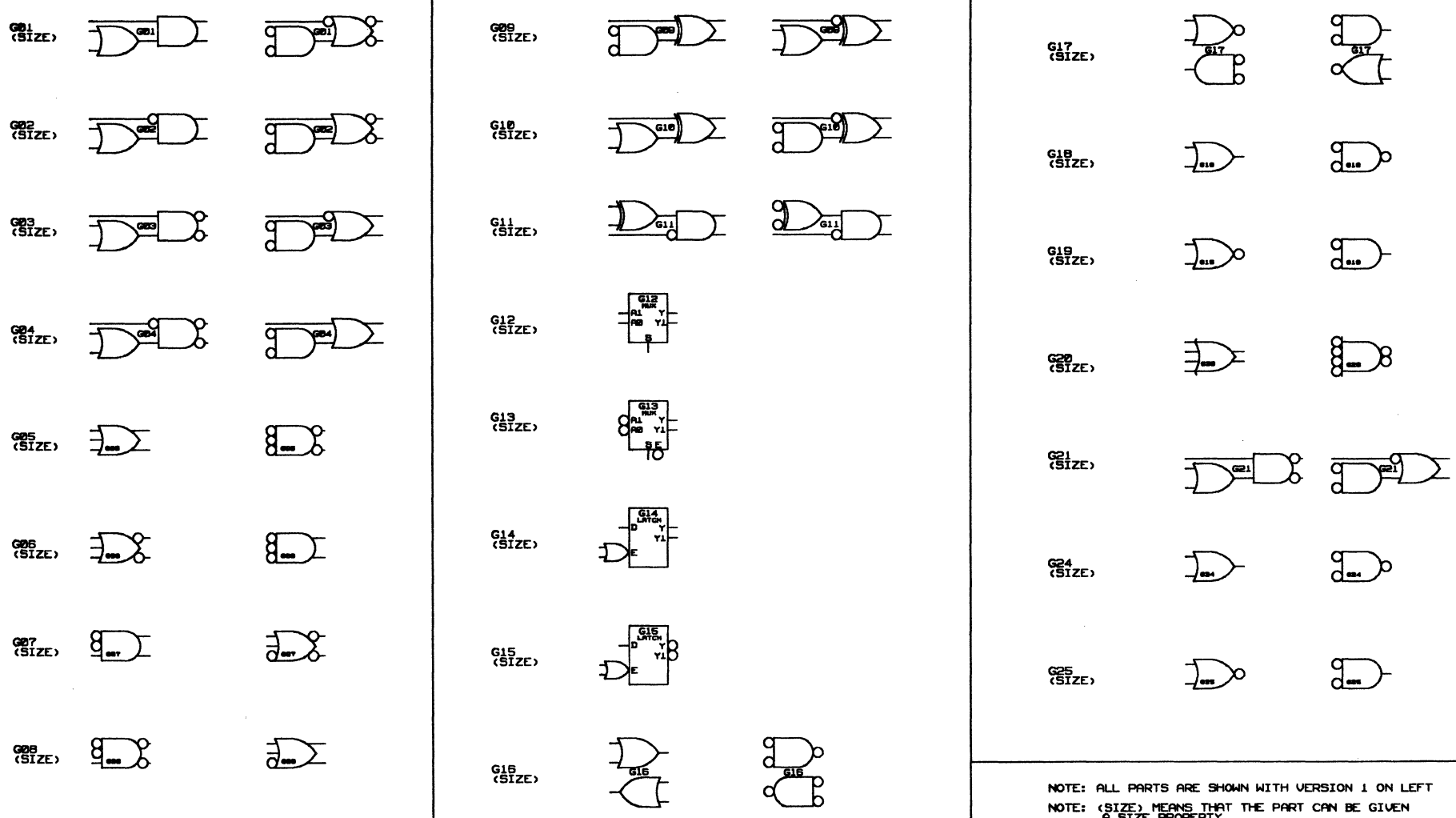
NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON LEFT
NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH MCA12 PART
ABBREV=EXMCA12

ENGINEER:
JIMMY

DATE:
THU SEP 22 17:42:04 1983



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH MCA12 PART
ABBREV=EXMCA12

ENGINEER:
JIMMY

DATE:
THU SEP 22 17:45:54 1983

THESE BODIES ARE USED TO REPRESENT WIRE-TIE CONNECTIONS.

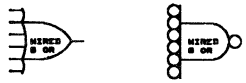
WIRED 2 OR (SIZE)

WIRED 3 OR (SIZE)

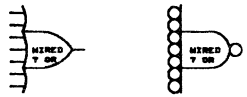
WIRED 4 OR (SIZE)

WIRED 5 OR (SIZE)

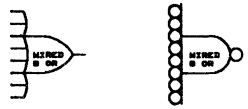
WIRED 6 OR (SIZE)



WIRED 7 OR (SIZE)



WIRED 8 OR (SIZE)



IO PAD (SIZE)

THIS BODY IS USED FOR BIDIRECTIONAL SIGNAL.
'PAD OUT' IS CONNECTED TO OUTPUT FLAG BODY,
'PAD IN' IS CONNECTED TO INPUT FLAG BODY,
'OUT' IS CONNECTED TO MACROCELL ARRAY OUTPUT, AND
'IN' IS CONNECTED TO MACROCELL ARRAY INPUT.

BOUT (SIZE)

THIS BODY IS USED FOR MACROCELL ARRAY OUTPUTS.
THE 'IN' PIN IS CONNECTED TO THE OUTPUT OF THE
MACROCELL ARRAY, AND THE 'OUT' PIN IS CONNECTED
TO AN OUTPUT FLAG BODY.

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON LEFT
NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN
A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH MCA12 PART
ABBREV=EXMCA12

ENGINEER:
RKM

DATE:
Wed Dec 28 14:20:00 1983

PAGE:
10 OF 10

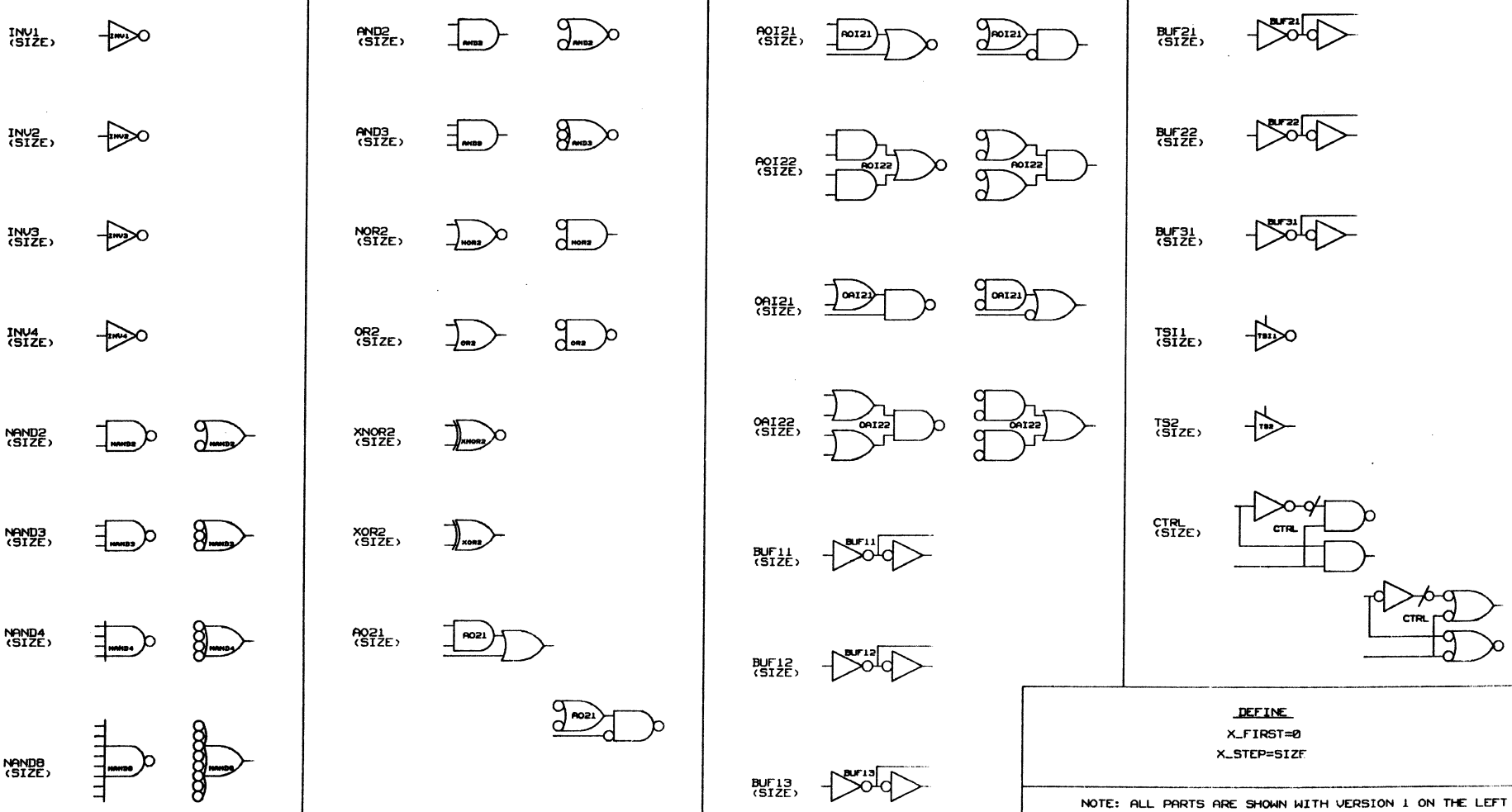
MCLDL Library

There have been no changes in the MCLDL Library since Release 3.3

The MCLDL Library requires approximately 0.26 MBy of disk storage on the S-32. It contains bodies and physical for the following components from the McLDL (United Technologies) Highland gate array cells:

AND2	2-input AND
AND3	3-input AND
AO21	3-input AND-OR
AOI21	3-input AND-OR-INVERT
AOI22	4-input AND-OR-INVERT
BUF11	buffer - true and complement outputs
BUF12	buffer - true (2 parallelled) and complement outputs
BUF13	buffer - true (3 parallelled) and complement outputs
BUF21	buffer - true and complement (2 parallelled) outputs
BUF22	buffer - true (2 parallelled) and complement (2 parallelled) outputs
BUF31	buffer - true and complement (3 parallelled) outputs
CGDVR	CMOS I/O driver
CIDVR	CMOS input buffer
CODVR	CMOS output driver
CSDVR	CMOS bidirectional I/O driver
CTRL	three-state output controller
DFF	D-type flip-flop
DFFAC	D-type flip-flop with active high asynchronous clear
DFFAP	D-type flip-flop with active low asynchronous preset
DFFAPC	D-type flip-flop with active low asynchronous preset and clear
DFFSC	D-type flip-flop with active high synchronous clear
DFFSP	D-type flip-flop with active low synchronous preset
DL	data latch
DLC	data latch with active high clear
DLP	data latch with active low preset
INV1	inverter
INV2	inverter-2 parallelled
INV3	inverter-3 parallelled
INV4	inverter-4 parallelled
IWPDVR	special-purpose input buffer
NAND2	2-input NAND
NAND3	3-input NAND
NAND4	4-input NAND
NAND8	8-input NAND
NOR2	2-input NOR
OAI21	3-input OR-AND-INVERT
OAI22	4-input OR-AND-INVERT
OR2	2-input OR

OSC	crystal-controlled oscillator
SR00	S-R latch - NAND implementation
SR11	S-R latch - NOR implementation
TDIDVR	TTL input buffer with pull-down
TGDVR	TTL I/O driver
TIDVR	TTL input buffer
TODVR	TTL output driver
TS2	three-state buffer
TSDVR	TTL bidirectional I/O driver
TSI1	three-state inverter
TUIDVR	TTL input buffer with pull-up
XNOR2	2-input exclusive NOR
XOR2	2-input exclusive OR



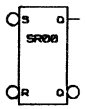
VALID LOGIC SYSTEMS INCORPORATED

DRAWING: TITLE=EXAMPLE OF EACH MCLDL PRIMITIVE ABBREV=EXMCLDL

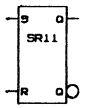
ENGINEER: JIMMY

DATE: Fri Sep 23 11:24:56 1983

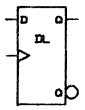
SR00 (SIZE)



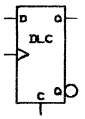
SR11 (SIZE)



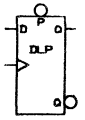
DL (SIZE)



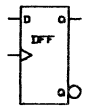
DLC (SIZE)



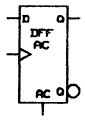
DLP (SIZE)



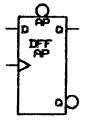
DFF (SIZE)



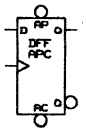
DFFAC (SIZE)



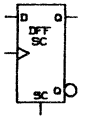
DFFAP (SIZE)



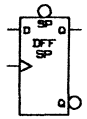
DFFAPC (SIZE)



DFFSC (SIZE)



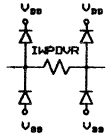
DFFSP (SIZE)



TIDVR (SIZE)

CIDVR (SIZE)

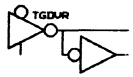
IWPDVR (SIZE)



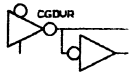
TUIDVR (SIZE)

TDIDVR (SIZE)

TGDVR (SIZE)

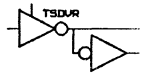


CGDVR (SIZE)



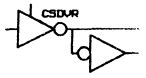
CODVR (SIZE)

TSDVR (SIZE)

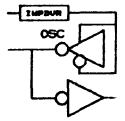


TODVR (SIZE)

CSDVR (SIZE)



OSC (SIZE)



NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH MCLDL PRIMITIVE
ABBREV=EXMCLDL

ENGINEER:
JIMMY

DATE:
Tue Sep 20 11:47:00 1983

AMD1850 Library

Release 4.4 is the first complete release of the AMD1850 Library.

The AMD1850 Library requires approximately 1.5 MBy of disk storage on the S-32. It contains bodies and physical for the following 200 components from the AMD 1250/1850 Gate Array family:

H01	3 & 4-input active low and/nand gates
H02	4-input or and/nand + 3-input and/nand
H03	2-wide 2-input or and/nand + 2-input or and/nand
H04	2-input active low and exclusive or + 2-input exclusive or gate
H05	7-input or/nor gate
H06	2-wide 2-input or 4-input and/nand gate
H07	6-input or and exclusive or gating structure
H08	6-input or and exclusive nor gating structure
H09	2-wide 2-input active low and exclusive or/nor
H10	exclusive or and/nand
H11	4-input exclusive or
H12	4-input exclusive nor
H15	2-wide 3-input active low or/nor
H16	6-input or and or/nor gating structure
H17	3-wide 3-2-2 and or/nor
H18	5-input gating structure
H27	3-wide 3-3-3 input and or/nor gating structure
H31	D flip-flop with reset and 2 clocks
H33	2-bit latch with 2 enable inputs
H34	latch with a 2-input data mux and 2 enable inputs
H35	dual latch
H40	dual 2-to-1 mux with column select
H41	dual 2-to-1 mux
H42	dual 2-to-1 mux with common enable and select inputs
H43	2-wide 2-input or 2-to-1 mux
H52	full adder
H54	half adder
H57	4-input gating structure
H58	4-input gating structure
H59	dual 4-input active low and/nand gates
H60	dual or and/nand gates
H61	dual 2-wide 2-input or and/nand gates
H62	dual 2-input active low and exclusive or/nor gates
H63	4-input active low and/nand gate
H64	2-input or 2-input active low and/nand gate
H65	2-wide 2-input or and/nand gate
H66	2-input active low or exclusive or/nor gate
H67	latch
H69	2-to-1 mux

H71	4-input exclusive or gate
H72	full adder
H73	half adder
H75	3-wide 3-2-2 active low and or/nor gate
H77	3-wide 3-input and or/nor gating structure
H78	D flip-flop with 2 clock inputs
H81	D flip-flop with differential clock and data
H82	D flip-flop with set & preset (async)
I400	TTL to PECL input buffer
I401	TTL to TECL input buffer
I402	TECL to TECL input buffer
I403	TTL to PECL 4-input or/nor gate
I404	TTL to PECL 2-wide 2-input or and/nand gate
I405	TTL to PECL 2-input or and/nand gate with enable
I406	TTL to PECL 2-input or and/nand gate with enable
I407	high speed TTL to PECL inverting buffer
LH201	D flip-flop with asynchronous reset
LH231	D flip-flop with 2 data inputs
LH293	latch with reset and 2 data and enable inputs
M13	12-input or/nor gate
M14	12-input or and/nand gating structure
M19	or exclusive or + and or exclusive nor gating structures
M20	4-wide 4-input active low and or/nor gating structure
M21	4-wide 3-input active low and or/nor gating structure
M22	4-wide 3-input and or/nor gating structure
M23	4-wide 3-3-2-1 and or/nor + 2-input exclusive or/nor gate
M24	6-wide 3-2-2-2-2-3 input and or/nor gating structure
M25	5-wide 2-2-3-3-3 input and or/nor gating structure
M26	5-wide 1-2-3-4-4 and or/nor gating structure
M28	3-wide 4-2-3 input and or/nor + 2 wide 2-3 and or/nor
M29	4-wide 5-3-4-2 input and or/nor gating structure
M30	4-wide 4-3-2-1 and or/nor + 5 input active low and/nand gate
M32	D flip-flop with a D-input mux and 2 clocks
M36	4-to-1 mux with enable
M37	4-to-1 mux with active low enable
M38	4-to-1 inverting mux with enable
M39	dual mux 4-to-1 and 2-to-1
M44	1-of-4 decoder (active high)
M45	1-of-4 decoder (active low)
M46	1-of-4 decoder (active high)
M47	1-of-4 decoder (active low)
M48	priority encoder
M49	priority expander
M50	full adder
M51	full adder
M53	full adder + half adder
M55	8-input gating structure
M56	9-input gating structure
M68	4-to-1 mux
M74	full adder
M76	6-wide 3-2-2-2-2-3 input and or/nor gating structure

M79	1-of-4 decoder (active low)
M80	4-to-1 mux with enable
0400	TECL to TECL buffer
0401	TECL to TECL inverting output buffer
0402	TECL to TECL 4-input or output buffer
0403	TECL to TECL 4-input nor output buffer
0404	TECL to TECL 2-wide 2-input or and gate
0405	TECL to TECL 2-wide 2-input or nand gate
0406	PECL to TTL output buffer 8 ma
0407	PECL to TTL inverting output buffer 8 ma
0408	PECL to TTL 16 ma output buffer with three-state
0409	PECL to TTL inverting 16 ma output buffer with three-state
0410	PECL to TTL 4-input or gate 8 ma
0411	PECL to TTL 4-input or gate 16 ma with three-state
0412	PECL to TTL 4-input nor gate 8 ma
0413	PECL to TTL 4-input nor gate 16 ma with three-state
0414	PECL to TTL 2-wide 2-input or and gate 8 ma
0415	PECL to TTL 2-wide 2-input or and gate 16 ma with three-state
0416	PECL to TTL 2-wide 2-input or nand gate 8 ma
0417	PECL to TTL 2-wide 2-input or nand gate 16 ma with three-state
0418	TECL to TTL output buffer 8 ma
0419	TECL to TTL inverting output buffer 8 ma
0420	TECL to TTL 16 ma output buffer with three-state
0421	TECL to TTL inverting 16 ma output buffer with three-state
0422	TECL to TTL 4-input or gate 8 ma
0423	TECL to TTL 4-input or gate 16 ma with three-state
0424	TECL to TTL 4-input nor gate 8 ma
0425	TECL to TTL 4-input nor gate 16 ma with three-state
0426	TECL to TTL 2-wide 2-input or and gate 8 ma
0427	TECL to TTL 2-wide 2-input or and gate 16 ma with three-state
0428	TECL to TTL 2-wide 2-input or nand gate 8 ma
0429	TECL to TTL 2-wide 2-input or nand gate 16 ma with three-state
0438	TECL to ECL output buffer with differential drive
PEI	pad for ecl input buffer
PEO	pad for ecl output buffer
PTI	pad for TTL input buffer
PTO8	pad for TTL output buffer 8ma
PTO16	pad for TTL output buffer 16ma

APPLICATION NOTE

This section describes some important usage notes.

1. Simulation

Two simulation models are provided. The first model is the regular Valid simulation model. The second model corresponds to the AMD TEGAS model which uses a 2 unit-delay for every cell.

To select which simulation model to use, add the following directive to the compiler.cmd file:

```
text_macro_file 'textmacro.dat';
```

Then create file 'textmacro.dat' with the following contents:

```
FILE_TYPE=TEXT_MACROS;
AMD_SIM_MODEL='X';
END.
```

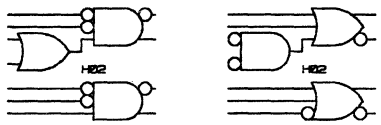
where X = 1 to select VALID,
and X = 2 to select AMD sim model

2. Special Bodies

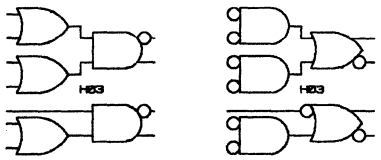
Several special models have been created to generate the proper TEGAS netlist. PEI and PTI are used to represent ECL and TTL input cell connections respectively. PEO is used to represent ECL output cell connections, and PTO8, and PTO16 are used to represent 8ma and 16ma TTL output cell connections respectively.

H01 (SIZE)

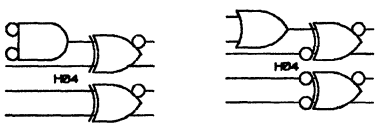
H02 (SIZE)



H03 (SIZE)

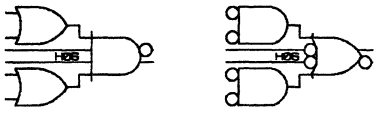


H04 (SIZE)

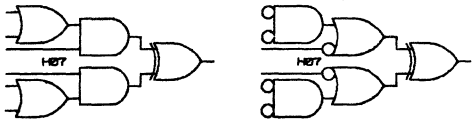


H05 (SIZE)

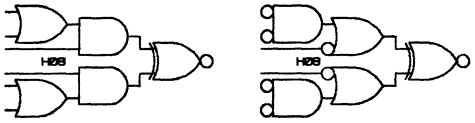
H06 (SIZE)



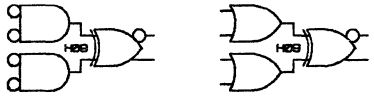
H07 (SIZE)



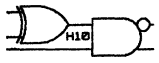
H08 (SIZE)



H09 (SIZE)



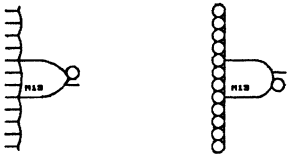
H10 (SIZE)



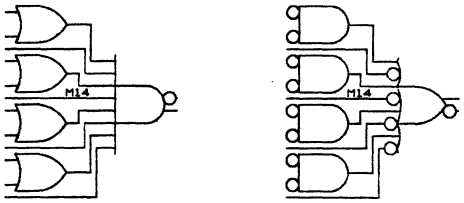
H11 (SIZE)

H12 (SIZE)

M13 (SIZE)

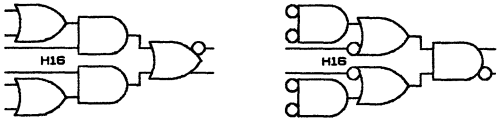


M14 (SIZE)

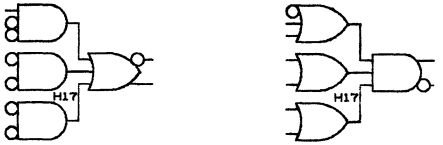


H15 (SIZE)

H16 (SIZE)



H17 (SIZE)



DEFINE

X_FIRST=0

X_STEP=SIZE

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

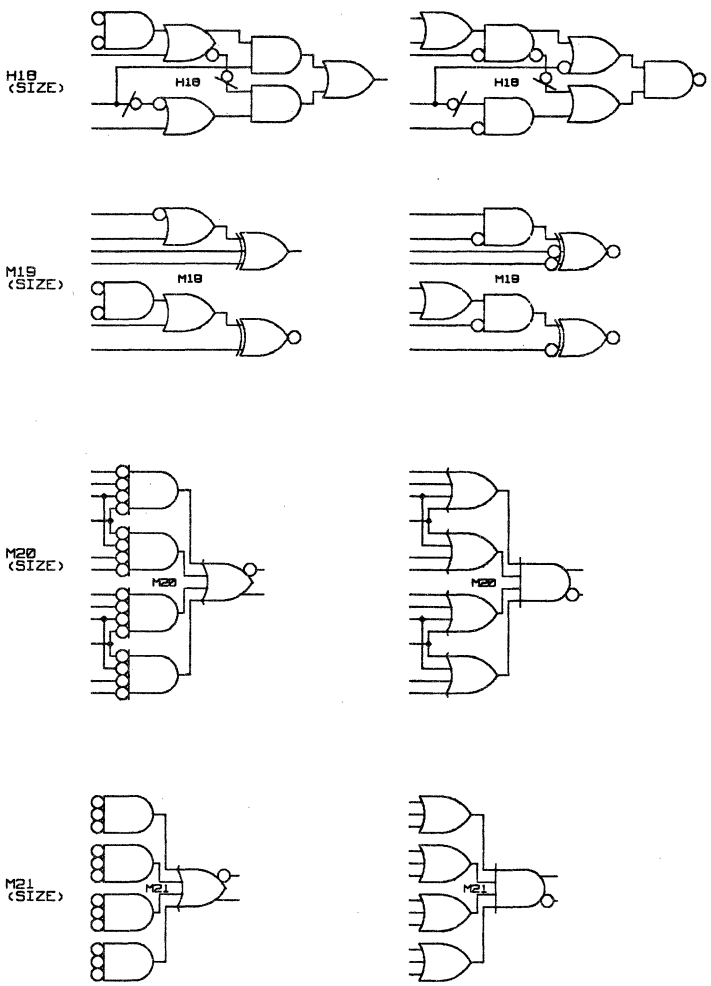
NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

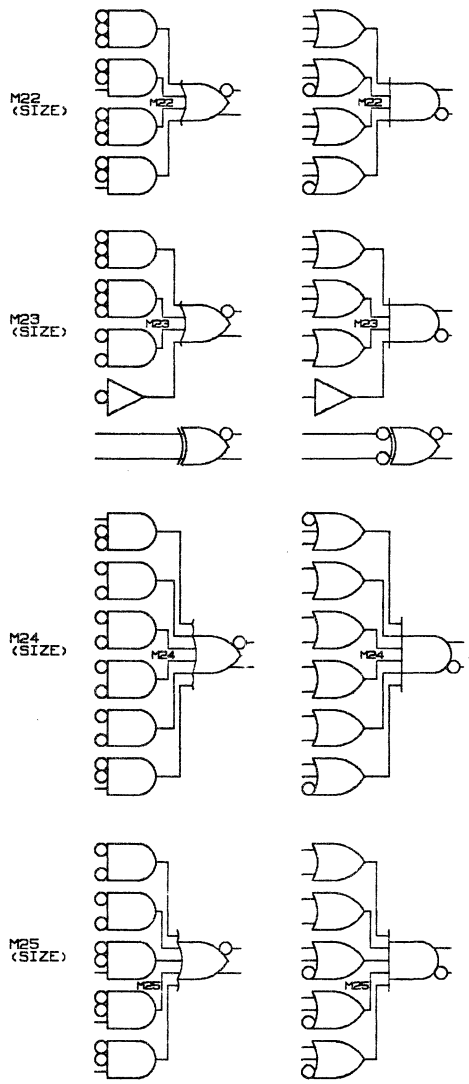
VALID LOGIC SYSTEMS INCORPORATED

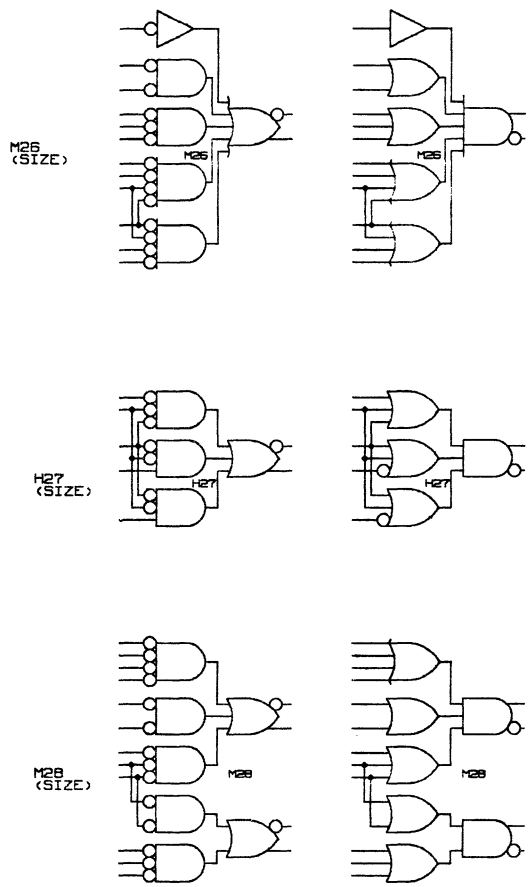
DRAWING:
TITLE=EXAMPLE OF EACH AMD1850 PART
ABBREV=EXAMD1850

ENGINEER: RRJ

DATE: Mon Mar 12 14:47:15 1984







NOTE: ALL PARTS ARE SHOW WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

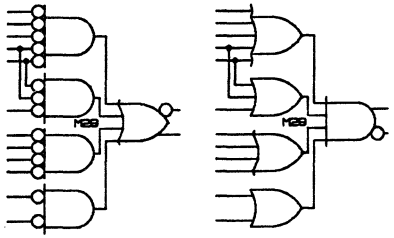
DRAWING:
TITLE=EXAMPLE OF EACH AMD1850 PART
ABBREV=EXAMD1850

ENGINEER:
RRJ

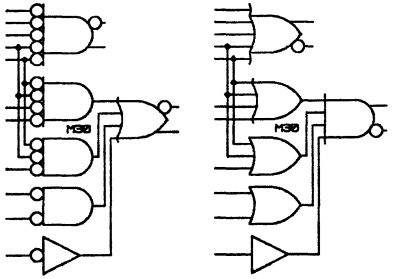
DATE:
Mon Mar 12 14:48:57 1984

PAGE:
2 OF 9

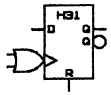
M29
(SIZE)



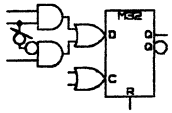
M30
(SIZE)



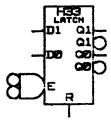
M31
(SIZE)



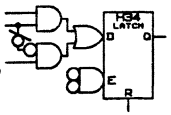
M32
(SIZE)



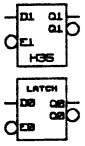
M33
(SIZE)



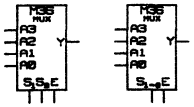
M34
(SIZE)



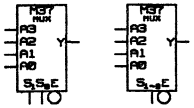
M35
(SIZE)



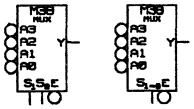
M36
(SIZE)



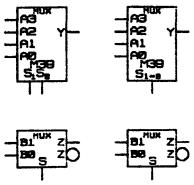
M37
(SIZE)



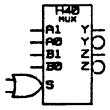
M38
(SIZE)



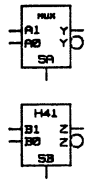
M39
(SIZE)



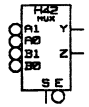
M40
(SIZE)



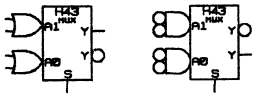
M41
(SIZE)



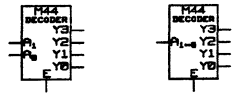
M42
(SIZE)



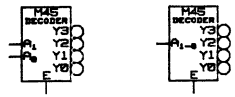
M43
(SIZE)



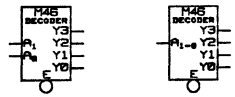
M44



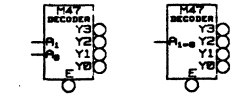
M45



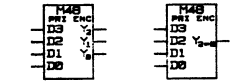
M46



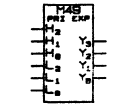
M47



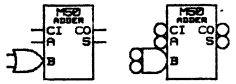
M48



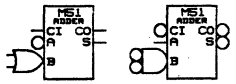
M49



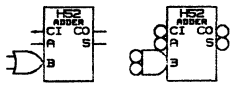
M50
(SIZE)



M51
(SIZE)



M52
(SIZE)



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID
LOGIC SYSTEMS INCORPORATED

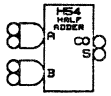
DRAWING:
TITLE=EXAMPLE OF EACH AMD1850 PART
ABBREV=EXAMD1850

ENGINEER:
RRJ

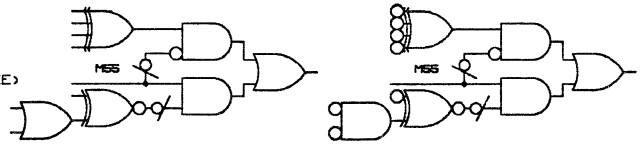
DATE:
Mon Mar 12 14:53:20 1984

M53 (SIZE)

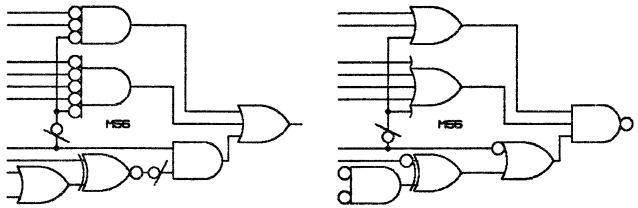
M54 (SIZE)



M55 (SIZE)

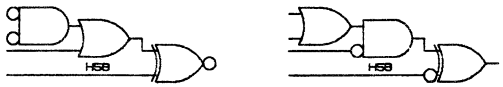


M56 (SIZE)



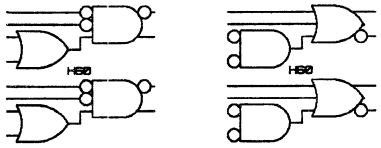
M57 (SIZE)

M58 (SIZE)

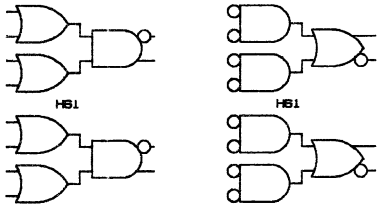


M59 (SIZE)

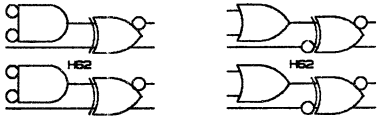
M60 (SIZE)



M61 (SIZE)

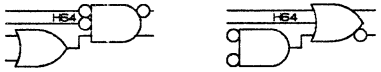


M62 (SIZE)

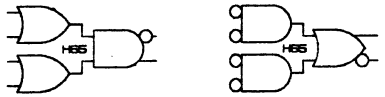


M63 (SIZE)

M64 (SIZE)



M65 (SIZE)



M66 (SIZE)

M67 (SIZE)

M68 (SIZE)

M69 (SIZE)

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: TITLE=EXAMPLE OF EACH AMD1850 PART ABBREV=EXAMD1850

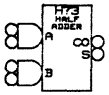
ENGINEER: RRJ

DATE: Mon Mar 12 14:56:24 1984

M71
(SIZE)

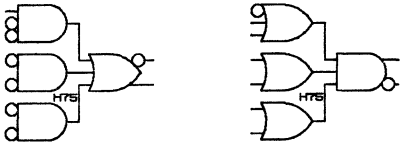
M72
(SIZE)

M73
(SIZE)

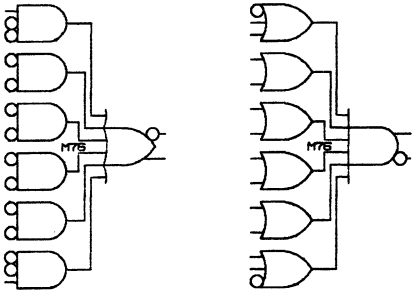


M74
(SIZE)

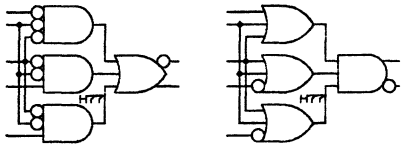
M75
(SIZE)



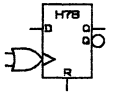
M76
(SIZE)



M77
(SIZE)



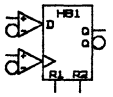
M78
(SIZE)



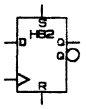
M79

M80
(SIZE)

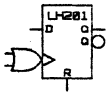
M81
(SIZE)



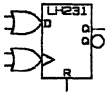
M82
(SIZE)



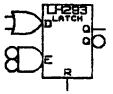
LH201
(SIZE)



LH231
(SIZE)



LH293
(SIZE)



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON LEFT
NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN
A SIZE PROPERTY

VALID
LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH AMD1850 PART
ABBREV=EXAMD1850

ENGINEER:
RRJ

DATE:
Mon Mar 12 14:58:31 1984

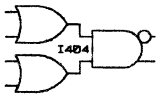
I400
(SIZE)

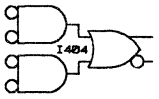
I401
(SIZE)

I402
(SIZE)

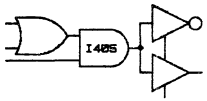
I403
(SIZE)

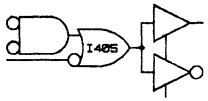
I404
(SIZE)



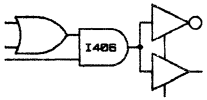


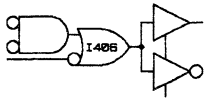
I405
(SIZE)





I406
(SIZE)





I407
(SIZE)

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH AMD1850 PART
ABBREV=EXAMD1850

ENGINEER:
RRJ

DATE:
Mon Mar 12 14:59:44 1984

PAGE:
6 OF 9

0400 (SIZE)

0401 (SIZE)

0402 (SIZE)

0403 (SIZE)

0404 (SIZE)

0405 (SIZE)

0406 (SIZE)

0407 (SIZE)

0408 (SIZE)

0409 (SIZE)

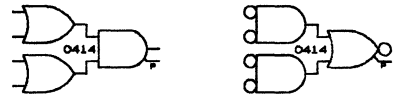
0410 (SIZE)

0411 (SIZE)

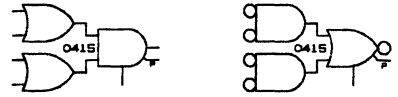
0412 (SIZE)

0413 (SIZE)

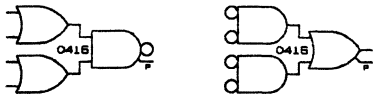
0414 (SIZE)



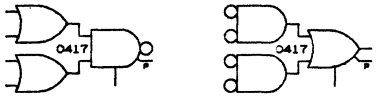
0415 (SIZE)



0416 (SIZE)



0417 (SIZE)



0418 (SIZE)

0419 (SIZE)

0420 (SIZE)

0421 (SIZE)

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH AMD1850 PART
ABBREV=EXAMD1850

ENGINEER:
RRJ

DATE:
Mon Mar 12 15:01:35 1984

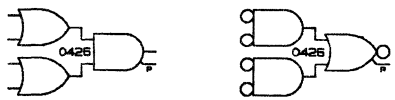
0422 (SIZE)

0423 (SIZE)

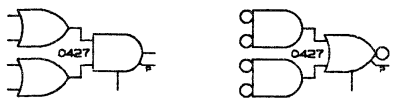
0424 (SIZE)

0425 (SIZE)

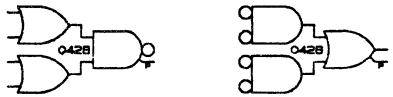
0426 (SIZE)



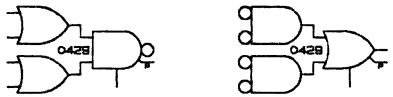
0427 (SIZE)



0428 (SIZE)



0429 (SIZE)



0438 (SIZE)

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH AMD1850 PART
ABBREV=EXAMD1850

ENGINEER: RRJ

DATE: Mon Mar 12 15:12:39 1984

PAGE: 8 OF 9

INPUT PAD CELLS

PEI (SIZE)

PAD FOR ECL INPUT CELL
'IN' IS CONNECTED TO AN INPUT FLAG BODY
'CELL' IS CONNECTED TO AN ECL INPUT CELL

PTI (SIZE)

PAD FOR TTL INPUT CELL
'IN' IS CONNECTED TO AN INPUT FLAG BODY
'CELL' IS CONNECTED TO A TTL-ECL INPUT CELL

OUTPUT PAD CELLS

PEO (SIZE)

PAD FOR ECL OUTPUT CELL
'CELL' IS CONNECTED TO AN ECL OUPUT CELL
'OUT' IS CONNECTED TO AN OUTPUT FLAG BODY

PTO8 (SIZE)

PAD FOR TTL OUTPUT CELL 8MA
'CELL' IS CONNECTED TO A 8MA ECL-TTL OUPUT CELL
'P' IS CONNECTED TO THE P PIN OF THE ECL-TTL OUTPUT CELL
'OUT' IS CONNECTED TO AN OUTPUT FLAG BODY

PTO16 (SIZE)

PAD FOR TTL OUTPUT CELL 16MA
'CELL' IS CONNECTED TO A 16MA ECL-TTL OUPUT CELL
'P' IS CONNECTED TO THE P PIN OF THE ECL-TTL OUTPUT CELL
'OUT' IS CONNECTED TO AN OUTPUT FLAG BODY

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH AMD1850 PART
ABBREV=EXAMD1850

ENGINEER:
RRJ

DATE:
Mon Mar 12 17:55:17 1984

HILO Library

There have been no changes in the HILO Library since Release 4.0.

The HILO Library requires approximately 0.3 MBy of disk storage on the S-32. It contains bodies and physical for the following 54 components:

1BUF	1-output buffer
1NOT	1-output not gate
2AND	2-input and gate
2BUF	2-output buffer
2NAND	2-input nand gate
2NOR	2-input nor gate
2NOT	2-output not gate
2OR	2-input or gate
3AND	3-input and gate
3BUF	3-output buffer
3NAND	3-input nand gate
3NOR	3-input nor gate
3NOT	3-output not gate
3OR	3-input or gate
4AND	4-input and gate
4BUF	4-output buffer
4NAND	4-input nand gate
4NOR	4-input nor gate
4NOT	4-output not gate
4OR	4-input or gate
5AND	5-input and gate
5BUF	5-output buffer
5NAND	5-input nand gate
5NOR	5-input nor gate
5NOT	5-output not gate
5OR	5-input or gate
6AND	6-input and gate
6BUF	6-output buffer
6NAND	6-input nand gate
6NOR	6-input nor gate
6NOT	6-output not gate
6OR	6-input or gate
7AND	7-input and gate
7BUF	7-output buffer
7NAND	7-input nand gate
7NOR	7-input nor gate
7NOT	7-output not gate
7OR	7-input or gate
8AND	8-input and gate
8BUF	8-output buffer
8NAND	8-input nand gate

8NOR	8-input nor gate
8NOT	8-output not gate
8OR	8-input or gate
BALR	2-input balanced line receiver
BUFIF0	tristate driver with inverting control input
BUFIF1	tristate driver with noninverting control input
CAPACITOR	capacitance element
CLOCK0	clock-generating oscillating element with noninverting output
CLOCK1	clock-generating oscillating element with inverting output
NOTIF0	tristate driver with inverting control input
NOTIF1	tristate driver with noninverting control input
TRANIF0	MOS transmission gate with inverting control input
TRANIF1	MOS transmission gate with noninverting control input

1BUF (SIZE)

2BUF (SIZE)

3BUF (SIZE)

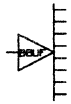
4BUF (SIZE)

5BUF (SIZE)

6BUF (SIZE)

7BUF (SIZE)

8BUF (SIZE)



1NOT (SIZE)

2NOT (SIZE)

3NOT (SIZE)

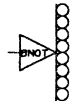
4NOT (SIZE)

5NOT (SIZE)

6NOT (SIZE)

7NOT (SIZE)

8NOT (SIZE)



CLOCK0 (SIZE)

CLOCK1 (SIZE)

BUFIF0 (SIZE)

BUFIF1 (SIZE)

NOTIF0 (SIZE)

NOTIF1 (SIZE)

TRANIF0 (SIZE)

TRANIF1 (SIZE)

BALR (SIZE)

CAPACITOR (SIZE)

DEFINE

X_FIRST=0

X_STEP=SIZE

DRAWING

TITLE=EXAMPLE OF EACH HILO PRIMITIVE

ABBREV=EXHILO

LAST_MODIFIED=Wed Oct 19 11:45:50 1983

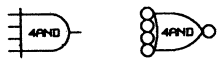
VALID LOGIC SYSTEMS INCORPORATED

TITLE:	DATE:
EXAMPLE OF EACH HILO PRIMITIVE	7/20/83
ENGINEER: AM	PAGE: 1 OF 3

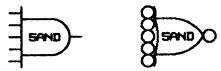
2AND (SIZE)

3AND (SIZE)

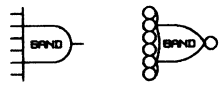
4AND (SIZE)



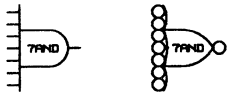
5AND (SIZE)



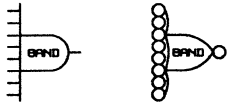
8AND (SIZE)



7AND (SIZE)



8AND (SIZE)



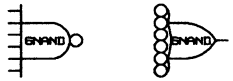
2NAND (SIZE)

3NAND (SIZE)

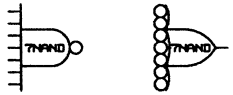
4NAND (SIZE)

5NAND (SIZE)

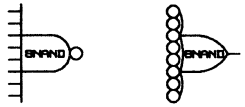
6NAND (SIZE)



7NAND (SIZE)



8NAND (SIZE)

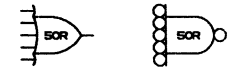


20R (SIZE)

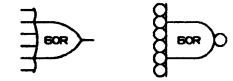
30R (SIZE)

40R (SIZE)

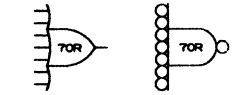
50R (SIZE)



60R (SIZE)



70R (SIZE)



DRAWING

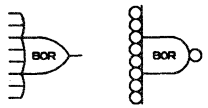
TITLE=EXAMPLE OF EACH HILO PRIMITIVE
ABBREV=EXHILO

LAST_MODIFIED=Wed Oct 19 11:48:05 1983

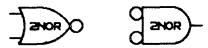
VALID LOGIC SYSTEMS INCORPORATED

TITLE:	DATE:
EXAMPLE OF EACH HILO PRIMITIVE	7/20/83
ENGINEER: AM	PAGE: 2 OF 3

8OR (SIZE)

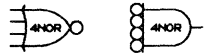


2NOR (SIZE)

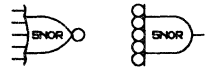


3NOR (SIZE)

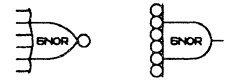
4NOR (SIZE)



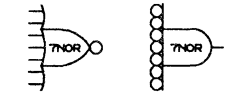
5NOR (SIZE)



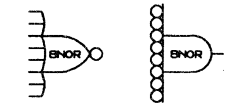
6NOR (SIZE)



7NOR (SIZE)



8NOR (SIZE)



DRAWING

TITLE=EXAMPLE OF EACH HILO PRIMITIVE
ABBREV=EXHILO
LAST_MODIFIED=Wed Oct 19 11:49:13 1983

VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH HILO PRIMITIVE	DATE: 7/20/83
ENGINEER: AM	PAGE: 3 OF 3

LOGCAP Library

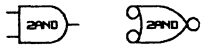
There have been no changes in the LOGCAP Library since Release 4.0.

The LOGCAP Library requires approximately 0.5 MBy of disk storage on the S-32. It contains bodies and physical for the following 65 components:

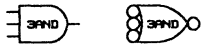
2,2 AO	2-2 input and-or gate
2,2 AOI	2-2 input and-or-invert gate
2,2 OA	2-2 input or-and gate
2,2 OAI	2-2 input or-and-invert gate
256x4 RAM	256 x 4 random access memory
256x8 ROM	256 x 8 read-only memory
2AND	2-input and gate
2NAND	2-input nand gate
2NOR	2-input nor gate
2OR	2-input or gate
2TRIBUS	2-input tristate bus
2WIRED	2-input wired and/or
2XNOR	2-input exclusive nor gate
2XOR	2-input exclusive or gate
3AND	3-input and gate
3NAND	3-input nand gate
3NOR	3-input nor gate
3OR	3-input or gate
3TRIBUS	3-input tristate bus
3WIRED	3-input wired and/or
4AND	4-input and gate
4NAND	4-input nand gate
4NOR	4-input nor gate
4OR	4-input or gate
4TRIBUS	4-input tristate bus
4WIRED	4-input wired and/or
5AND	5-input and gate
5NAND	5-input nand gate
5NOR	5-input nor gate
5OR	5-input or gate
5TRIBUS	5-input tristate bus
5WIRED	5-input wired and/or
6AND	6-input and gate
6NAND	6-input nand gate
6NOR	6-input nor gate
6OR	6-input or gate
6TRIBUS	6-input tristate bus
6WIRED	6-input wired and/or
7AND	7-input and gate
7NAND	7-input nand gate
7NOR	7-input nor gate

7OR	7-input or gate
7TRIBUS	7-input tristate bus
7WIRED	7-input wired and/or
8AND	8-input and gate
8NAND	8-input nand gate
8NOR	8-input nor gate
8OR	8-input or gate
8TRIBUS	8-input tristate bus
8WIRED	8-input wired and/or
8x8 PLA	8-in 8-out programmed logic array
BIGATE	bi-directional transfer gate (positive logic)
CTR	counter
DFF	D flip-flop
INV	inverter
JKF	JK flip-flop (active high set and reset)
JKFAC	JK flip-flop (edge sensing)
JKFDC	JK flip-flop
NBIGATE	bi-directional transfer gate (negative logic)
OSH	one-shot
SRF	set reset flip-flop
SRN	serial shift register (negative logic)
SSR	serial shift register (positive logic)
TRI	tristate gate (positive logic)
TRIN	tristate gate (negative logic)

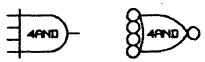
2AND (SIZE)



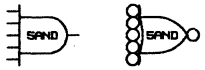
3AND (SIZE)



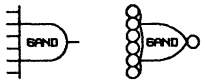
4AND (SIZE) (OFF GRID)



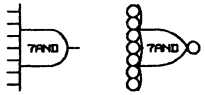
5AND (SIZE)



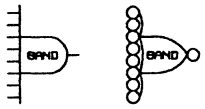
6AND (SIZE) (OFF GRID)



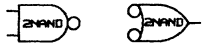
7AND (SIZE)



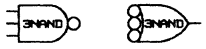
8AND (SIZE) (OFF GRID)



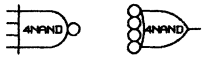
2NAND (SIZE)



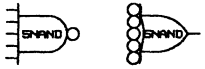
3NAND (SIZE)



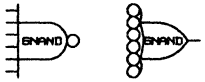
4NAND (SIZE) (OFF GRID)



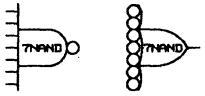
5NAND (SIZE)



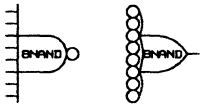
6NAND (SIZE) (OFF GRID)



7NAND (SIZE)

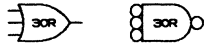


8NAND (SIZE) (OFF GRID)

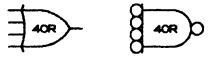


2OR (SIZE)

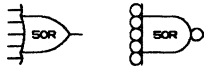
3OR (SIZE)



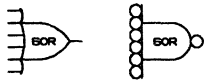
4OR (SIZE) (OFF GRID)



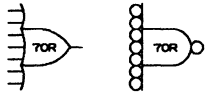
5OR (SIZE)



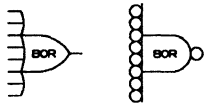
6OR (SIZE) (OFF GRID)



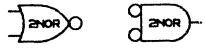
7OR (SIZE)



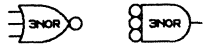
8OR (SIZE) (OFF GRID)



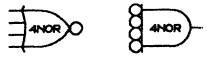
2NOR (SIZE)



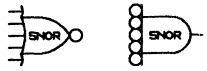
3NOR (SIZE)



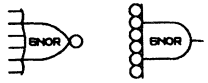
4NOR (SIZE) (OFF GRID)



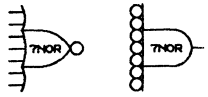
5NOR (SIZE)



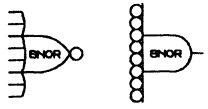
6NOR (SIZE) (OFF GRID)



7NOR (SIZE)



8NOR (SIZE) (OFF GRID)



2XOR (SIZE)

2XNOR (SIZE)

TRI (SIZE)

TRIN (SIZE)

SSR (SIZE)

SRN (SIZE)

INV (SIZE)

OSH (SIZE)

DEFINE

X_FIRST=0

X_STEP=SIZE

DRAWING

TITLE=EXAMPLE OF EACH LOGCAP PRIMITIVE
ABBREV=EXLOG

LAST_MODIFIED=Tue Oct 18 22:38:46 1983

NOTE: All primitives are shown with version 1 on the left

NOTE: (SIZE) means that the primitive can be given a SIZE property

NOTE: (OFF GRID) means that the output pin is off grid (0.1) The output pin should be wired with the blue cursor button (which snaps to the pin). The wire can then be placed down with the white cursor button.

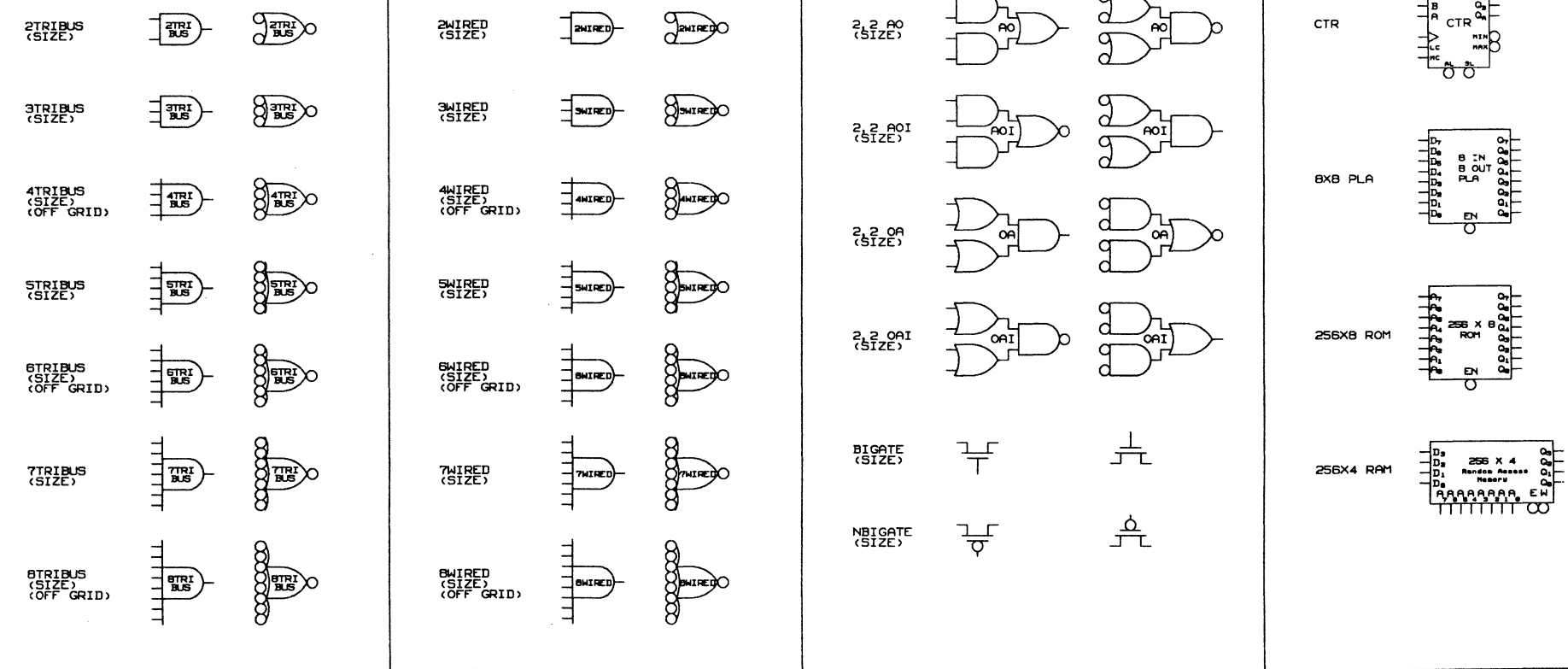
VALID LOGIC SYSTEMS INCORPORATED

TITLE: EXAMPLE OF EACH LOGCAP PRIMITIVE

DATE: 5-May-83

ENGINEER: PL

PAGE: 1 of 2



DRAWING

TITLE=EXAMPLE OF EACH LOGCAP PRIMITIVE
ABBREV=EXLOG

LAST_MODIFIED=Wed Oct 19 14:18:13 1983

NOTE: All primitives are shown with version 1 on the left

NOTE: (SIZE) means that the primitive can be given a SIZE property

NOTE: (OFF GRID) means that the output pin is off grid (0.1) The output pin should be wired with the blue cursor button (which snaps to the pin). The wire can then be placed down with the white cursor button.

VALID LOGIC SYSTEMS INCORPORATED

TITLE:	DATE:
EXAMPLE OF EACH LOGCAP PRIMITIVE	5-May-83
ENGINEER: PL	PAGE: 2 of 2

HCA6348 Library

Release 4.2 is the first release of the Motorola HCA6348 CMOS Gate Array Library.

The HCA6348 Library requires approximately 1.6 MBy of disk storage on the S-32. It contains bodies, physical, timing, and simulation models for the following 92 components:

B01D	3-state output buf - short ckt input w/pull-down
B01N	3-state output buf - short ckt input
B01U	3-state output buf - short ckt input w/pull-up
B02D	3-state output buf - ttl input w/pull-down (non-inv)
B02N	3-state output buf - ttl input (non-inv)
B02U	3-state output buf - ttl input w/pull-up (non-inv)
B03D	3-state output buf - cmos input w/pull-down (inv)
B03N	3-state output buf - cmos input (inv)
B03U	3-state output buf - cmos input w/pull-up (inv)
B04D	3-state output buf - cmos input w/pull-down (non-inv)
B04N	3-state output buf - cmos input (non-inv)
B04U	3-state output buf - cmos input w/pull-up (non-inv)
C001	triple 2-input nand
C002	dual 3-input nand
C003	dual 2-input nand/and
C004	triple 2-input nor
C005	dual 3-input nor
C006	dual 2-input nor/or
C007	triple inverting buffer
C008	quad inverter
C009	dual 3-state inverter
C010	3-state non-inverting buffer
C012	nand latch and 2-input nand
C013	nor latch and 2-input nor
C017	triple 4-input nand
C019	triple 3-input nand/and
C020	triple 4-input nor
C022	triple 3-input nor/or
C025	Schmitt trigger
C026	D latch w/reset(L) & enable(L)
C027	triple nand latch
C028	4-to-1 multiplexer w/3-state enable(L)
C029	4-to-1 data multiplexer
C030	4-bit parity checker
C031	triple nor latch
C032	full adder
C033	1-to-4 decoder w/outputs(L) & 2 inverters
C034	parallel load D flip-flop w/reset(H)
C035	multiplexed D flip-flop w/reset(L)
C036	toggle enable flip-flop w/reset(L)

```
C037      JK flip-flop w/reset & set
C038      1-bit presettable up/down counter w/set
C039      2-bit serial in/serial parallel out shift
          register w/reset
C040      1-bit ALU - 7 functions
C041      2-bit magnitude comparator
C042      2-bit serial/parallel shift register
C053      2-input xor buffer
C054      2-input 2-wide or-and/invert
C055      2-input 2-wide and-or/invert
C056      2-to-1 multiplexer buffer
C057      5-input nand/and
C058      5-input nor/or
C059      buffered D flip-flop
C060      D flip-flop w/reset(L) & set(L)
HCA 2 TRIBUS    2-input tri-state bus
HCA 3 TRIBUS    3-input tri-state bus
HCA 4 TRIBUS    4-input tri-state bus
HCA 5 TRIBUS    5-input tri-state bus
HCA 6 TRIBUS    6-input tri-state bus
HCA 7 TRIBUS    7-input tri-state bus
HCA 8 TRIBUS    8-input tri-state bus
HCA 2 WIRED     2-input wired-or
HCA 3 WIRED     3-input wired-or
HCA 4 WIRED     4-input wired-or
HCA 5 WIRED     5-input wired-or
HCA 6 WIRED     6-input wired-or
HCA 7 WIRED     7-input wired-or
HCA 8 WIRED     8-input wired-or
I01D      ttl input buf w/pull-down (non-inv)
I01N      ttl input buf (non-inv)
I01U      ttl input buf w/pull-up (non-inv)
I02D      cmos input buf w/pull-down (inv)
I02N      cmos input buf (inv)
I02U      cmos input buf w/pull-up (inv)
I03D      cmos input buf w/pull-down (non-inv)
I03N      cmos input buf (non-inv)
I03U      cmos input buf w/pull-up (non-inv)
I04D      short ckt input buf w/pull-down
I04N      short ckt input buf
I04U      short ckt input buf w/pull-up
I05D      Schmitt trigger input buf w/pull-down (non-inv)
I05N      Schmitt trigger input buf (non-inv)
I05U      Schmitt trigger input buf w/pull-up (non-inv)
I07D      clock buffer input w/pull-down (inv)
I07N      clock buffer input (inv)
I07U      clock buffer input w/pull-up (inv)
I08D      clock buffer input w/pull-down (non-inv)
I08N      clock buffer input (non-inv)
I08U      clock buffer input w/pull-up (non-inv)
I09N      oscillator buffer input
Y01N      output only buffer (non-inv)
```

Y03N short ckt output buffer

APPLICATION NOTE

This section describes some important usage notes.

1. Special Bodies

Several special models have been created to generate the proper Motorola (LOGCAP) netlist. The HCA WIRED bodies must be used to represent wire-tie connections and the HCA TRIBUS bodies must be used to represent tri-state bus structure. Please refer to the Motorola CMOS Macrocell Array CAD Manual for details.

2. Packaging

All drawings must be packaged prior to executing the LOGCAP interface program (glogcap). GLOGCAP is a physical interface program and requires the Packager to package the pertinent cells together.

C001 (SIZE)

C002 (SIZE)

C003 (SIZE)

C004 (SIZE)

C005 (SIZE)

C006 (SIZE)

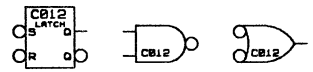
C007 (SIZE)

C008 (SIZE)

C009 (SIZE)

C010 (SIZE)

C012 (SIZE)



C013 (SIZE)

C017 (SIZE)

C019 (SIZE)

C020 (SIZE)

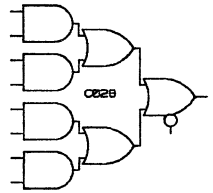
C022 (SIZE)

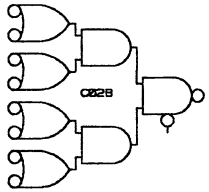
C025 (SIZE)

C026 (SIZE)

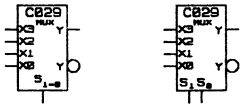
C027 (SIZE)

C028 (SIZE)





C029 (SIZE)



DEFINE

X_FIRST=0

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH HCA6348 PART
ABBREV=EXHCA6348

ENGINEER:
JIMMY

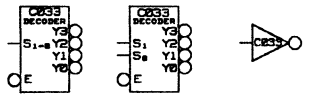
DATE:
Tue Dec 20 08:48:39 1983

C030

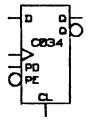
C031 (SIZE)

C032 (SIZE)

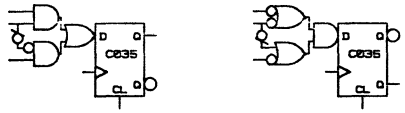
C033 (SIZE)



C034 (SIZE)

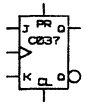


C035 (SIZE)

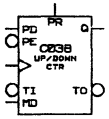


C036 (SIZE)

C037 (SIZE)

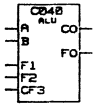


C038



C039

C040



C041

C042

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH HCA6348 PART
ABBREV=EXHCA6349

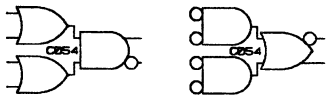
ENGINEER:
JIMMY

DATE:
Tue Dec 27 10:53:31 1983

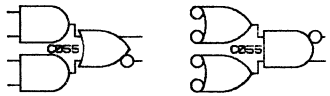
PAGE:
2 OF 6

C053
(SIZE)

C054
(SIZE)



C055
(SIZE)



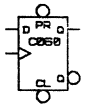
C056
(SIZE)

C057
(SIZE)

C058
(SIZE)

C059
(SIZE)

C050
(SIZE)



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

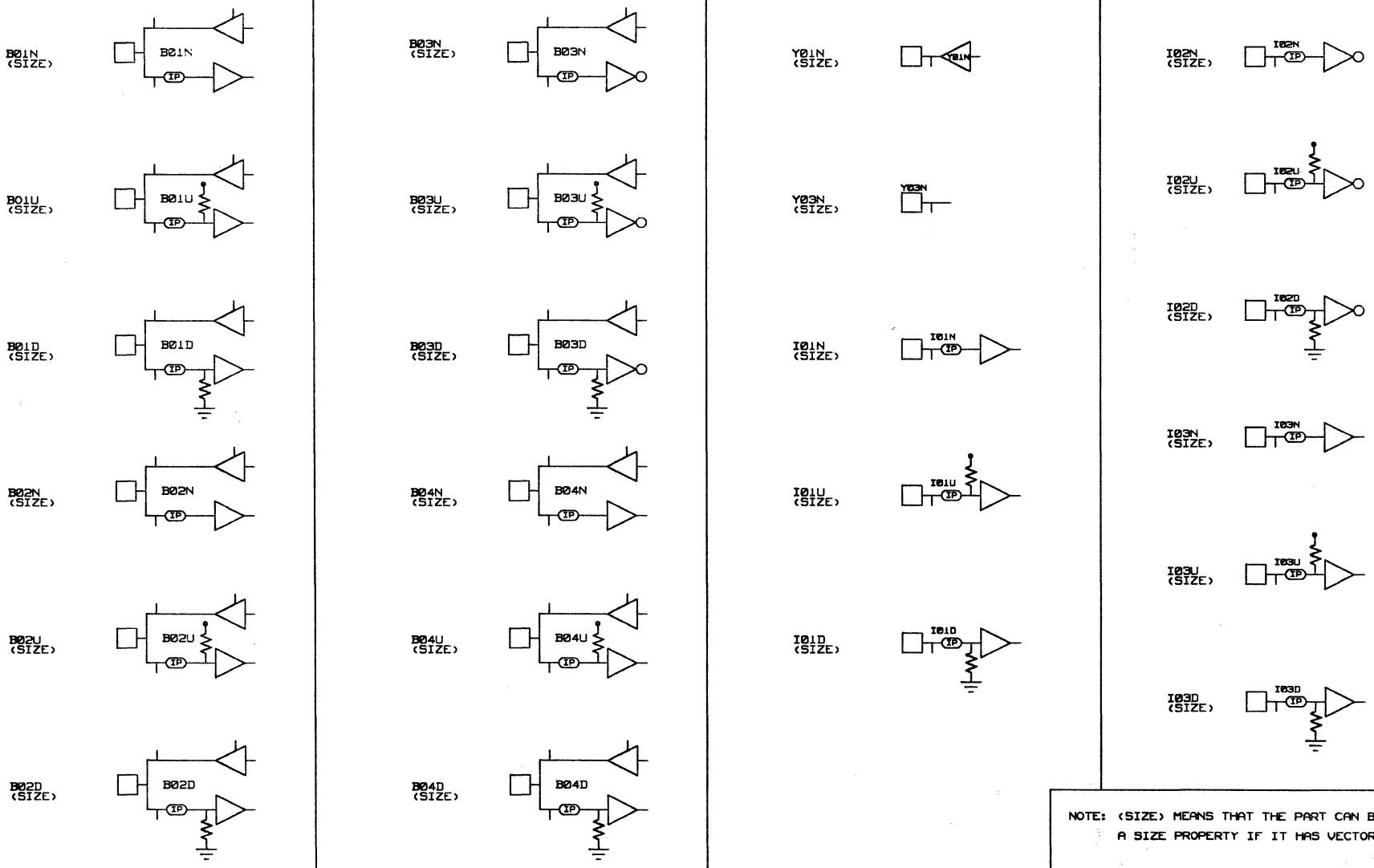
VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH HCA6348 PART
ABBREV=EXHCA6348

ENGINEER:
JIMMY

DATE:
Fri Dec 16 16:38:26 1983

PAGE:
3 OF 6



NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

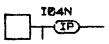
DRAWING:
TITLE=EXAMPLE OF EACH HCA6348 PART
ABBREV=EXHCA6348

ENGINEER:
JIMMY

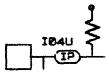
DATE:
Fri Dec 16 16:38:43 1983

PAGE:
4 OF 6

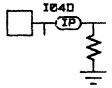
I04N (SIZE)



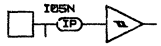
I04U (SIZE)



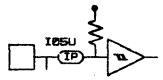
I04D (SIZE)



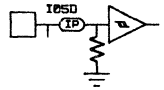
I05N (SIZE)



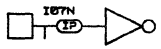
I05U (SIZE)



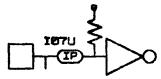
I05D (SIZE)



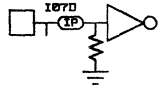
I07N (SIZE)



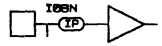
I07U (SIZE)



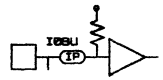
I07D (SIZE)



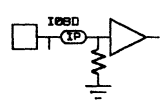
I08N (SIZE)



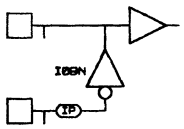
I08U (SIZE)



I08D (SIZE)



I09N (SIZE)



NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH HCA6348 PART
ABBREV=EXHCA6348

ENGINEER:
JIMMY

DATE:
Fri Dec 16 16:38:57 1983

PAGE:
5 OF 6

HCA_2 WIRED (SIZE)

HCA_3 WIRED (SIZE)

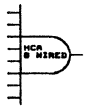
HCA_4 WIRED (SIZE)

HCA_5 WIRED (SIZE)

HCA_6 WIRED (SIZE)

HCA_7 WIRED (SIZE)

HCA_8 WIRED (SIZE)



HCA_2 TRIBUS (SIZE)

HCA_3 TRIBUS (SIZE)

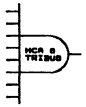
HCA_4 TRIBUS (SIZE)

HCA_5 TRIBUS (SIZE)

HCA_6 TRIBUS (SIZE)

HCA_7 TRIBUS (SIZE)

HCA_8 TRIBUS (SIZE)



NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

DRAWING: TITLE=EXAMPLE OF EACH HCA6348 PART ABBREV=EXHCA6348	ENGINEER: JIMMY
DATE: Fri Dec 16 16:39:08 1983	PAGE: 6 OF 6

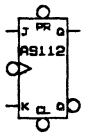
ASTTL Library

There have some changes in the ASTTL Library since Release 4.4.

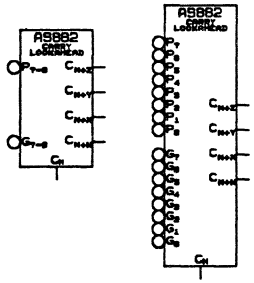
The ASTTL Library requires approximately 0.1 MBy (234 Blocks) of disk storage on the S-32. It contains bodies, physical, timing, and simulation models for the following 4 components:

74AS112	dual JK negative-edge-triggered flip-flop with clear and preset
74AS240	octal buffer and line driver with 3-state output
74AS374	octal D-type edge-triggered flip-flop
74AS882	32-bit look-ahead carry generator

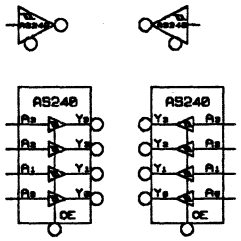
AS112
(SIZE)



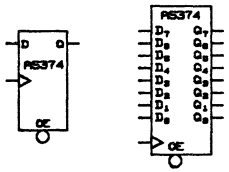
AS882



AS240
(SIZE)



AS374
(SIZE)



DEFINE
X_FIRST=0
X_STEP=SIZE

VALID
LOGIC SYSTEMS INCORPORATED

DRAWING:
EXAMPLE OF EACH ASTTL PART
EXAS

ENGINEER:
WANG

DATE:
SAT MAR 3 14:08:28 1984

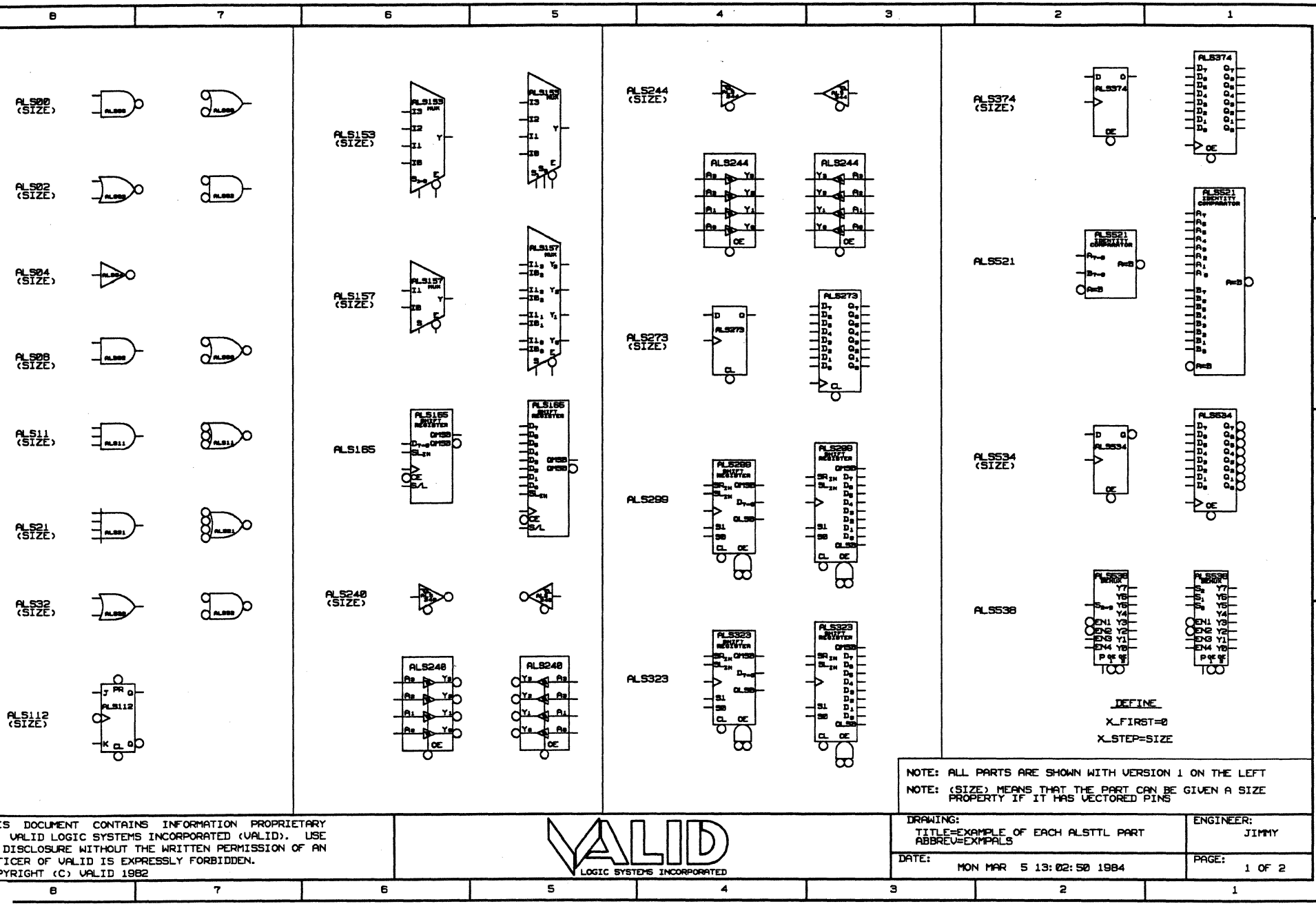
PAGE:
1 OF 1

ALSTTL Library

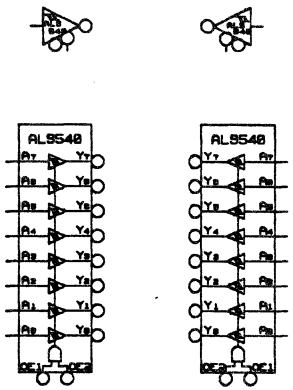
There have been some changes in the ALSTTL Library for Release 5.1.

The ALSTTL Library requires approximately 0.8 MBy (1522 Blocks) of disk storage on the S-32. It contains bodies, physical, timing, and simulation models for the following 31 components:

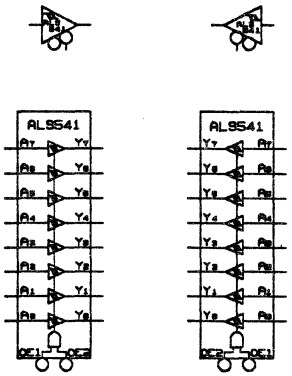
74ALS00	quad 2-input nand
74ALS02	quad 2-input nor
74ALS04	hex inverter
74ALS08	quad 2-input and
74ALS11	triple 3-input and
74ALS21	dual 4-input and
74ALS32	quad 2-input or
74ALS112	dual JK negative-edge-triggered flip-flop with clear and preset
74ALS153	dual-1-of 4 data selector/multiplexer
74ALS157	quad-1-of 2 data selector/multiplexer
74ALS165	parallel-load 8-bit shift register
74ALS240	octal buffer and line driver with 3-state output
74ALS244	octal buffer and line driver with 3-state output
74ALS273	octal D-type flip-flop with clear
74ALS299	8-bit universal shift/storage register with 3-state output
74ALS323	8-bit universal shift/storage register with 3-state output
74ALS374	octal D-type edge-triggered flip-flop
74ALS521	8-bit identity comparator
74ALS534	octal D-type edge-triggered flip-flop with 3-state output
74ALS538	3-line to 8-line decoder/demultiplexer with 3-state output
74ALS540	octal buffer and line driver with 3-state output
74ALS541	octal buffer and line driver with 3-state output
74ALS563	octal D-type transparent latch with 3-state output
74ALS564	octal D-type edge-triggered flip-flop with 3-state output
74ALS569	synchronous 4-bit up/down binary counter with 3-state output
74ALS573	octal D-type transparent latch with 3-state output
74ALS574	octal D-type edge-triggered flip-flop with 3-state output
74ALS576	octal D-type edge-triggered flip-flop with 3-state output
74ALS580	octal D-type transparent latch with 3-state output
74ALS804	hex 2-input nand driver
74ALS874	dual 4-bit D-type edge-triggered flip-flop


8
7
6
5
4
3
2
1
D
C
B
A
ALS00 (SIZE)
ALS02 (SIZE)
ALS04 (SIZE)
ALS08 (SIZE)
ALS11 (SIZE)
ALS21 (SIZE)
ALS32 (SIZE)
ALS112 (SIZE)
ALS153 (SIZE)
ALS157 (SIZE)
ALS165
ALS240 (SIZE)
ALS244 (SIZE)
ALS273 (SIZE)
ALS299
ALS323
ALS374 (SIZE)
ALS521
ALS534 (SIZE)
ALS538
DEFINE
X_FIRST=0
X_STEP=SIZE
NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT
NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS
THIS DOCUMENT CONTAINS INFORMATION PROPRIETARY TO VALID LOGIC SYSTEMS INCORPORATED (VALID). USE OR DISCLOSURE WITHOUT THE WRITTEN PERMISSION OF AN OFFICER OF VALID IS EXPRESSLY FORBIDDEN.
COPYRIGHT (C) VALID 1982
VALID
LOGIC SYSTEMS INCORPORATED
DRAWING:
TITLE=EXAMPLE OF EACH ALSTTL PART
ABBREV=EXMPALS
ENGINEER:
JIMMY
DATE:
MON MAR 5 13:02:50 1984
PAGE:
1 OF 2

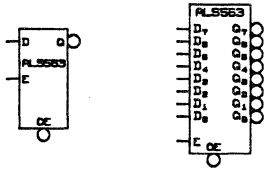
AL5540
(SIZE)



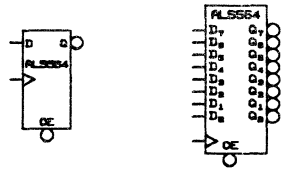
AL5541
(SIZE)



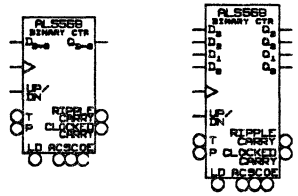
AL5563
(SIZE)



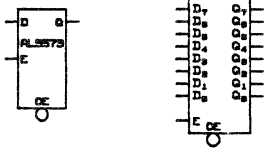
AL5564
(SIZE)



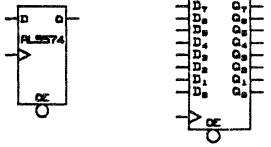
AL5569



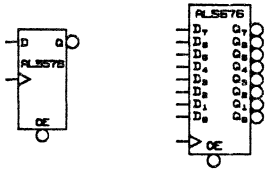
AL5573
(SIZE)



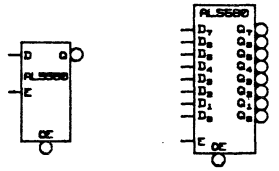
AL5574
(SIZE)



AL5576
(SIZE)

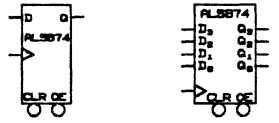


AL5580
(SIZE)



AL5804
(SIZE)

AL5874
(SIZE)



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID
LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH ALSTTL PART
ABBREV=EXMPALSTTL

ENGINEER:
JIMMY

DATE:
MON MAR 5 14:39:36 1984

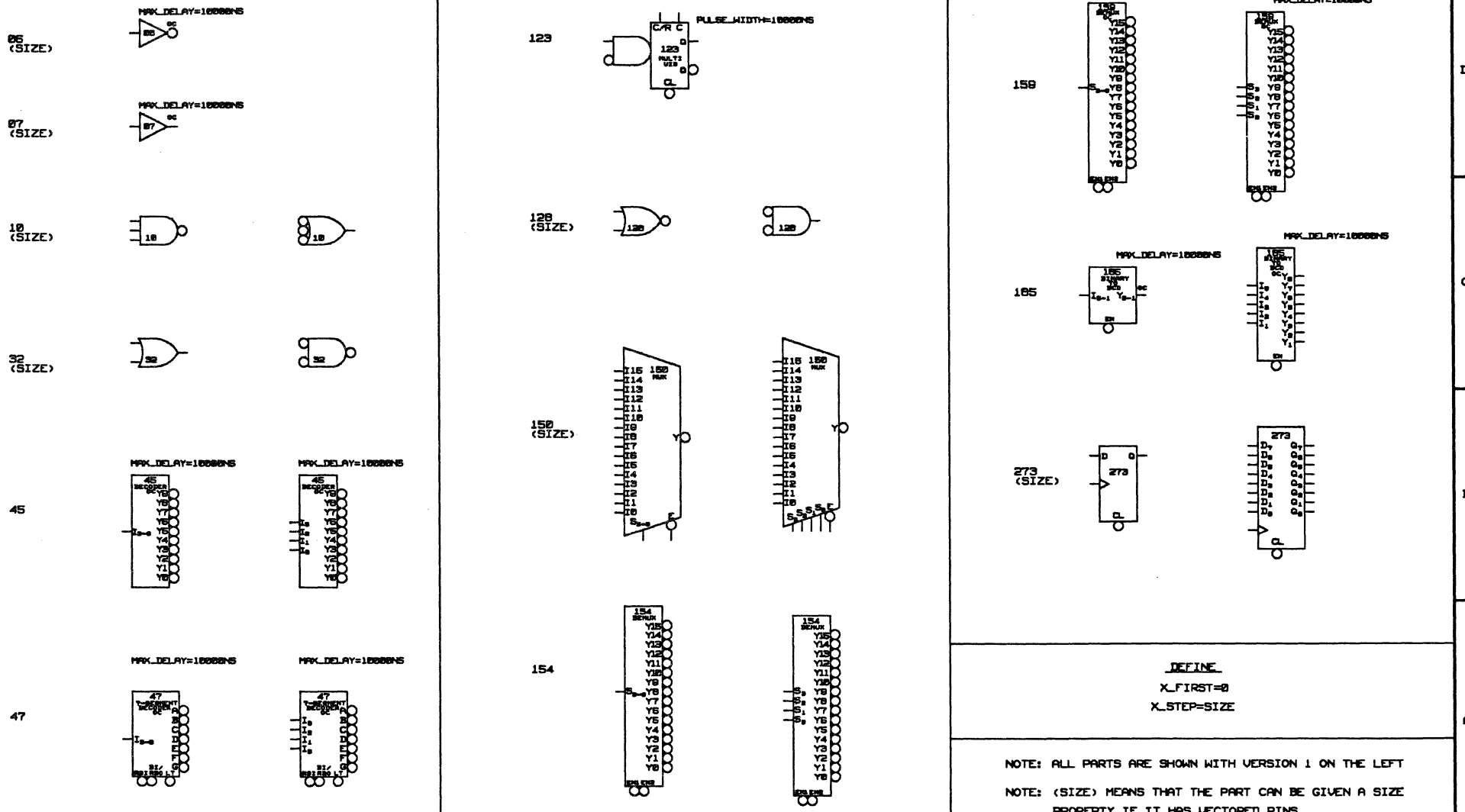
PAGE:
2 OF 2

TTL Library

There have been a few changes in the TTL Library for Release 5.1.

The TTL Library requires approximately 0.2 MBy (424 Blocks) of disk storage on the S-32. It contains bodies, physical, timing, and simulation models for the following 13 components:

7406	hex inverter buffer/driver w/open collector output
7407	hex buffer/driver w/open collector output
7410	triple 3-input positive nand gate
7432	quad 2-input positive or gate
7445	bcd to decimal decoder
7447	bcd to 7-segment decoder
74123	dual retriggerable monostable multivibrators w/clear
74128	50-ohm line driver
74150	1-of-16 data selector/multiplexer
74154	4-to-16 line decoder/demultiplexer
74159	4-to-16 line decoder/demultiplexer
74185	binary-to-bcd converter
74273	octal D-type flip-flop



DEFINE

X_FIRST=0

X_STEP=SIZE

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH TTL PART
ABBREV=EXTTTL

ENGINEER: RRJ

DATE: Tue Apr 10 13:01:43 1984

PAGE: 1 OF 1

HCMOS Library

There have been a few changes in the High-Speed CMOS Library for Release 5.1.

The HCMOS Library requires approximately 1.7 MBy (3257 Blocks) of disk storage on the S-32. It contains bodies, physical, timing and simulation models for the following 68 components:

74HC00	quad 2-input nand
74HC02	quad 2-input nor
74HC03	quad 2-input open-collector nand
74HC04	hex inverter
74HC08	quad 2-input and
74HC10	triple 3-input nand
74HC11	triple 3-input and
74HC14	hex schmitt-trigger inverter
74HC20	dual 4-input nand
74HC27	triple 3-input nor
74HC30	8-input nand
74HC32	quad 2-input or
74HC42	4-to-10-line decoder
74HC51	2-wide 3-input, 2-wide 2-input and-or-invert
74HC74	dual positive-edge-triggered D flip-flop
74HC75	4-bit bistable latch
74HC76	dual JK flip-flop w/ preset & clear
74HC86	quad 2-input exclusive-or
74HC107	dual JK negative-edge-triggered flip-flop
74HC109	dual JKbar positive-edge-triggered flip-flop
74HC112	dual JK negative-edge-triggered flip-flop
74HC113	dual JK negative-edge-triggered flip-flop
74HC125	quad bus buffer with three-state output
74HC126	quad bus buffer with three-state output
74HC133	13-input nand
74HC137	3-to-8 line decoder/demultiplexer w/address latch
74HC138	3-to-8 line decoder/demultiplexer
74HC139	dual 2-to-4 line decoder/multiplexer
74HC151	1-of-8 data selector/multiplexer
74HC153	dual 4-line to 1-line data multiplexer
74HC154	4-to-16 line decoder/demultiplexer
74HC157	quad 2-to-1-line non-inverting multiplexer
74HC158	quad 2-to-1-line inverting data multiplexer
74HC160	4-bit synchronous decade counters with direct clear
74HC161	4-bit synchronous binary counters with direct clear
74HC163	4-bit synchronous binary counters with synch clear
74HC164	8-bit parallel output serial shift register
74HC165	8-bit serial output shift register
74HC166	8-bit shift register
74HC173	quad D-type flip-flop

74HC174	hex D-type flip-flop
74HC175	quad D-type flip-flop
74HC190	synchronous BCD up/down counter
74HC191	synchronous binary up/down counter
74HC192	synchronous BCD up/down counter
74HC193	synchronous binary up/down dual clock counters
74HC194	4-bit bidirectional shift register
74HC195	4-bit universal shift register
74HC240	octal inverting 3-state bus transceiver
74HC241	octal non-inverting 3-state bus transceiver
74HC244	octal non-inverting 3-state bus transceiver
74HC245	octal non-inverting 3-state bus transceiver
74HC251	3-state data multiplexer
74HC253	dual data selector/multiplexer
74HC257	quad 3-state non-inverting data multiplexer
74HC273	octal D-type flip-flop
74HC280	9-bit odd/even parity generators/checker
74HC367	hex bus driver
74HC368	hex bus driver
74HC373	octal 3-state D-latch w/ common enable
74HC374	octal 3-state positive-edge-triggered D register
74HC390	dual 4-stage binary ripple counter
74HC393	dual 4-stage binary ripple counter
74HC573	octal latch with three-state output
74HC574	octal D-type flip-flop wih three-state output
74HC640	octal 3-state inverting bus transceiver
74HCT00	quad 2-input nand
74HCT04	hex inverter

HCT00 (SIZE)

HCT04 (SIZE)

HC00 (SIZE)

HC02 (SIZE)

HC03 (SIZE)

MAX_DELAY=10000NS

MAX_DELAY=10000NS

HC04 (SIZE)

HC08 (SIZE)

HC10 (SIZE)

HC11 (SIZE)

HC14 (SIZE)

HC20 (SIZE)

HC27 (SIZE)

HC30 (SIZE)

HC32 (SIZE)

HC42

HC51 (SIZE)

HC74 (SIZE)

HC75 (SIZE)

HC76 (SIZE)

HC86 (SIZE)

HC107 (SIZE)

HC109 (SIZE)

HC112 (SIZE)

HC113 (SIZE)

HC125 (SIZE)

HC126 (SIZE)

HC133 (SIZE)

DEFINE

X_FIRST=0

X_STEP=SIZE

NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

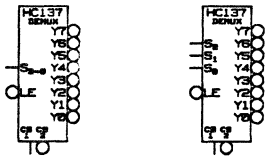
DRAWING:
TITLE=EXAMPLE OF EACH HCMOS PART
ABBREV=EXHC

ENGINEER: RKM

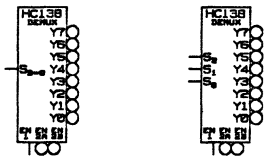
DATE: Wed Feb 1 13:20:21 1984

PAGE: 1 OF 5

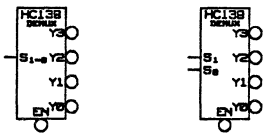
HC137



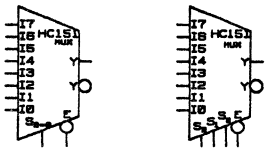
HC138



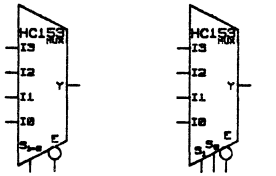
HC139



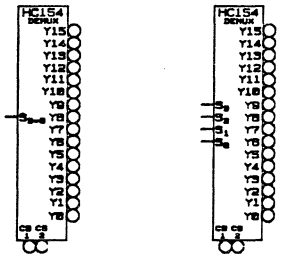
HC151
(SIZE)



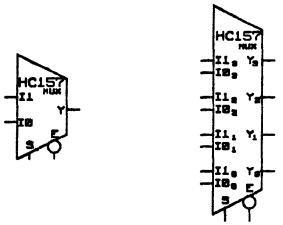
HC153
(SIZE)



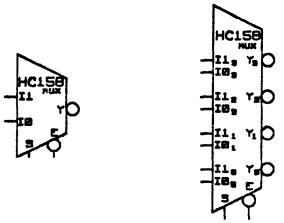
HC154
(SIZE)



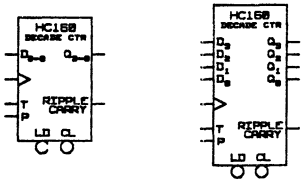
HC157
(SIZE)



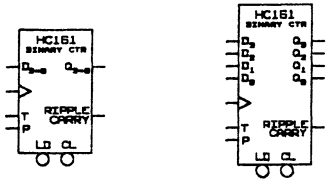
HC158
(SIZE)



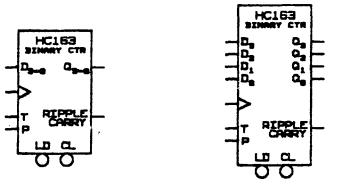
HC160



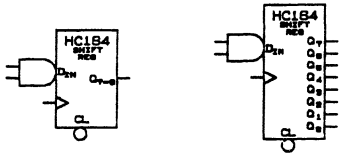
HC161



HC163



HC164



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

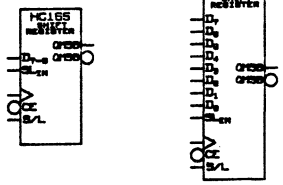
VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH HCMOS PART
ABBREV=EXHC

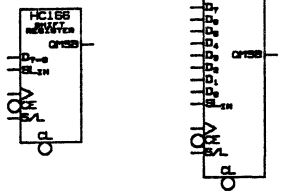
ENGINEER: RKM

DATE: Wed Feb 15 19:56:14 1984

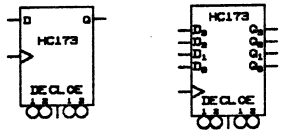
HC165



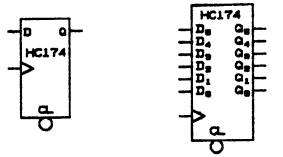
HC166



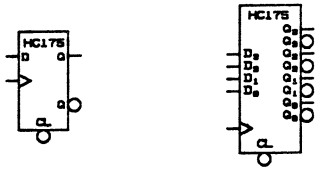
HC173



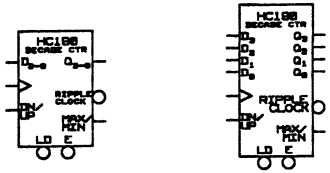
HC174
(SIZE)



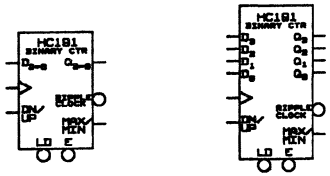
HC175
(SIZE)



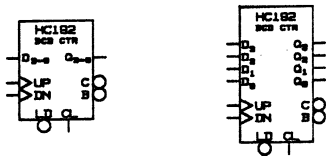
HC190



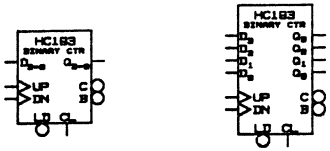
HC191



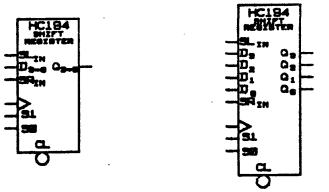
HC192



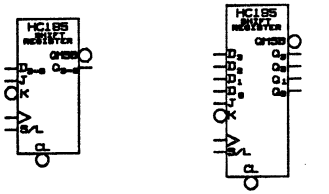
HC193



HC194



HC195



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

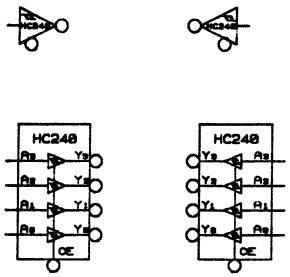
DRAWING:
TITLE=EXAMPLE OF EACH HCMOS PART
ABBREV=EXRC

ENGINEER:
RKM

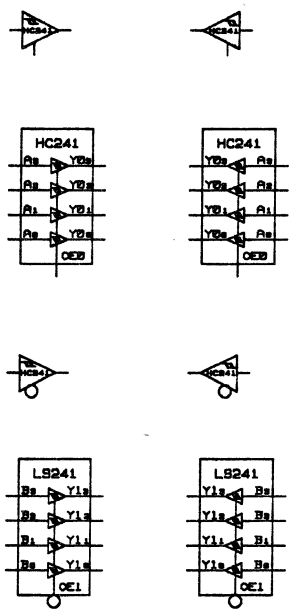
DATE:
Tue Jan 31 17:45:46 1984

PAGE:
3 OF 5

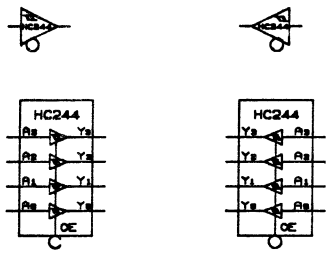
HC240 (SIZE)



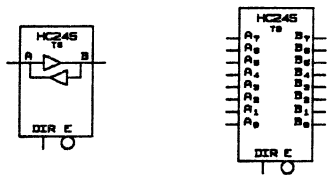
HC241 (SIZE)



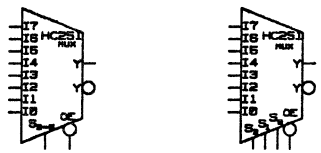
HC244 (SIZE)



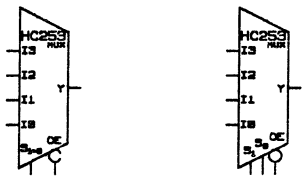
HC245 (SIZE)



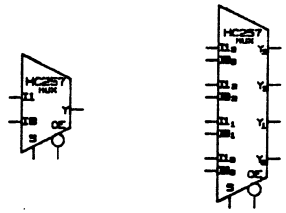
HC251 (SIZE)



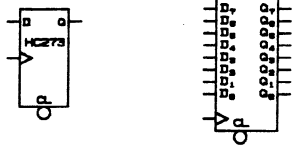
HC253 (SIZE)



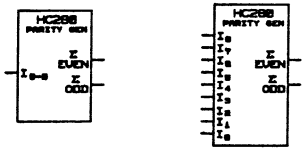
HC257 (SIZE)



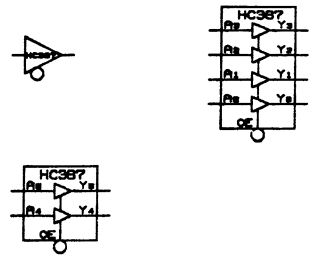
HC273 (SIZE)



HC280



HC367 (SIZE)



NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

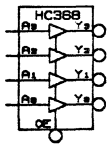
VALID LOGIC SYSTEMS INCORPORATED

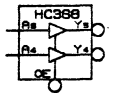
DRAWING: TITLE=EXAMPLE OF EACH HCMOS PART ABBREV=EXHC

ENGINEER: RKM

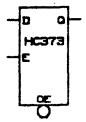
DATE: Tue Jan 31 17:46:57 1984

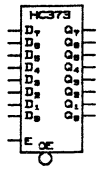
HC368 (SIZE)



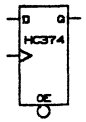


HC373 (SIZE)

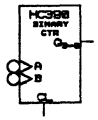


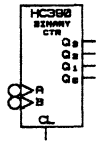


HC374 (SIZE)

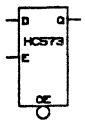


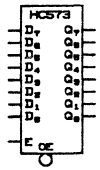
HC390



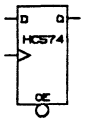


HC573 (SIZE)

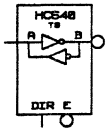


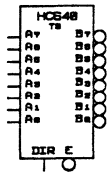


HC574 (SIZE)



HC640 (SIZE)





NOTE: ALL PARTS ARE SHOWN WITH VERSION 1 ON THE LEFT

NOTE: (SIZE) MEANS THAT THE PART CAN BE GIVEN A SIZE PROPERTY IF IT HAS VECTORED PINS

VALID LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH HCMOS PART
ABBREV=EXHC

ENGINEER:
RKM

DATE:
Wed Feb 15 19:14:30 1984

PAGE:
5 OF 5

CMOS Library

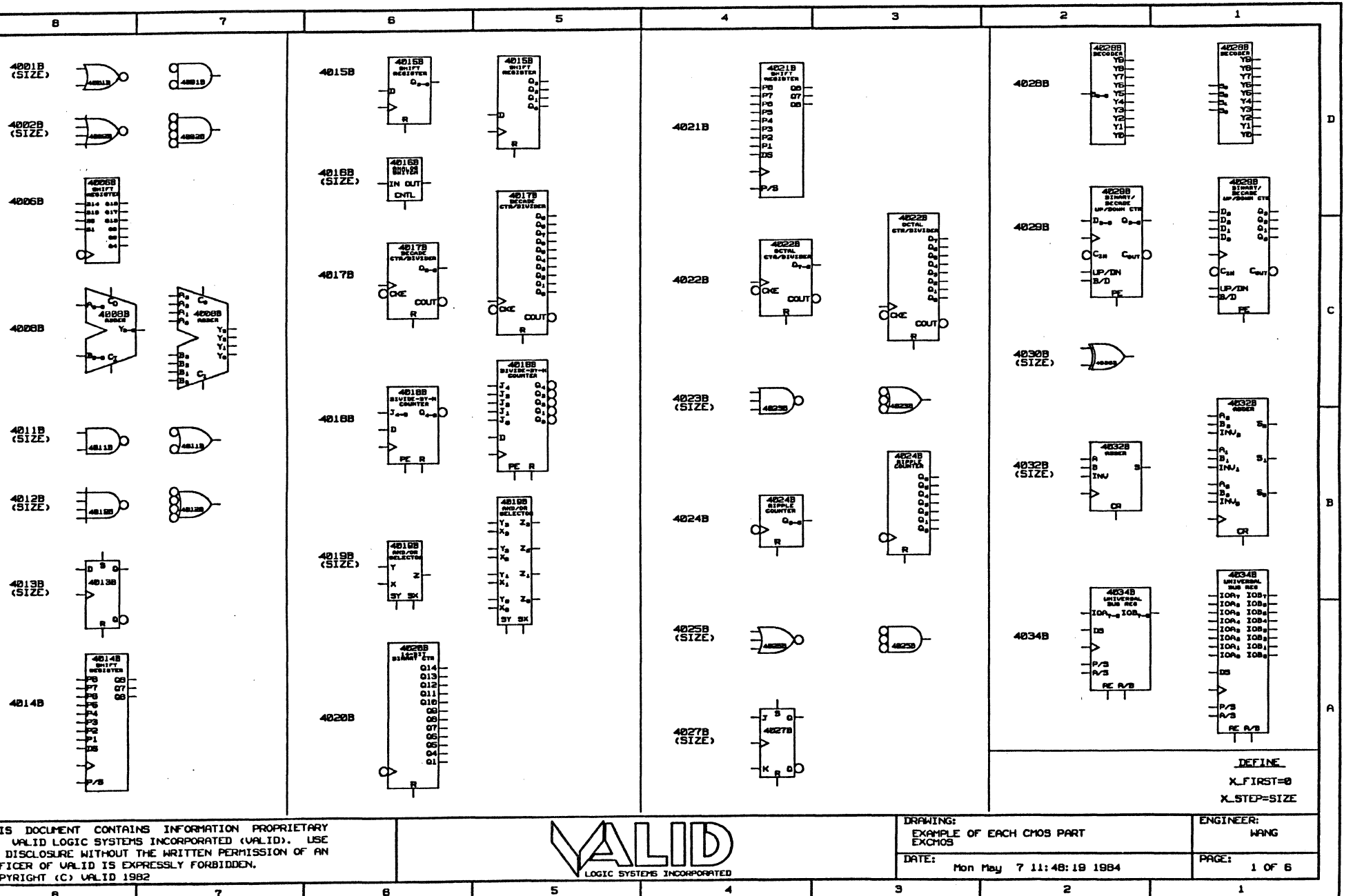
Release 4.5 is the first release of the CMOS Library.

The CMOS Library requires approximately 1.26MBy (2468 Blocks) of disk storage on the S-32. It contains bodies and physical for the following 99 components:

4001B	quad 2-input nor
4002B	dual 4-input nor
4006B	18-bit static shift register
4008B	4-bit full adder
4011B	quad 2-input nand
4012B	dual 4-input nand
4013B	dual D flip-flop
4014B	8-bit static shift register
4015B	dual 5-bit static shift register
4016B	quad analog switch/quad multiplexer
4017B	decade counter/divider
4018B	presettable divide-by-N counter
4019B	quad and/or select
4020B	14-bit binary counter
4021B	8-bit static shift register
4022B	octal counter/divider
4023B	triple 3-input nand
4024B	7-stage ripple counter
4025B	triple 3-input nor
4027B	dual JK flip-flop
4028B	BCD-to-decimal decoder
4029B	4-bit presettable up/down counter
4030B	quad exclusive-or
4032B	triple serial adder
4034B	8-bit universal bus register
4035B	4-bit shift register
4038B	triple serial adder
4040B	12-bit binary counter
4042B	quad latch
4043B	quad nor R-S latch
4049UB	hex inverter/buffer
4050B	hex buffer
4051B	8-channel analog multiplexer
4052B	dual 4-channel analog multiplexer
4053B	triple 2-channel analog multiplexer
4066B	quad analog switch
4067B	multiplexer/demultiplexer
4068B	8-input nand
4069UB	hex inverter
4070B	quad exclusive-or
4071B	quad 2-input or
4072B	dual 4-input or

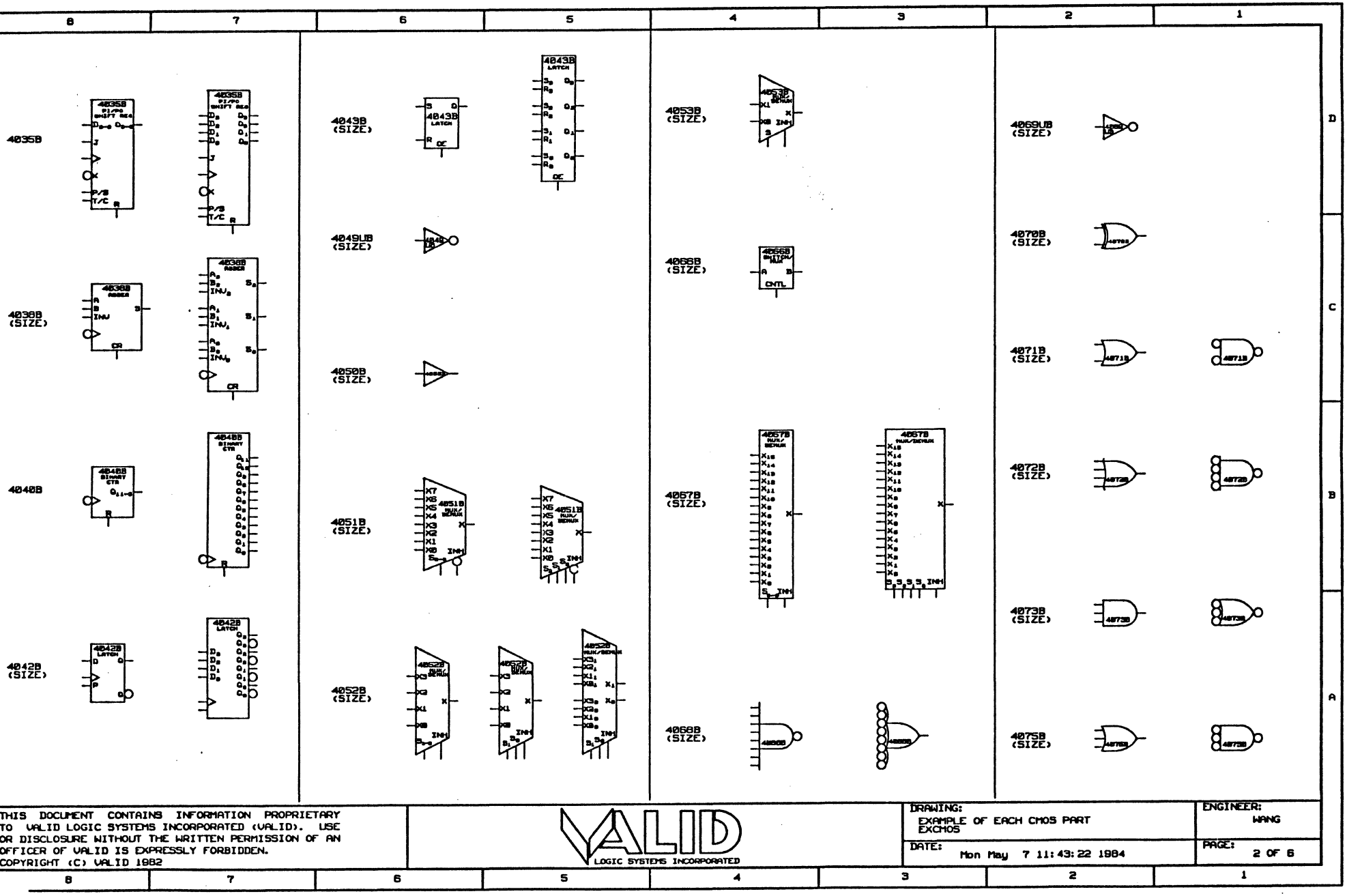
4073B	triple 3-input and
4075B	triple 3-input or
4076B	quad D-type register
4077B	quad exclusive-nor
4078B	8-input nor
4081B	quad 2-input and
4082B	dual 4-input and
4093B	quad 2-input nand schmitt trigger
4094B	8-bit bus compatible shift store latch
4098B	dual monostable multivibrator
4099B	8-bit addressable latch
4160B	decade counter w/ async clear
4161B	binary counter w/ async clear
4162B	decade counter w/ sync clear
4163B	binary counter w/ sync clear
4174B	hex D flip-flop
4175B	quad D flip-flop
4503B	hex 3-state buffer
4504B	hex TTL or CMOS to CMOS level shifter
4508B	dual 4-bit latch
4510B	BCD up/down counter
4511B	BCD to 7-segment latch/decoder/driver
4512B	8-channel data selector
4514B	4-bit latch/4-to-16 line decoder
4515B	4-bit latch/4-to-16 line decoder
4516B	binary up/down counter
4517B	dual 64-bit static shift register
4519B	4-bit and/or selector
4520B	dual binary up counter
4526B	programmable binary divide-by-N counter
4528B	dual monostable multivibrator
4529B	dual 4-channel analog data selector
4530B	dual 5-input majority logic gate
4532B	8-bit priority encoder
4538B	dual precision monostable multivibrator
4539B	dual 4-channel data selector/multiplexer
4551B	quad 2-input analog mux/demultiplexer
4552B	64 x 4 bit static RAM
4555B	dual binary to 1-of-4 decoder
4556B	dual binary to 1-of-4 decoder
4557B	1-to-64 bit variable length shift register
4562B	128-bit static shift register
4572UB	hex gate
4584B	hex schmitt trigger
4585B	4-bit magnitude comparator
4599B	8-bit addressable latch
4724B	8-bit addressable latch
40103B	8-stage presettable synchronous down counter
40105B	FIFO register
40109B	quad low-to-high voltage level shifter
40161B	binary counter w/ async clear
40162B	decade counter w/ sync clear

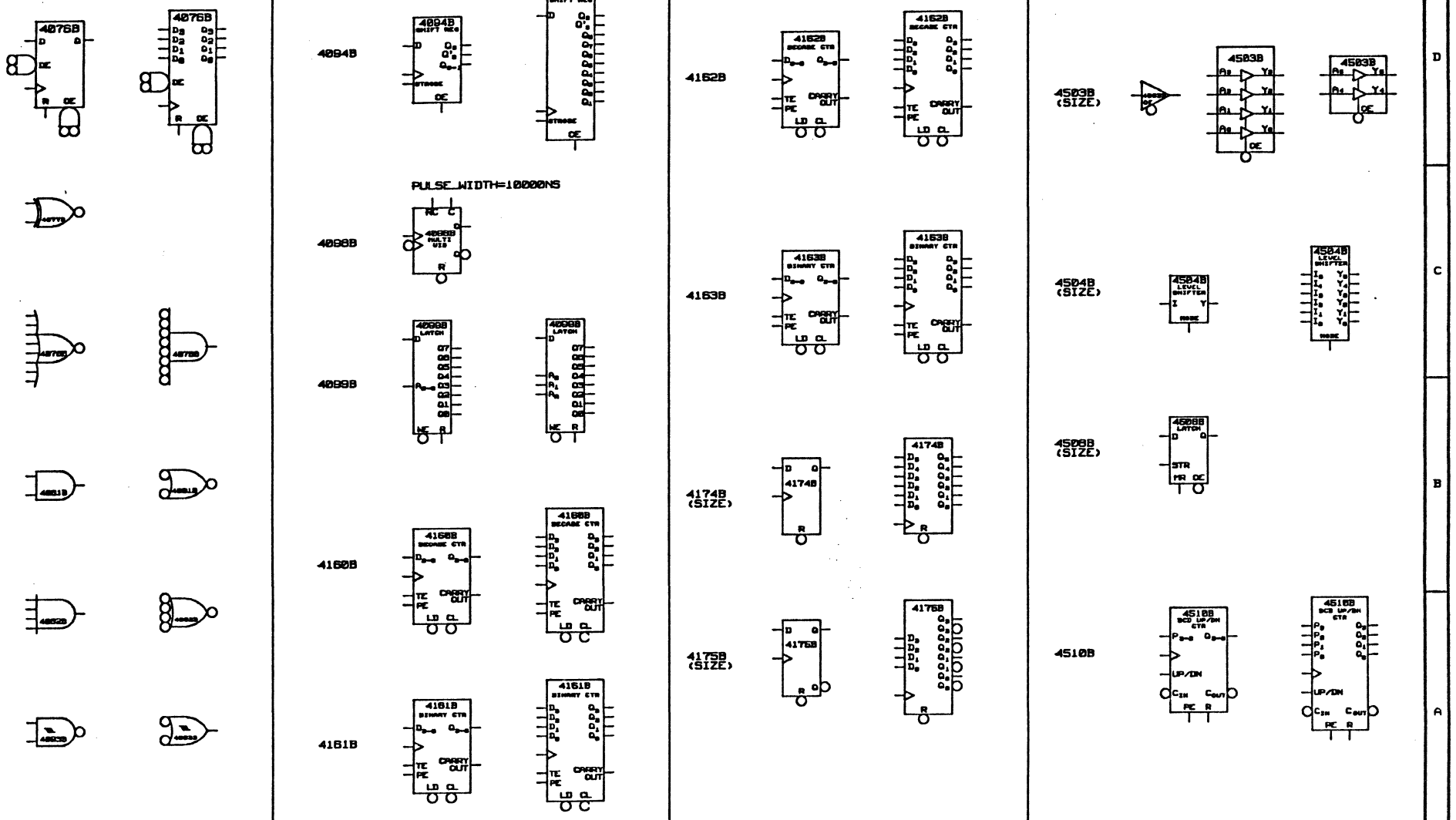
40163B	binary counter w/ sync clear
40174B	hex D flip-flop
40175B	quad D flip-flop
40192B	BCD up/down counter
40193B	binary up/down counter



VALID LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH CMOS PART EXCMOS	ENGINEER: WANG
DATE: Mon May 7 11:48:19 1984	PAGE: 1 OF 6


8
7
6
5
4
3
2
1
D
C
B
A
4035B
4038B
(SIZE)
4040B
4042B
(SIZE)
4043B
(SIZE)
4049UB
(SIZE)
4050B
(SIZE)
4051B
(SIZE)
4052B
(SIZE)
4053B
(SIZE)
4066B
(SIZE)
4067B
(SIZE)
4068B
(SIZE)
4069UB
(SIZE)
4070B
(SIZE)
4071B
(SIZE)
4072B
(SIZE)
4073B
(SIZE)
4075B
(SIZE)
THIS DOCUMENT CONTAINS INFORMATION PROPRIETARY TO VALID LOGIC SYSTEMS INCORPORATED (VALID). USE OR DISCLOSURE WITHOUT THE WRITTEN PERMISSION OF AN OFFICER OF VALID IS EXPRESSLY FORBIDDEN.
COPYRIGHT (C) VALID 1982
VALID
LOGIC SYSTEMS INCORPORATED
DRAWING:
EXAMPLE OF EACH CMOS PART
EXCMOS
ENGINEER:
WANG
DATE:
Mon May 7 11:43:22 1984
PAGE:
2 OF 6

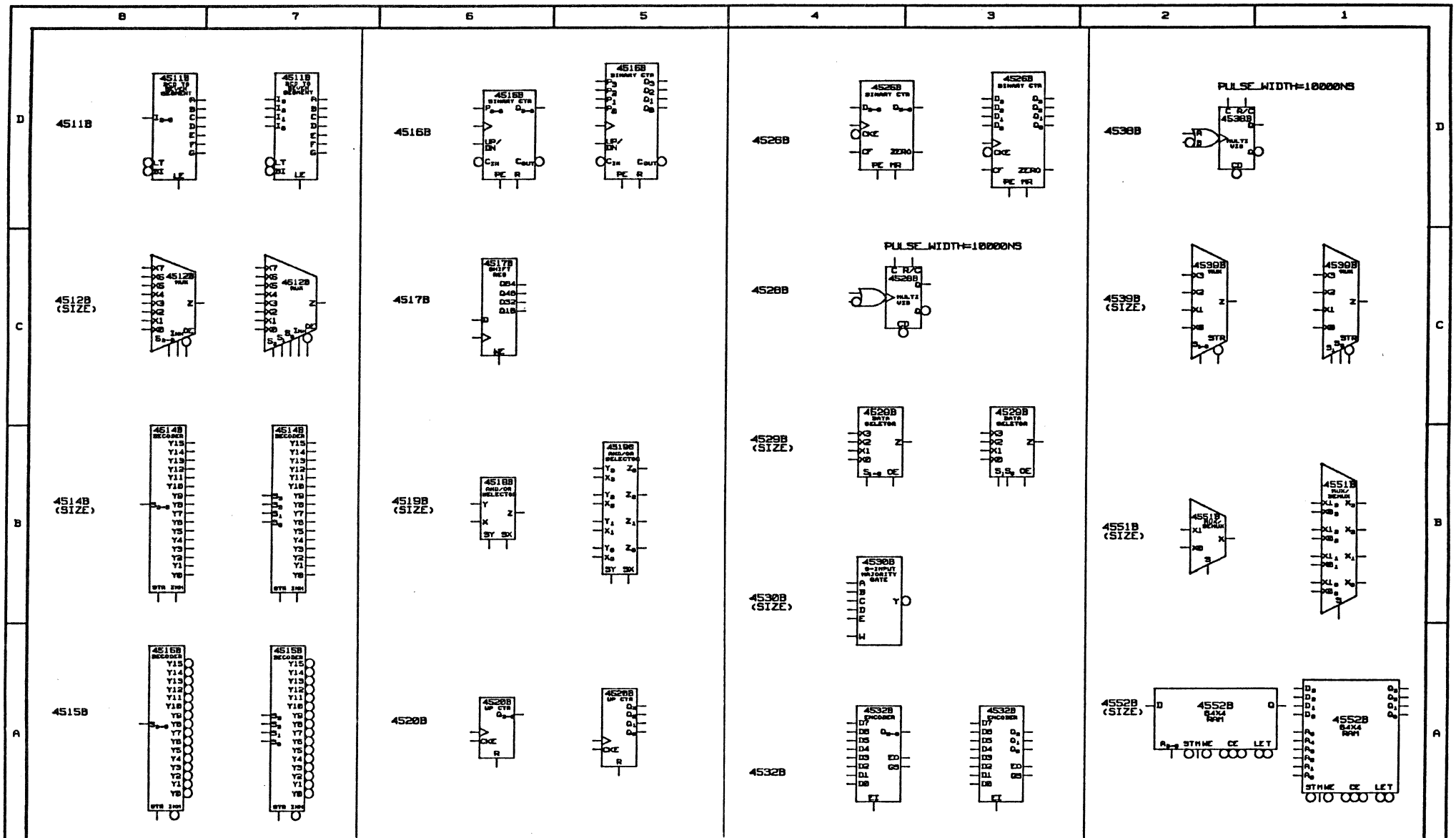


VALID
LOGIC SYSTEMS INCORPORATED

DRAWING:
TITLE=EXAMPLE OF EACH CMOS PART
ABBREV=EXCMOS

ENGINEER:
WANG

DATE:
Mon May 7 10:43:05 1984

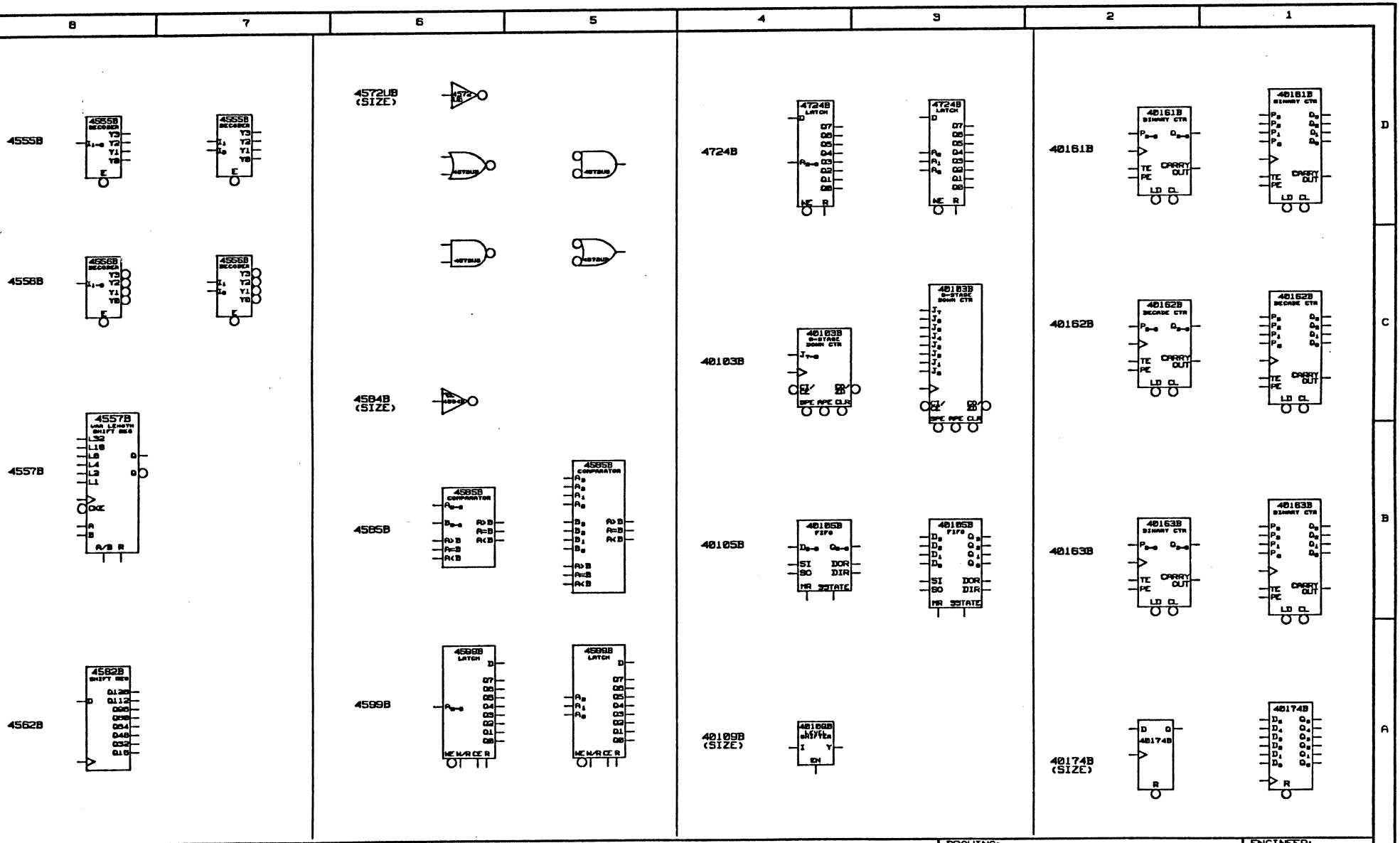


VALID
LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH CMOS PART EXCMOS

ENGINEER: WANG

DATE: Mon May 7 10:51:38 1984



VALID
LOGIC SYSTEMS INCORPORATED

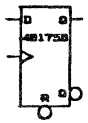
DRAWING:
TITLE=EXAMPLE OF EACH CMOS PART
ABBREV=EXCMOS

ENGINEER:
WANG

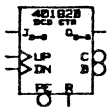
DATE:
Mon May 7 11:28:02 1984

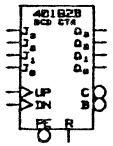
PAGE:
5 OF 6

40175B (SIZE)

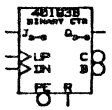


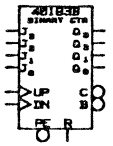
40192B





40193B





VALID
LOGIC SYSTEMS INCORPORATED

DRAWING: EXAMPLE OF EACH CMOS PART EXCMOS

ENGINEER: WANG

DATE: Mon May 7 11:37:18 1984

PAGE: 6 OF 6

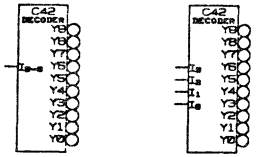
MM74C Library

Release 4.5 is the first release of the MM74C Library.

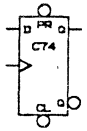
The MM74C Library requires approximately 0.14MBy (269 Blocks) of disk storage on the S-32. It contains bodies and physical for the following 10 components:

C42	bcd-to-decimal decoder
C74	dual D flip-flop
C154	4-line to 16-line decoder/demultiplexer
C157	quad 2-input multiplexers
C192	synchronous 4-bit up/down decade counter
C373	tri-state octal D-type latch
C374	tri-state octal D-type flip-flop
C902	hex non-inverting TTL buffer
C910	256 bit tri-state random access read/write memory
C914	hex schmitt trigger with extended input voltage

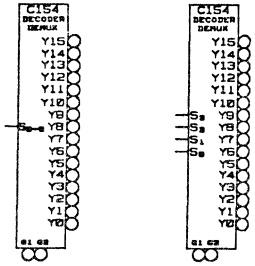
C42



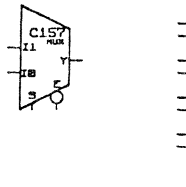
C74
(SIZE)



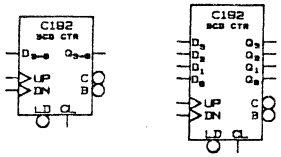
C154



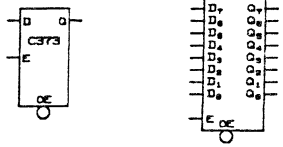
C157
(SIZE)



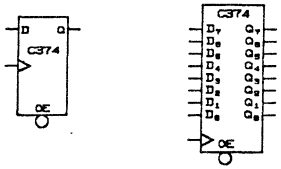
C192



C373
(SIZE)

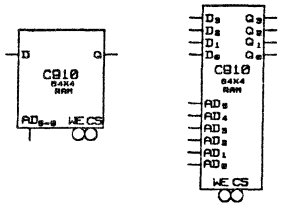


C374
(SIZE)



C902
(SIZE)

C910
(SIZE)



C914
(SIZE)

DEFINE
X_FIRST=0
X_STEP=SIZE

VALID
LOGIC SYSTEMS INCORPORATED

DRAWING:
EXAMPLE OF EACH MM74C PART
EXMM74C

ENGINEER:
WANG

DATE:
Thu May 3 14:40:51 1984

PAGE:
1 OF 1