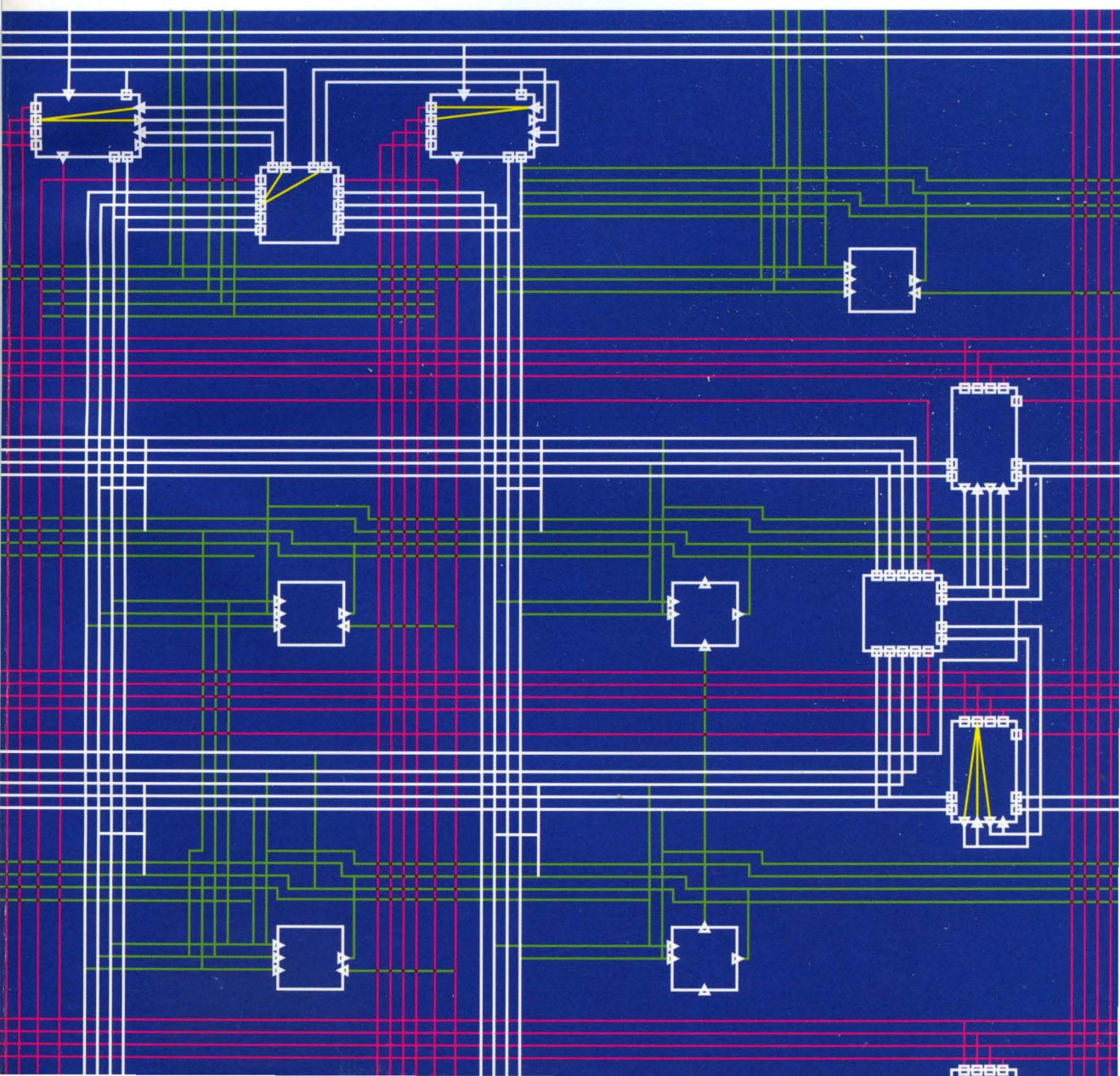
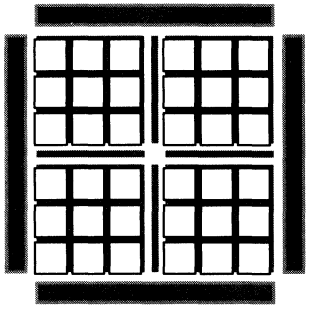
MOTOROLA

DL201/D

FPGA Data

Field Programmable Gate Arrays





Field Programmable Gate Arrays

Suggested References:

Packaging Manual for ASiC Arrays, Motorola Inc., 1993. Stock Code BR916/D

HDC Series Design Reference Guide, Motorola Inc., 1991. Stock Code HDCDM/D

Reliability and Quality Handbook, Motorola Inc., 1993. Stock Code BR518/D

MOTOROLA

FPGA Data
Field Programmable Gate Arrays

Advance Information

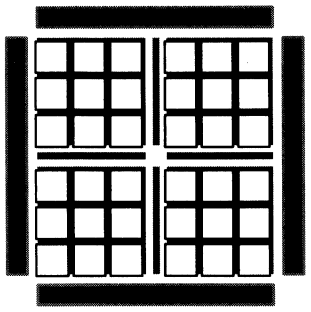
This databook contains device specifications for Motorola's Field Programmable Gate Arrays (FPGAs).

The MPA1000 series of FPGA is licensed under agreement with Pilkington Microelectronics, United Kingdom.

This document contains information on a new product. Specifications and information herein are subject to change without notice.

Printed in U.S.A.

CONTENTS



Field Programmable Gate Arrays

General Information 1

This section introduces the MPA1000 family, its architecture, programming modes, simple applications and a pin description.

CONTENTS

INTRODUCTION TO The MPA1000 FPGA Family

1

Introduction

The Field Programmable Gate Array, or FPGA is an outgrowth of the early days of bipolar PROMs in the 1970s. These early devices migrated into the early PLDs in bipolar and CMOS form. FPGAs are similar to the PLDs of yesteryear in that the user gets to specify the logic he wishes, and all devices provided by the semiconductor manufacturer are the same. Early FPGAs resemble their ancestors, the PROM, in that they use a look–up table approach to combinational logic. The early manufacturers used this approach and added flip–flops for storage elements to create an array of these large blocks of logic and storage.

The Motorola FPGA finds its roots in modern metal programmed gate arrays which have taken the form of a "sea of gates". The sea of gates architecture allows the designer to arrive at his logic with a large number of identical structures, the 2–Input NAND gate. All logic can be constructed using the 2–Input NAND as its building block. This is the basis of the MPA1000 family. Early attempts at creating an FPGA based on simple elements such as 2–Input NAND structures did not succeed because of limited routing capabilities. Most of the successful high density metal programmed gate arrays such as the Motorola H4C series, use three layer metal to aid routing (metal hierarchy). The problem has been solved for the MPA series of devices through hierarchical routing resources. The routing is accomplished through 3 levels: local, medium, and global buses.

The MPA1000 family allows the user to take advantage of the best features of metal programmed gate arrays without the need for nonrecurring engineering expense for these devices. Since the cells are fine grain they can be easily synthesized using logic synthesis third party software and open languages such as Verilog–HDL, and VHDL. The fine grain approach already resembles a standard sea of gates metal reprogrammed gate array, and the conversion procedure should be straight forward.

Features

Logic Array

- Fine Grain Structure, Flexible Logic Block Size
- High Logic and Register Density
- 8 Low–Skew (<1ns) Clocks
- Fast 0.65µm CMOS Process
- In–System Reconfigurable — SRAM Configuration Store

Input/Output

- Abundant Programmable I/O Cells with Input and Output Registers
- Programmable 3V/5V at Any I/O Site
- Programmable Output Drive — 6, 12mA Symmetrical
- 8–Bit Peripheral Bus for I/O Control, Decoders
- IEEE 1149.1 JTAG Boundary Scan Support

Development Tools

- 3rd Party Tools — Nonproprietary Commercial Quality Tools; Technology Independent Library, Maximum Reuse
- Front End — Standard Design Entry Tools Such as View*logic*®, Mentor, Exemplar and Synopsys
- Back End — Superior Auto Place and Route with NeoCAD™ Timing Wizard™
- Frequency–Driven Place and Route

Software

The MPA1000 series uses open third–party software for all functions. Motorola will sell low cost library locked versions of vendor independent software from View*logic* and NeoCAD. These tools may be "opened" through additions of library access by the original vendor or their representatives. The learning curve is transferable and not restricted to the Motorola MPA1000 family, thus eliminating the need for vendor proprietary tools.

Family Members

Part No.	Logic Cells	Internal Flip-Flops	I/O Cell Flip-Flops	FPGA Equivalent Gates[1]	Signal I/O Pads	Packages[2]
MPA1016	1600	400	160	3500	80	84PC, 128PQ
MPA1036	3600	900	240	8000	120	84PC, 128PQ 160PQ, 160PQµCool, 181PG
MPA1064	6400	1600	320	14200	160	160PQ, 160PQµCool, 208PQ, 224PG
MPA1100	10000	2500	400	22000	200	160PQ, 160PQµCool, 208PQ, 289PG

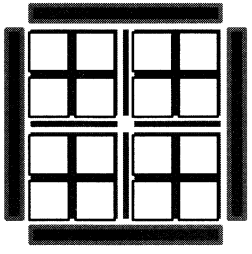
1 X–4000 Series gate array equivalents
2 PC=PLCC Package; PQ=PQFP Package; PQµCool = PQFP µCool Package; PG=PGA Package

MPA Field Programmable Gate Array Product Description

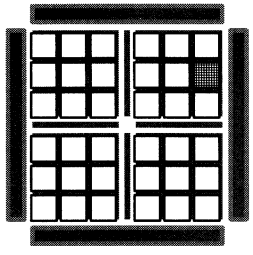
MPA1000 are members of a series of high density, high performance, reconfigurable field programmable gate arrays. When used with high performance automatic place and route tools, they offer an excellent solution to the demands of shrinking development cycles, rapid prototyping and applications that benefit from reprogramming in the field.

Logic Resources

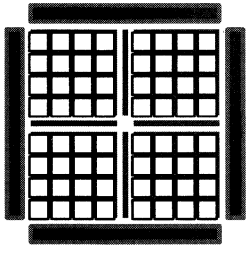
Logic resources in the Motorola FPGA are fine grain, each logic cell holds a single gate or predefined element, allowing logic block size to vary according to the specific function that is being implemented. The array is divided into four quadrants, each consisting of 2 x 2 (MPA1016) 3 x 3 (MPA1036), 4 x 4 (MPA1064) and 5 x 5 (MPA1100) zones (see Figure 1-1). Each zone consists of a 10 x 10 block of logic cells with port cells around the peripheral (see Figure 1-2). Port cells provide programmable connections between zones and global resources to the global buses. Between the quadrants are spaces for the inter–quadrant switches, which provide programmable connection points between quadrants. Around the chip periphery are I/O Cells.



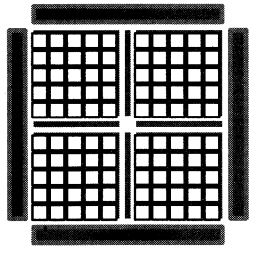
MPA1016 — 1600 Logic cells. Each quadrant consists of 2 x 2 zones.



MPA1036 — 3600 Logic cells. Each quadrant consists of 3 x 3 zones.



MPA1064 — 6400 Logic cells. Each quadrant consists of 4 x 4 zones.



MPA1100 — 10000 Logic cells. Each quadrant consists of 5 x 5 zones.

Figure 1-1. The MPA1016, MPA1036, MPA1064 and MPA1100

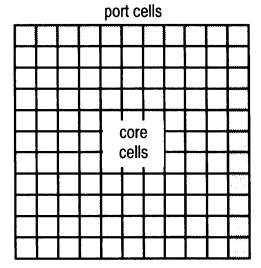


Figure 1-2. Logic Cells Within a Zone

Each input to the core cells has 2 inputs from 1 of 7 sources (see Figure 1-3). Five of the sources come from local level and two from Zone level interconnect. The output of each core cell may be connected to eight other core cells using Local interconnect, and may be connected to four Zone level buses. If the core cell is not driving a Zone level bus, then the structure may be used to join buses together to perform a 90° turn. The core cell may also be used to repower a signal, provide additional routing, and allowing a programmable means of adjusting routing resource to fit the needs of an individual design. The core cell has three states: repowering buffer, primary function and secondary function. All core cells contain the ability to invert the inputs and perform a 2–input AND gate function with programmable input inversion. Core cells are multifunctional. A group of four core cells are defined as a Tile (Figure 1-4). Within each tile, each of the core cells has one of four secondary functions. The secondary functions are: XOR (2 each), register (DFF or latch with CK enable and set or reset), and wire–OR. The device employs circuitry to allow core cells to be grouped together to form higher level functions such as: full adder, 2–input MUX, TFF, JKFF.

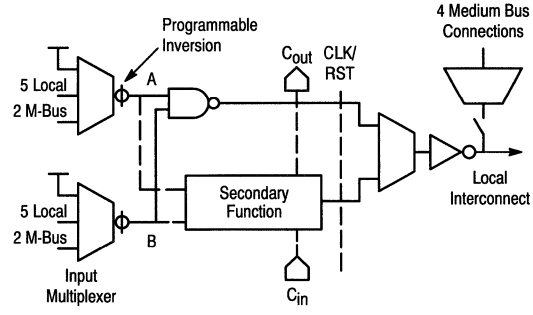


Figure 1-3. Core Cell

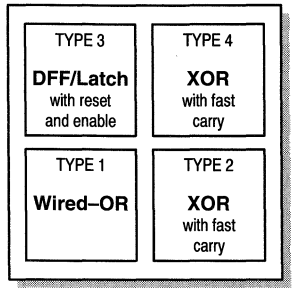
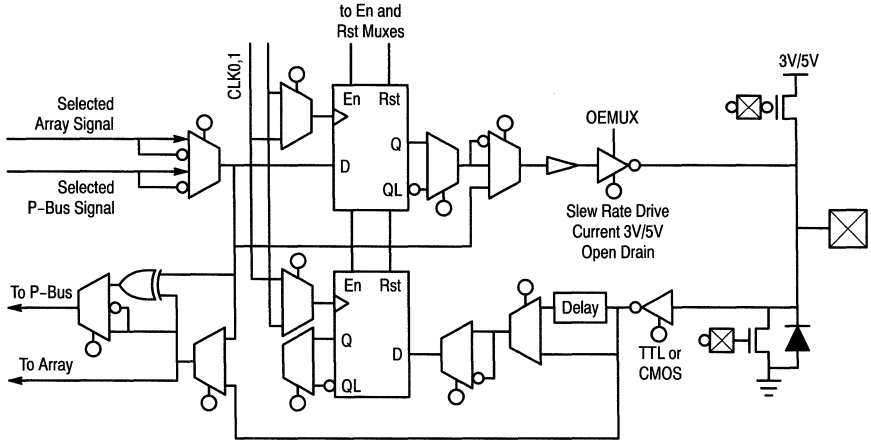


Figure 1-4. Secondary Function 2 x 2 Tiles



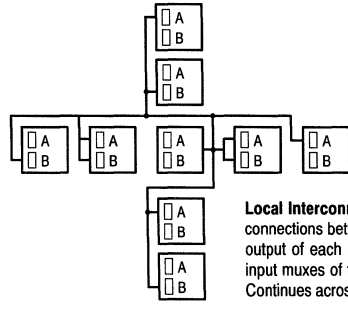
Input/Output Cells — there is one I/O cell for every two rows/columns of logic cells, thus the MPA1036 has a total of 120 signal I/O. Cells support bidirectional registering of signals, individually programmable 3V/5V levels, TTL or CMOS, programmable drive (6, 12mA) and programmable slew. An 8-bit peripheral bus runs between the array and I/O cells. JTAG boundary scan is supported with dedicated circuitry.

Figure 1-5. I/O Cells

The periphery of the device contains I/O cells (Figure 1-5), with 80, 120, 160 and 200 cells in the MPA1016, 1036, 1064 and 1100 respectively. Each I/O cell can be direct input or output, bidirectional, register input or output and registered I/O. Each register is configurable as either a latch or DFF. Hold time can be adjusted to compensate for the delay of the clocks to the I/O cell element. The input level may be either TTL or CMOS level. Output drive is configurable to 6 or 12 mA. Individual outputs may be either 3 volt or 5 volt level. Fully compliant IEEE 1149.1 boundary scan is also provided (Chapter 2, Page 2-7).

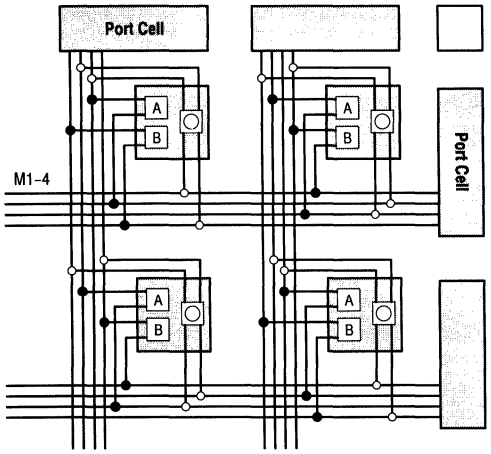
Routing Resources

Coupled with fast, hierarchical routing, a library of hard macros and advanced soft macro capability, the fine grain approach offers full flexibility for optimizing logic placement while maintaining maximum performance in critical areas. The core cell is partitioned hierarchically as are the local, medium and global routing resources. This minimizes bus loading and maximizes performance. Connections within zones are made using local interconnect (to each cell's 8 nearest perpendicular neighbors, see Figure 1-6), or medium range interconnect (to anywhere in the zone, see Figure 1-7). Horizontal and vertical Medium buses run the length and breadth of each zone (4 per row and column), via the medium buses, to neighboring zones or to the global bus network (see Figure 1-8).



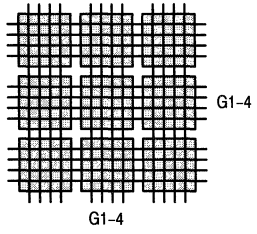
Local Interconnect — provides the fastest connections between neighboring cells. The output of each cell connects directly to the input muxes of the eight nearest neighbors. Continues across zone boundaries.

Figure 1-6. Local Interconnect



Medium Buses — used for intra-zone connections that are not possible with local interconnect; also for zone-zone connections via the port cells. Running the length and breadth of a zone, the are four medium buses per row and column of logic cells (left). Through the port cells, two of the medium buses can connect to the global routing network, while the other two can connect to the neighboring zones. Medium buses connect to the logic cells' input and output muxes (right), as does the local interconnect.

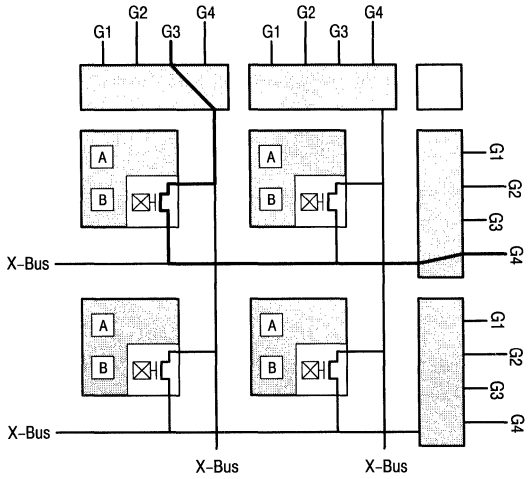
Figure 1-7. Medium Buses



Global Buses — run the length of a quadrant, and are the fastest interconnect for long journeys. The programmable inter-quadrant switches connect two buses for full width connections. There are four global buses for each row and column of zones.

Figure 1-8. Global Buses

Horizontal and vertical 'X' buses (Figure 1-9) also run the length and breadth of each zone (1 per row and column). In connection with a switch in each zone cell, they are used for making right angle turns on global buses. Global buses run the length and breadth of each quadrant (4 per row and column of core cells). Switches between the quadrants allow global buses to run the whole length of the device, if required. Global buses connect to zones via the port cells. Dedicated buses provide very low skew (<1.0nS) clock and reset distribution to the D flip–flop core cells. Eight of the user I/O cells may also be used as clock input cells. On–chip 3–states are implemented using a combination of dedicated zone wired–OR buses and the normal global interconnect. The user configurable I/O cells provide connections to and from the array (via global, medium and 'X' buses). All I/O cells also connect to the 8–bit peripheral bus, running around the edge of the device.



X-Bus — is a special bus for facilitating 90° turns between global buses. There is one X-bus per row and column of logic cells, running the length/breadth of a zone. Turns are made by the X-bus switch. The X-buses also serve other purposes, such as connecting the medium buses of two non-neighboring zones without using a global bus.

Figure 1-9. Global Buses

The peripheral bus is divided into 4 sections — 1 for each edge of the device. Sections can be connected using inter–side switches in the corners of the array. Inputs and outputs can be selected, inverted and/or latched as required. The peripheral bus can be used as either a conventional bus, or as a wired–OR bus for address decoding etc. The pull–ups are situated in the corners, along with the inter–side switches.

Configuration

The configuration logic is responsible for loading external configuration data into the device. The interfaces to the configuration logic are shown in Figure 1-10. Four dedicated input pins, Mode [3:0], determine the configuration Mode, the configuration clock source and also whether JTAG is to be enabled in this mode. There are two basic modes: Micro Mode and Boot From ROM (BFR) Mode. Micro Mode allows the devices to be configured from a Microprocessor, or similar, system, through a conventional peripheral interface. BFR Mode supports three sub modes. The mode programming is shown in Table 1-1.

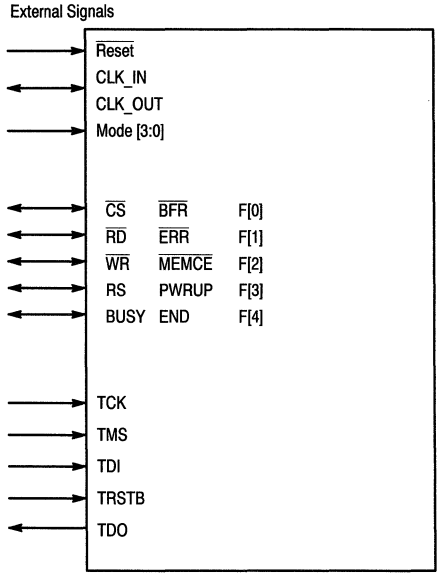


Figure 1-10. Interface Signal Description

Micro Mode

The table below shows the function of the external pins when microprocessor mode is selected (Table 1-2)

In micro mode the device behaves as a asynchronous microprocessor peripheral. A chip select ($\overline{CS}$) is derived from the processor address and enables a particular MPA device. In a multiple device subsystem, a chip select for each MPA device is required. When a device is selected, the data bus is used to write commands and data to the device or read current device status. There are two registers, the function register and the data/status register. Registers are selected using RS signal. RS is normally connected to the least significant address line. Reading the data register (RS=1) returns the status register contents. Configuration data is provided by writing the data register. Writing the function register (RS=0) causes the device to perform a specific configuration function. The configuration data format expected for each function is briefly described in Table 1-3. A more detailed description can be found in the section entitled "Data Vector Format". Configuration data is normally automatically generated by the bitstream generation software (BITGEN) in conjunction with a user design and an associated 'layout'.

The maximum data transfer rate is governed by the R/W timing described in Figure 1-13 and Figure 1-14. The processor should only write data to the device when BUSY is inactive. BUSY is only asserted when data cannot be accepted at the maximum rate and is not asserted for each byte transferred. The specific behavior of BUSY for each micro mode function is described in Table 1-3. When the device is first powered up, an internal reset sequence is initiated and BUSY is asserted (see "Behavior During Power–On–Reset"). BUSY will be de–asserted when the internal reset sequence completes and the $\overline{RST}$ pin is HIGH. The processor can monitor BUSY directly via the busy pin or indirectly by reading the status register.

If processor R/W cycles are faster than the timing shown, external circuitry can be used to inject wait states. Figure 1-11 and Figure 1-12 show an application circuit consisting of one or more MPA devices and the unspecified logic necessary to lengthen R/W timing based on $\overline{CS}$, $\overline{MEMW}$, $\overline{MEMR}$, BUSY, $\overline{RST}$, and an externally provided clock.

Note – for proper circuit operation the internal oscillator must be enabled and the bootstrap circuit should be activated 100µ s before the user inputs or outputs are enabled.

Table 1-1. Mode Programming

Mode Bits		
[1]	[0]	Description
0	0	**Micro Mode** — Micro interface circuitry with parallel (byte wide) data.
0	1	**BFR Mode (1)** — Boot From ROM using internal address generator and byte wide data.
1	0	**BFR Mode (2)** — Boot From ROM using external address generator and serial data. (Low pin count serial EEPROM)
1	1	**BFR Mode (3)** — Boot From ROM using external address generator and byte wide data.
Mode Bits		
[3]	[2]	Description
X	1	Use external clock for configuration.
1	X	Enable JTAG circuitry and pins.

Table 1-2. External Pins in Micro Mode

Pin Name	Micro	I/O		Description
Mode [3:0]	Mode [3:0]	I	D*	**Mode Pins**
$\overline{\text{Reset}}$	$\overline{\text{Reset}}$	I	D*	**General configuration reset**
CLK	CLK	I/O	D*	**Clock for configuration circuitry** — If external clock is selected, pin is an input. If not selected internal configuration is used and output through this pin.
F0	$\overline{\text{CS}}$	I	D*	**Chip select** for device in Micro mode.
F1	$\overline{\text{RD}}$	I	D*	**Micro read signal**
F2	$\overline{\text{WR}}$	I	D*	**Micro write signal**
F3	RS	I	D*	**Register select** — Two register locations are active: Function Register (RS = 0) and Data (RS = 1).
F4	Busy	O	D*	**Busy signal** — Active high when device is not ready to accept data, i.e. while device is resetting data in array or a data register to array transfer is taking place.
Data [7:0]	Data [7:0]	I/O	D*	**Micro data port** — for configuration logic.
JTAG [4:0]	J [4:0]	I/O	User/ JTAG	**JTAG pins** — JTAG or User I/O is selected by Mode [3].

* Dedicated — Pins that cannot be used for User I/O and are only used for controlling the function of the device are referred to as dedicated.

Table 1-3. Micro Mode Function Register Map

Data [7:4]	Data [3:0]	Function
XXXX	0000	**Normal operation** — No specified function.
XXXX	0001	**Reset device configuration** — Internal reset state machine initiates reset sequence. Busy signal is asserted immediately after access and remains active until reset is completed.
XXXX	0010	**Load configuration vector** — Vector is loaded by writing vector to data register. (Configuration format should be identical to BFR mode) A read access to the data register during this operation returns the current status register contents. Busy is asserted as soon as the Error Check Byte (ECB) is loaded at the end of each row, then deasserted if an error has been detected or the internal transfer of the data to the array has been completed. **The row address register contents are reset and then incremented as each successive row of data is loaded. Writing this value resets the internal ID and ECB error flags. If an error is detected during this operation, NO write accesses to the array can take place.
XXXX	0011	**Reset row** — Indicates that the next data value to be written to the data register is a device row address and that a row reset sequence is to be initiated when that value is written. Busy is asserted immediately when the row address is loaded and deasserted when the operation is complete.
XXXX	0100	**Load row** — Indicates that the next data value to be written to the data register is a row address and that the data following this is row configuration data, including the ECB. A read access to the data register during this operation returns the current status register contents. Again, busy is asserted and deasserted in the manor described above.

Table 1-3. Micro Mode Function Register Map (continued)

Data [7:4]	Data [3:0]	Function
XXXX	0101	**Read row** — Indicates that the next data value to be written to the data register is a row address and that a row read access is to be initiated when this has been written. Busy is asserted during this operation. Row data is accessed by successive reads to the data register. (No ECB value is appended).
XXXX	0110	**Read ID** — 4 reads to the data register will return the 4 byte device ID. Primarily for test. If user cannot configure device, this gives some diagnostic capability as to suitability of configuration data.
—	0111	**Bits [3:0]** — Reserved pattern
—	1XXX	**Bits [3:0]** — Reserved pattern
1XXX	XXXX	**Bootstrap circuitry enabled**
X1XX	XXXX	**Internal oscillator disabled** — Disable is only effective if external clock mode is selected.
XX1X	XXXX	**User inputs enabled**
XXX1	XXXX	**User outputs enabled**

Table 1-4. Micro Mode Status Register

Bit Position								Function
[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]	
0	0	0	0	0	X	X	1	Incorrect Device IDs
0	0	0	0	0	X	1	X	Row configuration data error
0	0	0	0	0	1	X	X	Busy signal asserted. Allows software handshaking of data if hardware wait states are not to used.

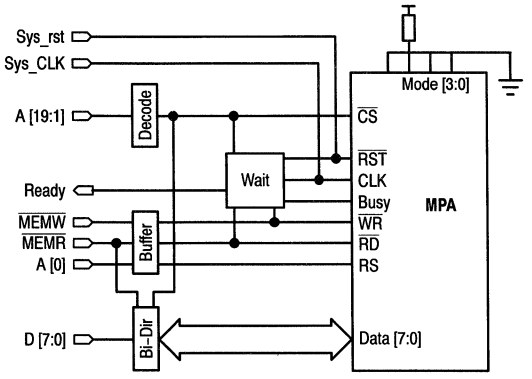


Figure 1-11. Micro Mode Application Circuit
(Single Device)

1

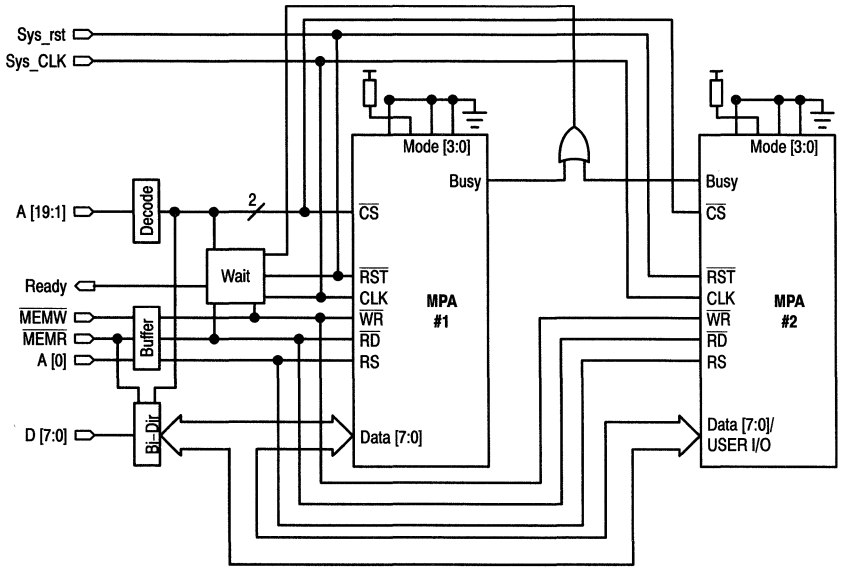


Figure 1-12. Micro Mode Application Circuit (Multiple Devices)

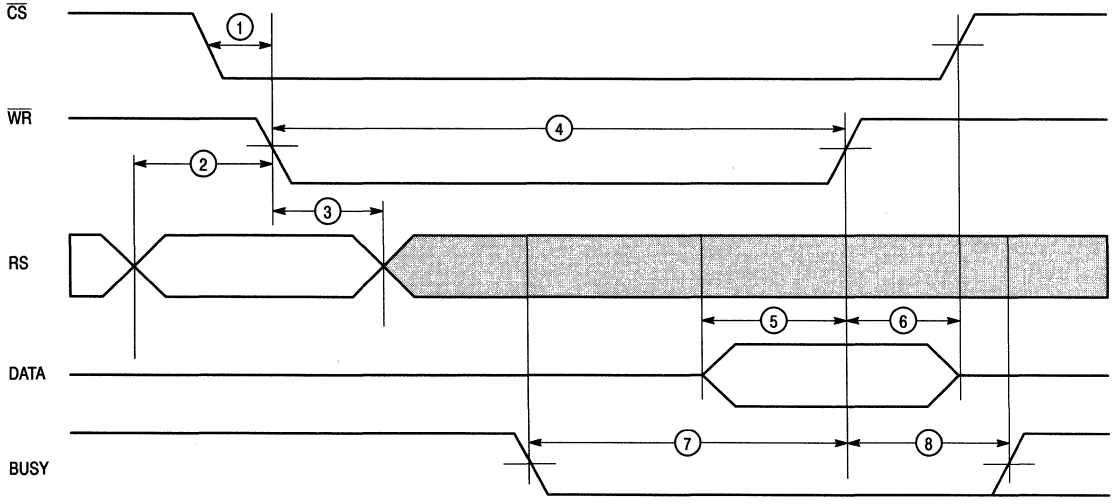


Figure 1-13. Micro Mode External Timings (Write Cycle)

Number	Characteristic	Min	Max	Unit	Notes
1	$\overline{CS}$ Setup before Write	10		ns	
2	RS Setup before Write	10		ns	
3	RS Hold after Write Falling Edge	10		ns	
4	Write Pulse Width	50		ns	
5	Data Setup to End of Write	20		ns	
6	Data Hold after Write	10		ns	
7	Busy Inactive before End of Write	50		ns	
8	Busy Active after Write	0	20	ns	

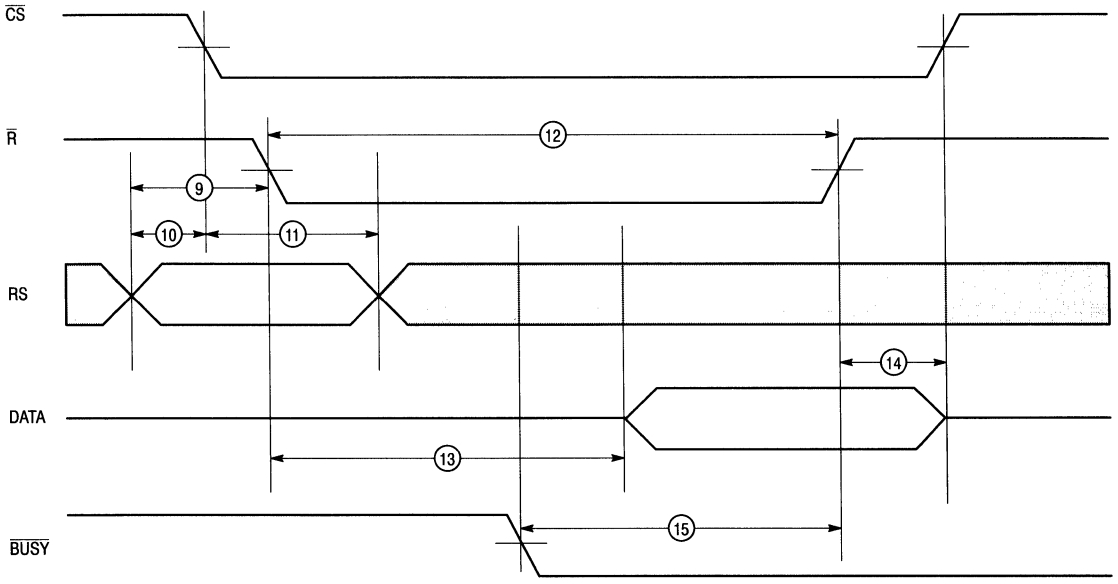


Figure 1-14. Micro Mode External Timings (Read Cycle)

Number	Characteristic	Min	Max	Unit	Notes
9	$\overline{CS}$ Setup before Read	10		ns	
10	RS Setup before Read	10		ns	
11	RS Hold after Read Falling Edge	10		ns	
12	Read Pulse Width	50		ns	
13	Data Access Time	20	40	ns	
14	Data Hold Time after Read	0	10	ns	
15	Busy Inactive before End of Read	50		ns	

Boot From ROM Modes (BFR)

Table 1-5. Description of External Pins in BFR Mode

Pin Name	Micro	I/O		Description
Mode[3:0]	Mode[3:0]	I	Dedicated*	**Mode Pins** – 16 modes supported.
$\overline{Reset}$	$\overline{Reset}$	I	Dedicated	**General configuration reset**
CLK	CLK	I/O	Dedicated	**Clock for configuration circuitry** — If external clock is selected, pin is an input. If not selected internal configuration clock is used and output through this pin.
F0	$\overline{BFR}$	I	Dedicated	**Boot From ROM initiate** — A falling edge on this signal initiates an internal reset sequence. When the reset sequence is completed a configuration sequence is initiated.
F1	$\overline{ERR}$	I	Dedicated	**Error** — Internal checksum error on device ID mismatch has been detected.
F2	$\overline{MEMCE}$	I	Dedicated	**Chip enable for external memory signal** — Inactive during reset, active during configuration.
F3	PWRUP	I	Dedicated	**Power up** — Disables User I/O and internal bootstrap.
F4	END	O	Dedicated	**End of configuration** — Active low during reset, goes high when device is configured.

* Dedicated — Pins that cannot be used for User I/O and are only used for controlling the function of the device are referred to as dedicated.

Table 1-5. Description of External Pins in BFR Mode (continued)

Pin Name	Micro	I/O		Description
DCLK	DCLK	I/O	Dedicated	**External data clock** — Data is latched internally on rising edge. DCLK is driven by device currently being configured and tristate at all other times. A pull–up on line is required when chaining devices together each device in turn drives DCLK. If the first device in the chain is using an internal address generator, then the DCLK for that device is both an input & output. This input drives the internal address counter of first device.
Data [7:0]	Data [7:0]	I	User/Data	**Data port** — for configuration logic.
Add [17:0]	Add [17:0]	O	User/Data	**Address output** — If internal address generation is selected.
JTAG[4:0]	J [4:0]	I/O	User/JTAG	**JTAG pins** — JTAG or User I/O is selected by Mode [3].

* Dedicated — Pins that cannot be used for User I/O and are only used for controlling the function of the device are referred to as dedicated.

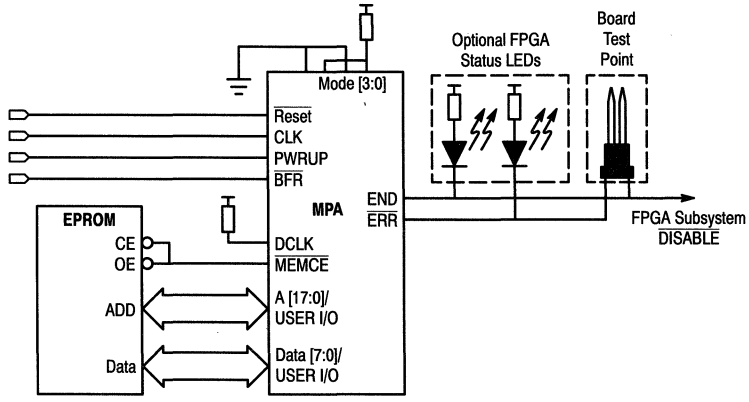


Figure 1-15. BFR Mode (1) Application Circuits
(Single Device)

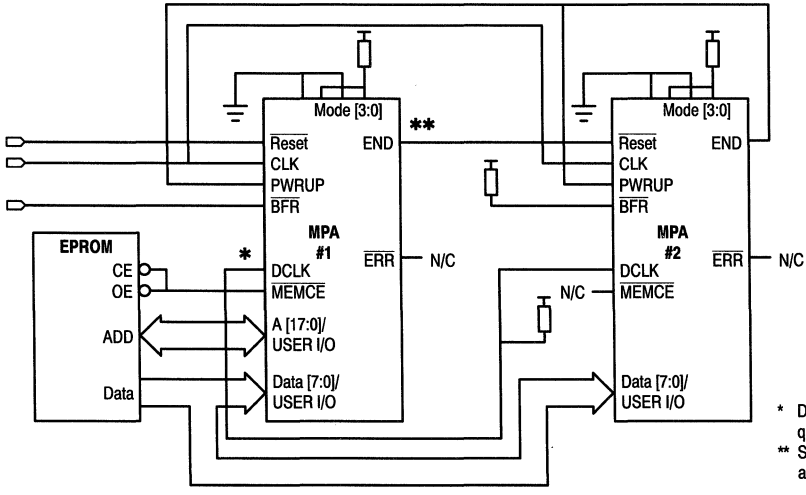


Figure 1-16. BFR Mode (1) Application Circuits
(Multiple Devices)

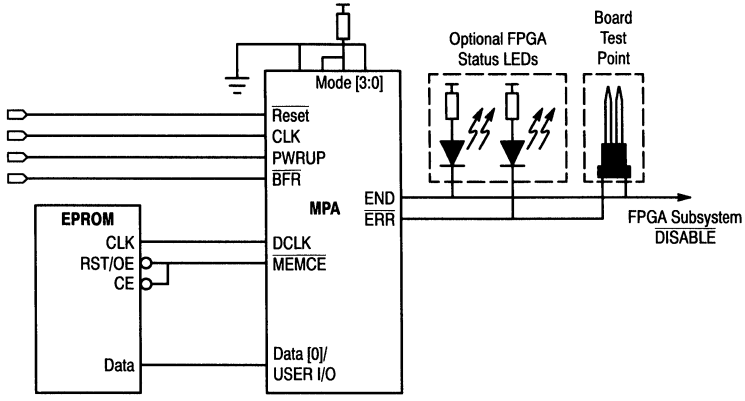


Figure 1-17. BFR Mode (2) Application Circuits
(Single Device)

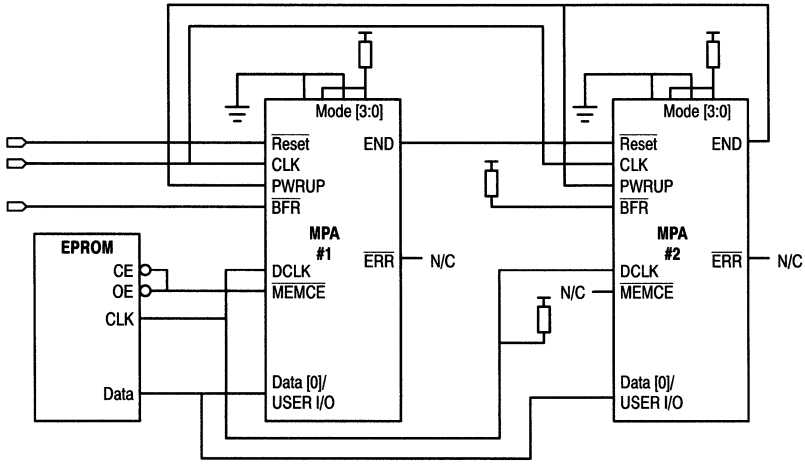


Figure 1-18. BFR Mode (2) Application Circuits
(Multiple Devices)

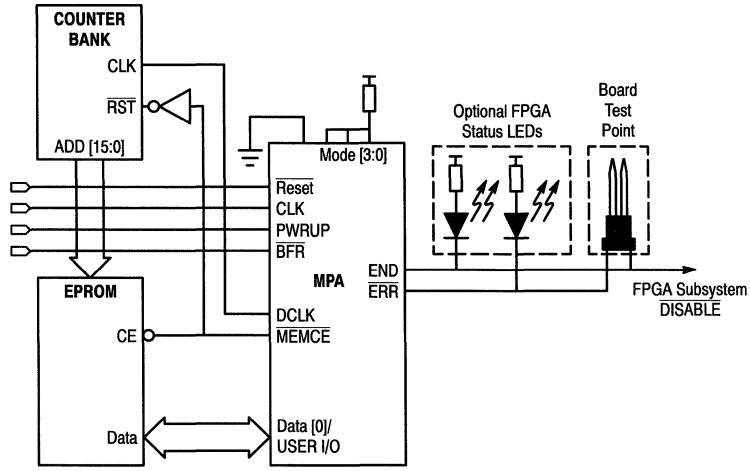


Figure 1-19. BFR Mode (3) Application Circuit
(Single Device)

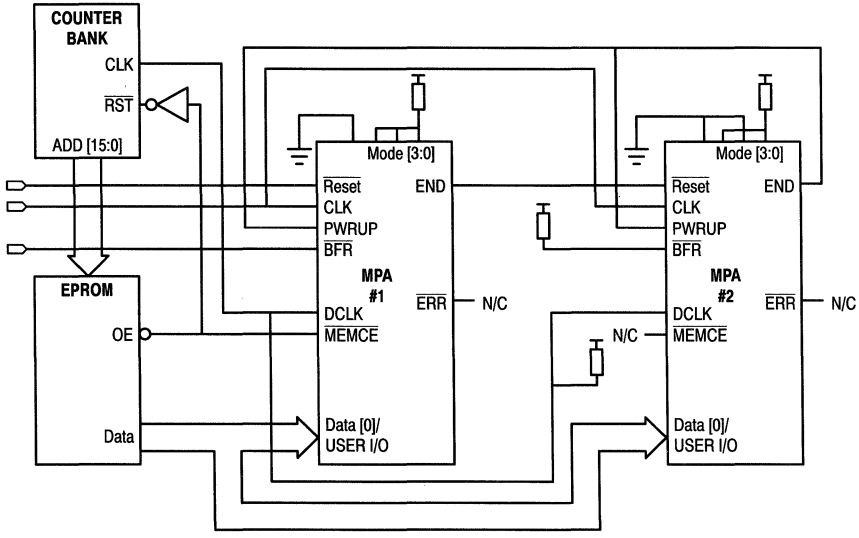


Figure 1-20. BFR Mode (3) Application Circuit
(Multiple Devices)

Boot From ROM Sequence

An overview of BFR timing is shown in Figure 1-21. This particular sequence has been initiated by the BFR signal being asserted. The configuration behavior for sequences initiated for Power–On–Reset and also the external reset are described later.

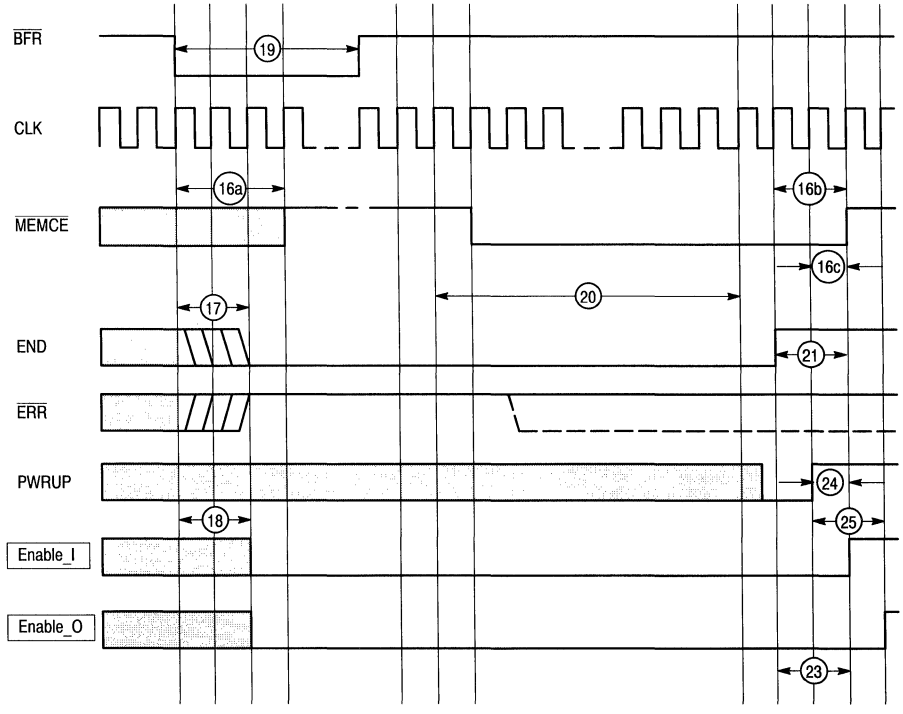


Figure 1-21. BFR Sequence

Number	Characteristic	Min	Max	Unit	Notes
16a	$\overline{BFR}$ Low to $\overline{MEMCE}$ High		3		
16b	END High to $\overline{MEMCE}$ High	1		CLK	
16c	PWRUP to $\overline{MEMCE}$ High	1		CLK	
17	$\overline{BFR}$ Low to END Low	0	3	CLK	
18	$\overline{BFR}$ Low to Internal Disable	0	3	CLK	
19	$\overline{BFR}$ Pulse Width	50		CLK	
20	Configuration Sequence Duration			CLK	Configuration sequence dependent on device size
21	END to Enable User Inputs	1		CLK	
22	END to Enable User Outputs	2		CLK	
23	End to Enable Bootstrap	1		CLK	
24	PWRUP to Enable User Inputs	1		CLK	
25	PWRUP to Enable User Outputs	2		CLK	

The assertion of the $\overline{BFR}$ signal triggers an internal array reset sequence. The diagram shows a Reset Sequence Initiate (RSI) initiating this reset, a second pulse, Reset Sequence End (RSE), is generated upon completion of this sequence. During this time the $\overline{MEMCE}$ signal is deasserted (This is necessary to reset the counter for the externally generated address modes) and the END signal is deasserted to indicate that the device is unconfigured. The User I/Os are all disabled for input and output.

Once the reset is completed, the configuration starts and the external memory enabled by the $\overline{MEMCE}$ signal.

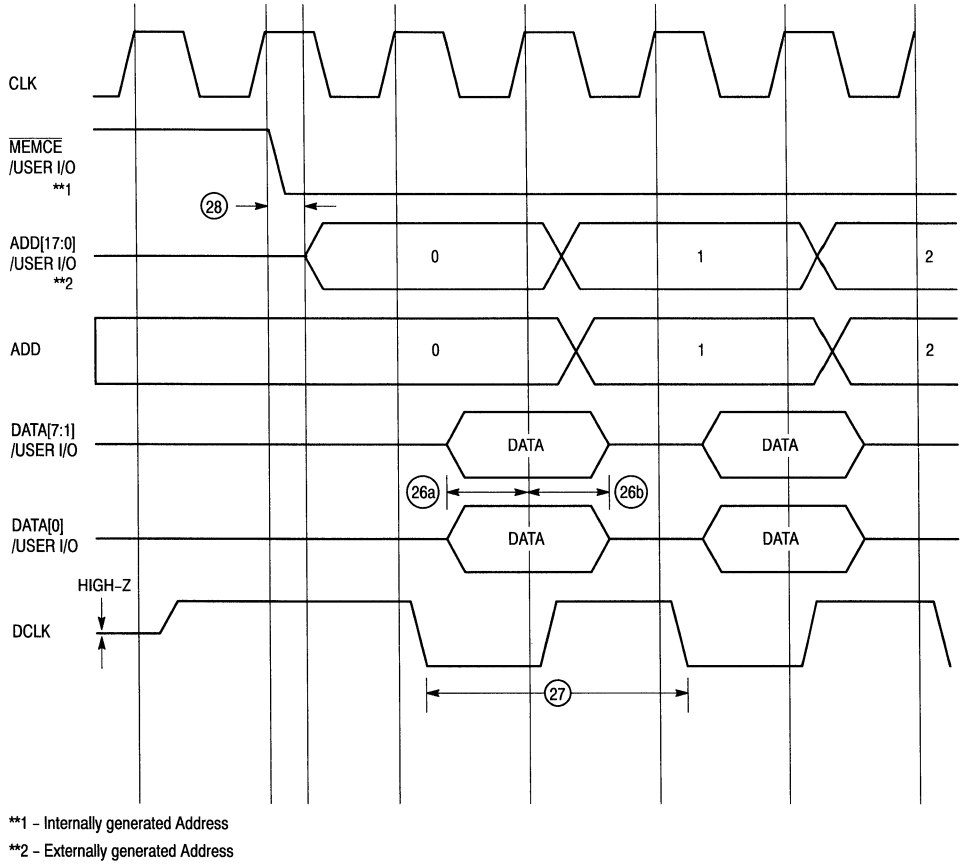
If the device is successfully configured, the Configuration Sequence End (CSE) signal is asserted. If the PWRUP signal is high and configuration is completed the device becomes active, and the I/Os are enabled. The device remains configured but in the power down state if PWRUP is low. PWRUP is ignored during reset and configuration.

BFR Timing Detail

All the timing in BFR Mode is relative to the configuration clock. Figure 1-22 below shows the timing detail for all three BFR Mode accesses. The CLK wave form is the configuration clock internal to the device. ADD[17:0]/User I/O and ADD are the internally and externally generated Address signals respectively. The data bus is shown as Data[7:1] and Data[0] this represents the parallel and serial configuration modes, the access timing is identical for both cases. The data clock signal (DCLK) can be an input or output depending on the Mode selected.

In BFR mode 1 (internal address generation, Figure 1-15 and Figure 1-16) the ADD[17:0] are active during configuration, tristate during reset and conventional User I/O during normal operation. The DCLK signal is an input and must be tied high, unless devices are daisy chained together. Two internal configuration clock cycles are required to access data. The address changes and the data is latched on a rising edge of the internal clock. In BFR mode 2 (External address generation, serial ROM, Figure 1-17 and Figure 1-18) the internal address generator is inactive and the ADD[17:0] pads inactive or User I/O. The DCLK signal is now a tristate output which is active when the device is configuring. A weak pull–up resistor is required on the DCLK signal, so spurious clk pulses are not produced when the output is not enabled. The data shift internal register at the bottom of the array is byte wide, circuitry is used in BFR mode 2 to convert the incoming serial data into a parallel stream.

The external address counter is incremented on the rising edge of DCLK and the data is latched on the rising edge of the internal clk signal corresponding to that DCLK edge. BFR mode 3 (Figure 1-19 and Figure 1-20) has the same timing as mode 2 but with parallel data.



**1 - Internally generated Address

**2 - Externally generated Address

Figure 1-22. BFR Data Access Detail

Number	Characteristic	Min	Max	Unit	Notes
26a	Data Setup to CLK	20		ns	
26b	Data Hold after CLK	0		ns	
27	DCLK Period (When Active)	2	2	clk	
28	CLK to Address Valid (Internal Generator)	15		ns	

BFR Timing Detail — End of Row Behavior

The device continues to access data every DCLK cycle during the configuration of a Row. Data is transferred to the array when the contents of an entire row has been loaded into the data shift register. The DCLK cycles halt until the internal data transfer has taken place. The $\overline{Err}$ signal goes low 2 clock cycles after the last byte (or bit in serial mode) has been loaded if an error has been detected. (see Figure 1-23).

Internal data transfers are initiated by the Start Access flag (SA), the End access flag (EA) indicates that the transfer has been completed.

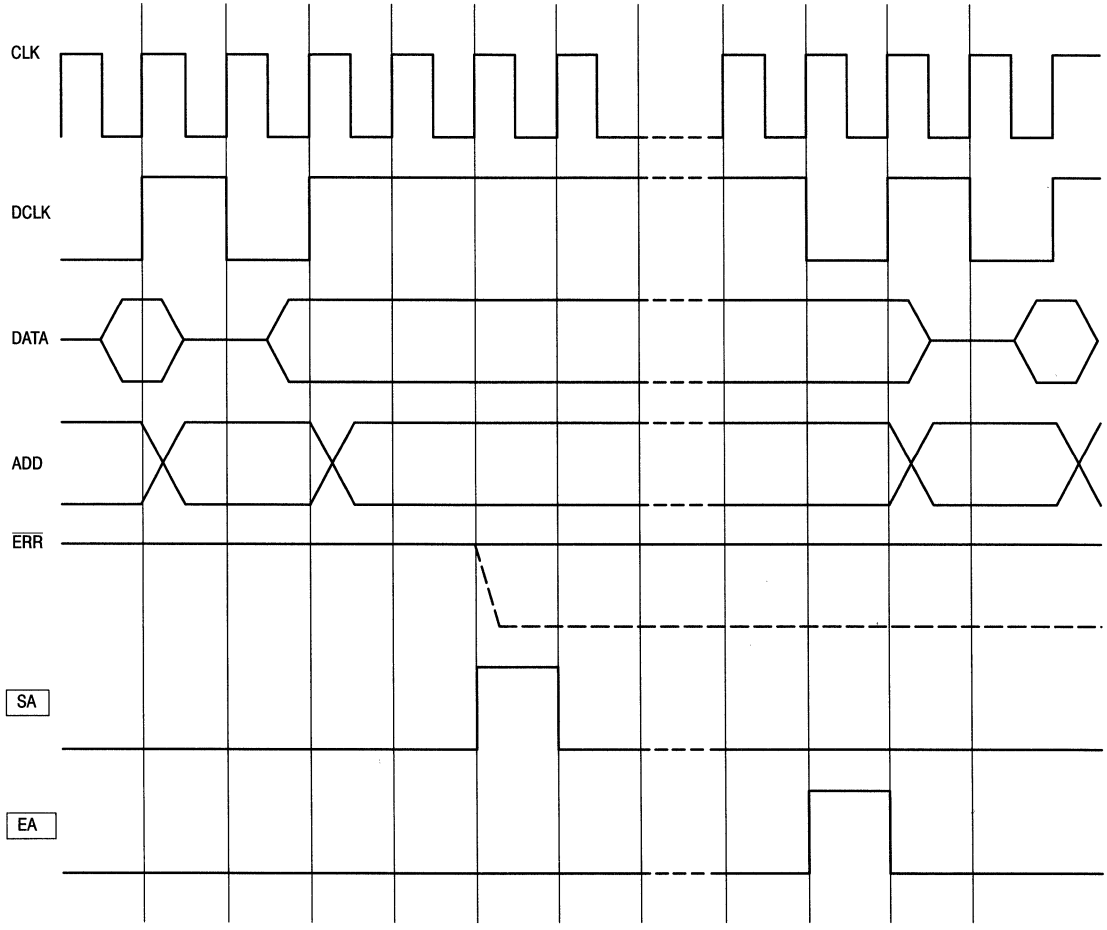


Figure 1-23. End of Row Behavior Timing

BFR Timing Detail — Daisy Chaining Devices

Many devices can be configured from the same ROM. To do this devices have to be 'daisy chained'. Effectively, the first device configures itself and, on completion, passes control to a second device for configuration, which in turn enable a third etc. All devices can be enabled at the same time by use of the PWRUP signal. Figure 1-24 shows the timing for this transfer of control. Labels #1 & #2 refer to the signals of FPGA #1 and #2.

FPGA #1 inhibits FPGA#2 from configuring by holding the RST #2 input low during FPGA #1 configuration. FPGA #1 is active during this time and FPGA #2 is inactive.

When #1 is configured END #1 is asserted, FPGA #1 becomes inactive and control passes to #2.

In BFR mode (1), the internal address counter of #1 remains active after configuration until PWRUP #1 is asserted.

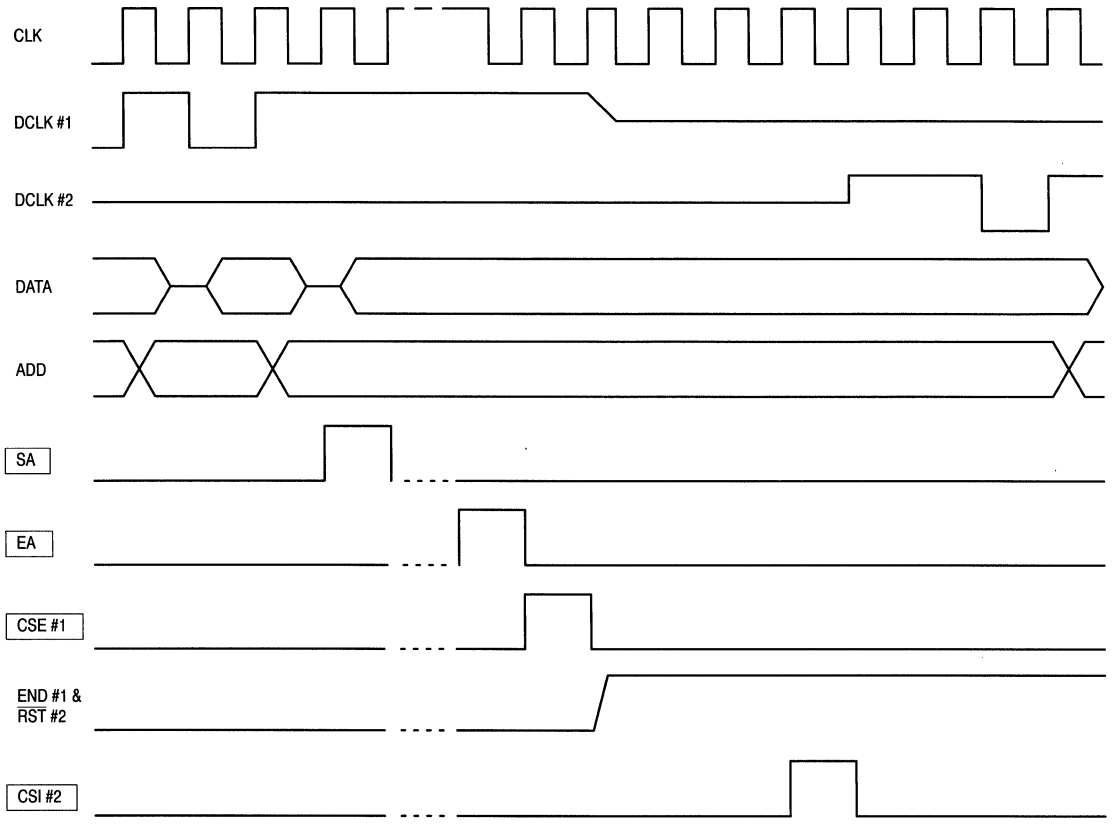


Figure 1-24. BFR Daisy Chaining

Clock Specification

Two clock sources are possible, Internal and External. The clock source for configuration is determined by Mode pin [2]. A '0' on this input signifies that an internal clock is to be used for configuration and the CLK pin becomes an output. A '1' on this output signifies that an External clock is to be used and the CLK pin becomes an input.

Internal Clock Specification

The internal ring oscillator has a frequency range from 10MHz to 40MHz, as shown in Figure 1-25.

The 10MHz to 40MHz clock output is divided by 8 internally and the resulting clock may be used to control configuration circuitry.

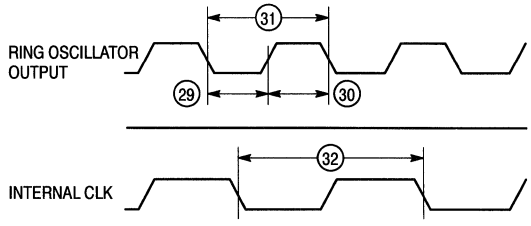


Figure 1-25. Internal Ring Oscillator

Num	Characteristic	Min	Typ	Max	Unit
29	Ring Oscillator Low	10	25	50	ns
30	Ring Oscillator High	10	25	50	ns
31	Ring Oscillator Period	25	20	100	ns
32	Internal Clock Period	200	400	800	ns

Behavior During Power–On–Reset

During power on reset the internal clock is always enabled, i.e. the state of mode pin [2] is ignored. When the power on reset sequence is completed, control for the configuration logic may be transferred to the external clock.

The ring oscillator may be disabled in the $\overline{\text{PWRUP}}$ (i.e. power down) state if an external clock is selected. This is only possible if the power on reset sequence is completed, as described above. If the internal clock mode is selected the ring oscillator and clock circuitry can never be disabled.

This applies equally to the BFR modes and Micro modes.

External Clock Specification

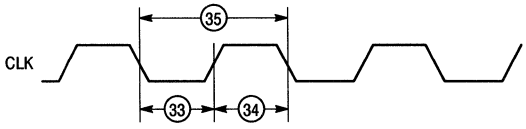


Figure 1-26. External Clock Timing

Num	Characteristic	Min	Max	Unit	Notes
33	External Clock Low	10		ns	
34	External Clock High	10		ns	
35	External Clock Period	25		ns	

Reset Specification

An internal reset sequence can be initiated in 4 ways: as a result of internal power–on–reset, an external reset, the BFR signal going low, or in micro mode, writing a Reset Device Configuration byte to the function register.

Internal Power–On–Reset

To ensure that the Power–On–Reset signal has a duration in excess of the supply settling time, an internal counter clocked by the internal configuration clock is used. The timing for this given in Figure 1-27.

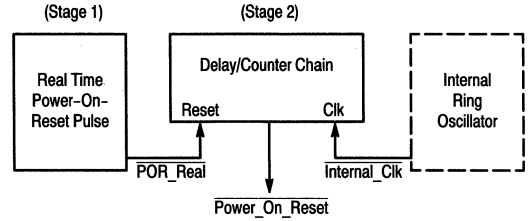


Figure 1-27. 2–Stage Power–On–Reset

The first stage of the power on reset circuitry generates an internal signal with a duration of ~10µS (labelled POR_Real), this resets the internal counter and holds off the internal count. This internal counter has fourteen stages, resulting in a power on reset pulse duration as shown in Figure 1-28.

Internal configuration clock Range: 1.25 MHz – 5 MHz
POR duration: 128 mS – 32 mS

During the Power–On–Reset, Reset sequence only, the internal circuitry continuously cycles through all row addresses writing '0' to all SRAM locations. A final single reset sequence is performed when the internal Power–On–Reset signal is deasserted.

In BFR mode, if the External $\overline{\text{reset}}$ signal is not active the device starts to configure immediately the final reset sequence is completed.

In Micro Mode, the Busy signal is asserted until the internal reset sequences are completed, the device is then ready to be configured under the control of the processor system.

External Reset

A Reset sequence is initiated by the falling edge of $\overline{\text{Reset}}$. Once the single reset sequence is initiated it can not be terminated by $\overline{\text{Reset}}$ going high again.

In BFR mode once the internal reset sequence is completed and the external $\overline{\text{reset}}$ signal is deasserted the device will start to configure. The external $\overline{\text{reset}}$ may be used to hold off configuration. Configuration is initiated on the rising edge of the external $\overline{\text{reset}}$. This is shown in Figure 1-29. The clock controlling this reset sequence is the configuration clock which may be either the internal clock or an external clock, depending on level of the mode[2] pin.

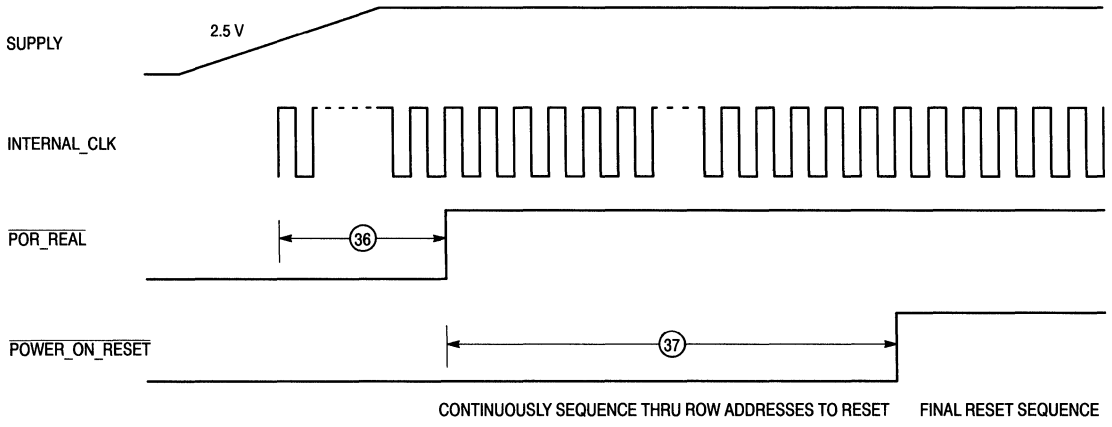


Figure 1-28. Power–On–Reset Circuitry Timing

Number	Characteristic	Min	Max	Unit	Notes
36	POR Real	10	1000	µs	
37	Power–On–Reset (Active)	32.7	131	ms	

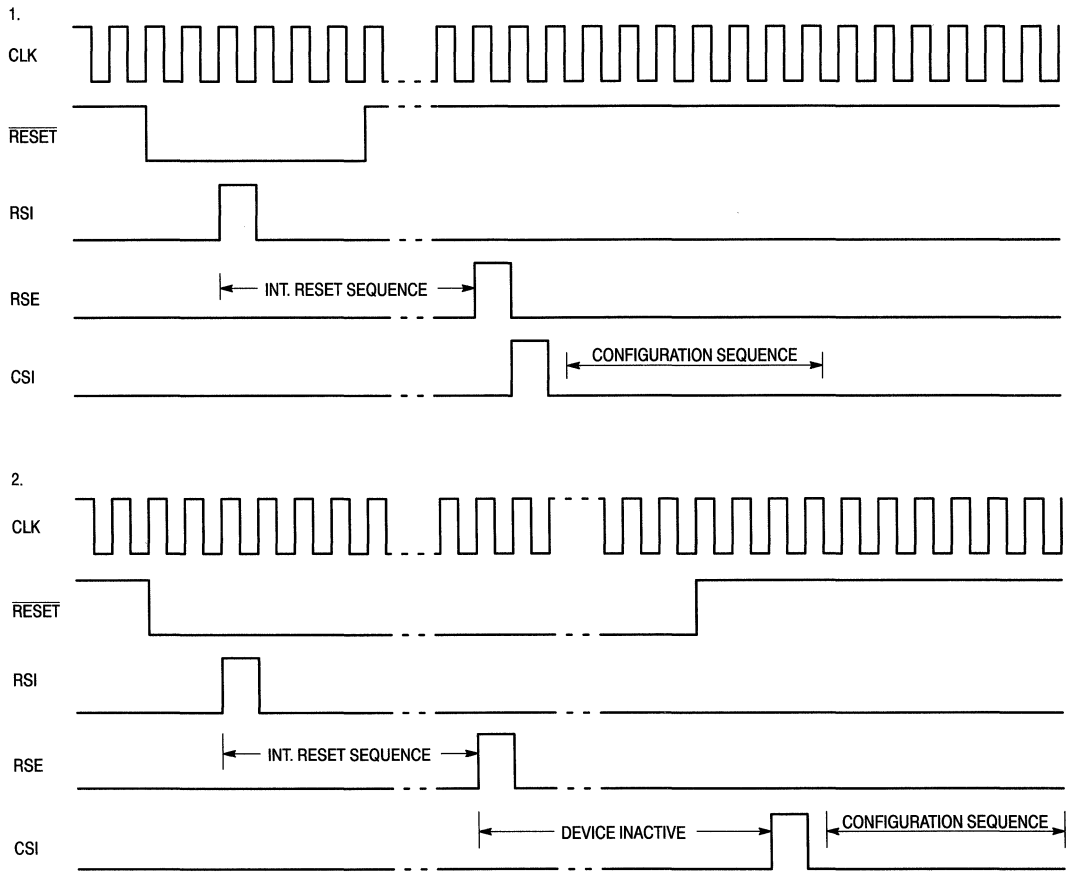


Figure 1-29. External Reset Behavior

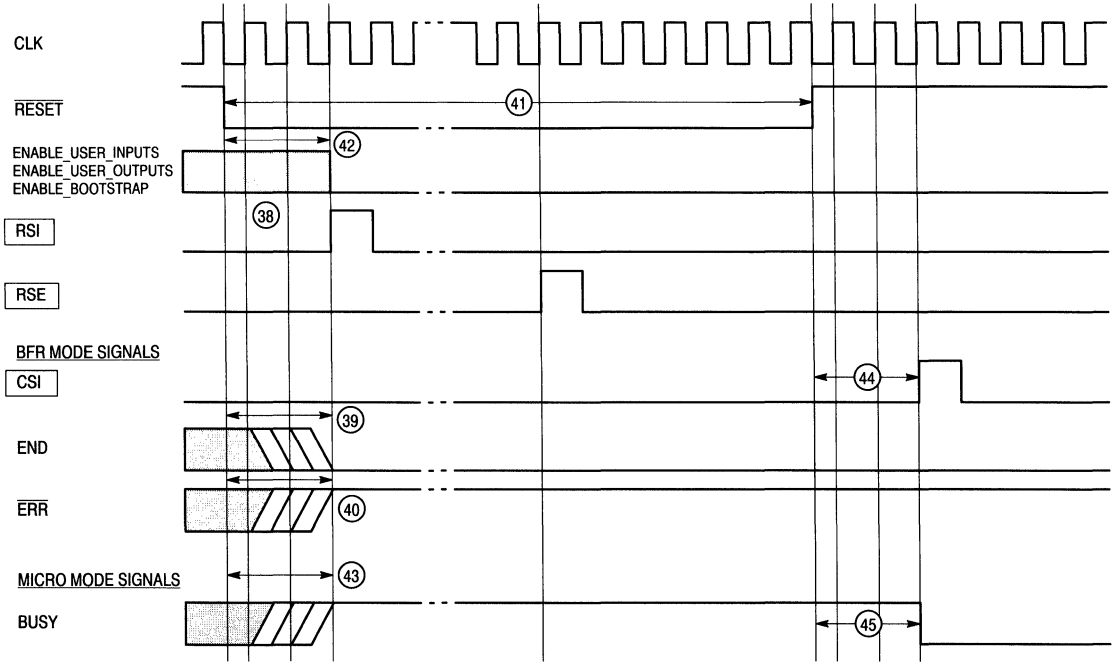


Figure 1-30. External Reset Timing

Number	Characteristic	Min	Max	Unit	Notes
38	$\overline{\text{RESET}}$ Low to Reset Sequence	2	3	clk	
39	$\overline{\text{RESET}}$ Low to END Low	0	3	clk	
40	$\overline{\text{RESET}}$ Low to $\overline{\text{ERR}}$ High	0	3	clk	
41	$\overline{\text{RESET}}$ Pulse Width	50		ns	
42	$\overline{\text{RESET}}$ Low to Internal Disable	0	3		
43	$\overline{\text{RESET}}$ Low to Busy Active	0	3		
44	$\overline{\text{RESET}}$ High to CSI Pulse	2			
45	$\overline{\text{RESET}}$ High to Busy Inactive	2			

In Micro mode, the busy signal remains high while the reset signal is asserted and until the internal reset sequence is completed.

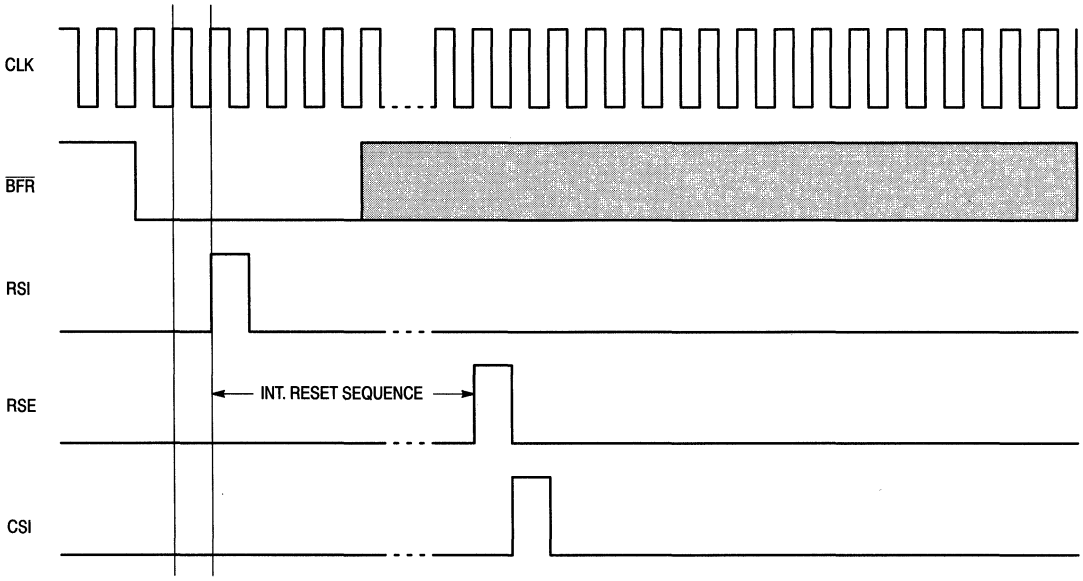


Figure 1-31. BFR Reset Behavior

BFR Signal (In BFR Mode Only)

The falling edge of BFR initiates an internal reset sequence. Once the sequence is completed the device will try to configure itself. The rising edge of BFR is ignored. This is shown in Figure 1-31.

This behavior is similar to that of the external reset signal, other than the response to a rising edge. The External reset signal is a true device reset and whilst this is asserted the device remains unconfigured. The BFR signal is a reset and configure signal.

Data Vector Format

The configuration Data Vector has two main components: The Header Block followed by Configuration Data, shown below:

Header Block

Device ID [3]
Device ID [2]
Device ID [1]
Device ID [0]
Data Type

Configuration Data Block (Normal Data)

Data 0 (*Row 0*)	Data 1 (*Row 0*)	~	~	~	ECB 0
Data 0 (*Row 1*)	Data 1 (*Row 1*)	~	~	~	ECB 1
		"	"	"	
		"	"	"	
Data 0 (*Row x*)	Data 1 (*Row x*)	~	~	~	ECB x
Data 0 (*Row y*)	Data 1 (*Row y*)	~	~	~	ECB y

Header Block

The header block is further subdivided into two areas:

1. A 32–bit (4 byte) device identification value. This is a value unique to a manufacturer and the product. For convenience, it has been decided that this should be the same as the JTAG identification word. This 4 byte word is loaded into the device and compared with the internally stored value. If these are not equal, an internal error is flagged:

   In BFR mode the configuration halts and the END signal remains permanently low and Err goes low.

   In Micro Mode, no accesses (read or write) to the array configuration data are permitted and the internal ID error flag of the status register is set.

If the values are equal, the configuration sequence may proceed.

2. A byte to specify what data 'type' is to follow. A table showing the data type allocation is shown below.

[7:3]	[2]	[1]	[0]	Data Type
00000	X	X	0	Sequential data (Normal data)
00000	X	X	1	Test data – Multiple row access
00000	X	0	X	Unencrypted data
	X	1	X	Encrypted data – Not supported on first product. **Reserved** for future implementations
	0	X	X	Uncompressed data
	1	X	X	Compressed data – Not supported on first product. **Reserved** for future implementations

Configuration Data Component

On the first Product only two data types are supported:

Normal Configuration

The configuration data is loaded a row at a time. An overview of the Configuration Map is given below:

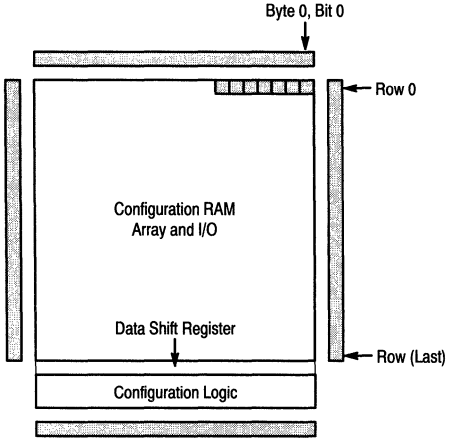


Figure 1-32. Configuration Map

This shows that the Upper right hand corner is Bit 0 of Byte 0 on Row 0. The number of Rows and Byte length for each Row varies between devices. The Data Sequence is as follows:

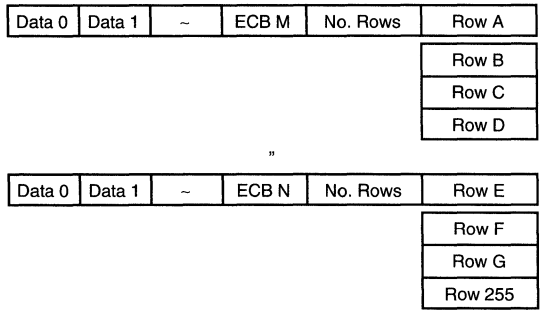
Configuration data for an entire row is loaded into the shift register at the foot of the array. During the loading of this data an Error Check is performed. The last byte to be written in a row is the Error Check Byte (ECB), this is compared with an internally generated value. If no error is detected, the entire row of data is transferred to the array. When this operation is completed, the internal row address counter is incremented and the next row loaded.

Test Data Configuration

This data type allows the size of the configuration data component to be greatly reduced and is used primarily for test. The majority of the test time for the device is spent loading test configurations onto the device rather than exercising the circuits when loaded.

Test configurations are highly regular with many rows being loaded with identical configuration data. By allowing the same data to be written to multiple rows the overall test time can be reduced.

The format is as follows:



(Row 255 = Configuration Terminating Byte)

The row configuration data is loaded as per the previous data type with a terminating ECB. Instead of the Data being transferred to the array once a correct ECB is loaded from the configuration vector, this byte indicates the number of rows to which the contents of the data register should be written. Each Row address byte is then loaded, separated by the internal transfer of data. Once the last Row address has been loaded and the transfer taken place, the internal state machine moves on to the next multiple row data.

In normal data configuration, the configuration sequence is terminated by an access to the last row. In the test mode the rows are not accessed sequentially, to indicate to the internal state machine that the configuration is terminated, an extra row address byte is added to the final multiple row configuration data. When a Row address value of 255 is loaded the configuration is terminated.

1

Pinout for MPA1036

Pad	Pad Type	Pin Location		
		84–Pin PLCC	160–Pin QFPs	181 PGA
1	5V Int Vdd			A2
2	Ext Vss			VSSE
3	3V Ext Vdd	12	40	A1
4	5V Ext Vdd			VDD
5	Ext Vss			VSSE
6	I/O L29, A16	13	39	B2
7	I/O L28		38	C2
8	I/O L27, A15	14	37	D4
9	I/O L26		36	B1
10	I/O L25, A14	15	35	C3
11	5V Ext Vdd			VDD
12	I/O L24, A13	16	34	D3
13	I/O L23		33	C1
14	I/O L22, A12	17	32	D2
15	I/O L21		31	D1
16	I/O L20, A11	18	30	E3
17	Ext Vss		29	VSSE
18	I/O L19		28	F3
19	I/O L18, A10	19	27	E1
20	I/O L17		26	E2
21	I/O L16		25	F1
22	I/O Clk, L15	20	24	G3
23	5V Int Vdd		23	G1
24	Int Vss	21	22	VSSI
25	I/O Clk, L14	22	21	G2
26	I/O L13		20	F2
27	I/O L12		19	H1
28	I/O L11, A9	23	18	H3
29	I/O L10		17	H2
30	5V Ext Vdd		16	VDD
31	I/O L9, A8	24	15	J1
32	I/O L8		14	J2
33	I/O L7, A7	25	13	K1
34	I/O L6		12	K2
35	I/O L5		11	L1
36	Ext Vss	26	10	VSSE
37	I/O L4		9	M1
38	I/O L3, A6	27	8	L2
39	I/O L2		7	N1
40	I/O L1, A5	28	6	J3
41	I/O L0		5	P1

Pad	Pad Type	Pin Location		
		84–Pin PLCC	160–Pin QFPs	181 PGA
42	F[4]	29	4	K3
43	Int Vss			VSSI
44	F[3]	30	3	M2
45	Ext Vss			VSSE
46	F[2]	31	2	L3
47	5V Ext Vdd			VDD
48	F[0]	32	1	M3
49	3V Ext Vdd			P2
50	Ext Vss			VSSE
51	Ext Vss			VSSE
52	Ext Vss			VSSE
53	5V Ext Vdd			VDD
54	/reset	33	160	R1
55	3V Ext Vdd			N2
56	F[1]	34	159	R2
57	I/O B0, A4	35	158	N3
58	I/O B1		157	R3
59	I/O B2, A3	36	156	N4
60	I/O B3		155	R4
61	I/O B4, A2	37	154	P3
62	Ext Vss		153	VSSE
63	I/O B5		152	N5
64	I/O B6, A1	38	151	R5
65	I/O B7		150	P4
66	I/O B8, A0	39	149	R6
67	I/O B9		148	N6
68	5V Ext Vdd	40	147	VDD
69	I/O B10		146	P5
70	I/O B11		145	R7
71	I/O B12, D7	41	144	N7
72	I/O B13		143	R8
73	I/O Clk, B14	42	142	N8
74	Int Vss		141	VSSI
75	5V Int Vdd	43	140	P6
76	I/O Clk, B15	44	139	P8
77	I/O B16		138	P7
78	I/O B17		137	R9
79	I/O B18, D6	45	136	P9
80	I/O B19		135	R10
81	Ext Vss	46	134	VSSE
82	I/O B20, D5	47	133	R11

Pinout for MPA1036 (continued)

Pad	Pad Type	Pin Location		
		84–Pin PLCC	160–Pin QFPs	181 PGA
83	I/O B21		132	N9
84	I/O B22, D4	48	131	R12
85	I/O B23		130	P10
86	I/O B24, D3	49	129	P11
87	5V Ext Vdd			VDD
88	I/O B25		128	R13
89	I/O B26, D2	50	127	N10
90	I/O B27		126	R14
91	I/O B28, D1	51	125	N11
92	I/O B29		124	P13
93	Mode[0]	52	123	P12
94	Ext Vss			VSSE
95	Mode[1]	53	122	N12
96	3V Ext Vdd		121	P14
97	5V Ext Vdd			VDD
98	Probe Pad	DO NOT BOND		
99	Ext Vss			VSSE
100	Ext Vss			VSSE
101	5V Ext Vdd			VDD
102	Mode[2]	54	120	M12
103	Quiet Vdd			R15
104	Mode[3]	55	119	N13
105	Vpp			P15
106	Clk	56	118	L13
107	3V Ext Vdd	57	117	N15
108	Ext Vss			VSSE
109	I/O R0, Dclk	58	116	L14
110	I/O R1		115	M13
111	I/O R2, D0	59	114	M15
112	I/O R3		113	N14
113	I/O R4, TDO	60	112	K14
114	Ext Vss		111	VSSE
115	I/O R5, TDI	61	110	L15
116	I/O R6		109	K13
117	I/O R7		108	K15
118	I/O R8, TMS	62	107	M14
119	I/O R9		106	J15
120	5V Ext Vdd		105	VDD
121	I/O R10		104	H14
122	I/O R11, TRSTB	63	103	J13
123	I/O R12		102	H15
124	I/O R13		101	J14

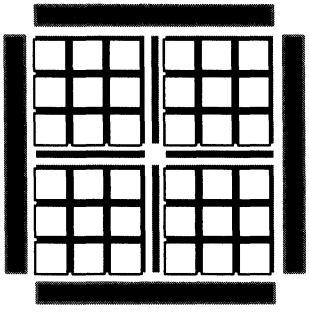
Pad	Pad Type	Pin Location		
		84–Pin PLCC	160–Pin QFPs	181 PGA
125	I/O Clk, R14	64	100	G14
126	Int Vss	65	99	VSSI
127	5V Int Vdd		98	G15
128	I/O Clk, R15	66	97	H13
129	I/O R16		96	F15
130	I/O R17	67	95	G13
131	I/O R18		94	E15
132	I/O R19	68	93	F14
133	Ext Vss		92	VSSE
134	I/O R20, TCK	69	91	F13
135	I/O R21		90	D15
136	I/O R22	70	89	E14
137	I/O R23		88	C15
138	I/O R24	71	87	E13
139	5V Ext Vdd			VDD
140	I/O R25		86	D13
141	I/O R26	72	85	D14
142	I/O R27		84	C13
143	I/O R28		83	B15
144	I/O R29	73	82	D12
145	Ext Vss			VSSE
146	3V Ext Vdd			C12
147	Ext Vss	74	81	VSSE
148	5V Ext Vdd			VDD
149	5V Int Vdd			C14
150	Int Vss			VSSI
151	Ext Vss			VSSE
152	5V Ext Vdd	75	80	VDD
153	3V Ext Vdd			A15
154	Ext Vss		79	VSSE
155	I/O T29	76	78	B14
156	I/O T28		77	C11
157	I/O T27	77	76	B13
158	I/O T26		75	B12
159	I/O T25	78	74	A14
160	5V Ext Vdd		73	VDD
161	I/O T24	79	72	A13
162	I/O T23		71	C10
163	I/O T22	80	70	A12
164	I/O T21		69	B11
165	I/O T20	81	68	A11
166	Ext Vss		67	VSSE

Pinout for MPA1036 (continued)

Pad	Pad Type	Pin Location		
		84–Pin PLCC	160–Pin QFPs	181 PGA
167	I/O T19	82	66	A10
168	I/O T18		65	B10
169	I/O T17	83	64	A9
170	I/O T16		63	C9
171	I/O Clk, T15	84	62	B8
172	5V Int Vdd	1	61	B9
173	Int Vss		60	VSSI
174	I/O Clk, T14	2	59	C8
175	I/O T13		58	A8
176	I/O T12	3	57	B7
177	I/O T11		56	A7
178	I/O T10	4	55	C7
179	5V Ext Vdd			VDD
180	I/O T9	5	54	B6
181	I/O T8		53	A6
182	I/O T7	6	52	C6
183	I/O T6		51	A5

Pad	Pad Type	Pin Location		
		84–Pin PLCC	160–Pin QFPs	181 PGA
184	I/O T5	7	50	B5
185	Ext Vss		49	VSSE
186	I/O T4	8	48	C5
187	I/O T3		47	A4
188	I/O T2	9	46	B4
189	I/O T1		45	A3
190	I/O T0, A17	10	44	C4
191	Ext Vss		43	VSSE
192	5V Ext Vdd		42	VDD
193	Ext Vss	11		VSSE
194	3V Ext Vdd		41	B3
195	Int Vss			VSSI
	NC			E5

181PGA NOTES:
VSSE: G12, E12, K12, D10, M10, G4, E4, K4, D6, M6
VSSI: E8, L8, H11, M11, H5, D5
VDD: D8, M8, H12, F12, J12, L12, D9, M9, D11, H4, F4, M4, J4, L4, D7, M7, M5



Field Programmable Gate Arrays

This section contains features and benefits, Electrical Specifications, Boundary Scan information, a description of the software support tools, a datasheet on the first of the serial support EPROMs, and case/package outline information.

CONTENTS

Device Data 2

MPA Features and Characteristics

2

The features of each FPGA member are tabulated to show logic cells, internal flip–flops, I/O cell flip–flops, equivalent gates, signal I/O pads and packages (see Table 2-2).

In addition, FPGA densities can only be meaningfully compared by taking account of the ability of the automatic place and route tool, and only then by using guiding preferences that are non–architecture–specific. Figure 2-1 shows the relative density obtained with the MPA1036 using only push button place and route with identical performance preferences.

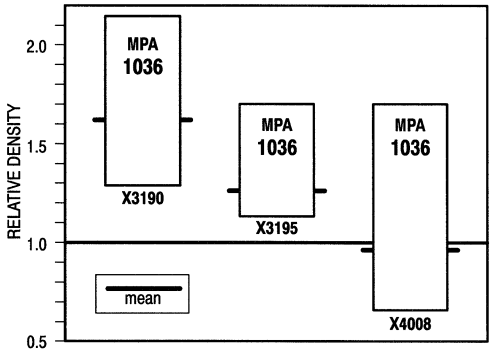


Figure 2-1. How Many Gates in an FPGA?

Motorola FPGA are designed to provide abundant programmable I/O cells with input and output registers. At any I/O state, it is programmable at 3/5V and programmable output drive at 6, 12mA, symmetrically. In addition, IEEE 1149.1 JTAG boundary scan is supported.

The speed characteristics of Motorola MPA1000 are shown in Table 2-1:

Table 2-1. MPA1000 — Performance With Flexibility

LOGIC BLOCK DELAY (2 INPUT AND)	Typical 1.0ns
LOGIC PLUS LOCAL INTERCONNECT	1.2ns
D FLIP–FLOP CLK TO Q	1.2ns
TOGGLE FREQUENCY	265MHz
MEDIUM BUS	1.2ns

Table 2-2. Family Members

Part No.	Logic Cells	Internal Flip-Flops	I/O Cell Flip-Flops	FPGA Equivalent Gates[1]	Signal I/O Pads	Packages[2]
MPA1016	1600	400	160	3500	80	84PC, 128PQ
MPA1036	3600	900	240	8000	120	84PC, 128PQ 160PQ, 160PQμCool, 181PG
MPA1064	6400	1600	320	14200	160	160PQ, 160PQμCool, 208PQ, 224PG
MPA1100	10000	2500	400	22000	200	160PQ, 160PQμCool, 208PQ, 289PG

1 X–4000 Series gate array equivalents
2 PC=PLCC Package; PQ=PQFP Package; PQμCool = PQFP μCool Package; PG=PGA Package

Electrical Specifications

ABSOLUTE MAXIMUM RATINGS*

Symbol	Parameter	Value	Unit
V_{DD}	DC Supply Voltage	–0.5 to +6.5	V
V_{IN}	DC Input Voltage	–0.5 to V_{DD} + 0.5	V
V_{OUT}	DC Output Voltage	–0.5 to V_{DD} + 0.5	V
I	DC Current Drain per Pin, Any Single Input or Output	50	mA
T_A	Commercial Operating Temperature Range (In Free Air)	0 to +70	°C
T_{STG}	Storage Temperature Range	–65 to +150	°C

* Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation should be restricted to the Recommended Operating Conditions.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Unit
V_{DD}	DC Supply Voltage	4.75	5.25	V
V_{DDO}	Output Voltage Levels	3/4.75	3.6/5.25	
V_{IHT}	High Level Input Voltage, TTL	2.0	V_{DD}	V
V_{ILT}	Low Level Input Voltage, TTL	0	0.8	V
V_{IHC}	High Level Input Voltage, CMOS	70	100	%V_{DDO}
V_{ILC}	Low Level Input Voltage, CMOS	0	20	%V_{DDO}

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation, it is recommended that V_{IN} and V_{OUT} be constrained to the range $V_{SS} \le (V_{IN}$ or $V_{OUT}) \le V_{DD}$. Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}.)

DC ELECTRICAL CHARACTERISTICS

Symbol	Characteristic	Min	Max	Unit	Condition
V_{OH}	Output Voltage Level High	3.7	V_{DD}	V	I_{OH}=–6 or 12mA; $V_{DD} \ge 4.75V$
V_{OL}	Output Voltage Level Low		0.4	V	I_{OL} = +6 or 12mA
I_Q	Quiescent Supply Current TTL (No Active Wired–OR Buses) CMOS		TBD TBD	mA	
I_L	Leakage Current	–10	10	µA	
I_{pu}	Pad Pull–Up (When Selected) at V_{IN} = 0V		–75	µA	
I_{pd}	Pad Pull–Down		+110	µA	
C_{in}	Input Capacitance		10	pF	

Input: CMOS or TTL/3, 3.3 volt CMOS levels, programmable pull–ups/pull–downs.
Output: Symmetrical end programmable drive either 6mA or 12mA, open drain is programmable, slew rate control.

MPA1036 CELL AC CHARACTERISTICS

Symbol	Characteristic	Min	Typ	Max	Unit
T_{cn}	Core Cell Delay – AND		1.0	1.75	ns
T_{cx}	Core Cell Delay – XOR		1.5	2.65	ns
T_{ioi}	I/O Cell Delay (Pad to Medium/Global Bus)		1.6	2.80	ns
T_{ioo}	I/O Cell Delay (Medium/Global Bus to Pad)			10	ns
T_{fc}	Fast Carry Path Delay (Per Tile)			2.0	ns
T_{dsu}	DFF Data Setup		1.5	2.65	ns
T_{dho}	DFF Data Hold		0		ns
T_{esu}	DFF Enable Setup		1.0		ns
T_{eh}	DFF Enable Hold		0		ns
T_{clkh}	DFF Clock High Time		2.0		ns
T_{clkl}	DFF Clock Low Time		2.0		ns
$T_{clk\ to\ Q}$	Clock to Q Delay		1.2	2.1	ns
T_{rec}	DFF Reset Recovery Time		1.0		ns
T_{mux}	2:1 Multiplexer Delay		2.5	4.5	ns
T_{pdlu}	Pad Input Delay		TBD		ns
T_{ppd}	Pad–to–Pad Delay Direct		TBD		ns
T_{ppdr}	Pad–to–Pad Delay Resistor Out		TBD		ns

MPA1036 INTERCONNECT CHARACTERISTICS

Symbol	Characteristic	Min	Typ	Max	Unit
T_{ccl}	Cell–to–Cell (Local Interconnect)	0.6		1.1	ns
T_{ccm}	Cell–to–Cell (Same Zone, Medium Bus)	1.2		2.1	ns
T_{ccg}	Cell–to–Cell (Same Quadrent, Global Bus)	4.0		7.0	ns
T_{ccgx}	Cell–to–Cell (Same Quadrent, Global + 'X')	8.0		14	ns
T_{ccq}	Cell–to–Cell (Adjacent Quadrents, Global Bus)	5.0		8.75	ns
T_{ccqx}	Cell–to–Cell (Adjacent Quadrents, Global + 'X')	10.5		18	ns
T_{piod}	Adjacent I/O–to–I/O Cell Delay Through Peripheral Bus			TBD	ns
T_{pbad}	Peripheral Bus Delay per Section			TBD	ns

MPA1036 TRISTATE NET CHARACTERISTICS

Symbol	Characteristic	Min	Typ	Max	Unit
T_{cwqr}	Cell–to–Cell via Inter–Quadrent Global Wired–OR Bus Rising			6.0	ns
T_{cwqf}	Cell–to–Cell via Inter–Quadrent Global Wired–OR Bus Falling			11	ns
T_{pwdr}	I/O–to–I/O Cell via Peripheral Bus Wired–OR Rising			TBD	ns
T_{pwdf}	I/O–to–I/O Cell via Peripheral Bus Wired–OR Falling			TBD	ns

MPA1036 PRIMARY CLOCK CHARACTERISTICS

Symbol	Characteristic	Min	Typ	Max	Unit
T_{pclk}	Primary Clocks — Pad–to–Core Cell Delay			5.0	ns
T_{skw}	Primary Clocks — Skew			1.0	ns

MPA1036 JTAG CHARACTERISTICS

Symbol	Characteristic	Min	Typ	Max	Unit
F_{cjtag}	Shift Clock Frequency			16	MHz

MPA1036 CONFIGURATION LOGIC CHARACTERISTICS

Symbol	Characteristic	Min	Typ	Max	Unit
MICRO MODE EXTERNAL TIMINGS					
Write Cycle					
T_{csw}	$\overline{CS}$ Setup Before Write	10			ns
T_{rsw}	RS Setup Before Write	10			ns
T_{rshw}	RS Hold After Write Falling Edge	10			ns
T_{pww}	Write Pulse Width	50			ns
T_{dsw}	Data Setup to End of Write	20			ns
T_{dhw}	Data Hold After Write	10			ns
T_{busw}	Busy Inactive Before End of Write	50			ns
T_{busaw}	Busy Active After Write	0		20	ns
Read Cycle					
T_{csr}	$\overline{CS}$ Setup Before Read	10			ns
T_{rsr}	RS Setup Before Read	10			ns
T_{rshr}	RS Hold After Read Falling Edge	10			ns
T_{rpw}	Read Pulse Width	50			ns
T_{dacc}	Data Access Time	20		40	ns
T_{dhr}	Data Hols Time After Read	0		10	ns
T_{busr}	Busy Inactive Before End of Read	50			ns
BFR MODE					
	$\overline{BFR}$ Low to Internal	2		3	tclk
	$\overline{BFR}$ Low to $\overline{MEMCE}$ High			3	tclk
	END High to $\overline{MEMCE}$ High	1			tclk
	PWRUP to $\overline{MEMCE}$ High	1			tclk
	$\overline{BFR}$ Low to END Low	0		3	tclk
	$\overline{BFR}$ Low to Internal Disable	0		3	tclk
	$\overline{BFR}$ Pulse Width	50			tclk
	Reset Sequence Duration				tclk

MPA1036 CONFIGURATION LOGIC CHARACTERISTICS (continued)

Symbol	Characteristic	Min	Typ	Max	Unit
BFR MODE (continued)					
	Configuration Sequence Duration				tclk
	CSE to END High	1		1	tclk
	END to Enable User Inputs	1			tclk
	END to Enable User Outputs	2			tclk
	END to Enable Use Bootstrap	1			tclk
	PWRUP to Enable User Inputs	1			tclk
	PWRUP to Enable User Outputs	2			tclk
	PWRUP to Enable User Bootstrap	1			tclk
BFR DATA ACCESS					
	Data Setup to CLK	20			clk
	Data Hold to CLK	0			clk
	DCLK Period When Active	2		2	clk
	CLK to Address Valid	15			ns

MPA1036 CLOCK SPECIFICATION

Symbol	Characteristic	Min	Typ	Max	Unit
INTERNAL					
	Ring Oscillator Low	10	25	50	ns
	Ring OscillatorHigh	10	25	50	ns
	Ring Oscillator Period	25	50	100	ns
	Internal Clock Period	200	400	800	ns
	Internal Ring Oscillator Clock Frequency Range	10		40	MHz
EXTERNAL					
	External Clock Low	10			ns
	External Clock High	10			ns
	External Clock Period	50			ns

MPA1036 EXTERNAL $\overline{\text{RESET}}$ SPECIFICATION

Symbol	Characteristic	Min	Typ	Max	Unit
	$\overline{\text{RESET}}$ Low to Reset Sequence	2		3	clk
	$\overline{\text{RESET}}$ Low to END Low	0		3	clk
	$\overline{\text{RESET}}$ Low to ERR High	0		3	clk
	$\overline{\text{RESET}}$ Pulse Width	50			ns
	$\overline{\text{RESET}}$ Low to Internal Disable	0		3	
	$\overline{\text{RESET}}$ Low to Busy Active	0		3	
	$\overline{\text{RESET}}$ High to to CSI Pulse	2			
	$\overline{\text{RESET}}$ High to Busy Inactive	2			

JTAG Boundary Scan

JTAG Boundary Scan Functions

JTAG is a standardized boundary scan methodology used for board level testing to detect faults in package and board connections, as well as internal circuitry. The JTAG boundary scan cell in Motorola's FPGAs is designed to meet the IEEE std. 1149.1 for testability test of an integrated circuit.

IEEE 1149.1 Architecture

Figure 2-2 shows the general diagram of an JTAG systems of the IEEE 1149.1 in Motorola's FPGA. It's design is compatible to Motorola ASICs H4C and H4C+ family of arrays. Motorola's FPGA JTAG design is hard wired, and it virtually supports all basic instructions as the Motorola HDC and H4C gate arrays.

TAP and I/O Periphery Signals

The TAP (Test Access Port) consists of five externally accessible signals which are used to control and observe boundary scan data. These five pins, namely, TCK, TMS, TDI, TRSTB, and TDO are multiplexed with normal signal pins. After testability test, these pins can be programmed as normal I/O pins and the JTAG systems will be shut off automatically. The test clock pin, TCK, is used to synchronize all JTAG functions. The TCK, TMS and TRSTB control the TAP controller. TDI is the test data input pin and TDO is the test data output pin.

JTAG Control and Test Register

The **TAP Controller** is a synchronous, 16–state finite machine, which selects the mode of operation for the test circuitry. An example of the operation of the TAP controller is shown in Figure 2-3 where the TAP controller is sequenced through most of its test states.

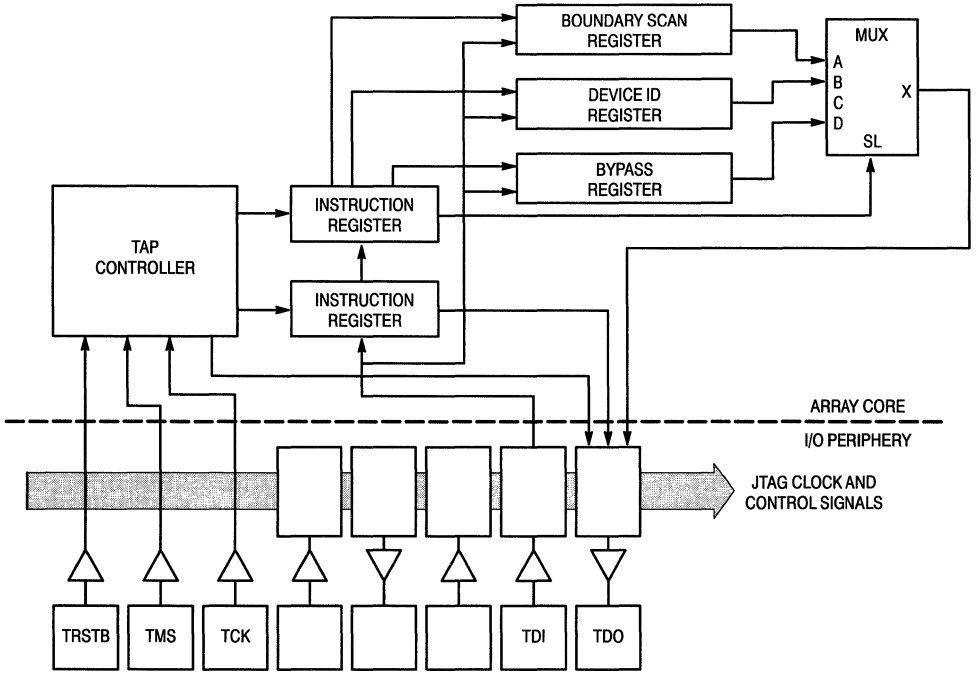
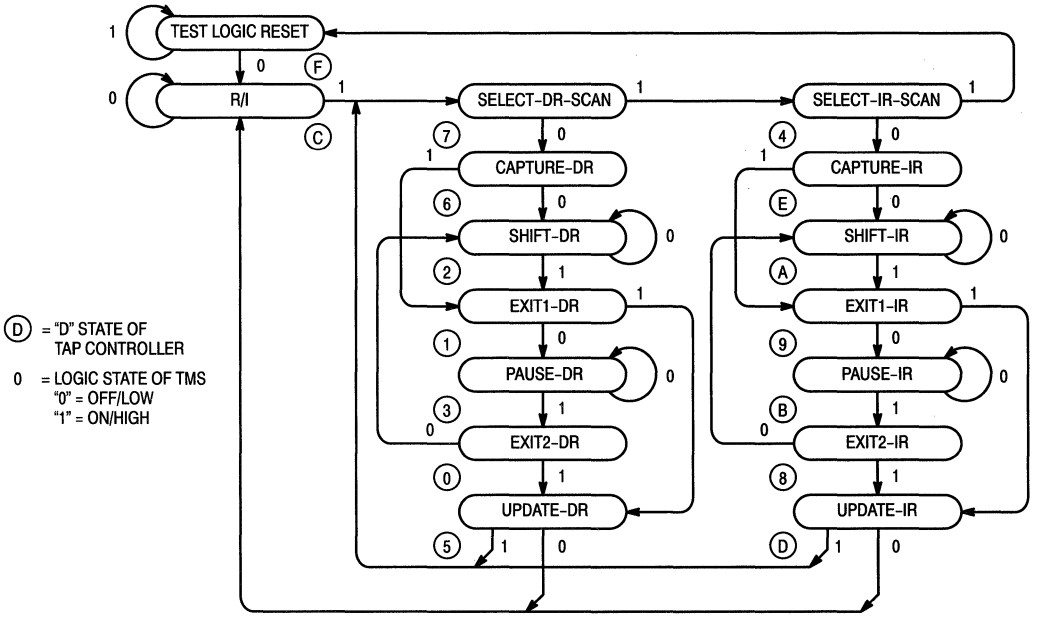


Figure 2-2. JTAG Systems



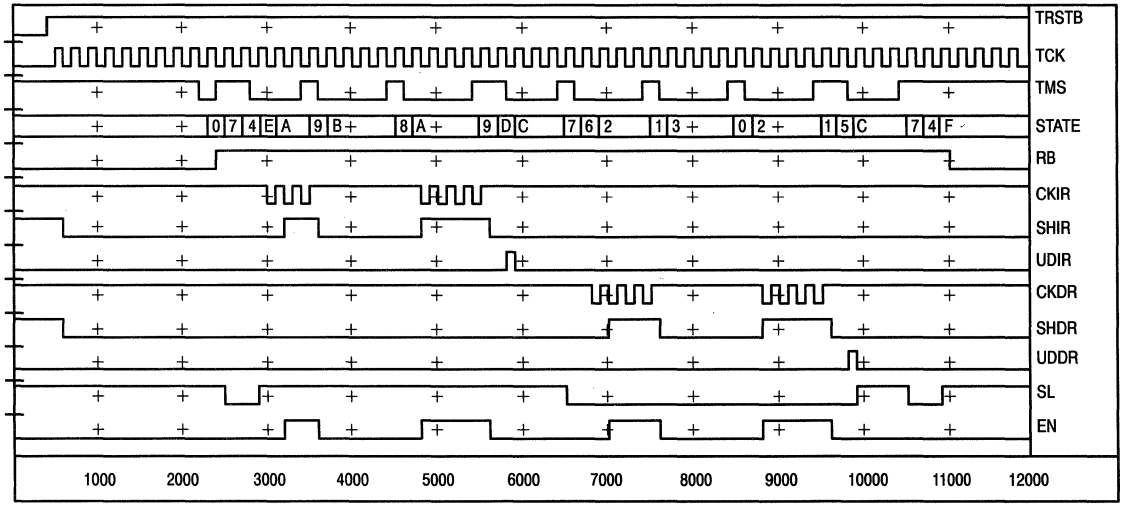


Figure 2-3. TAP Controller and Test Cycle

The **Instruction Register** is a 3–bit shift register, which permits an instruction to be shifted into the design to select the test to be performed. The **Instruction Decode** translates the instruction into separate control signals. Table 2-3 shows the basic public instructions supported by Motorola's FPGA:

Table 2-3. Basic Public Instruction

I_2	I_1	I_0	Public Instruction	Register Selected
0	0	0	EXTEST	Boundary Scan Cell
0	0	1	INTEST	Boundary Scan Cell
0	1	0	SAMPLE	Boundary Scan Cell
1	0	0	IDCODE	Device Register
1	1	1	BYPASS	Bypass Register

- **EXTEST** (external test) is the boundary scan test that checks the board interconnections between integrated circuits(I.C.s).
- **INTEST** (internal test) checks the logic internal to I.C.s.
- **SAMPLE** test samples data at the I/O pins of an I.C. during normal operating mode.
- **IDCODE** instruction outputs the identification code of the I.C.
- **BYPASS** instruction redirects the test data from TDI directly to TDO, effectively removing the I.C. from the boundary scan chain.

The **Bypass Register** is a single–bit shift register used to provide a shortest path between TDI and TDO.

The **Device Identification Register** is a 32–bit register which holds a manufacturer's identity code, part number and version code. The bit assignment for the ID code is given in Table 2-4.

Table 2-4. Device Register ID Codes

Bit Number	Code Use
0–11	Motorola Identification
12–21	Array Identification
22–27	Programmable Logic Products Identification
28–31	Version Number

For example, for MPA1036 & MPA1064, the ID codes are listed as follows:

Array	ID code
MPA1016	0001 001110 0100001110 000000011101
MPA1036	0001 001110 0100011110 000000011101
MPA1064	0001 001110 0100110100 000000011101
MPA1100	0001 001110 0100100000 000000011101

The **Boundary Scan Register** is the chain of JTAG boundary scan cells that are linked together to form a shift register around the periphery of the array. The test data enters the boundary scan register through the TDI pin, the rising edge of CKDR when SHDR is asserted, then is shifted around the array through each I/O cell in a counter clockwise direction, and finally exits through the TDO pin. Since each I/O pin is designed as a bidirectional pin, a 2–bit shift register resides in each I/O cell, one for monitor either the input or output, and the other to monitor the enable pin of the 3–state output buffer. For every two clock cycles, the data shifts from one I/O site to the other. The boundary scan cell resides in every I/O site with the exception of TDI, TCK, TMS, TRSTB and TDO pins.

Support Tools

2

Motorola is the first FPGA vendor to support a total third party vendor solution to software support. Some of the early FPGA products were supported by proprietary tools out of necessity, but since these tools are a major source of revenue for these vendors they are reluctant to give them up. An FPGA vendor that is able to convince the designer to purchase the vendor specific tools has accomplished a virtual "lock" into the design, since tools are expensive and take an enormous amount of time and energy to learn.

By selecting the best and most universal of tools in all categories, Motorola frees the designer and encourages the use of tools that have wide appeal allowing the development of Motorola MPA1000 FPGAs and competitive parts. Motorola and its franchised distributors will sell a Motorola version of the Viewlogic and NeoCAD tools. For information about the logic synthesis tools the designer is encouraged to contact the individual software suppliers or their agents.

Schematic Capture and Simulation

Viewlogic's Viewdraw and Viewsim are recommended for a PC environment, and Viewlogic and Mentor Graphics are both recommended for a Unix workstation. Both tools are supported on Sun and H.P. workstations.

Logic Synthesis

Exemplar's Core and Synopsys are the two recommended packages for logic optimization and synthesis. Both packages allow the use of either VHDL or Verilog–HDL for representation. Synopsys is available for the Unix workstation and Exemplar is available on both the workstation and PC platforms.

Place and Route

NeoCAD is the Motorola recommended and supported tool. NeoCAD takes the design file in from various formats and does mapping and place and route. The place and route is timing driven, allowing the user to specify net delays in MHz or nanosecs, not merely some "guess" as to what might be important. NeoCAD is sold by Motorola, its franchised distributors, Mentor Graphics and Value Added reseller of NeoCAD. Please contact NeoCAD for a list of these VARS.

For further information Please Contact:

NeoCAD Inc.
2585 Central Avenue
Boulder, Colorado
Tel: (303) 442–9121
1–800–862–3143

Viewlogic Systems
Marlboro, Massachusetts
Tel: 508 480–0881 or
1–800–873–8439

Synopsys Inc.
700 East Middlefield Road
Mountain View, California 94043
Tel: (415) 962–5000

Exemplar Logic Inc.
2550 Ninth Street, Suite 102
Berkeley, California 94710
Tel: (510) 849–0937
Fax: (510) 849–9935

Product Preview

128K Serial EPROM

MCP17128

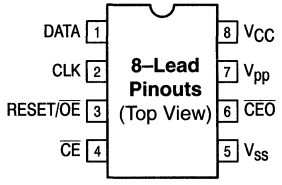
128K SERIAL EPROM

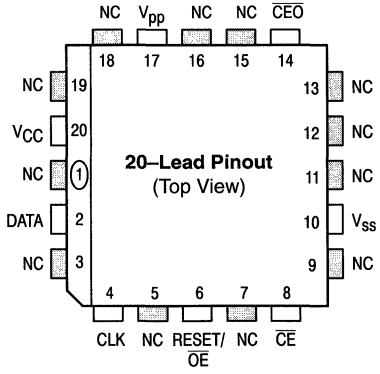
2

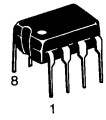
The MCP17128 serial OTP EPROM is the companion serial EPROM to the MPA1036 Field Programmable Gate Array. When used with the MPA1036, it provides a compact, low pin count, non–volatile configuration code store, with the FPGA automatically configuring on power–up or BFR (Boot From ROM) request.

The MCP17128 can also be cascaded for increased memory storage when needed. It is available in the standard 8–pin plastic DIP (N suffix), 8–pin SOIC (D suffix) and 20–pin PLCC (FN suffix) packages.

- Configuration EPROM for MPA1036 Field Programmable Gate Array
- Voltage Range — 4.5 to 6.0V
- Maximum Read Current of 10mA at 5.0V
- Standby Current of 10μA, Typical
- Industry Standard Synchronous Serial Interface/1 Bit per Rising Edge of Clock
- Full Static Operation
- Sequential Read/Program
- Cascadable Output Enable
- 10MHz Maximum Clock Rate at 5.0Vdc
- Programmable Polarity on Hardware Reset
- Programming With Industry Standard EPROM Programmers
- Electrostatic Discharge Protection > 2000 Volts
- 8–Pin PDIP and SOIC; 20–Pin PLCC Packages
- Commercial (0 to +70°C) and Industrial (–40 to +85°C) Temperature Ranges

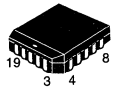






N SUFFIX
PLASTIC PACKAGE
CASE 626–05

D SUFFIX
PLASTIC SOIC PACKAGE
CASE 751–05



FN SUFFIX
PLCC PACKAGE
CASE 775–02

PIN NAMES

Pins	Function
DATA	Data I/O
CLK	Clock
RESET/$\overline{OE}$	Reset Input and Output Enable
$\overline{CE}$	Chip Enable Input
V_{SS}	Ground
$\overline{CEO}$	Chip Enable Output
V_{PP}	Programming Voltage Supply
V_{CC}	+4.5 to 6.0V Power Supply
NC	Not Connected

This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

MOTOROLA

MAXIMUM RATINGS*

Symbol	Parameter	Value	Unit
	V_{CC} and Input Voltages W.R.T. V_{SS}	–6.0 to V_{DD} + 0.6	V
	V_{PP} Voltage W.R.T. V_{SS} During Programming	–0.6 to +14.0	V
	Output Voltage W.R.T. V_{SS}	–0.6 to V_{CC} + 0.6	V
	Storage Temperature Range	–65 to +150	°C
	Ambient Temperature With Power Applied	–65 to +125	°C
	Soldering Temperature of Leads (10 Seconds)	+300	°C
	ESD Protection on All Leads	≥2	kV

2

NOTE: Maximum Ratings are those values beyond which damage to the device may occur. Functional operation should be restricted to the Recommended Operating Conditions.

DC CHARACTERISTICS (V_{CC} = 4.5 to 6.0V; Commercial (C) T_A = 0 to +70°C; Industrial (I) T_A = –40 to +85°C)

Symbol	Characteristic		Min	Max	Unit	Condition
V_{IH}	Input Voltage High	DATA, $\overline{CE}$, $\overline{CEO}$, Reset	2.0	V_{CC}	V	
V_{IL}	Input Voltage Low	DATA, $\overline{CE}$, $\overline{CEO}$, Reset	–0.3	0.8	V	
V_{OH}	Output Voltage High	DATA, $\overline{CE}$, $\overline{CEO}$, Reset	3.86 2.40		V	I_{OH} = –4mA; V_{CC} ≥ 4.5V
V_{OL}	Output Voltage Low	DATA, $\overline{CE}$, $\overline{CEO}$, Reset		0.32	V	I_{OL} = 4.0mA
I_{LI}	Input Leakage Current		–10	10	µA	V_{IN} = 0.1V to V_{CC}
I_{LO}	Output Leakage Current		–10	10	µA	V_{OUT} = 0.1V to V_{CC}
C_{INT}	Internal Capacitance	(All Inputs/Outputs)		10	pF	V_{CC} = 5.0V (Note 1); T_A = 25°C; f_{clk} = 1MHz
I_{CC} Read	Operating Current			10	mA	V_{CC} = 6.0V; CLK = 10MHz
I_{CCS}	Standby Current			500	µA	V_{CC} = 6.0V

1. This parameter is initially characterized and not 100% tested.

Applications Information

DATA

Three–state DATA output for reading and function as the input during programming.

CLOCK

Clock input. Used to increment the internal address and bit counters for reading and programing.

RESET/$\overline{OE}$

Reset and Output Enable input. A Low level both the $\overline{CE}$ and RESET/$\overline{OE}$ inputs enables the data output driver. A High level on RESET/$\overline{OE}$ resets both the address and bit counters. In the MCP17128, the logic polarity of this input is programmable as either RESET/$\overline{OE}$ or $\overline{OE}$/RESET. This document describes the pin as RESET/$\overline{OE}$ although the opposite polarity is also possible, this option is defined and set at device program time.

$\overline{CE}$

Chip Enable input. Used for device selection. A Low level on both $\overline{CE}$ and $\overline{OE}$ enables the data output driver. A High level on $\overline{CE}$ disables both the address and bit counters and forces the device into a low power mode.

$\overline{CEO}$

Chip Enable Out output. This signal is asserted Low on the clock cycle following the last bit read from the memory. It will stay Low as long as $\overline{CE}$ and $\overline{OE}$ are both Low. It will then follow $\overline{CE}$ until $\overline{OE}$ goes High. Thereafter $\overline{CEO}$ will stay High until the entire PROM is read again. This pin also used to sense the status of RESET polarity when program mode is entered.

V_{PP}

Programming Voltage Supply. Used to enter programming mode (+10V) and to program the memory (+13V) Must be connected directly to V_{CC} for normal Read operation. No overshoot above +15.5V permitted.

CONTROLLING THE MCP17128 SERIAL PROMS

The connections between the FPGA device and the Serial PROM are as follows:

- The DATA output of the MCP17128 drives DO (data in) of the FPGA devices.
- The CLK input of the MCP17128 is driven by the master FPGA DCLK (configuration clock) output.
- The MCP17128 can be cascaded by using the $\overline{CEO}$ output to drive the $\overline{CE}$ input of the next MCP17128.
- For normal Read operations V_{PP} must be connected to V_{CC}. Do not leave V_{PP} open.

There are two different ways to use the inputs $\overline{CE}$ and $\overline{OE}$:

1. The simplest connection is to have the FPGA $\overline{MEMCE}$ or LDC output drive both $\overline{CE}$ and $\overline{OE}$ in parallel, but it fails when a user applies RESET to the FPGA during the FPGA configuration process. This method must never be used when there is any chance of external reset during configuration. The FPGA will abort the configuration and then restart a new configuration, but the MCP17128 does not reset its address counter, since there was never a High level on the $\overline{OE}$ input. The new configuration reads the remaining data in the PROM and sees it as preamble, length count, etc. Since the FPGA device is the master, it issues the necessary number of CLOCK pulses, up to 16 million (2^{24}) and $\overline{MEMCE}$ goes High. The FPGA configuration will then be wrong, with potential contentions inside the FPGA device and on its output pins.
2. The recommended connection is to have the FPGA $\overline{MEMCE}$ or LDC output drive only the $\overline{CE}$ input of the MCP17128, while the $\overline{OE}$ input is driven by the inverse of the FPGA RESET input. This works under all normal circumstances, even when the configuration is reset before $\overline{MEMCE}$ has gone High. The High level on the $\overline{OE}$ input during RESET clears the PROM internal address pointer, and the reconfiguration starts at the beginning. The polarity of the RESET pin must be programmed to the correct edge.

FPGA MASTER SERIAL MODE SUMMARY

The I/O, logic functions, and associated interconnections of the application FPGA are established by a configuration program. The program is loaded either automatically upon power up, or on command, depending on the individual configuration of the FPGA. In a Master Serial Mode, the FPGA automatically loads the configuration program from an external memory on power up. The MCP17128 family Serial Configuration PROM has been designed to be compatible with the Master Serial Mode.

Data is read from the Serial Configuration SCPROM sequentially on a single data line. Synchronization is provided by the rising edge of the temporary signal DCLK, which is generated during configuration.

Master Serial Mode provides a simple configuration interface as only a serial data line and two control lines are required to configure an FPGA. Data from the Serial Configuration SCPROM is read sequentially, accessed via the internal address and bit counters which are incremented on every valid rising edge of CLK driven by the FPGA DCLK.

Programming the FPGA with Counters Reset Upon Completion

The connections between an FPGA device and its SCPROM are shown in Figure 1. The DATA line from the SCPROM is connected to the CLK input of the SCPROM. At power–up or upon reconfiguration, the $\overline{MEMCE}$ signal goes Low (pulled low by the FPGA device at reset, or by external circuitry for reconfiguration), enabling the SCPROM and its DATA output. During the configuration process, DO reads data from the SCPROM on every rising clock edge. The $\overline{MEMCE}$ signal goes High at the end of configuration and resets the internal address counters of the SCPROM.

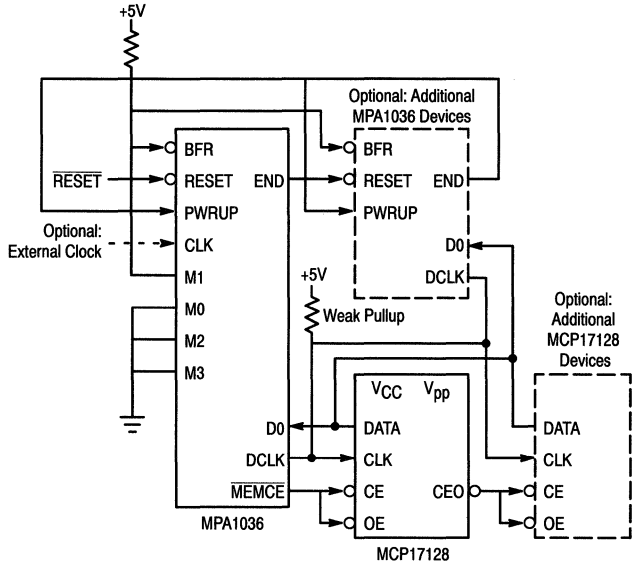


Figure 1. MPA1036 Configuration Using MCP17128 Serial EPROM

Programming the FPGA with Counters Unchanged Upon Completion

The $\overline{OE}$ pin should be tied Low when multiple FPGA configurations for a single FPGA are stored in a Serial Configuration PROM, as shown in Figure 2. The internal address counters are reset upon power–up and configuration begins with the first program stored in memory. Since the $\overline{OE}$ in is held Low, the address counters remain unchanged after configuration is complete. Then to reprogram the FPGA with another program, the $\overline{MEMCE}$ line is pulled Low and configuration begins at the last value of the address counters.

Cascading Serial Configuration PROMs

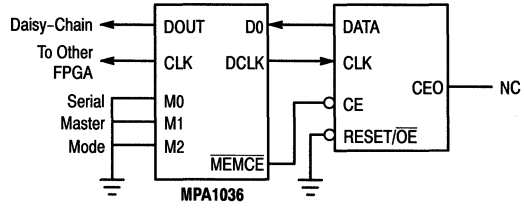
Cascading SCPROMs provide additional memory for multiple FPGAs configured as a daisy–chain, or for future FPGAs requiring larger configuration memories.

When the last bit from the first SCPROM is read, the next clock signal to the SCPROM asserts its $\overline{CEO}$ output Low and disables its DATA line. The second SCPROM recognizes the Low level on its $\overline{CE}$ input and enables its DATA output. (See Figure 1).

When configuration is complete, the address counters of all cascaded SCPROMs are reset if RESET goes Low forcing the RESET/$\overline{OE}$ on each SCPROM to go High.

If the address counters are not to be reset upon completion, then the RESET/$\overline{OE}$ inputs can be tied to ground, as shown in Figure 2. To reprogram the FPGA device with another program, the $\overline{MEMCE}$ line goes Low and configuration begins where the address counters had stopped. In this case, avoid contention between DATA and the configured I/O use of DO if DO was configured as an output.

Additional logic may be required if cascaded memories are so large that the rippled chip enable is not fast enough to activate successive SCPROMs.



Notes:
1. M2 should be programmed as an input during operation if it is tied to Ground.
2. If the FPGA is reset during configuration, it will abort back to initialization state. $\overline{MEMCE}$ will not go High, so an external signal is required to reset the MCP17128 counters.

Figure 2. Address Counters Not Reset

STANDBY MODE

The MCP17128 enters a low power standby mode whenever $\overline{CE}$ is High. In standby mode, the SCPROM consumes less than 500µA of current. The output will remain in a high impedance state regardless of the state of the $\overline{OE}$ input.

PROGRAMMING MODE

Programming mode is entered by holding V_{PP} High for at least two clock edges and is exited by removing power from the device or by a Low on both $\overline{CE}$ and $\overline{OE}$. Figure 5 through Figure 10 shows the programming algorithm.

MCP17128 RESET POLARITY

The MCP17128 lets the user choose the reset polarity as either RESET/$\overline{OE}$ or $\overline{OE}$/RESET. Any third–party commercial programmer should prompt the user for the desired reset polarity.

The programming of the overflow word should be handled transparently by the PROM programmer; it is mentioned here as supplemental information only.

The polarity is programmed into the first overflow word location, max address+1. 00000000 in these locations makes the reset active Low, FFFFFFFF in these locations makes the reset active High. The default condition is RESET active High.

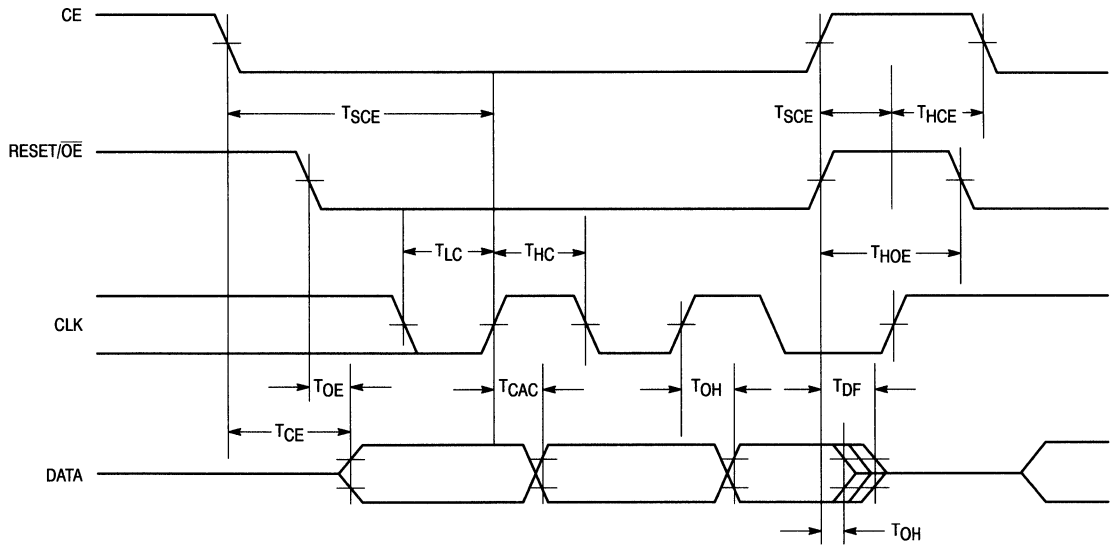


Figure 3. AC Characteristics Over Operating Conditions

AC OPERATING CONDITIONS

Symbol	Parameter	Limit 4.5V ≤ V_{CC} ≤ 6.0V		Unit	Condition
		Min	Max		
T_{OE}	$\overline{OE}$ to Data Delay		45	ns	
T_{CE}	$\overline{CE}$ to Data Delay		50	ns	
T_{CAC}	CLK to Data Delay		60	ns	
T_{OH}	Data Hold From $\overline{OE}$, $\overline{CE}$ or CLK	0		ns	
T_{DF}	$\overline{OE}$ or $\overline{CE}$ to Data Float Delay		50	ns	Note 1
T_{LC}	CLK Low Time	25		ns	Note 2
T_{HC}	CLK High Time	25		ns	Note 2
T_{SCE}	$\overline{CE}$ Setup Time to CLK (To Guarantee Proper Counting)	25		ns	
T_{HCE}	$\overline{CE}$ Hold Time to CLK (To Guarantee Proper Counting)	0		ns	Note 2
T_{HOE}	$\overline{OE}$ High Time (Guarantees Counters are Reset)	20		ns	Note 2
CLK_{max}	Clock Frequency		10	MHz	

1. Float delays are measured with minimum tester AC load and maximum DC load.
2. Guarantee by design, not tested.

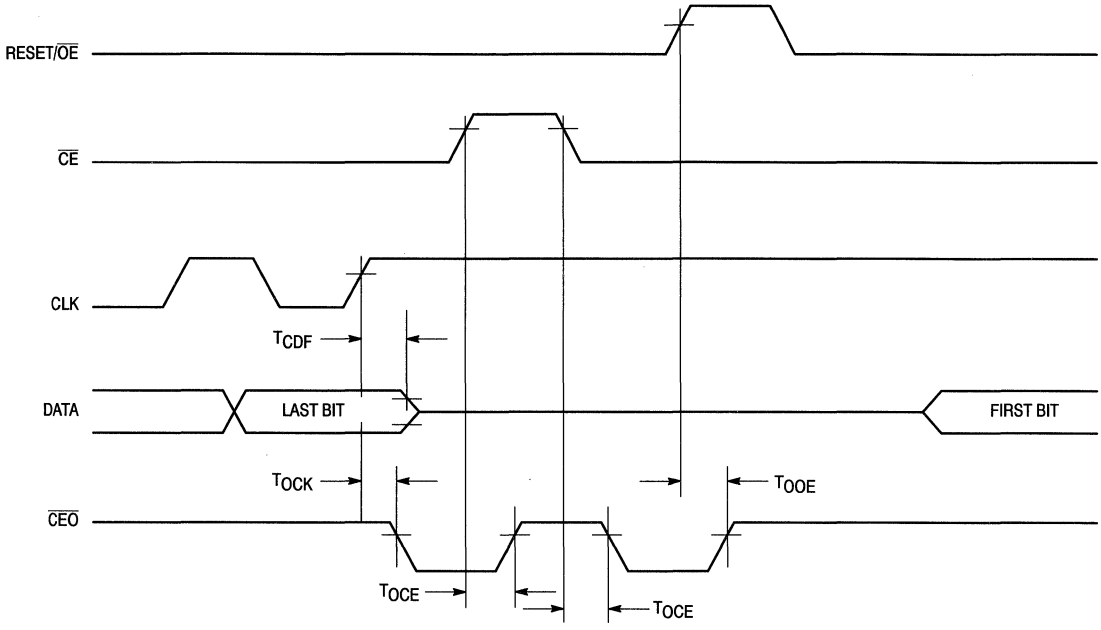


Figure 4.

Symbol	Parameter	Limit 4.5V ≤ V_{CC} ≤ 6.0V		Unit	Condition
		Min	Max		
T_{CDF}	CLK to Data Float Delay		50	ns	
T_{OCK}	CLK to $\overline{CEO}$ Delay		40	ns	
T_{OCE}	$\overline{CE}$ to $\overline{CEO}$ Delay		40	ns	
T_{OOE}	RESET/$\overline{OE}$ to $\overline{CEO}$ Delay		40	ns	

PIN ASSIGNMENTS IN THE PROGRAMMING MODE

Pin Name	DIP	PLCC	I/O	Function
DATA	1	2	I/O	The rising edge of the clock shifts a data word in or out of the PROM one bit at a time.
CLK	2	4	I	Clock input. Used to increment the internal address/word counter for reading and programming operation.
RESET/$\overline{OE}$	3	6	I	The rising edge of CLK shifts a data word into the PROM when $\overline{CE}$ and $\overline{OE}$ are High; it shifts a data word out of the PROM when $\overline{CE}$ is Low and $\overline{OE}$ is High. The address/word counter is incremented on the rising edge of CLK while $\overline{CE}$ is held High and $\overline{OE}$ is held Low. Note: Any modified polarity of the RESET/$\overline{OE}$ pin is ignored in the programming mode.
$\overline{CE}$	4	8	I	The rising edge of CLK shifts a data word into the PROM when $\overline{CE}$ and $\overline{OE}$ are High; it shifts a data word out of the PROM when $\overline{CE}$ is Low and $\overline{OE}$ is High. The address/word counter is incremented on the rising edge of CLK while $\overline{CE}$ is held High and $\overline{OE}$ is held Low.
GND	5	10	—	Ground pin.
$\overline{CEO}$	6	14	O	The polarity of the RESET/$\overline{OE}$ pin can be read by sensing the $\overline{CEO}$ pin. Note: The polarity of the RESET/$\overline{OE}$ pin is ignored while in the programming mode. In final verification, this pin must be monitored to go Low one clock cycle after the last data bit has been read.
V_{PP}	7	17	—	Programming Voltage Supply. Programming mode is entered by holding $\overline{CE}$ and $\overline{OE}$ High and V_{PP} at V_{PP1} for two rising clock edges and then lowering V_{PP} to V_{PP2} for one more rising clock edge. A word is programmed by strobing the device with V_{PP} for the duration TPGM V_{PP} must be tied to V_{CC} for normal operation.
V_{CC}	8	20	—	+5 V power supply input.

DC PROGRAMMING SPECIFICATIONS

Symbol	Parameter	Limit		Unit	Condition
		Min	Max		
V_{CCP}	Supply Voltage During Programming	5.0	6.0	V	
V_{IL}	Input Voltage Low	0	0.5	V	
V_{IH}	Input Voltage High	2.4	V_{CC}	V	
V_{OL}	Output Voltage Low		0.4	V	
V_{OH}	Output Voltage High	3.7		V	
V_{PP1}	Programming Voltage	12.5	13.5	V	Note 1
V_{PP2}	Programming Mode Access Voltage	V_{CCP}	V_{CCP} + 1	V	
I_{PPP}	Supply Current in Programming Mode		100	mA	
I_L	Input or Output Leakage Current	–10	10	µA	
V_{CCL}	First Pass Supply Voltage Low for Final Verification	2.8	3.0	V	
V_{CCH}	Second Pass Supply Voltage High for Final Verification	6.0	6.2	V	

1. No overshoot is permitted on this signal. V_{PP} must not be allowed to exceed V_{PP1} max.

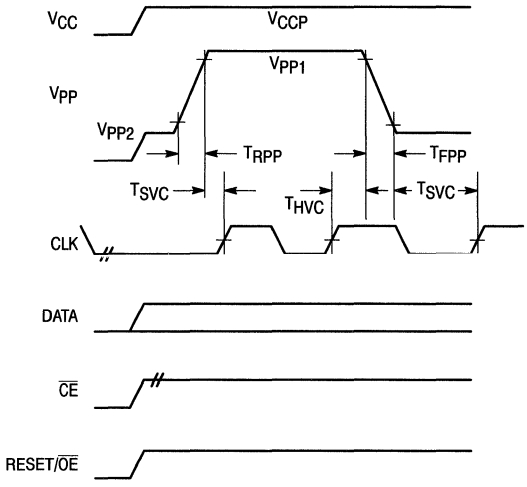


Figure 5. Enter Programming Mode

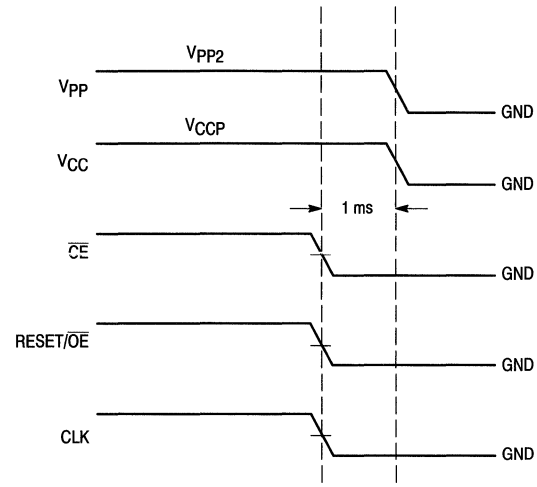


Figure 6. Exit Programming Mode

2

AC PROGRAMMING SPECIFICATIONS

Symbol	Parameter		Limit		Unit	Condition
			Min	Max		
T_{RPP}	Rise Time of V_{PP}	(10 to 90%)	50		ns	
T_{FPP}	Fall Time of V_{PP}	(90 to 10%)	50		ns	
T_{PGM}	V_{PP} Programming Pulse Width		0.95	1.05	ms	
T_{SVC}	V_{PP} Setup to CLK for Entering Programming Mode		100		ns	
T_{HVC}	V_{PP} Hold from CLK for Entering Programming Mode		300		ns	
T_{SDP}	Data Setup to CLK for Programming		50		ns	
T_{HDP}	Data Hold from CLK for Programming		0		ns	
T_{SCC}	$\overline{CE}$ Setup to CLK for Programming/Verifying		100		ns	Note 1
T_{HCC}	$\overline{CE}$ Hold from CLK for Programming/Verifying		200		ns	
T_{SCV}	$\overline{CE}$ Setup to V_{PP} for Programming		100		ns	
T_{HCV}	$\overline{CE}$ Hold from V_{PP} for Programming		50		ns	
T_{SIC}	$\overline{OE}$ Setup to CLK for Incrementing Address Counter		100		ns	
T_{HIC}	$\overline{OE}$ Hold from CLK for Incrementing Address Counter		0		ns	
T_{CAC}	CLK to Data Valid			400	ns	
T_{OH}	Data Hold from CLK		0		ns	
T_{CE}	$\overline{CE}$ Low to Data Valid			250	ns	

1. While in programming mode, $\overline{CE}$ should only be changed while CLK is High and has been High for 200ns.

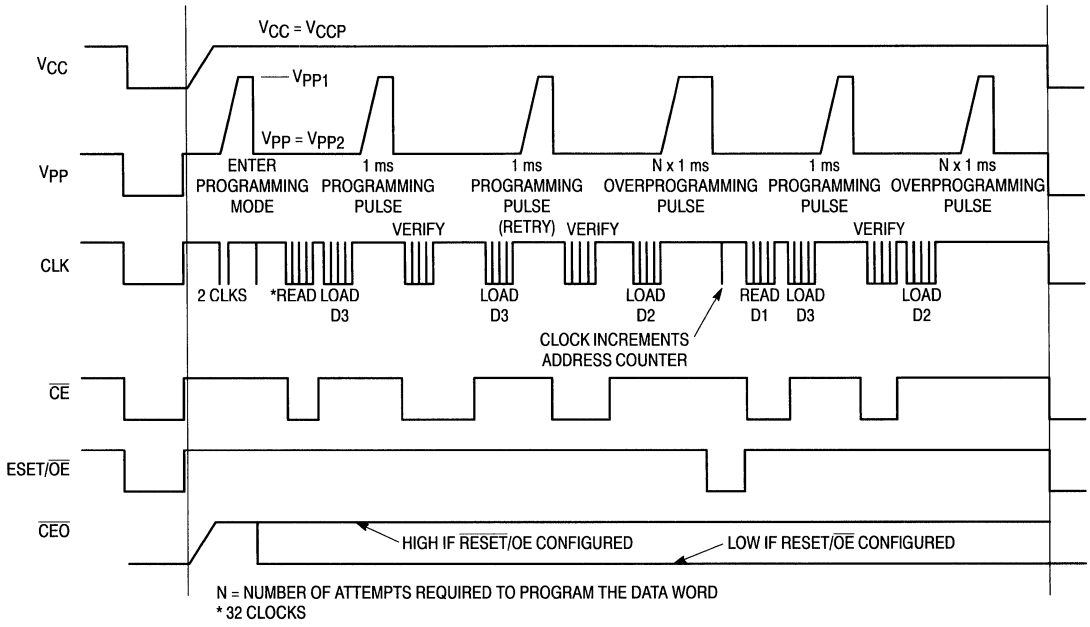


Figure 7. Programming Cycle Overview

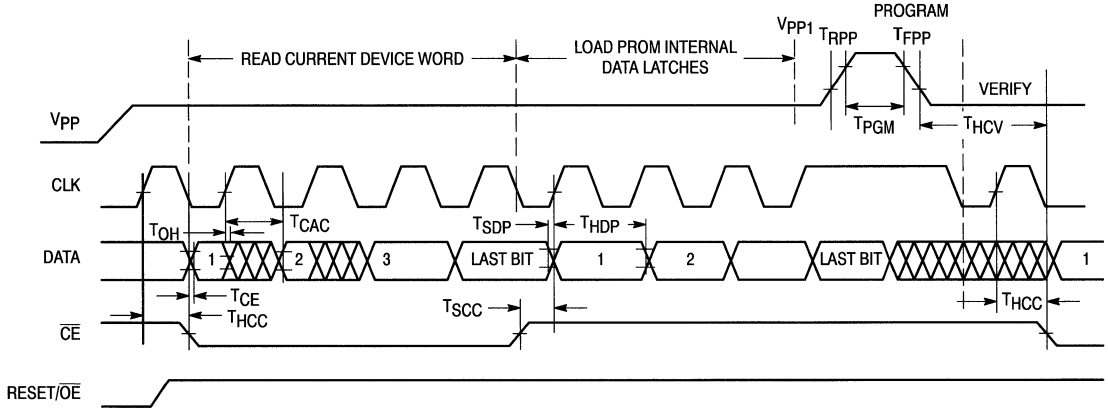


Figure 8. Details of Read/Program/Verify Cycle

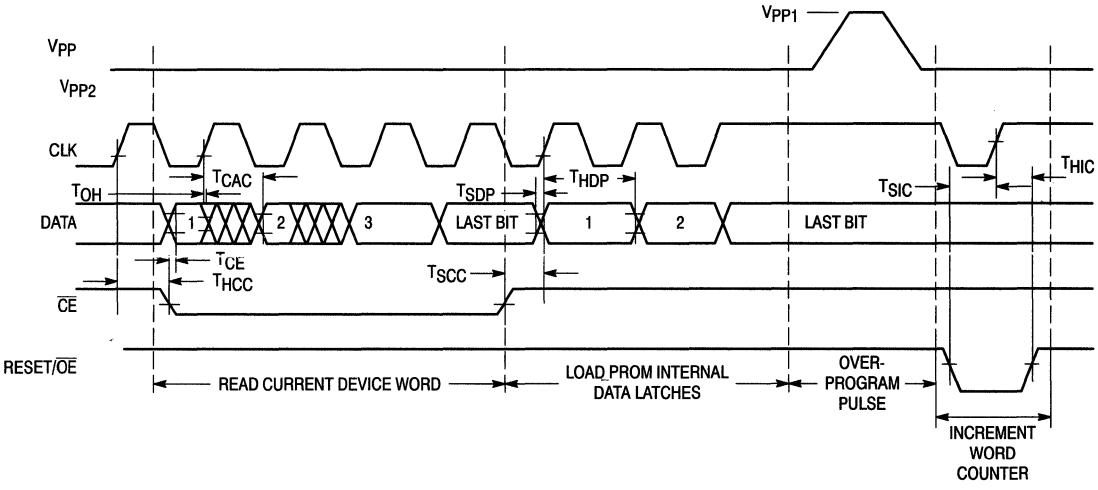


Figure 9. Overprogramming Detail

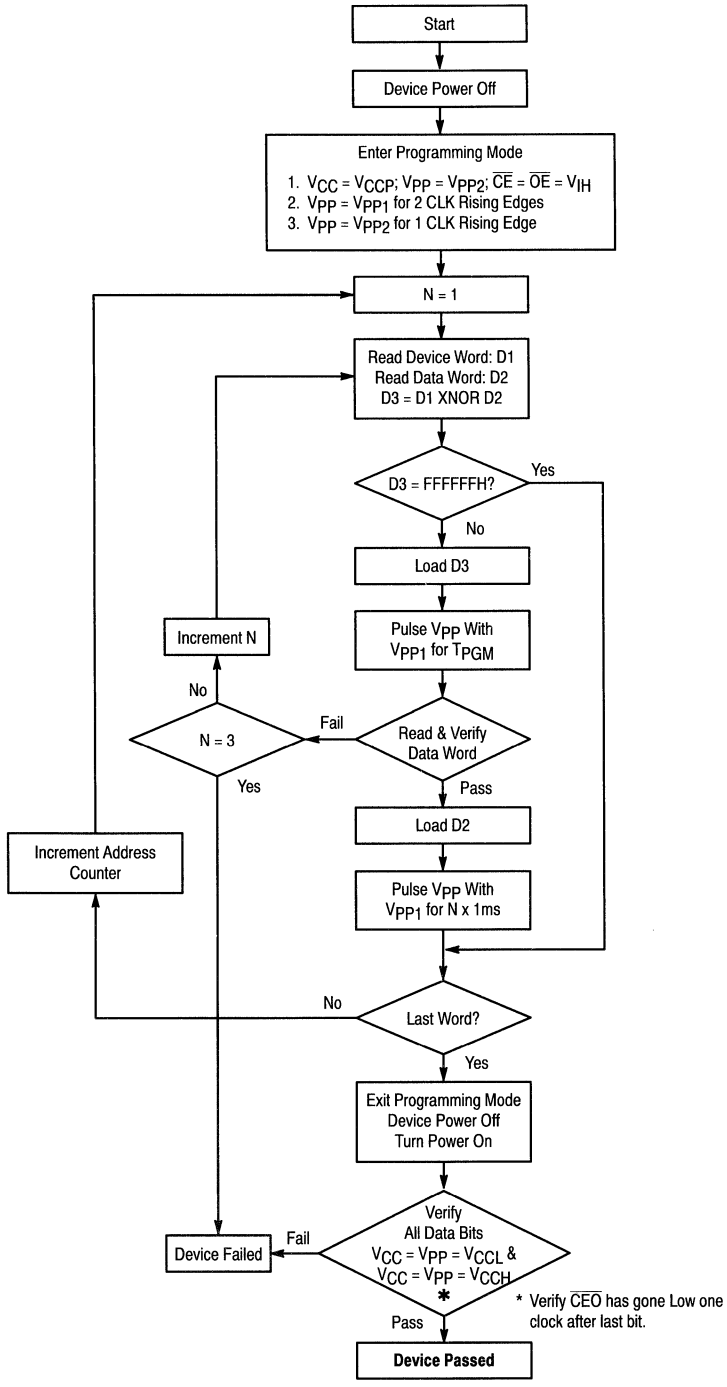
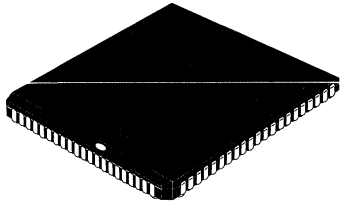
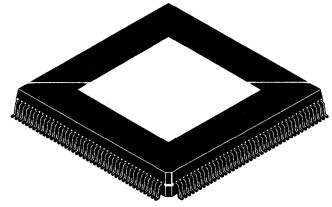


Figure 10. MCP17128 Programming Spec

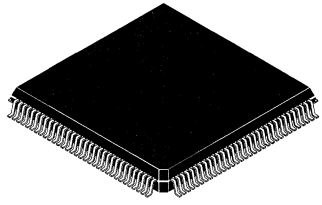
Case Information



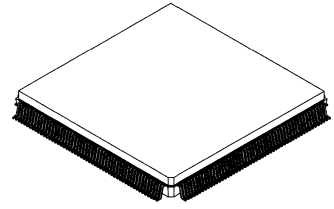
84-Pin PLCC
CASE 780-01
PI SUFFIX



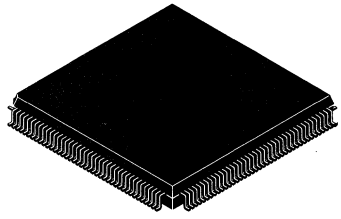
160-Pin μCQFP
CASE 864D-02
MZ SUFFIX



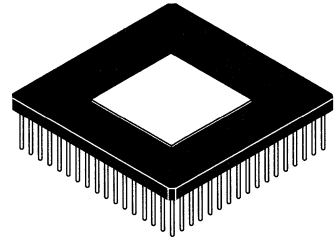
128-Pin PQFP
CASE 862A-01
DD SUFFIX



208-Pin FQFP
CASE 872D-01
DK SUFFIX

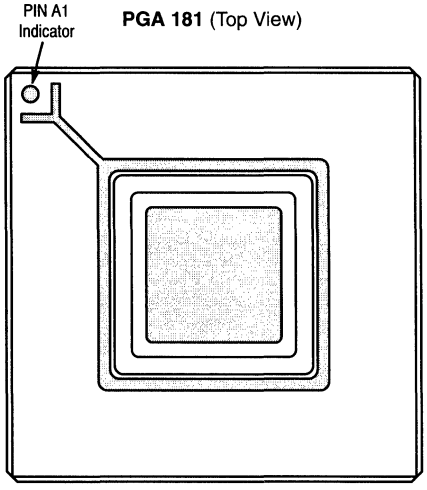


160-Pin PQFP
CASE 864A-01
DH SUFFIX



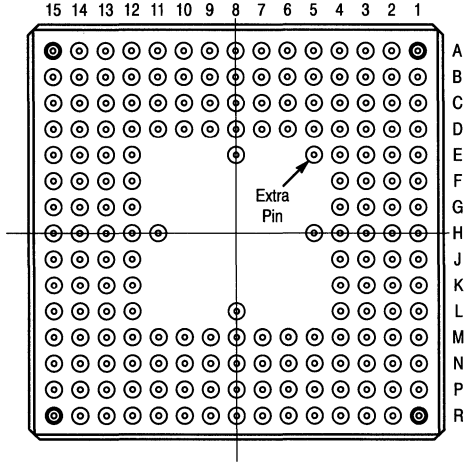
224-Pin PGA
CASE 823B-01
KE SUFFIX

HI SUFFIX
181 PGA PACKAGE
CASE TBD



PGA 181 (Top View)

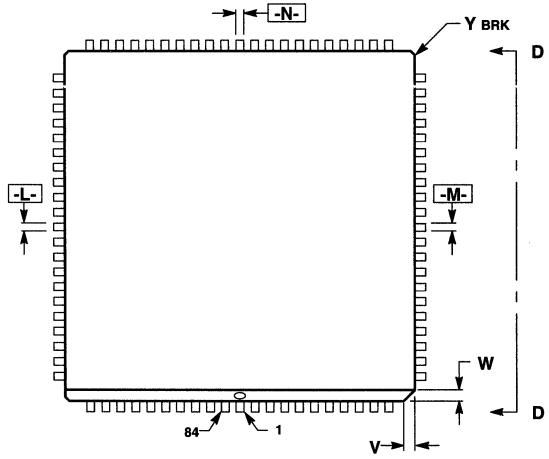
Pinout: PGA 181 (Bottom View)

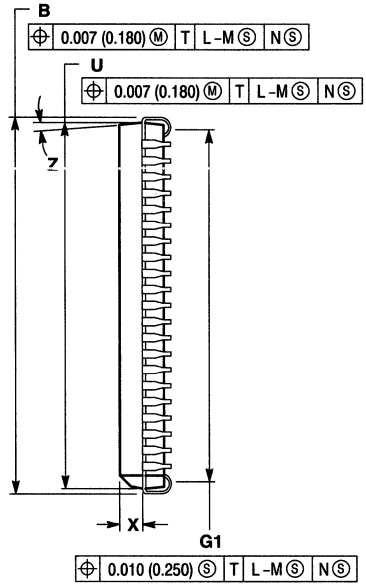


NOTE: Drawing not to scale.

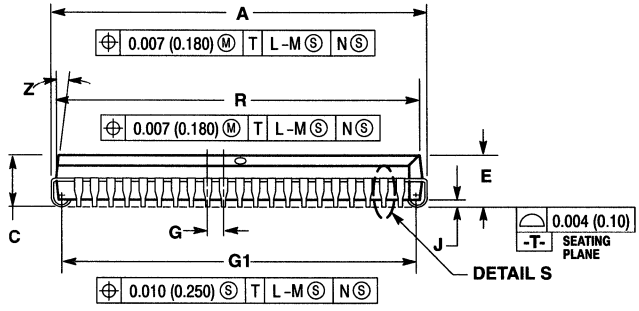
2

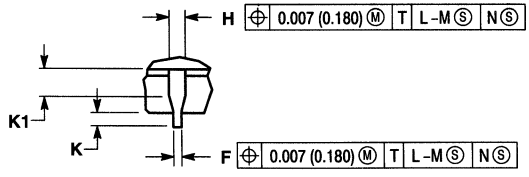
PI SUFFIX
PLASTIC PLCC PACKAGE
CASE 780-01
ISSUE A





VIEW D-D



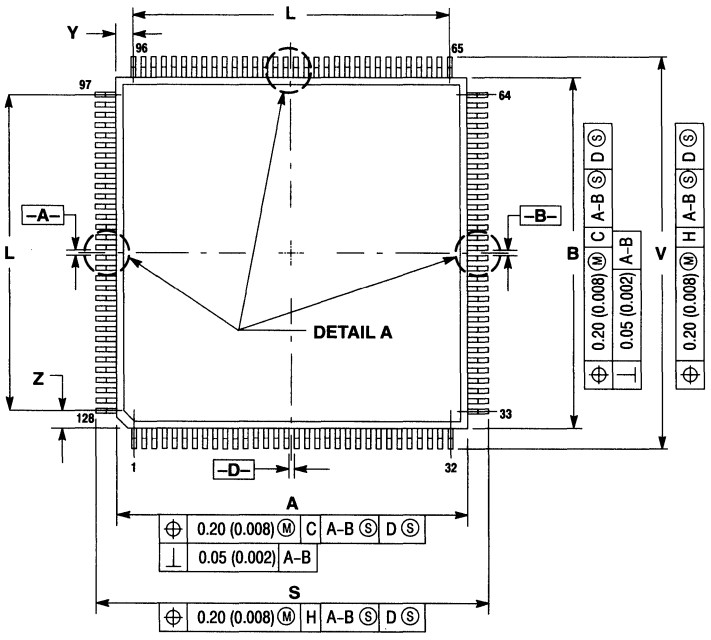


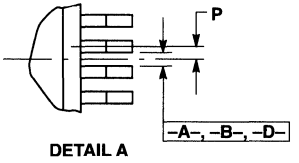
DETAIL S

NOTES:
1. DATUMS -L-, -M-, AND -N- DETERMINED WHERE TOP OF LEAD SHOULDER EXITS PLASTIC BODY AT MOLD PARTING LINE.
2. DIM G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE.
3. DIM R AND U DO NOT INCLUDE MOLD FLASH. ALLOWABLE MOLD FLASH IS 0.010 (0.250) PER SIDE.
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
5. CONTROLLING DIMENSION: INCH.
6. THE PACKAGE TOP MAY BE SMALLER THAN THE PACKAGE BOTTOM BY UP TO 0.012 (0.300). DIMENSIONS R AND U ARE DETERMINED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY EXCLUSIVE OF MOLD FLASH, TIE BAR BURRS, GATE BURRS AND INTERLEAD FLASH, BUT INCLUDING ANY MISMATCH BETWEEN THE TOP AND BOTOM OF THE PLASTIC BODY.
7. DIMENSION H DOES NOT INCLUDE DAMBAR PROTRUSION OR INTRUSION. THE DAMBAR PROTRUSION(S) SHALL NOT CAUSE THE H DIMENSION TO BE GREATER THAN 0.037 (0.940). THE DAMBAR INTRUSION(S) SHALL NOT CAUSE THE H DIMENSION TO BE SMALLER THAN 0.025 (0.635).

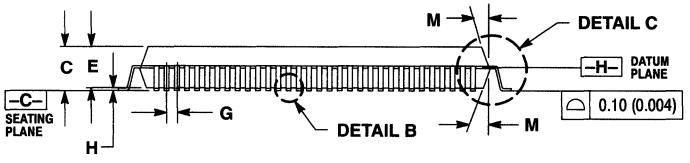
	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	1.185	1.195	30.10	30.35
B	1.185	1.195	30.10	30.35
C	0.165	0.180	4.20	4.57
E	0.090	0.110	2.29	2.79
F	0.013	0.019	0.33	0.48
G	0.050 BSC		1.27 BSC	
H	0.026	0.032	0.66	0.81
J	0.020	—	0.51	—
K	0.025	—	0.64	—
R	1.150	1.156	29.21	29.36
U	1.150	1.156	29.21	29.36
V	0.042	0.048	1.07	1.21
W	0.042	0.048	1.07	1.21
X	0.042	0.056	1.07	1.42
Y	—	0.020	—	0.50
Z	2°	10°	2°	10°
G1	1.110	1.130	28.20	28.70
K1	0.040	—	1.02	—

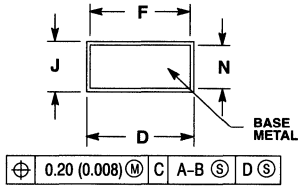
DD SUFFIX
PLASTIC QFP PACKAGE
CASE 862A-01
ISSUE A



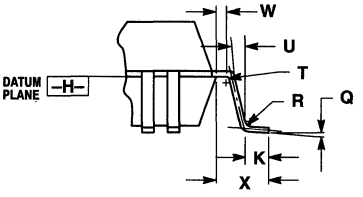


NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER
3. DATUM PLANE -H- IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS -A-, -B- AND -D- TO BE DETERMINED AT DATUM PLANE -H-.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE -C-.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 (0.003) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OF THE FOOT.





DETAIL B

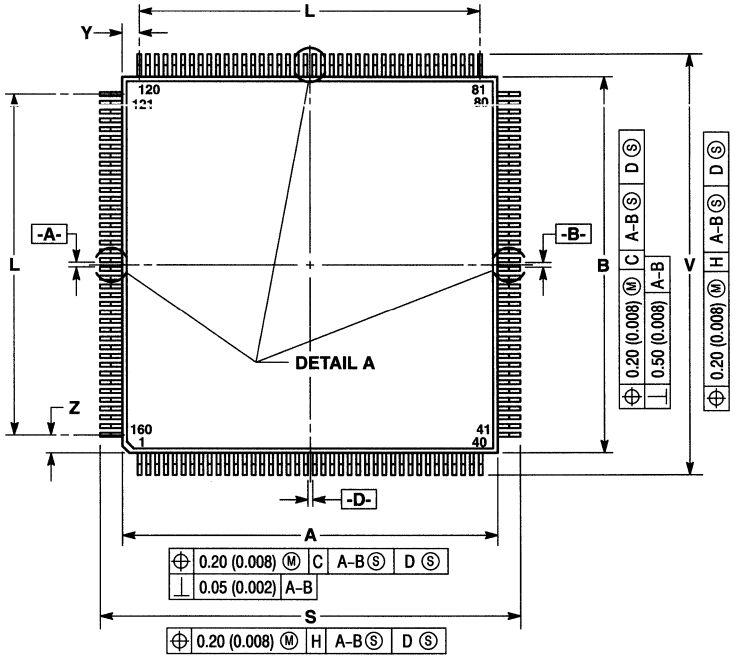


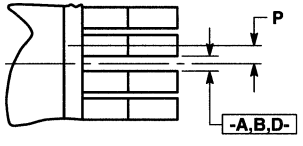
DETAIL C

DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	27.90	28.10	1.098	1.106
B	27.90	28.10	1.098	1.106
C	3.45	3.85	0.136	0.152
D	0.30	0.45	0.012	0.018
E	3.17	3.67	0.125	0.144
F	0.30	0.40	0.012	0.016
G	0.80 BSC		0.032 BSC	
H	0.25	0.35	0.010	0.014
J	0.13	0.23	0.005	0.009
K	0.75	0.92	0.030	0.036
L	24.80 REF		0.976 REF	
M	5 °	16 °	5 °	16 °
N	0.13	0.17	0.005	0.007
P	0.40 BSC		0.016 BSC	
Q	0 °	7 °	0 °	7 °
R	0.13	0.30	0.005	0.012
S	31.10	31.37	1.224	1.235
T	0.13	---	0.005	---
U	0 °	---	0 °	---
V	31.10	31.37	1.224	1.235
W	0.40	---	0.016	---
X	1.60 REF		0.063 REF	
Y	1.60 REF		0.063 REF	
Z	1.60 REF		0.063 REF	

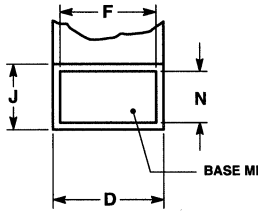
2

DH SUFFIX
PLASTIC QFP PACKAGE
CASE 864A-01
ISSUE B

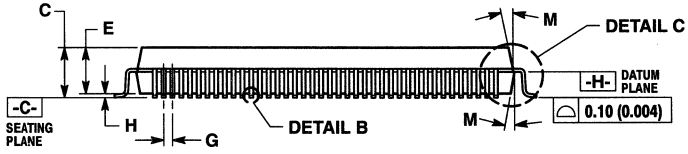


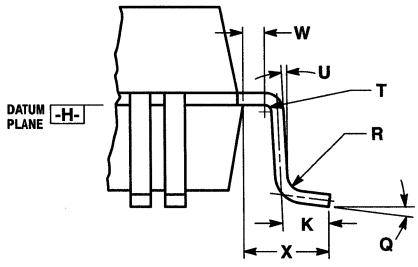


DETAIL A



DETAIL B
VIEW ROTATED 90° CLOCKWISE



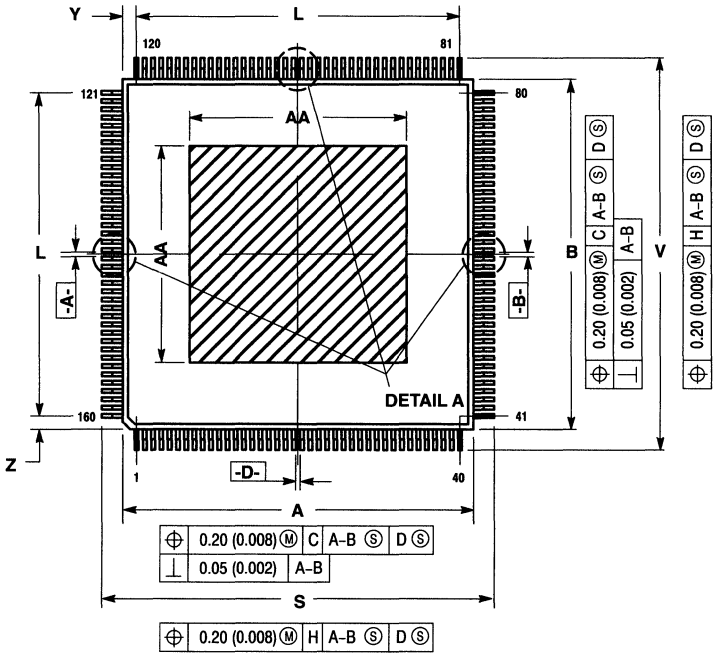


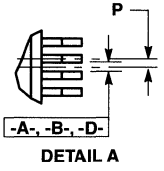
DETAIL C

NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE -H- IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS -A-, -B- AND -D- TO BE DETERMINED AT DATUM PLANE -H-.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE -C-.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 (0.003) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	27.90	28.10	1.098	1.106
B	27.90	28.10	1.098	1.106
C	3.35	3.85	0.132	0.152
D	0.22	0.38	0.009	0.015
E	3.20	3.50	0.126	0.138
F	0.22	0.33	0.009	0.013
G	0.650 BSC		0.0256 BSC	
H	0.25	0.35	0.010	0.012
J	0.11	0.23	0.004	0.009
K	0.70	0.90	0.028	0.035
L	25.35 REF		0.998 REF	
M	5°	16°	5°	16°
N	0.11	0.19	0.004	0.007
P	0.325 BSC		0.0130 BSC	
Q	0°	7°	0°	7°
R	0.13	0.30	0.005	0.012
S	31.00	31.40	1.220	1.236
T	0.13	—	0.005	—
U	0°	—	0°	—
V	31.00	31.40	1.220	1.236
W	0.40	—	0.016	—
X	1.60 REF		0.063 REF	
Y	1.33 REF		0.052 REF	
Z	1.33 REF		0.052 REF	

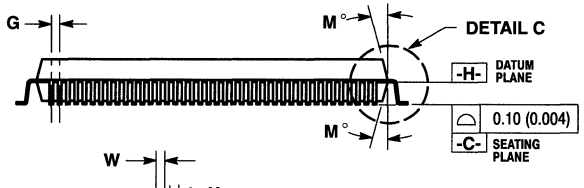
MZ SUFFIX
PLASTIC MICROCOOL QFP PACKAGE
CASE 864D-02
ISSUE B

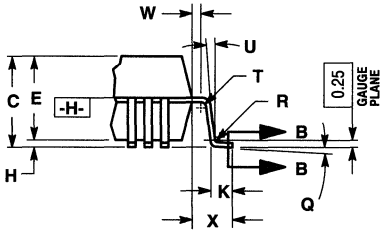




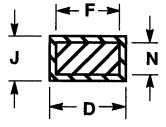
DETAIL A

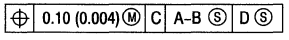
NOTES:
3. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
4. CONTROLLING DIMENSION: MILLIMETER.
5. DATUM PLANE -H- IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
6. DATUMS -A-, -B- AND -D- TO BE DETERMINED AT DATUM PLANE -H-.
7. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE -C-.
8. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.
9. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 (0.003) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.





DETAIL C



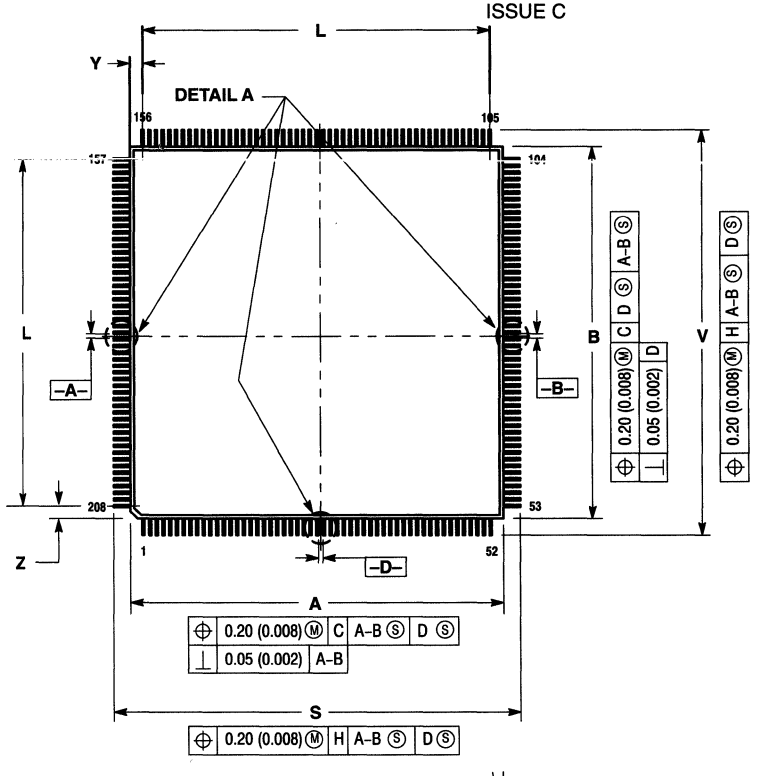


SECTION B-B

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	27.90	28.10	1.098	1.106
B	27.90	28.10	1.098	1.106
C	3.45	3.85	0.136	0.152
D	0.22	0.38	0.009	0.015
E	3.20	3.50	0.126	0.138
F	0.22	0.33	0.009	0.013
G	0.650 BSC		0.0256 BSC	
H	0.25	0.35	0.010	0.012
J	0.13	0.23	0.005	0.009
K	0.75	0.95	0.030	0.037
L	25.35 REF		0.998 REF	
M	5 °	16 °	5 °	16 °
N	0.13	0.17	0.005	0.007
P	0.325 BSC		0.0130 BSC	
Q	0 °	7 °	0 °	7 °
R	0.13	0.30	0.005	0.012
S	31.10	31.37	1.224	1.235
T	0.13	---	0.005	---
U	0 °	---	0 °	---
V	31.10	31.37	1.224	1.235
W	0.40 REF		0.016 REF	
X	1.60 REF		0.063 REF	
Y	1.33 REF		0.052 REF	
Z	1.33 REF		0.052 REF	
AA	16.30	16.30	0.642	0.720

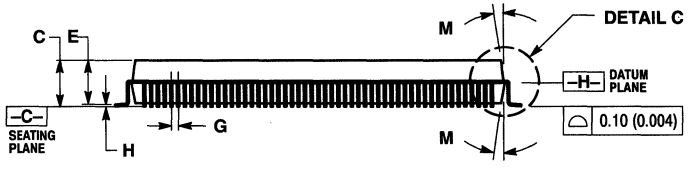
2

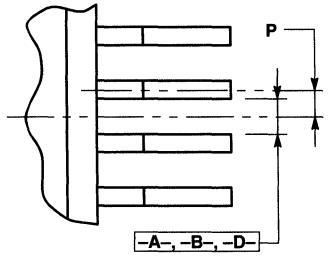
DK SUFFIX
CERAMIC QFP PACKAGE
CASE 872D-01
ISSUE C



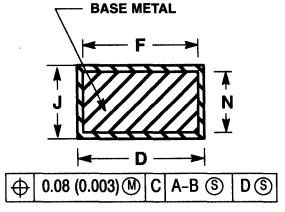
NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE -H- IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS -A-, -B- AND -D- TO BE DETERMINED AT DATUM PLANE -H-.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE -C-.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE D DIMENSION TO EXCEED 0.38 (0.015).

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	27.90	28.10	1.098	1.106
B	27.90	28.10	1.098	1.106
C	3.45	4.10	0.136	0.161
D	0.14	0.30	0.007	0.010
E	3.20	3.60	1.126	0.142
F	0.14	0.26	0.005	0.012
G	0.50 BSC		0.020 BSC	
H	0.25	0.35	0.010	0.014
J	0.09	0.20	0.003	0.008
K	0.70	0.90	0.027	0.036
L	25.50 REF		1.004 REF	
M	5°	16°	5°	16°
N	0.09	0.18	0.003	0.007
P	0.25 BSC		0.010 BSC	
Q	0°	7°	0°	7°
R	0.13	0.30	0.005	0.012
S	31.00	31.40	1.220	1.236
T	0.13	---	0.005	---
U	0°	---	0°	---
V	31.00	31.40	1.197	1.236
W	0.40	---	0.016	---
X	1.60 REF		0.063 REF	
Y	1.25 REF		0.049 REF	
Z	1.25 REF		0.049 REF	

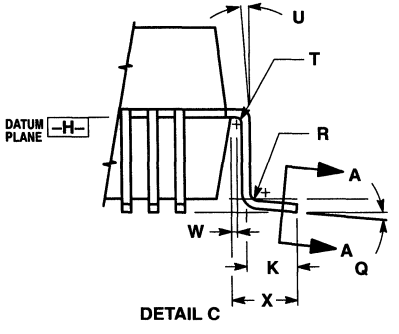




DETAIL A

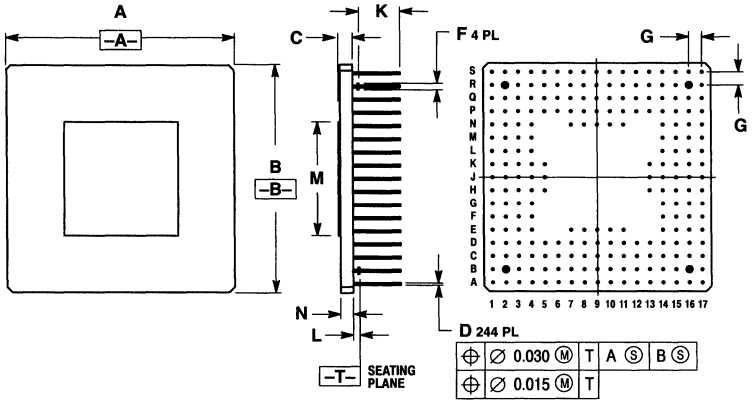


SECTION AA



DETAIL C

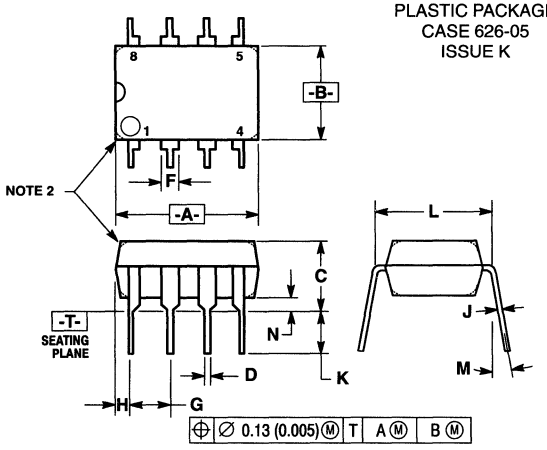
KE SUFFIX
PGA PACKAGE
CASE 823B-01
ISSUE O



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	1.726	1.744	43.84	44.30
B	1.726	1.744	43.84	44.30
C	0.095	0.120	2.41	3.05
D	0.018		0.46	
F	0.050		1.27	
G	0.100 BSC		2.54 BSC	
K	0.283	0.339	7.19	8.61
L	0.043	0.057	1.09	1.45
M	0.865	0.885	21.97	22.48
N	0.080	0.100	2.03	2.54

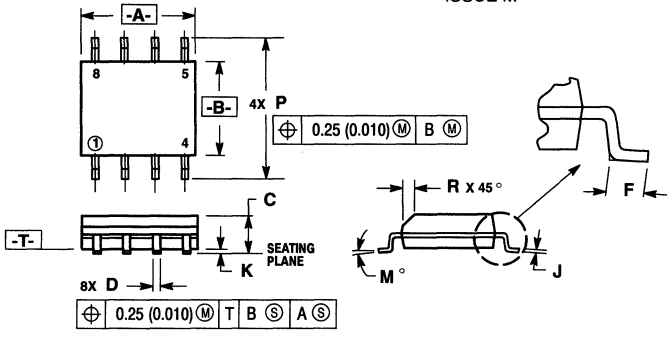
P SUFFIX
PLASTIC PACKAGE
CASE 626-05
ISSUE K



NOTES:
1. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
2. PACKAGE CONTOUR OPTIONAL (ROUND OR SQUARE CORNERS).
3. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	9.40	10.16	0.370	0.400
B	6.10	6.60	0.240	0.260
C	3.94	4.45	0.155	0.175
D	0.38	0.51	0.015	0.020
F	1.02	1.78	0.040	0.070
G	2.54 BSC		0.100 BSC	
H	0.76	1.27	0.030	0.050
J	0.20	0.30	0.008	0.012
K	2.92	3.43	0.115	0.135
L	7.62 BSC		0.300 BSC	
M	—	10°	—	10°
N	0.76	1.01	0.030	0.040

D SUFFIX
PLASTIC SOIC PACKAGE
CASE 751-05
ISSUE M

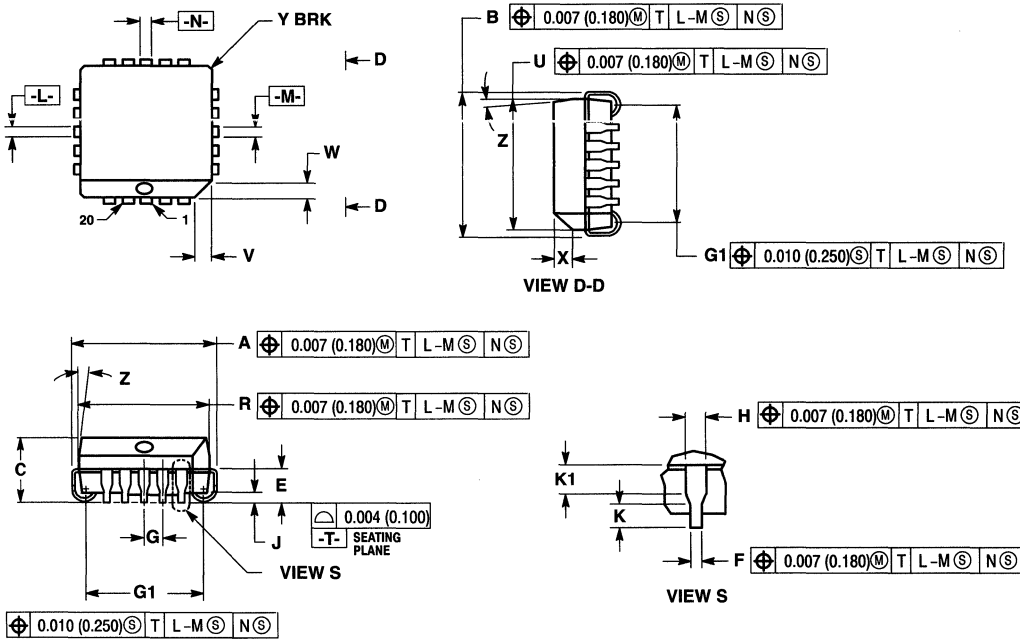


NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.127 (0.005) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	4.80	5.00	0.189	0.196
B	3.80	4.00	0.150	0.157
C	1.35	1.75	0.054	0.068
D	0.35	0.49	0.014	0.019
F	0.40	1.25	0.016	0.049
G	1.27 BSC		0.050 BSC	
J	0.18	0.25	0.007	0.009
K	0.10	0.25	0.004	0.009
M	0°	7°	0°	7°
P	5.80	6.20	0.229	0.244
R	0.25	0.50	0.010	0.019

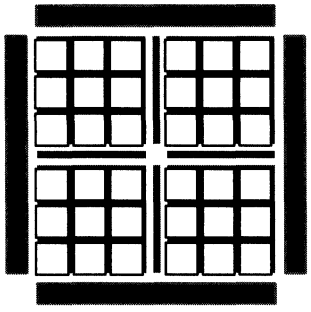
2

FN SUFFIX
PLCC PACKAGE
CASE 775-02
ISSUE C



NOTES:
1. DATUMS -L-, -M-, AND -N- DETERMINED WHERE TOP OF LEAD SHOULDER EXITS PLASTIC BODY AT MOLD PARTING LINE.
2. DIM G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE.
3. DIM R AND U DO NOT INCLUDE MOLD FLASH. ALLOWABLE MOLD FLASH IS 0.010 (0.250) PER SIDE.
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
5. CONTROLLING DIMENSION: INCH.
6. THE PACKAGE TOP MAY BE SMALLER THAN THE PACKAGE BOTTOM BY UP TO 0.012 (0.300). DIMENSIONS R AND U ARE DETERMINED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY EXCLUSIVE OF MOLD FLASH, TIE BAR BURRS, GATE BURRS AND INTERLEAD FLASH, BUT INCLUDING ANY MISMATCH BETWEEN THE TOP AND BOTTOM OF THE PLASTIC BODY.
7. DIMENSION H DOES NOT INCLUDE DAMBAR PROTRUSION OR INTRUSION. THE DAMBAR PROTRUSION(S) SHALL NOT CAUSE THE H DIMENSION TO BE GREATER THAN 0.037 (0.940). THE DAMBAR INTRUSION(S) SHALL NOT CAUSE THE H DIMENSION TO BE SMALLER THAN 0.025 (0.635).

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.385	0.395	9.78	10.03
B	0.385	0.395	9.78	10.03
C	0.165	0.180	4.20	4.57
E	0.090	0.110	2.29	2.79
F	0.013	0.019	0.33	0.48
G	0.050 BSC		1.27 BSC	
H	0.026	0.032	0.66	0.81
J	0.020	—	0.51	—
K	0.025	—	0.64	—
R	0.350	0.356	8.89	9.04
U	0.350	0.356	8.89	9.04
V	0.042	0.048	1.07	1.21
W	0.042	0.048	1.07	1.21
X	0.042	0.056	1.07	1.42
Y	—	0.020	—	0.50
Z	2°	10°	2°	10°
G1	0.310	0.330	7.88	8.38
K1	0.040	—	1.02	—



Field Programmable Gate Arrays

Quality and Reliability 3

Quality and Reliability

Quality

The Motorola culture is a culture of quality. Throughout all phases of product development, from defining and designing to shipping the product, Motorola strives for total customer satisfaction through "Six Sigma" and "On Time Delivery" programs.

Designing Products

Extensive work was done on the 75% UDR CMOS process to ensure a solid platform for quality products. Process reliability studies were performed to uncover any weaknesses in the initial process so that enhancements could be made to strengthen it before it was released to production. In addition, comprehensive characterization and correlation work was completed on the process to ensure the utmost in modeling parameter accuracy.

The design of the products strictly adhered to the design rules set forth by the process designers. Conservative, manufacturable layout rules were followed to minimize the performance variability due to a marginally manufacturable product.

Manufacturing Process

Through SPC and continual engineering work, the manufacture of the 75% UDR CMOS process is both monitored and enhanced on a continuous basis. Statistical data is gathered at both probe and final test through the device data collection to monitor the distribution of a parameter to its specification limits. In addition, final quality assurance gates are set up to guarantee the quality of outgoing product.

Product Characterization

Products are both DC and AC characterized for all data book environmental conditions prior to the release of the product to production. The distributions of the parameters are compared to their specification limits to ensure that Motorola "manufactures" quality products as opposed to "testing" quality products through distribution truncation. In addition, ongoing AC characterization is performed to enhance the distributions of the AC parameters of the device. In doing so, as the distributions warrant, further enhancements to the AC specifications can be achieved.

Reliability

To ensure the long term reliability of MPA products, extensive accelerated life testing is performed prior to production release. This qualification work is performed by Logic Reliability Engineering, an organization specifically dedicated to monitoring and guaranteeing the quality and and reliability of logic products. The accelerated life test consists of the following:

Operating Life Test: 145°C, 5.5V Man Supply
Temperature Cycle: –65°C to 150°C
Pressure, Temperature, Humidity (Hermeticity)

A minimum of 35 lots, 100 die per lot taken from seven different waters in the lot constitute a qualification sample. Various intermediate readouts are taken to monitor the performance more closely. In addition, the devices are tested beyond the specification limits to determine where and how they will fail.

Another responsibility of the reliability group is that of failure analysis. This failure analysis service is supported for both internal purposes and for servicing the needs of our customers. Analysis entails everything from simple package examination to internal microprobing to SEM analysis of IC structures. The results of the analysis are returned to the customer and if the analysis suggests a potential problem with the device the information is also passed to the internal product groups.

RAP: Reliability Audit Program

The Reliability Audit Program (RAP) devised in March 1977 is the Motorola internal reliability audit which is designed to assess outgoing product performance under accelerated stress conditions. Logic Reliability Engineering has overall responsibility for RAP, including updating its requirements, interpreting its results, administration at offshore locations and monthly reporting of results. These reports are available at all sales offices. Also available is the "Reliability and Quality Handbook" which contains data for all Motorola semiconductors (BR518/D).

Rap is a system of environmental and electrical tests performed periodically on randomly selected samples of standard products. Each sample receives the tests specified in Figure 3-1. Frequency of testing is specified per internal document 12MRM15301A.

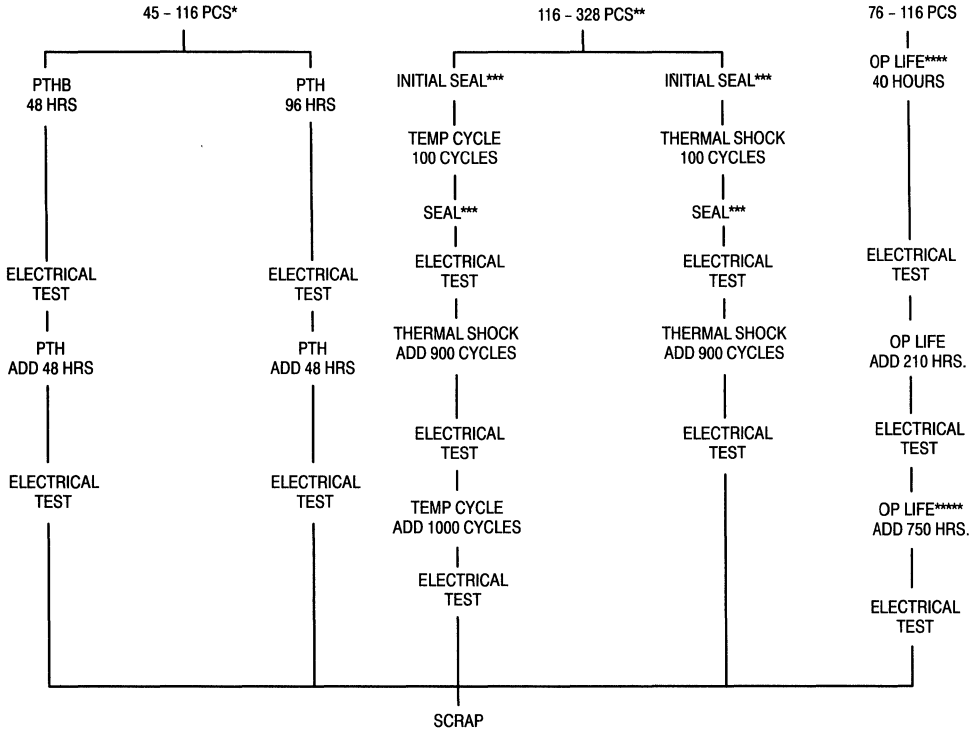


Figure 3-1. Reliability Audit Program Test Flow

* PTH will be run as a substitute if PTHB sockets are not available. Only required on plastics packages.

** Thermal Shock will be run if Temp Cycle is not available.

*** Seal (fine and gross) only required on hermetic packages.

**** All units for Op Life to be AC/DC tested before and after being stressed. All units failing AC after stress will be analyzed.

***** One sample per month

PTHB

15psig/121°C/100% RH at rated V_{CC} or V_{EE} – to be performed on plastic encapsulated devices only.

Temp Cycle

Mil. Std. 883, Method 1010, Condition C, – 65°C to 150°C

Op Life

Mil. Std. 883, Method 1005, Condition C (Power plus Reverse Bias), T_A = 145°C.

Notes:

1. All standard 25°C DC and functional parameters will be measured Go/NoGo at each readout.
2. Any indicated failure is first verified and then submitted to the Product Analysis Lab for detailed analysis.
3. Sampling to include all package types routinely.
4. Device types sampled will be by generic type and will include all assembly locations.
5. 16 hrs. PTHB is equivalent to ≈ 800 hrs. of 85°C/85% RH THB for $V_{CC} \leq 15V$.
6. Only moisture related failures (like corrosion) are criteria for failure on PTHB test.
7. Special device specifications (48A's) for digital products will reference 12MRM15301A as a source of generic data for any customer requiring monthly audit reports.

Data Integrity

To provide stability during readback, the output path inverter is slightly stronger than that of the input path. In addition, the pass transistor used when writing to the cell (controlled by 'w') is stronger than the readback device (controlled by 'wb').

The RAM cell is layed out to a width of approximately 3 metal pitches, to support one vertical data programming line plus two potential interconnect 'channels'. The P–devices are connected to the bootstrapped Vpp line (Vdd+Vt).

Latch up

Latch up will not a problem for most designs, but the designer should be aware of it, what causes it, and how to prevent it.

Figure 3-3 shows the cross–section of a typical CMOS inverter and Figure 3-2 shows the parasitic bipolar devices. The circuit formed by the parasitic transistors and resistors is the basic configuration of a silicon controlled rectifier, or SCR. In the latch up condition, transistors Q1 and Q2 are turned ON, each providing the base current necessary for the other to remain in saturation, thereby latching the devices in the ON state. Unlike a conventional SCR, where the device is turned ON by applying a voltage to the base of the NPN transistor, the parasitic SCR is turned ON by applying a voltage to the emitter of either transistor. The two emitters that trigger the SCR are at the same point, the CMOS output. Therefore, to latch up the CMOS device, the output voltage must be greater than V_{DD}+0.5V or less than V_{SS}–0.5V and have sufficient current to trigger the SCR. The latch–up mechanism is similar for the inputs.

To reduce the current for triggering the SCR, guard rings are formed and act as dummy collectors to collect charges directly through V_{CC} and ground, rather than through active circuitry, thereby shunting the parasitic transistors. The guard rings are connected to V_{CC} and ground near the input and output diodes to short out the parasitic SCR. Guard ring diffusion also creates additional parasitic transitors and reduces effective substrate resistance which makes the SCR harder to turn on.

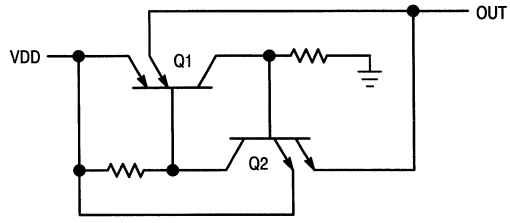


Figure 3-2. Latch–Up Circuit Schematic

Once a CMOS device is latched up, if the supply current is not limited, the device will be destroyed. Ways to prevent such occurrences are listed below:

10. Insure that inputs and outputs are limited to the maximum rated values, as follows: $-0.5V \le V_{in}$ or $V_{out} \le V_{DD} + 0.5V$ (referenced to V_{SS}) Iin ot Ioutl ≤ 10mA (unless otherwise indicated on the data sheet).
11. If voltage transients with sufficient energy to latch up the device is expected on the inputs or outputs, external protection diodes can be used to clamp the voltage. Another method of protection is to use a series resistor to limit the expected worst case current to the maximum rating of 10mA.
12. Sequence power supplies so that the inputs or outputs of CMOS devices are not active before the supply pins are powered up (e.g., recessed edge connectors and/or series resistors may be used in plug–in board applications).
13. Voltage regulating or filtering should be used in board design and layout to insure that power supply lines are free of excessive noise.
14. Limit the available power supply current to the devices that are subject to latch–up conditions. This can be accomplished with a power supply filtering network or a current limiting regulator.

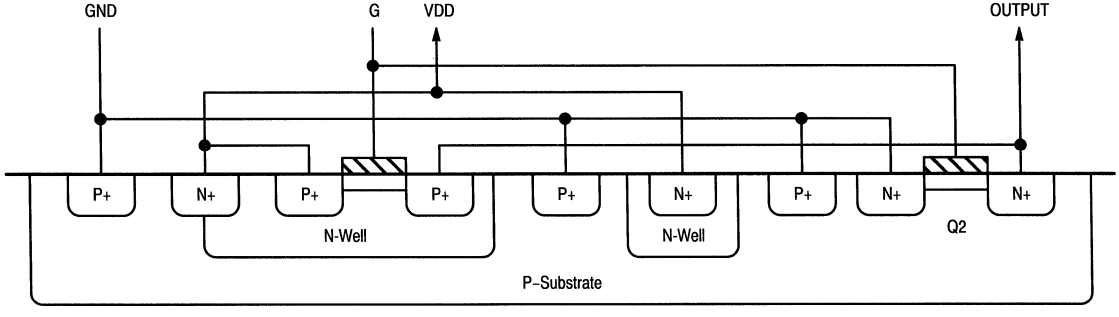


Figure 3-3. CMOS Wafer Cross–Section

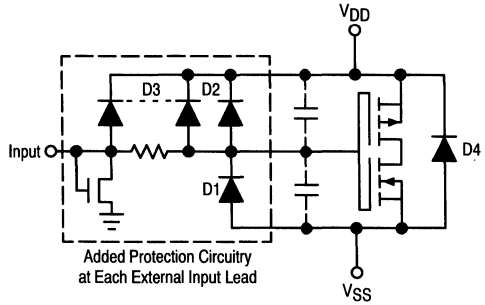
Electrostatic Discharge

The gate electrode of a CMOS circuit is completely isolated from the substrate by a silicon dioxide layer, which forms the dielectric of the gate–to–substrate capacitor. The thickness of the oxide insulator between the gate and the substrate of a MOS device is about 1000Å and has a typical breakdown voltage in the range of 100 to 120V.

If a voltage higher than the breakdown voltage is applied to the gate, the silicon dioxide beneath the gate will rupture. This can result in permanent damage of the device, causing a short between gate metal and either the substrate or a P or N region. Because of the extremely high resistance of the gate oxide, even a very low energy source (i.e., stray electrostatic charges) is capable of developing this breakdown voltage. The possibility that a CMOS device will be destroyed by static overvoltage exists only during handling and testing. Once the device is mounted in a circuit, normal circuit impedances and voltages make this danger virtually impossible. In order to avoid destruction of CMOS devices by static discharge, various input protection circuits were developed.

The protection system used for ESD is the Double Diode Plus Resistor Protection Circuit as shown in Figure 3-4.

It consists of a series isolation resistor Rs, whose average value is 1.5kohm, and diodes D1 and D2 for clamping excess input voltages to the power supply pins, VDD or VSS. Diode D3 is a distributed parasitic structure resulting from the diffusion fabrication of Rs.

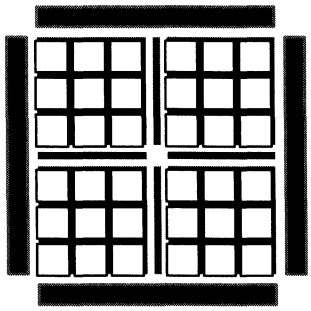


NOTES: R_S = 1.5kΩ Nominal
Avalanche Voltages
BVD_1 = 30V BVD_2 = 30V
BVD_3 = 80V BVD_4 = 90V

Figure 3-4. Double Diode Plus Resistor Protection Circuit

3

3



Field Programmable Gate Arrays

This section describes the supported Motorola products for Schematic Capture and Simulation, and Place and Route tools. The subject of logic synthesis and optimization is discussed and two recommended tools for this purpose.

Software Support and Tools 4

CONTENTS

Motorola MPA1000 Design System
View*logic* Schematic Capture and Simulation

Highlights

- Motorola sold and supported system to help assure first–pass success
- Integrates cleanly with the Motorola/NeoCAD backend software
- Simulation vectors are easily generated via timing diagrams, tabular data or a powerful command language
- Logic values displayed on the schematic provide the industry's best simulation debugging environment
- Post layout simulation verifies correct timing
- The Motorola design environment is available on the PC, Sun Sparc and HP workstations
- Upgrade options provide growth path into View*logic*'s full suite of system–level design tools

4

Product Overview

The Motorola design system is a turnkey design entry and simulation environment that helps to assure success in implementing Motorola FPGA designs. Its intuitive graphical interface makes the system easy to learn and use.

ViewDraw provides advanced design capture and analysis

View*logic*'s ViewDraw goes beyond traditional schematic capture by supporting a wide range of graphical and textual design entry methods, and by serving as an effective front end for any design methodology. ViewDraw provides a single point of design entry for all designs, which can be entered graphically.

ViewSim for powerful, interactive simulation

ViewSim is an interactive, event–driven, digital simulator. Its versatile evaluation mechanism uses the complete IEEE 1076 VHDL language for defining arbitrary states and strengths. ViewSim also supports a unique 28–state algorithm to assure high–speed and high accuracy evaluation of designs. Designs can be entered through a wide variety of input, including schematics, EDIF, JEDEC, and IEEE 1076 VHDL. These forms can be mixed and matched within ViewSim and tied together through hierarchical blocks on the master schematic.

ViewGen

ViewGen generates schematic drawings from structural or netlist descriptions. This powerful tool creates any needed component symbols and can generate a top–level symbol for the entire design, which can be used to include the netlist in other design hierarchies. Buses are supported by ViewGen to create readable and efficient schematic drawings. ViewGen streamlines design processes such as logic synthesis and design retargetting FPGA and PLD technologies by automatically creating the schematics of the created design.

Growth path to system–level design tools

Upgrade options allow you to expand your MPA1000 design system for system–level implementation of designs. View*logic*'s family of tools provide Motorola MPA1000 designers with a growth path that spans a wide range of design solutions that run on standard hardware platforms. All designs created with the MPA1000 system are upwardly compatible with the Workview environment, and customers who decide to upgrade to system–level design capabilities receive full credit from View*logic* toward the purchase of new software.

Motorola MPA1000 Design System

- 386/486 PC
- Sun Sparc station
- HP 700 Series
- Printer plotter support — standard

Contact

Contact your local Motorola sales office or authorized distributor, or View*logic* at:

View*logic* Systems
Marlboro, Massachusetts
Tel: 508 480–0881 or
1–800–873–8439

Motorola MPA1000 Design System NeoCAD Place and Route Software

Highlights

- Motorola sold and supported FPGA Place and Route Software
- For use with View*logic* and Mentor Graphics Schematic Capture tools
- May be used with Synopsys and Exemplar Logic synthesis tools
- Provides back annotation for ViewSim and Quickdraw simulators, and SDF file interchange for Verilog and VHDL simulators*
- Available on PC–386/486 Windows 3.0 or higher, Sun (O.S. 4.1.3 or higher) and H.P. workstations (O.S. 8.0 or higher)
- May be upgraded to other semiconductor vendors' FPGA

Product Overview

The MPA1000 design system sold by Motorola and its franchised distributors, consists of a front–end schematic capture and simulation package from View*logic* and a back–end timing driven place and route tool from NeoCAD. The two packages team up to provide the user with an easy to use, low cost set of tools for use with either PC or Unix workstations. The NeoCAD tool may be used with Mentor Graphics Quickdraw or with logic synthesis tools such as Synopsys and Exemplar.

FPGA Foundry™

This set of tools was developed to be a device independent tool–set designed to meet the needs of today's FPGA designer. The design team of the nineties and beyond needs to select the best FPGA devices. Just as the ASIC world has migrated to third party tools from Vendor specific tool, Foundry offers solutions for Motorola and many other key FPGA vendors. For a complete list of all vendors supported by foundry please contact NeoCAD.

Capture, Mapping and Optimization Support Technology–Transparent Design and Device Retargetting

FPGA Foundry allows designs to be captured using device–specific libraries, vendor–independent libraries or a combination of both. No other design toolset lets you designate the specific design capture method which best supports your requirements. Vendor–independent libraries and industry standard netlists, for example, give you greater design flexibility by permitting easier migration among FPGA architectures or between technologies (e.g., TTL, PLDs, ASICs).

FPGA Foundry's device– and architecture–specific optimization combined with superior place and route capabilities produce consistently higher utilization than that provided by the vendor's proprietary mappers. Of course, Foundry fully supports device–specific features such as hard macros, RAM and automatic assignment of global clock buffers.

With complete back annotation, incremental mapping and the ability to preserve hierarchy throughout the design process, Foundry gives you as much help in updating and debugging your design as it does in implementing it.

Advanced Place and Route Capabilities (PAR)

Customer benchmarks have shown FPGA Foundry's Place and Route (PAR) program to be significantly faster than other place and route tools. Using the most powerful combination of algorithms available, PAR consistently completes designs with the fewest iterations and with no manual intervention. And, PAR's fast execution time (up to ten times faster than conventional tools) and built–in incremental change capability result in the shortest possible design cycle.

With the addition of NeoCAD's Timing Wizard module, designers can specify frequency and timing requirements up front. Timing Wizard then drives PAR to meet those requirements, delivering higher performance devices with the fastest possible operating frequencies while shortening design cycles even further.

Powerful Interactive Layout Editor (EPIC™)

NeoCAD's Editor for Programmable ICs (NeoCAD EPIC) is a powerful, interactive layout editor that streamlines the debugging and tuning of FPGA designs. EPIC's easy–to–use graphical interface provides a choice of push–button, menu–driven or command–line editing capabilities that can be customized to suit any set of requirements. In addition, EPIC has been tuned to guarantee the fastest graphics response, eliminating the unproductive waiting while a large design is panning, zooming or simply highlighting a net.

Many advanced features have been designed into EPIC to make working with complex devices easier. Among these are manual place and routing, autoplacement, autorouting and integration of NeoCAD's powerful timing analyzer. EPIC's on–line DRC can be used in logical mode (allowing changes to placement and routing, but preventing any changes to the logic during the editing session) or in physical mode (allows logic and signals to be added and deleted while guaranteeing that changes are valid within the physical constraints of the specified FPGA).

FPGA Specific Timing Analyzer (TRACE™)

NeoCAD's TRACE provides complete analysis of a circuit's timing characteristics. Using actual component and interconnect delays, TRACE exhaustively examines every signal path and automatically evaluates the circuit for setup

and hold violations, race conditions and adherence to specified timing preferences.

TRACE runs its analysis using user–specified timing preferences (such as desired operating frequency) and feeds back detailed results that identify specifically where the design fails to meet those requirements, thus eliminating the need to read through reams of paper to pinpoint potential timing problems.

NeoCAD Foundry Upgrades

Motorola and its franchised distributors will sell a *"library–locked"* version of NeoCAD foundry and View*logic*'s ViewDraw and ViewSim. Both software vendors are offering very attractive pricing for these tools. If the designer has purchased a library locked version, locked to Motorola MPA series devices and feels the need to try one or more vendors' devices, he can easily upgrade through the software vendor or the Value Added Reseller (VAR) in his geographic area. Since the tools were designed to be vendor independent there is no learning curve with the added vendor modules. the designer only needs to learn one tool for all of his or her designs. Each of the vendors will credit a generous portion of the original purchase price towards and "open–system" configuration with one or more additional FPGA vendors included. For specifics of this purchase credit please contact:

NeoCAD Inc.
2585 Central Avenue
Boulder, Colorado
Tel: (303) 442–9121
1–800–862–3143

4

Motorola MPA1000 Design System Logic Synthesis Tools

Introduction

Motorola has been working with the major software vendors to supply logic synthesis tools for the MPA series of FPGAs. The MPA architecture is fine–grained, consisting of small easily synthesizable elements, perfect for logic synthesis. our plan is to support a minimum of four third party vendors: Synopsys and Mentor for workstation software, and Exemplar and View*logic* for the PC. Motorola will not sell or directly support this software, but our strong partnership relationships with these vendors will assure a consistently high level of service.

High Level Design Language

Mazor and Langstraat [1] discuss the use of High Level Design Language (HDL) for documenting a design, simulating behavior of a design, and direct logic synthesis. The two popular languages supported by most third party vendors are Verilog–HDL and VHDL, both IEEE standard descriptive languages. The four supported vendors either include simulation capability or work with other standard third party vendor simulation packages (please contact the individual vendors for specifics of simulation support). The use of an HDL allows for technology independent design procedures. In general, code written in HDL is portable and transferrable across many vendors tools and since it is not specifically targeted at either an FPGA, gate array, or standard cell, it can be re–compiled with several target libraries for different goals. Rauba [2} describes the design of a dram controller for a microprocessor. He describes using Verilog–HDL language and the Exemplar tool, and third party simulation, and then take the same device file in Verilog, and re–targets using Motorola's OACS bundled tools, that includes either Synopsys or Mentor Graphics Autologic and re–targets the chip to a Motorola H4C gate array.

HDL, Logic Synthesis and Simulation

These tools range in price from a few thousand dollars each to about $100,000 depending on complexity and speed. Tools such as Exemplar and View*logic*'s Viewsynth are available on a PC for under $10,000 and is extremely viable for many designs. The MPA series of products is extremely amenable to this design flow and we encourage its use. Please contact the individual vendors for complete details of their software offerings.

[1] Mazor and Langstraat, *A Guide to VHDL*, Kluwer Academic Publishers, 1992.

[2] Phil Rauba, *68030 DRAM Controller Design Using Verilog–HDL*, Application Notes included in this Databook, Motorola Inc., 1994.

Motorola MPA1000 Design System
Synopsys

High Level Design from Synopsys

Many thousands of designers are developing their next generation systems and ICs in record time, while increasing price performance and significantly lowering development risk. They are using design tools supplied by Synopsys. Any high level design has three parts: technology independent specification, constraint driven implementation and gate level verification. The designer can begin to take advantage of design for reuse by incorporating behavioral models for predefined high level building blocks.

Synopsys takes the designer through architectural or RTL models that serve as the functional specification of the chips. The tools generate an optimized gate–level specification from the design constraint and the high level specification. The last step includes gate–level verification in the same simulation environment that was used earlier in the process.

4

Simulation

- State–of–the art interpreted simulation
- High speed compiled simulation
- Fast gate level simulation

Synthesis

- More popular than all other synthesis tools combined
- Translation and optimization for VHDL and Verilog(R),
- Floorplan Manager, allows high level design while synthesizing to actual floorplan information.

Test

- Ensures predictable test results, saving months of cycle time
- Full or partial scan insertion
- Integration of design for test with ATPG software
- Provides for automated boundary scan IEEE 1149.1

Services

Synopsys offer a complete range of services to help the designer get started including "Jumpstart" installation and a number of training classes. Synopsys also offers design consulting services, and the ACES program allows the user to receive the same training a Synopsys field engineer receives.

For further information, please contact:

Synopsys Inc.
700 East Middlefield Road
Mountain View, California 94043
Tel: (415) 962–5000

Motorola MPA1000 Design System
Exemplar CORE™ Logic Synthesis System

Product Overview

Exemplar Logic's CORE Logic Synthesis System allows FPGA designers to define, synthesize, and simulate logic designs and target the results to a wide range of FPGA, CPLD and ASIC technologies. Designs can be entered in industry standard formats; including Verilog–HDL, VHDL, OpenABEL, and PALASM. Designs can also be entered in mixed mode and combined into a single technology.

- Robust support for VHDL and Verilog–HDL input languages. VHDL is IEEE Std. 1076, 1164, and VHDL–93 compliant
- *Customized Synthesis* insures optimized results for the target technology
- RTL–VHDL and Verilog–HDL output supported for post–optimization simulation
- Production proven
- Output netlists are compatible with downstream tools such as NeoCAD and vendor proprietary place and route tools and filters.

CORE performs logic synthesis, optimization, and technology mapping into EDIF netlists compatible with NeoCAD's FPGA Foundry. A mapping library co–developed with Motorola insures designs synthesized by CORE can be successfully mapped into Motorola's MPA1000 device family.

CORE translates abstract logic descriptions into proprietary netlists which are thoroughly tested to be compatible with downstream tools; including simulators and Place–and–Routers. The optimization performed is customized to each supported target technology to insure "Best–in–Class" results. Area or speed trade–offs can be made at run time.

For the Motorola MPA1000 device family CORE includes a mapping library that produces EDIF netlists, timing and critical path reports for each synthesis run. Nine separate optimization strategies can be run to insure excellent results for any design, no matter how complex it may be. The EDIF netlists can be directly utilized to physically implement the design using Motorola supplied APR tools or NeoCAD's FPGA Foundry for place and route.

Module Generation (MODGEN™) incorporates special libraries for data path logic such as adders, comparators and multipliers. By analyzing the source VHDL and Verilog, MODGEN calls these libraries to work with the random logic synthesis routines to produce better results for data path applications than ordinary synthesis tools.

Because of its ability to read a variety of industry standard and proprietary netlist formats, designers can also retarget designs between FPGA, CPLD and ASIC technologies. This increases the utility of the tool and completes the simple, but elegant and productive Exemplar design Environment.

Platforms

PC, DOS and Windows, SUN and HP UNIX.

For further information, please contact:

Exemplar Logic Inc.
2550 Ninth Street, Suite 102
Berkeley, California 94710
Tel: (510) 849–0937
Fax: (510) 849–9935
email: info@exemplar.com

FPGA Software Product List

View*logic* Schematic Capture and Simulation Products

Part Number	Description
MOTO/3150	View*logic* PROCapture Schematic/Symbol Editor (PC version running under DOS/Windows)
MOTO/6150/NL	View*logic* Viewdraw Schematic/Symbol Editor (UNIX version for PowerView: Node Locked)
MOTO/6150/NW	View*logic* Viewdraw Schematic/Symbol Editor (UNIX version for PowerView: Floating Network License)
MOTO/3350H	View*logic* PROSIM Logic Simulator, 20K Gate Limit and PROWave Waveform Processor (PC version running under DOS/Windows)
MOTO/6350H/NL	Viewsim Logic Simulator, 20K Gate Limit and Viewwave Waveform Processor (UNIX version for PowerView: Node Locked)
MOTO/6350H/NW	Viewsim Logic Simulator, 20K Gate Limit and Viewwave Waveform Processor (UNIX version for PowerView: Floating Network License)
MOTO/3PKG/BASIC	Includes the MOTO/3150 and MOTO3350H
MOTO/6PKG/NL	Includes the MOTO/6150/NL and MOTO/6350H/NL
MOTO/6PKG/NW	Includes the MOTO/6150/NW and MOTO/6350H/NW

4

NeoCAD Place and Route Software

FPGA Design Systems (MPA1000 Development System)

Part Number	Description
NFD/M1K/P	PC
NFD/M1K/SN	Sun Workstation (Node Locked)
NFD/M1K/SF	Sun Workstation (Floating)
NFD/M1K/HN	HP Workstation (Node Locked)
NFD/M1K/HF	HP Workstation (Floating)

The MPA1000 Development Systems (NFV/M1K/xx) contain:

1. NeoLPM libraries and provisions for netlist translation, physical mapping and optimization routines, delay based place and route, graphical editing timing analysis and device programming.
2. Timing and frequency driven place and route capabilities.
3. One of the following NeoCAD CAE Vendor Integration Kits including NeoLPM, NeoCAD's LPM Complaint Device–Independent library.
   - View*logic* Integration Kit (NFK/V00/xx)
   - Mentor Graphics Integration Kit (NFK/M00/xx)
   - Synopsys Integration Kit (NFK/S00/SN)
   - Verilog Simulation Integration Kit (NV/xxx/xx)
4. A down load cable (NF2/000), Evaluation System (NFE2/000), and Electronic On–Line documentation.

FPGA Foundry Modules (MPA1000 Family Module)

Part Number	Desription
NFV/M1K/P	PC
NFV/M1K/SN	Sun Workstation (Node Locked)
NFV/M1K/SF	Sun Workstation (Floating)
NFV/M1K/HN	HP Workstation (Node Locked)
NFV/M1K/HF	HP Workstation (Floating)

When combined with a compatible NeoCAD FPGA Design System, the MPA1000 Family Module provides physical mapping/optimization, place and route, graphical editing, timing analysis and device programming.

Mentor Integration Kit

Part Number	Description
NFK/M00/SN	Sun Workstation (Node Locked)
NFK/M00/SF	Sun Workstation (Floating)
NFK/M00/HN	HP Workstation (Node Locked)
NFK/M00/HF	HP Workstation (Floating)

Provides complete Mentor Graphics 8.x integration. The kit includes NeoLPM, NeoCAD's generic library for schematic capture and front–end functional simulation.

View*logic* Integration Kit

Part Number	Description
NFK/V00/P	PC
NFK/V00/SN	Sun Workstation (Node Locked)
NFK/V00/SF	Sun Workstation (Floating)
NFK/V00/HN	HP Workstation (Node Locked)
NFK/V00/HF	HP Workstation (Floating)

Provides complete View*logic* PRO Series, Workview Plus and Powerview integration. The kit includes NeoLPM, NeoCAD's generic library for schematic capture and front–end functional simulation.

Synopsys High Level Design Link

Part Number	Description
NFS/M1K/SN	Sun Workstation (Node Locked)
NFS/M1K/SF	Sun Workstation (Floating)
NFS/M1K/HN	HP Workstation (Node Locked)
NFS/M1K/HF	HP Workstation (Floating)

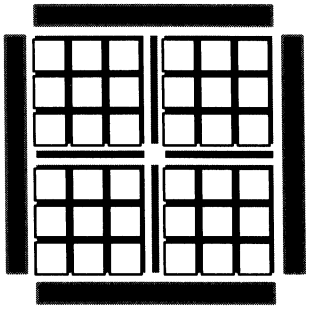
Provides timing constraint interface from Synopsys FPGA compiler and Design Compiler to Timing Wizard. When used in conjunction with an applicable optimization library, it provides complete integration with the Synopsys synthesis environment.

Verilog Simulation Integration Kit — Planned introduction in 1995.

Miscellaneous Product Options

Part Number	Description
NF1/000	Documentation Set
NF2/000	Download Cable
NFE1/000	Evaluation Board – 181PGA
NFE2/000	Evaluation Board – 84PLCC

4



Field Programmable Gate Arrays

This section contains a high level design philosophy that allows FPGAs to be converted to Gate Arrays and an actual application using the methodology.

Application Notes 5

CONTENTS

APPLICATION NOTE
How to Convert FPGA to Gate Arrays With a Minimum of Pain

by Fred Zlotnick, Marketing Manager, Motorola FPGA

Introduction

Many designers believe they need to convert their FPGAs to hard–wired ASICs thinking they will save money in the process. Some look at the per unit cost, which in 1994 is a factor of 8 to 10 times, if you calculate on a per gate basis. This article will deal with the factors involved in making the choice, and if the decision to convert is made, a strategy that will aid the reader with a flow and a methodology.

FPGA vs. ASICs

By now almost everyone knows the benefits of using a programmable product. Successful end equipment companies will innovate products and constantly obsolete themselves. Smith and Reinertsen [1] explain the effects of being late with a product in the market place. In the case where the product is vying for market share competing with other companies with similar strengths, their assumption is that a company would lose opportunity if they were late, and not achieve the same results as if they would have been on time. The product would still end its life at its normal time in the product life cycle (Figure 5-1). The loss in revenue is the shaded area (sales units/month multiplied by time). FPGAs can save time in the development cycle of a new product by:

5

1. No waiting time to get silicon from ASIC vendor
2. Development time is shortened, since there is no need for writing test vectors
3. Complex interactions of hardware and software can be worked on sooner, changes to hardware can be tried almost instantly.
4. Design cycle may be shortened, since the risk of failure of the first design is low. Also, there is no NRE and turn around time to fix a problem is quick.

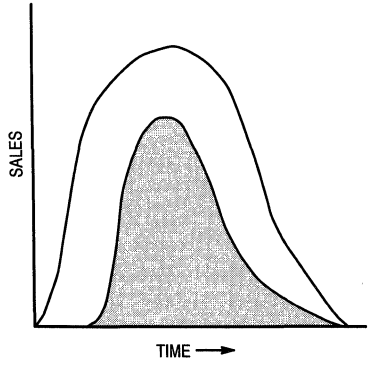


Figure 5-1. Life Cycle

Many designers have adopted the strategy that FPGAs are viable for prototyping and early market entry, but believe they must migrate into hard–wired ASICs as soon as possible. There are only four good reasons to convert an FPGA to a Metal Programmed Gate Array (MPGA):

1. FPGA speed is too slow
2. The volume is high enough that it is clearly worthwhile to switch
3. Density or Power: one ASIC will replace several FPGAs resulting in lower power and improved density
4. Combinations of the above, i.e. speed/density/ lower unit cost.

The critical issue in replacing an FPGA with an ASIC is:

Will the volume really justify the initial NRE expenditure?

FPGA pricing is such that it takes about 5000 units to justify the typical NRE. Even at that the user just breaks even. The trend is expected reduce the gap between FPGAs and MPGAs. For the most part, many MPGAs already have more usable gates relative to I/O than they need. When higher density technology is commonplace, it is likely the only concern will be I/O. The number of gates on a device will far exceed the requirements for most applications. As we approach geometries <0.5µ both FPGAs and MPGAs will contain enough gates for most applications to be I/O bound. If this is the case then cost would not be a motivating factor, i.e. the user pays for square millimeters of silicon, not what is on it.

Defining the ASIC

Assuming one or more of the criteria are met and the user must convert from FPGA to MPGA then he should plan a priori that he wants to convert to an MPGA. The first thing that should be established is a pure synchronous design. Asynchronous designs are nearly impossible for ASIC conversion. Given that we have a synchronous design, the user should determine if he wants simply to convert from one FPGA to one ASIC. Although this is certainly possible,the user should scrutinize the economics of this decision. The user incurs an NRE charge for each ASIC he chooses to convert. We highly recommend that the user takes advantage of the larger gate arrays that are available and perhaps convert 2 or more FPGAs into a gate. Motorola's HDC series and H4C series of gate arrays offer densities that can easily incorporate two or more of the largest planned products. This holds for competitive FPGAs as well. Since Motorola and many other suppliers cannot guarantee a perfect correspondence of pins the multiple FPGA to one MPGA continues to make sense. Not only is

the density improved and NRE costs lower, but the cost of laying out the board again should no longer be a factor. The problem of I/O shows up, even with today's 0.8µ products. If we were to try to match a 160 pin MPA1036 to an HDC, the smallest device that is compatible is the 27,000 gate HDC027. This part has 8 fixed power and ground pins. The HDC series has similar I/O structures, but somewhat higher drive capabilities. It is also important to note that Motorola requires the user to supply test vectors for the gate arrays in order to assure the quality of the product.

Partitioning

If the user is going to convert several FPGAs to an ASIC the board design will be different and therefore not critical that the part matches pin for pin. It is recommended the designer partitions the design into functional subsets that have some meaning to him. There are a few third–party software tools that take a large design and divide the logic into smaller pieces to target to multiple FPGAs. Until we have more experience with these tools it is recommended that the user partition the design manually.

EDA Tools

If the user is familiar with an HDL language such as Verilog–HDL or VHDL, this makes a very convenient, technology independent way of describing the logic. The user can describe the logic in either of these languages and compile the logic into the Motorola MPA series FPGAs using a tool such as Synopsys, Exemplar, Autologic or Viewsyn. These tools will target the FPGA through its library and produce a net–list. The NeoCAD tool will place and route the logic and provide back annotation to allow the after placement timing to be evaluated. After the FPGAs are successfully put into place, there is a chance to evaluate all the features and make any changes necessary, the FPGA can be rerun and the user has working prototypes in hours. If the same HDL is now aimed at the Motorola ASIC family with the appropriate Motorola tools, the user should have a retargeted FPGA to ASIC. If he had the foresight to manage several FPGAs to one ASIC up front then his efforts may be rewarded with significantly lower unit cost, and a possibility of improved system performance.

Open Tools

Motorola endeavors to use third–party open software tools wherever possible. The following is a list of third party EDA tools and where they may be used:

1. **Mentor Graphics** — Schematic Capture, Simulation, Logic Synthesis*
2. **Viewlogic** — Schematic Capture, Simulation, Logic Synthesis*
3. **Synopsys** — Logic Synthesis, Simulation
4. **Exemplar** — Logic Synthesis
5. **NeoCAD** — Timing Driven Place and Route, Back Annotation

* Planned support – Late 1994

If the designer truly wants to accomplish a conversion, he should try to remain "technology independent" throughout as much of the design as possible. VHDL or Verilog–HDL are both technology independent representations of the circuit. Logic synthesis tools are highly recommended. It should be noted that the logic synthesis tools were developed for fine grain architectures as found in MPGAs, and will do a good job synthesizing the MPA family. Schematics or Boolean equations, can be entered as well, through the use of these tools. After the technology independent portion is completed, the designer can use the NeoCAD place and route tools to target to the Motorola MPA family. The tools will provide a back annotation file to allow post place and route synthesis and bit stream generation for testing the part. When the design is stable i.e. no more "fixes" or marketing changes, the user can then go through the gate array procedure. If he is using logic synthesis tools such as Synopsys, he can go back to his original VHDL or Verilog file, recompile using HDC libraries, and throughout he use of the rest of the OACS™ tool set, can assure a high chance for success.

How not to succeed with an ASIC conversion

If the designer tries to design a system based on Motorola or competitive FPGAs and does any or all of the following, his chances for success are small

a. Asynchronous design
b. Try to "hand off" conversion process to another engineer or to the ASIC vendor
c. Optimize the FPGA taking advantage of peculiarities of the logic cell block
d. Have little or no understanding of test vector generation
e. Underestimate the task

Note on OACS™

This is a tool suite that enables the user to develop MPGAs and have a high confidence that the product will have first pass success. It provides design rule checks, automatic test vector generation and provides the "glue" for the designer's third party tools to bind them properly to the Motorola ASICS. The reader should contact his nearest Motorola sales office, for further information on this software product.

References

[1] Smith and Reinersten, *Designing Products in Half the Time*, Van Nostraud Reinhold, pp 32–40.

5

APPLICATION NOTE
68030 DRAM Controller Design Using Verilog HDL Tools

by Phil Rauba, Motorola Field Applications Engineer

Purpose

This article is intended to give a hardware engineer insight into the design methodology of using the Verilog Hardware Descriptive Language (HDL), targeting Motorola's Field Programmable Gate Array (FPGA) and H4C gate array families. The advantage of using an HDL, such as Verilog, is the ability to retarget the design to other device technologies, by only resynthesizing the design description. A 68030 Dynamic Ram Controller design was used to demonstrate the portability of the Verilog language, and included all of the circuits necessary to interface DRAM to a 68030 microprocessor including: memory decoding, STERM generation, refresh request generation, CAS before RAS refresh, burst address sequencing, DRAM address multiplexing, and bus error timeout.

Design Methodology

The DRAM Logic was designed with a synchronous state machine design technique and described using the Verilog Hardware Descriptive Language, with the intent of providing a portable and easily maintainable design. The design tools used for this project are listed in Appendix A. The steps included in the design process include the system block diagram definition, state diagram generation, Verilog HDL logic definition, Verilog logic simulator verification, Verilog logic synthesis, place and route, and final simulation.

Two design directories were maintained for this design, one for the simulation code and the other for the synthesis code. A flow diagram of the design methodology is shown in Figure 5-2. A separate simulation model was used for the logic verification of the design and included a simple subset of the bus controller features of the 68030 microprocessor, modeled with the behavioral modeling capabilities of Verilog HDL. As the Verilog design progressed, a simulation model was used as the original coding for the DRAM controller. Each individual submodule of the design was verified via simulation and then cut and pasted into the synthesis code.

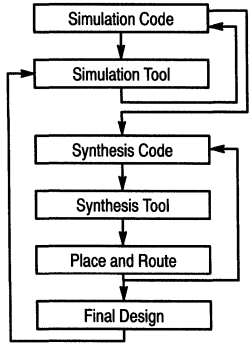


Figure 5-2. Verilog Design Method

The Verilog design used a hierarchical module development methodology, which partitioned the design into eight submodules and instantiated each of the submodules into the design through a top module description designated as module glue68k. Submodules were easily modified and resynthesized as needed, saving considerable time by not having to resynthesize the entire design.

Since different tools were used for the logic synthesis, when targeting to an FPGA and gate array, different methods were used during the synthesis process. For FPGA development, the Exemplar tool was used for synthesizing each of the submodules individually. When synthesizing the top glue68k module, two passes were needed by the Exemplar tools, one to generate submodule connectivity and the other to read and reformat the Verilog netlist. Empty submodules are instantiated in the design during the first pass of the Exemplar logic synthesizer with a Verilog netlist being generated. The Verilog glue68k netlist is then edited to add the links to the submodules by using the include command, referencing each of the submodule's file pathname. A final netlist is output from the second pass of the synthesizer, which reads the eight Verilog netlists from the links in the top module, and reformats the file to an EDIF netlist.

For gate array development, Synopsys was used for synthesis of the design. Each of the eight submodules and the top module, glue68k were read into Synopsys and synthesized all at one time without having the need to use the include command.

The EDIF netlist is used by the FPGA and gate array place and route tools to generate the final design files.

System Description

Bursting is a feature in the newer generation of CISC/RISC microprocessors that is comprised of a memory access of four long words of 32 bits each. The DRAM burst cycle is initiated by first generating a RAS cycle access and a CAS cycle access for the first long word, and then fetching the next three long words by generating only CAS cycles thereafter. The intention of the burst cycle is to divide the RAS cycle generation overhead of the first access amongst all four longword fetches; thereby, providing an overall access performance improvement as compared to single RAS generation for each longword.

The system block diagram is shown in Figure 5-3 and includes a 68030 microprocessor, a 16MByte dram array using 4Mx4 DRAMs, data bus drivers, and the FPGA (or gate array). The FPGA provides all of the DRAM interface circuitry needed to support 68030 bursting.

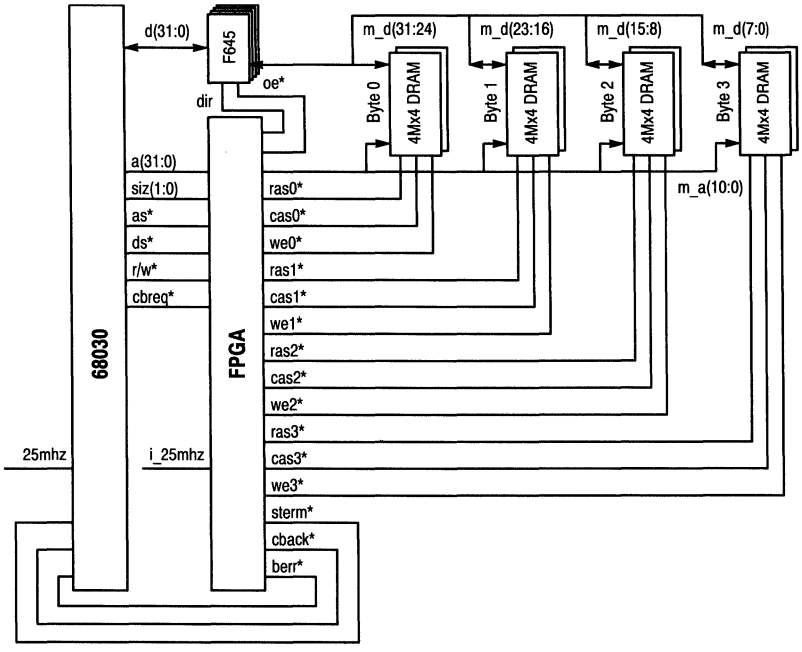


Figure 5-3. DRAM Controller

5

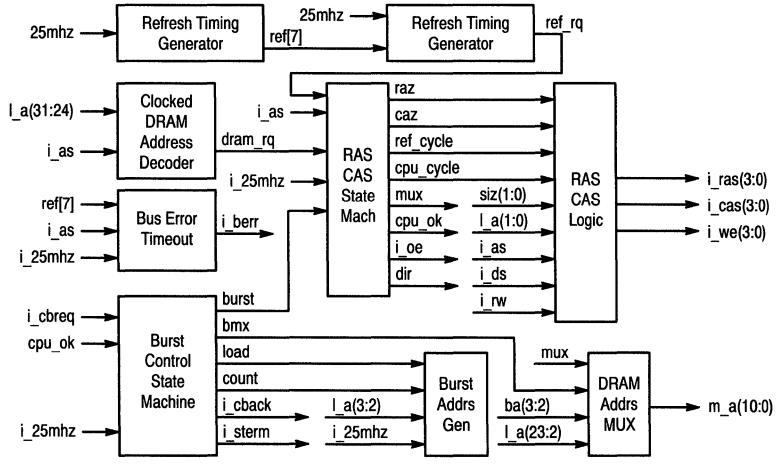


Figure 5-4. FPGA DRAM Controller

FPGA Functional Description

The block diagram functional description of the FPGA is shown in Figure 5-4 and shows all of the modules within the design, including the refresh timing generator, the refresh request state machine, the address decoder, the RAS/CAS state machine, the RAS/CAS decoder logic, the burst control state machine, the burst address generator, the DRAM address multiplexer, and the bus error timeout state machine.

FPGA Timing Synchronization

As indicated in the FPGA functional block diagram Figure 5-4, the main clock for all the state machines is the inverted 25MHz clock to the 68030. Since all of the output timing out of the 68030 is referenced to the falling edge of the processor's 25MHz clock, the clock is inverted and is used for clocking the FPGA's internal registers. A delay line (not shown) will be needed for moving the assertion point of the address strobe signal with respect to the internal clock skew within the target device to prevent flip–flop metastable conditions. The delay line value will be dependent on the actual clock skew within the FPGA or gate array.

Refresh Timing Generator

The refresh timing generator provides a 97.656 KHz refresh request square wave with a period of 10.24 usec for the refresh request state machine. The generator is comprised of a eight bit free running up counter with a 25 MHz clock source.

5

Refresh Request State Machine

The refresh state machine receives the 97.656KHz refresh square wave and generates a ref_rq signal to the RAS/CAS state machine. The refresh state machine requests a refresh cycle only once when ref[7] is asserted high and inhibits the request after the RAS/CAS state machine has initiated a CAS before RAS refresh cycle.

Figure 5-8 shows the start of a refresh DRAM cycle initiated by the rising edge of ref[7] and by the assertion of ref_rq to the RAS/CAS Controller. The RAS/CAS Controller generates the timing for a CAS before RAS refresh cycle with the refresh request state machine starting to sequence through the request operation. At the end of the refresh cycle the refresh request state machine is looking for negation of ref[7] and remains in REFEND state.

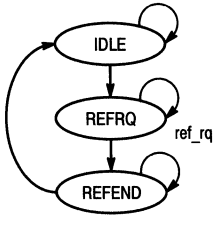


Figure 5-5. Refresh Request State Diagram

Synchronized DRAM Address Decode

The DRAM Address decoder is used to decode the 68030 address 01xxxxxh with qualification of address strobe to generate a access request to the RAS/CAS state machine. An intermediate gating equation is used with address strobe to negate the dram_rq as soon as address strobe is negated. **figure** 5 shows the dram_rq timing.

RAS/CAS State Machine

The RAS/CAS State machine arbitrates between refresh requests and 68030 access requests via signals, ref_rq and dram_rq respectively. The arbitration between the two requests occurs in the IDLE state, where refresh has the highest priority. If a refresh request is pending the state machine will take the refresh branch and generate a CAS before RAS refresh timing sequence.

If a DRAM request is pending from the 68030 and a refresh request is not present, the state machine will take the 68030 access branch and generate the RAS, MUX, and CAS timing for a random access into the DRAM array. The assertion of RAS latches the row address into the DRAMs, the MUX signal will present the column address to the DRAM array, and the CAS signal will then latch the column address into the DRAMs. The signal cpu_ok will indicate to the burst controller to start wait state generation. If a burst request sequence is requested by the burst controller via the signal burst, the RAS/CAS state machine will sequence through the burst control states. For a full four long word burst, the burst address generator will provide the column addresses for each of the long word accesses. The RAS/CAS controller state machine will exit bursting upon the negation of address strobe, and has the capability of exiting a burst upon a premature ending of a full four long word burst.

RAS/CAS Logic

The RAS/CAS logic is comprised of combinational logic that encodes the CAS signals for selecting which byte lanes of the DRAM array that are going to be accessed during a cycle. For a CPU write access, the logic supports the misalignment capabilities of the 68030, providing CAS signals only to the bytes of the DRAM array that will be accessed for the write operation. For a CPU read cycle all of the CAS signals will be asserted. During refresh cycles all of the CAS and RAS lines will be asserted for the DRAM.

Burst Control State Machine

The burst control state machine provides all of the bursting control for a 68030 DRAM access and is synchronized to the RAS/CAS controller. Upon the receipt of a dram_rq, the RAS/CAS controller will generate RAS and CAS timing to the DRAM array and will assert the signal cpu_ok, indicating to the burst controller state machine to start a burst cycle. The burst controller will leave an idle state and assert the i_back signal indicating a synchronous burst access. The burst controller will then insert wait states during the burst operation and be responsible for asserting i_sterm indicating the availability of DRAM data. The burst controller will also generate the counting and load controls for the burst address generator

and provide the burst multiplexing control to the DRAM address multiplexer.

Burst Address Generator

The burst address generator provides the two least significant bits of the DRAM address during a 68030 burst cycle. The burst controller will initiate the loading of the first burst address into the burst address generator and then control the incrementing of the addresses for the next three long word accesses of the burst cycle.

The burst address generator sequences through the long word addresses, which are generated from the ba(3:2) signals. Entry into the counter state machine can occur at any state and will be defined as the starting 68030 starting address plus one. During the first long word access of the burst, the first address will be supplied by the 68030; the next three long word addresses will be supplied by the burst address generator. After the first 68030 address, the address generator will enter the state of the next address from the load signal from the burst controller. The burst address generator will then be incremented two more times for the next two long word addresses.

DRAM Address MUX

The DRAM address MUX provides the row and column addresses on a eleven bit multiplexed address bus to the DRAM array. The 68030 provides the row address to the DRAM array, and the column address of the first long word access of a burst cycle. The two least significant bits of the column address will be supplied by the burst address generator for the next three long word accesses in the burst cycle. Gating of the addresses onto the DRAM multiplexed address bus is controlled by both the burst controller and the RAS/CAS controller. The DRAM address MUX generates a 68030 burst access to long word address locations 01xxxxx0h to 01xxxxxCh, when the starting 68030 address is 01xxxxx0h.

Bus Error Timeout

A bus error watch dog time out function is provided by the DRAM controller to keep the 68030 from locking up due to accesses into unused memory. The bus error timeout controller monitors the assertion of address strobe and will generate a bus error to the 68030 if it has kept address strobe asserted from 40.96 usec to 46.08 usec. This timeout may vary depending on where the 68030 started a memory access in relationship to ref[7] of the refresh timing generator. The bus error timeout controller monitors ref[7] for its state transitions, while watching the assertion of address strobe. If address strobe is negated before reaching the state NOACK of the bus error timeout state machine, a bus error will not generated.

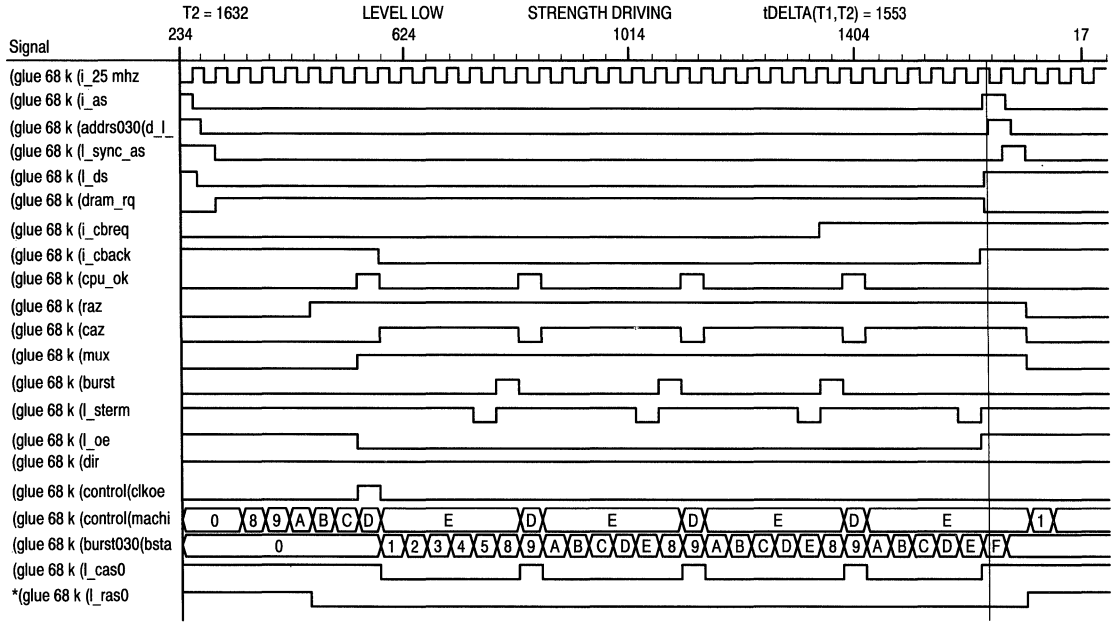


Figure 5-6. DRAM Burst Timing

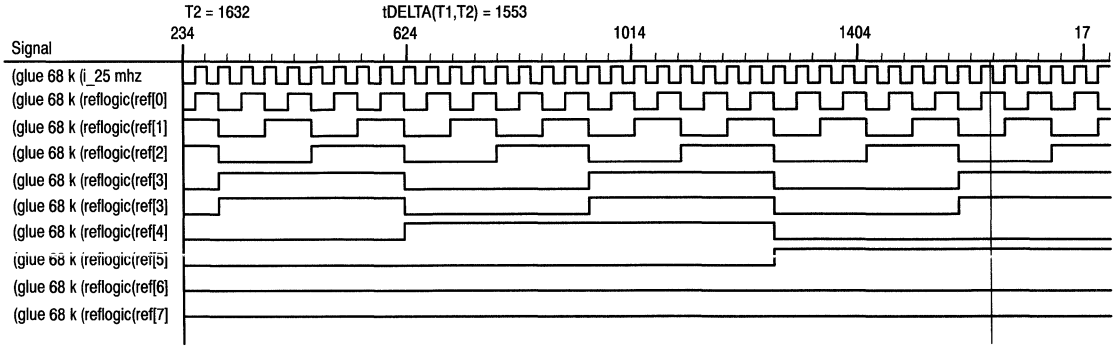


Figure 5-7. Refresh Counter Timing

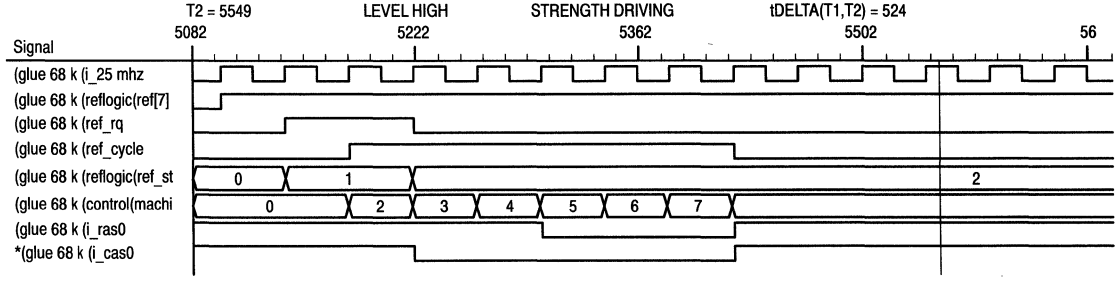


Figure 5-8. Refresh Timing

5

System Timing

The system timing is shown in Figure 5-6 and gives the overall operation of the modules within the DRAM design for a DRAM array access. The diagram shows the logical implementation of the design with zero propagation delay and is meant to give a relationship of the signal handshaking between the submodules of the design. The system timing diagram shows a four long word burst read access to the DRAM array and is comprised of a 14–7–7–7 burst for a total of 35 cycles. The design has not been optimized for speed at this time, with the intent of generating a reasonable amount of logic for timing verification targeting lower cost designs using slower DRAMs.

Figure 5-6 to Figure 5-87 timing diagrams are logical simulations of the design copied from the SILOS III timing analyzer and do not include final place and route timings.

Verilog Coding Example

The following Verilog synthesis code describes the refresh module:

```
module refresh (i_25mhz, ref_cycle,
                timerclk, ref_rq);
input i_25mhz;
output timerclk;
reg [7:0] ref;
assign timerclk = ref[7];
always @ (posedge i_25mhz)
begin
      ref[0] = ~ref[0];
      ref[1] = ref[1] ^ ref[0];
      ref[2] = ref[2] ^ (&ref[1:0]);
      ref[3] = ref[3] ^ (&ref[2:0]);
      ref[4] = ref[4] ^ (&ref[3:0]);
      ref[5] = ref[5] ^ (&ref[4:0]);
      ref[6] = ref[6] ^ (&ref[5:0]);
      ref[7] = ref[7] ^ (&ref[6:0]);
end
// Refresh request state machine
input ref_cycle;
reg [1:0] ref_states;
output ref_rq;
reg ref_rq;
parameter IDLE = 'b00; // idle state
parameter REFRQ = 'b01; // assert ref rqst
parameter REFEND = 'b10; // wait for end
always @ (posedge i_25mhz)
begin
      case (ref_states)
      IDLE:begin
            if (ref[7])
                  begin
                  ref_states = REFRQ; ref_rq = 1;
                  end
            if (~ref[7])
                  begin
                  ref_states = IDLE; ref_rq = 0;
                  end
            end
      REFRQ:begin
            if (ref_cycle)
                  begin
                  ref_states = REFEND; ref_rq = 0;
                  end
            if (~ref_cycle)
                  begin
                  ref_states = REFRQ; ref_rq = 1;
                  end
            end
      REFEND:begin
            if (~ref[7])
                  begin
                  ref_states = IDLE; ref_rq = 0;
                  end
            if (ref[7])
                  begin
                  ref_states = REFEND; ref_rq = 0;
                  end
            end
      endcase
end
endmodule
```

FPGA – Verilog Simulation

As indicated in Figure 5-9, a separate Verilog model was used for simulation and included a clock module for generating a system clock and a modified address decoding module for generating the timing for a 68030 burst cycle. Figure 5-6 to Figure 5-8 show the results of the simulation, which was run from 0ns to 60,000ns. The following Verilog simulation code was used for generating a 25MHz free running clock:

```
module clk25 (clock);
output clock;
reg clock;
initial
      #5 clock = 0;
always
      #20 clock = ~clock;
endmodule
```

The primary reason for using separate simulation code, was that the Verilog initial statements and simulation time # statements, as shown in the clock module example, were not supported by the Exemplar tool and caused syntax errors during synthesizes.

The decoding module was modified in the simulation code to include vectors for generating the 68030 bus signal timing for a burst read cycle. Once the simulation code was verified logically by the SILOS III simulator and analyzer, the code was determined to be free from syntax errors and matched the expected timing for the design. Note that at this point, the design has not been verified with the gate and path delays generated from the final place and route.

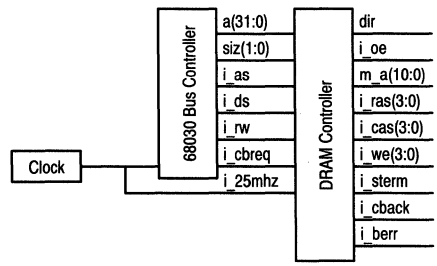


Figure 5-9. Simulation Model

FPGA — Exemplar Synthesis

The refresh submodule example is synthesized with the DOS command "fpga refresh.v refresh.vg –target=moto –save –macro". The save option stores all the optimization passes of the logic synthesizer allowing the user to select the best pass based on timing and cell count size. The

macro option is used for not assigning IO pads to the inputs and outputs of the submodules, reserving IO pad assignments to the top module. Each submodule of the design is synthesized separately using the same command, but with different filenames that are identical to the submodule name. The top module glue68k is synthesized with "fpga glue68k.v glue68k.vg –target=moto –pass=2". The glue68k.vg Verilog netlist is edited manually, adding include commands to the end of the file to read in all the Verilog netlists into the top hierarchical module during the Exemplar reformat pass. The include command "'include "c:\fpga\refresh.vg" reads in the refresh submodule into the glue68k.vg module when the DOS command "fpga glue68k.vg glue68k.edif –source=moto –target=moto –effort=reformat", is executed in the fpga directory. The final design resulted in 360 cells including 254 ANDs, 30 CARRY1s, 30 CARRY2s, 43 DFFs, and 64 IOBs, when targeted to a Motorola FPGA.

The types of gates that were synthesized by the Exemplar tool for the DRAM Controller design included:

AND2	DFC	OR2
AND2I1	INV	OR2I2
AND2I2	MUXI2	ORI1
BUFF	MUXI2I1	XOR2
DFA	MUXI2I2	XOR2I1
DFAD	MUXIA2I1	

The synthesis times for the modules varied with the complexity of the logic. For instance the dram module, which is a fairly complex state machine design, took longer to synthesize than the refresh module, which contained simpler circuitry. Synthesis times (–save option) for the dram module are shown:

Pass	Cells	Levels	Min:Sec
1	152	23	08:48
2	92	6	02:10
3	154	23	05:00
4	87	14	02:08
5	152	22	03:25
6	87	14	01:31
7	154	23	04:05
8	87	14	01:47
9	186	21	05:55
10	99	8	27:03
11	86	14	25:09

The passes of the Exemplar synthesizer are related to eleven different types of optimization algorithms. The best pass for the dram module, based upon timing, is pass 2, with an estimated gate delay level of 6. In general all of the modules synthesized in this design, had pass two consistently generate the least amount of level delays. One may save considerable CPU time by specifying that a particular pass be executed by the Exemplar tool. If only, for example, the –pass=2 option was used, the synthesis time would only be two minutes as compared to one and a half hours.

FPGA – NeoCAD Place and Route

The design needs to be processed by three DOS command steps, before the place and route tool is invoked. First, the glue68k.edi file is converted to a NeoCAD binary file with "edif2ngd –l –lpm glue68k.edf glue68k.ngo". Second, the glue68k.ngo is converted to a NeoCAD primitives data base with "ngdbuild –v motorola:mpa1000 –p \neocad\data\motorola\mpa1000\macro glue68k.ngo glue68k.ngd". Third, the glue68k.ngd is mapped to a Motorola MPA1036, 160 pin device with "mmap –n –u –p 1036PQ160 glue68k.ngd glue68k.ncd".

The final place and route was invoked using the NeoCAD's PAR graphical user interface, setting the Placement Cost Tables option to ten, the Router Pass option to ten, and saving the best three routes.

By analyzing the .dly files of the three saved designs, the designer can pick the best design based on timing and placement. NeoCAD's EPIC editor was used to verify the final timing of the three .par design files by selecting critical pin paths and using the Delay tool to calculate a total path delay, which includes component and wire delays. The designer also has the option of invoking a timing driven router, by using the .prf file and specifying the operating frequency desired.

H4C Gate Array

The Verilog netlist files for the design were transferred on a DOS disk to a Sun workstation. Synopsys was used to read in the top hierarchical glue68k module and each of the submodules of the design. The design was synthesized and targeted to the H4C gate array family without any errors. The combinational area of the design was 446 and the noncombinational area was 344 (43 flip–flops using 8 gates per flip–flops) with a total used area of 790.

The types of gates generated by Synopsys include:

AND2	INV2	NOR8H
AND2H	INVB	OA211H
AND3	MUX2A	OA21H
AO22H	MUX2I	OA22H
AOI22H	MUX2IH	OAI211H
DFFP	NAN2	OAI22H
DFFRP	NAN3	ONDAI22H
EXNORA	NAN4	OR2
EXORA	NOR2B	OR3
INV	NOR2H	OR4

One observation of the synthesized design was that the Synopsys synthesizer added buffers to heavily loaded signals to minimize the wire delays and edge rates in the design. Another observation indicated that the gates that were generated for the Motorola FPGA and H4C gate array were similar in their fine grain structure.

At this point the design was not simulated with estimated timing, and was not placed and routed. The purpose here was to verify the portability of the Verilog design and check for syntax errors. Final place and route for the gate array was held off to wait for the verification of the final design with the Motorola MPA1036 FPGA.

Final Simulation

The final placed and routed FPGA design has not yet been verified via the Verilog simulator, pending the release (4Q94) of the NeoCAD tool set, which will feature the enhancement of backannotating to the Verilog Standard Delay Format (SDF).

Appendix A – Design Tools The FPGA design tools were selected for a 486 PC Platform and included SIMUCAD, Inc.'s SILOS III Verilog HDL Logic and Fault Simulation Environment, Exemplar's CORE–TD–DOS PC Topdown Verilog Synthesizer, and NeoCAD's FPGA Place and Router. The PC was upgraded to 32MBytes of DRAM memory, of which 16MBytes were the minimum required to run the Exemplar software. A CD–ROM drive was required for loading the NeoCAD software.

For targeting H4C gate arrays, the design development tool kit was Motorola's Open Architecture CAD System (OACS). Synopsys was used for Verilog logic synthesis on a Sun platform in one of Motorola's ASIC design centers.

5

MOTOROLA DISTRIBUTOR AND WORLDWIDE SALES OFFICES

AUTHORIZED NORTH AMERICAN DISTRIBUTORS

UNITED STATES

ALABAMA

Huntsville
- Arrow/Schweber Electronics .... (205)837-6955
- Future Electronics ..... (205)830-2322
- Hamilton Hallmark ..... (205)837-8700
- Newark ..... (205)837-9091
- Time Electronics 1-800-789-TIME
- Wyle Laboratories ..... (205)830–1119

ARIZONA

Phoenix
- Future Electronics ..... (602)968-7140
- Hamilton Hallmark ..... (602)437-1200
- Newark Electronics ..... (602)864-9905
- Wyle Laboratories ..... (602)437-2088

Tempe
- Arrow/Schweber Electronics .... (602)431-0030
- Time Electronics 1-800-789-TIME

CALIFORNIA

Agoura Hills
- Time Electronics Corporate ... 1-800-789-TIME

Belmont
- Richardson Electronics ..... (415)592-9225

Calabassas
- Arrow/Schweber Electronics .... (818)880-9686
- Wyle Laboratories ..... (818)880-9000

Chatsworth
- Future Electronics ..... (818)865-0040
- Time Electronics 1-800-789-TIME

Costa Mesa
- Hamilton Hallmark ..... (714)641-4100

Culver City
- Hamilton Hallmark ..... (213)558-2000

Irvine
- Arrow/Schweber Electronics ... (714)587-0404
- Future Electronics ..... (714)250-4141
- Wyle Laboratories Corporate .... (714)753-9953
- Wyle Laboratories ..... (714)863-9953

Los Angeles
- Wyle Laboratories ..... (818)880-9000

Mountain View
- Richardson Electronics ..... (415)960-6900

Orange
- Newark ..... (714)634-8224

Rocklin
- Hamilton Hallmark ..... (916)624-9781

Sacramento
- Newark ..... (916)721-1633
- Wyle Laboratories ..... (916)638-5282

San Diego
- Arrow/Schweber Electronics ... (619)565-4800
- Future Electronics ..... (619)625-2800
- Hamilton Hallmark ..... (619)571-7540
- Newark ..... (619)569-9877
- Wyle Laboratories ..... (619)565-9171

San Francisco
- Newark ..... (415)571-5300

San Jose
- Arrow/Schweber Electronics .... (408)441-9700
- Arrow/Schweber Electronics .... (408)428-6400
- Future Electronics ..... (408)434-1122

Santa Clara
- Wyle Laboratories ..... (408)727-2500

Sunnyvale
- Hamilton Hallmark ..... (408)435-3500
- Time Electronics 1-800-789-TIME

Torrance
- Time Electronics 1-800-789-TIME

Tustin
- Time Electronics 1-800-789-TIME

West Hills
- Newark ..... (818)888-3718

Woodland Hills
- Hamilton Hallmark ..... (818)594-0404
- Richardson Electronics ..... (615)594-5600

COLORADO

Lakewood
- Future Electronics ..... (303)232-2008

Colorado Springs
- Newark ..... (719)592-9494

Denver
- Newark ..... (303)757-3351

Englewood
- Arrow/Schweber Electronics .... (303)799-0258
- Hamilton Hallmark ..... (303)790-1662
- Time Electronics 1-800-789-TIME

Thornton
- Wyle Laboratories ..... (303)457-9953

CONNECTICUT

Chesire
- Future Electronics ..... (203)250-0083
- Hamilton Hallmark ..... (203)271-2844

Southbury
- Time Electronics 1-800-789-TIME

Wallingfort
- Arrow/Schweber Electronics .... (203)265-7741

Windsor
- Newark ..... (203)683-8860

FLORIDA

Altamonte Springs
- Future Electronics ..... (407)767-8414

Clearwater
- Future Electronics ..... (813)530-1222

Deerfield Beach
- Arrow/Schweber Electronics .... (305)429-8200
- Wyle Laboratories ..... (305)420-0500

Ft. Lauderdale
- Future Electronics ..... (305)436-4043
- Hamilton Hallmark ..... (305)484-5482
- Time Electronics 1-800-789-TIME

Lake Mary
- Arrow/Schweber Electronics .... (407)333-9300

Largo/Tampa/St. Petersburg
- Hamilton Hallmark ..... (813)541-7440
- Newark ..... (813)287-1578
- Wyle Laboratories ..... (813)576-3004

Orlando
- Newark ..... (407)896-8350
- Time Electronics 1-800-789-TIME

Plantation
- Newark ..... (305)424-4400

Winter Park
- Hamilton Hallmark ..... (407)657-3300
- Richardson Electronics ..... (407)644-1453

GEORGIA

Atlanta
- Time Electronics 1-800-789-TIME
- Wyle Laboratories ..... (404)441-9045

Duluth
- Arrow/Schweber Electronics .... (404)497-1300
- Hamilton Hallmark ..... (404)623-5475

Norcross
- Future Electronics ..... (404)441-7676
- Newark ..... (404)448-1300
- Wyle Laboratories ..... (404)441-9045

ILLINOIS

Addison
- Wyle Laboratories ..... (708)620-0969

Bensenville
- Hamilton Hallmark ..... (708)860-7780

Chicago
- Newark Electronics Corp. ..... (312)784-5100

Hoffman Estates
- Future Electronics ..... (708)882-1255

Itasca
- Arrow/Schweber Electronics ... (708)250-0500

LaFox
- Richardson Electronics ..... (708)208-2401

Schaumburg
- Newark ..... (708)310-8980
- Time Electronics 1-800-789-TIME

INDIANA

Indianapolis
- Arrow/Schweber Electronics .... (317)299-2071
- Hamilton Hallmark ..... (317)872-8875
- Newark ..... (317)259-0085
- Time Electronics 1-800-789-TIME

Ft. Wayne
- Newark ..... (219)484-0766

IOWA

Cedar Rapids
- Newark ..... (319)393-3800
- Time Electronics 1-800-789-TIME

KANSAS

Lenexa
- Arrow/Schweber Electronics .... (913)541-9542
- Hamilton Hallmark ..... (913)888-4747

Overland Park
- Future Electronics ..... (913)649-1531
- Newark ..... (913)677-0727
- Time Electronics 1-800-789-TIME

MARYLAND

Beltsville
- Newark ..... (301)604-1700

Columbia
- Arrow/Schweber Electronics .... (301)596-7800
- Future Electronics ..... (410)290-0600
- Hamilton Hallmark ..... (410)988-9800
- Time Electronics 1-800-789-TIME
- Wyle Laboratories ..... (410)312-4844

MASSACHUSETTS

Boston
- Arrow/Schweber Electronics .... (617)271-9953

Bolton
- Future Corporate ..... (508)779-3000

Burlington
- Wyle Laboratories ..... (617)271-9953

Methuen
- Newark ..... (508)683-0913

Norwell
- Richardson Electronics ..... (617)871-5162

Peabody
- Time Electronics 1-800-789-TIME
- Hamiltion Hallmark ..... (508)532-3701

MICHIGAN

Detroit
- Newark ..... (313)967-0600

Grand Rapids
- Future Electronics ..... (616)698-6800

Livonia
- Arrow/Schweber Electronics .... (313)462-2290
- Future Electronics ..... (313)261-5270
- Hamilton Hallmark ..... (313)347-4020
- Time Electronics 1-800-789-TIME

MINNESOTA

Bloomington
- Wyle Laboratories ..... (612)853-2280

Eden Prairie
- Arrow/Schweber Electronics .... (612)941-5280
- Future Electronics ..... (612)944-2200
- Hamilton Hallmark ..... (612)881-2600
- Time Electronics 1-800-789-TIME

Minneapolis
- Newark ..... (612)331-6350

AUTHORIZED DISTRIBUTORS – continued

UNITED STATES – continued

MINNESOTA – continued

Earth City
Hamilton Hallmark ..... (314)291-5350

MISSOURI

St. Louis
Arrow/Schweber Electronics .... (314)567-6888
Future Electronics ..... (314)469-6805
Newark ..... (314)298-2505
Time Electronics 1-800-789-TIME

NEW JERSEY

Cherry Hill
Hamilton Hallmark ..... (609)424-0100

Fairfield
Newark ..... (201)882-0300

Marlton
Arrow/Schweber Electronics .... (609)596-8000
Future Electronics ..... (609)596-4080

Pinebrook
Arrow/Schweber Electronics .... (201)227-7880
Wyle Laboratories ..... (201)882-8358

Parsippany
Future Electronics ..... (201)299-0400
Hamilton Hallmark ..... (201)515-1641

Wayne
Time Electronics 1-800-789-TIME

NEW MEXICO

Albuquerque
Alliance Electronics ..... (505)292-3360
Hamilton Hallmark ..... (505)828-1058
Newark ..... (505)828-1878

NEW YORK

Commack
Newark ..... (516)499-1216

Hauppauge
Arrow/Schweber Electronics .... (516)231-1000
Future Electronics ..... (516)234-4000
Hamilton Hallmark ..... (516)434-7400

Konkoma
Hamilton Hallmark ..... (516)737-0600

Melville
Wyle Laboratories ..... (516)293-8446

Pittsford
Newark ..... (716)381-4244

Rochester
Arrow/Schweber Electronics .... (716)427-0300
Future Electronics ..... (716)272-1120
Hamilton Hallmark ..... (716)475-9130
Richardson Electronics ..... (716)264-1100
Time Electronics 1-800-789-TIME

Rockville Centre
Richardson Electronics ..... (516)872-4400

Syracuse
Future Electronics ..... (315)451-2371
Time Electronics 1-800-789-TIME

NORTH CAROLINA

Charlotte
Future Electronics ..... (704)455-9030
Richardson Electronics ..... (704)548-9042

Greensboro
Newark ..... (919)292-7240

Raleigh
Arrow/Schweber Electronics .... (919)876-3132
Future Electronics ..... (919)790-7111
Hamilton Hallmark ..... (919)872-0712
Time Electronics 1-800-789-TIME

OHIO

Centerville
Arrow/Schweber Electronics .... (513)435-5563

Cleveland
Newark ..... (216)391-9330
Time Electronics 1-800-789-TIME

Columbus
Newark ..... (614)431-0809
Time Electronics 1-800-789-TIME

Dayton
Future Electronics ..... (513)426-0090
Hamilton Hallmark ..... (513)439-6735
Newark ..... (513)294-8980
Time Electronics 1-800-789-TIME

Mayfield Heights
Future Electronics ..... (216)449-6996

Solon
Arrow/Schweber Electronics .... (216)248-3990
Hamilton Hallmark ..... (216)498-1100

Worthington
Hamilton Hallmark ..... (614)888-3313

OKLAHOMA

Tulsa
Hamilton Hallmark ..... (918)254-6110
Newark ..... (918)252-5070

OREGON

Beaverton
Arrow/Almac Electronics Corp. .. (503)629-8090
Future Electronics ..... (503)645-9454
Hamilton Hallmark ..... (503)528-6200
Wyle Laboratories ..... (503)643-7900

Portland
Newark ..... (503)297-1984
Time Electronics 1-800-789-TIME

PENNSYLVANIA

King of Prussia
Newark ..... (215)265-0933

Mt. Laurel
Wyle Laboratories ..... (609)439-9110

Montgomeryville
Richardson Electronics ..... (215)628-0805

Philadelphia
Time Electronics 1-800-789-TIME
Wyle Laboratories ..... (609)439-9110

Pittsburgh
Arrow/Schweber Electronics .... (412)963-6807
Newark ..... (412)788-4790
Time Electronics 1-800-789-TIME

TENNESSEE

Franklin
Richardson Electronics ..... (615)791-4900

Knoxville
Newark ..... (615)588-6493

TEXAS

Austin
Arrow/Schweber Electronics .... (512)835-4180
Future Electronics ..... (512)502-0991
Hamilton Hallmark ..... (512)258-8818
Newark ..... (512)338-0287
Time Electronics 1-800-789-TIME
Wyle Laboratories ..... (512)345-8853

Carollton
Arrow/Schweber Electronics .... (214)380-6464

Dallas
Future Electronics ..... (214)437-2437
Hamilton Hallmark ..... (214)553-4300
Richardson Electronics ..... (214)239-3680
Time Electronics 1-800-789-TIME
Wyle Laboratories ..... (214)235-9953

Ft. Worth
Allied Electronics ..... (817)336-5401

Houston
Arrow/Schweber Electronics .... (713)530-4700
Future Electronics ..... (713)785-1155
Hamilton Hallmark ..... (713)781-6100
Newark ..... (713)270-4800
Time Electronics 1-800-789-TIME
Wyle Laboratories ..... (713)879-9953

Richardson
Newark ..... (214)235-1998

UTAH

Salt Lake City
Arrow/Schweber Electronics .... (801)973-6913
Future Electronics ..... (801)467-4448
Hamilton Hallmark ..... (801)266-2022
Newark ..... (801)261-5660
Wyle Laboratories ..... (801)974–9953

West Valley City
Time Electronics 1-800-789-TIME
Wyle Laboratories ..... (801)974-9953

WASHINGTON

Bellevue
Almac Electronics Corp. ..... (206)643-9992
Newark ..... (206)641-9800
Richardson Electronics ..... (206)646-7224

Bothell
Future Electronics ..... (206)489-3400

Redmond
Hamilton Hallmark ..... (206)881-6697
Time Electronics 1-800-789-TIME
Wyle Laboratories ..... (206)881-1150

Seattle
Wyle Laboratories. ..... (206)881-1150

Spokane
Arrow/Almac Electronics Corp. .. (509)924-9500

WISCONSIN

Brookfield
Arrow/Schweber Electronics .... (414)792-0150
Future Electronics ..... (414)879-0244

Milwaukee
Time Electronics 1-800-789-TIME

New Berlin
Hamilton Hallmark ..... (414)780-7200

Wauwatosa
Newark ..... (414)453-9100

Waukesha
Wyle Laboratories ..... (414)879-0434

CANADA

ALBERTA

Calgary
Electro Sonic Inc. ..... (403)255-9550
Future Electronics ..... (403)250-5550
Hamilton Hallmark ..... (604)420-4101

Edmonton
Future Electronics ..... (403)438-2858
Hamilton Hallmark ..... (800)663-5500

BRITISH COLUMBIA

Vancouver
Arrow Electronics ..... (604)421-2333
Electro Sonic Inc. ..... (604)273-2911
Future Electronics ..... (604)294-1166
Hamilton Hallmark ..... (604)420-4101
Newark ..... (800)463-9275

MANITOBA

Winnipeg
Electro Sonic Inc. ..... (204)783-3105
Future Electronics ..... (204)944-1446
Hamilton Hallmark ..... (800)663-5500

ONTARIO

Ottawa
Arrow Electronics ..... (613)226-6903
Electro Sonic Inc. ..... (613)728-8333
Future Electronics ..... (613)820-8313
Hamilton/Avnet Electronics ..... (613)226-1700

Toronto
Arrow Electronics ..... (416)670-7769
Electro Sonic Inc. ..... (416)494-1666
Future Electronics ..... (416)612-9200
Hamilton Hallmark ..... (905)564-6060
Newark ..... (800)463-9275
Richardson Electronics ..... (800)348-5530

QUEBEC

Montreal
Arrow Electronics ..... (514)421-7411
Future Electronics ..... (514)694-7710
Hamilton Hallmark ..... (514)335-1000
Newark ..... (800)463-9275
Richardson Electronics ..... (800)348-5530

Quebec City
Future Electronics ..... (418)877-6666

SALES OFFICES

UNITED STATES

ALABAMA, Huntsville ..... (205)464-6800
ALASKA, ... (800)635-8291
ARIZONA, Tempe ..... (602)302-8056
CALIFORNIA, Agoura Hills .. (818)706-1929
CALIFORNIA, Los Angeles . (310)417-8848
CALIFORNIA, Irvine ..... (714)753-7360
CALIFORNIA, San Diego ... (619)541-2163
CALIFORNIA, Sunnyvale ... (408)749-0510
COLORADO,
Colorado Springs ..... (719)599-7497
COLORADO, Denver ..... (303)337-3434
CONNECTICUT,
Wallingford ..... (203)949-4100
FLORIDA, Maitland ..... (407)628-2636
FLORIDA, Pompano Beach/
Ft. Lauderdale ..... (305)351-6040
FLORIDA, Clearwater ..... (813)538-7750
GEORGIA, Atlanta ..... (404)729-7100
IDAHO, Boise ..... (208)323-9413
ILLINOIS, Chicago/
Hoffman Estates ..... (708)413-2500
Shaumburg ..... (708)413-2500
INDIANA, Fort Wayne ..... (219)436-5818
INDIANA, Indianapolis ..... (317)571-0400
INDIANA, Kokomo ..... (317)455-5100
IOWA, Cedar Rapids ..... (319)378-0383
KANSAS, Kansas City/
Mission ..... (913)451-8555
MARYLAND, Columbia .... (410)381-1570
MASSACHUSETTS,
Marlborough ..... (508)481-8100
MASSACHUSETTS,
Woburn ..... (617)932-9700
MICHIGAN, Detroit ..... (313)347-6800
MINNESOTA, Minnetonka ... (612)932-1500
MISSOURI, St. Louis ..... (314)275-7380
NEW JERSEY, Fairfield .... (201)808-2400
NEW YORK, Fairport ..... (716)425-4000
NEW YORK, Hauppauge ... (516)361-7000
NEW YORK, Fishkill ..... (914)896-0511
NORTH CAROLINA,
Raleigh ..... (919)870-4355
OHIO, Cleveland ..... (216)349-3100
OHIO, Columbus/
Worthington ..... (614)431-8492
OHIO, Dayton ..... (513)495-6800
OKLAHOMA, Tulsa ..... (800)544-9496
OREGON, Portland ..... (503)641-3681
PENNSYLVANIA, Colmar ... (215)997-1020
Philadelphia/Horsham .... (215)957-4100
TENNESSEE, Knoxville .... (615)690-5593
TEXAS, Austin ..... (512)502-2100
TEXAS, Houston ..... (800)343-2692
TEXAS, Plano ..... (214)516-5100
TEXAS, Seguin ..... (210)372-7620
VIRGINIA, Richmond ..... (804)285-2100
UTAH, CSI @ ..... (801)561-5099
WASHINGTON, Bellevue ... (206)454-4160
Seattle Access ..... (206)622-9960
WISCONSIN, Milwaukee/
Brookfield ..... (414)792-0122

Field Applications Engineering Available
Through All Sales Offices

CANADA

BRITISH COLUMBIA,
Vancouver ..... (604)293-7650
ONTARIO, Toronto ..... (416)497-8181
ONTARIO, Ottawa ..... (613)226-3491
QUEBEC, Montreal ..... (514)333-3300

INTERNATIONAL

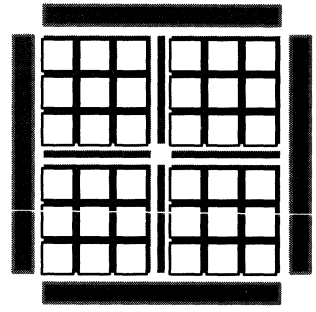
AUSTRALIA, Melbourne ... (61-3)887-0711
AUSTRALIA, Sydney ..... 61(2)906-3855
BRAZIL, Sao Paulo ..... 55(11)815-4200
CHINA, Beijing 86-505-2180
FINLAND, Helsinki 358-0-351 61191
car phone ..... 358(49)211501
FRANCE, Paris/Vanves .... 33(1)40 955 900
GERMANY, Langenhagen/
Hannover ..... 49(511)786880
GERMANY, Munich 49 89 92103-0
GERMANY, Nurenberg 49 911 96-3190
GERMANY, Sindelfingen ... 49 7031 79 710
GERMANY, Wiesbaden 49 611 973050
HONG KONG, Kwai Fong ... 852-489-1111
Tai Po 852-6668333
INDIA, Bangalore ..... (91-812)627094
ISRAEL, Tel Aviv ..... 972(3)753-8222
ITALY, Milan ..... 39(2)82201
JAPAN, Atsugi 81-462-23-0761
JAPAN, Gotanda 81-3-5487-8448
JAPAN, Kumagaya 81-485-26-2600
JAPAN, Kyushu 81-92-771-4212
JAPAN, Nagoya 81-52-232-1621
JAPAN, Osaka 81-06-305-1801
JAPAN, Sendai 81-22-268-4333
JAPAN, Tachikawa 81-425-23-6700
JAPAN, Takamatsu 81-878-37-9972
JAPAN, Tokyo 81-3-3440-3341
JAPAN, Yokohama 81-45-472-2751
KOREA, Pusan ..... 82(51)4635-035
KOREA, Seoul ..... 82(2)554-5118
MALAYSIA, Penang ..... 60(4)374514
MEXICO, Mexico City ..... 52(5)282-0230
MEXICO, Guadalajara ..... 52(36)21-8977
Marketing ..... 52(36)21-2023
Customer Service ..... 52(36)669-9160
NETHERLANDS, Best .... (31)4998 612 11
PUERTO RICO, San Juan .. (809)793-2170
SINGAPORE ..... (65)2945438
SPAIN, Madrid ..... 34(1)457-8204
or ..... 34(1)457-8254
SWEDEN, Solna ..... 46(8)734-8800
SWITZERLAND, Geneva .. 41(22)799 11 11
SWITZERLAND, Zurich .... 41(1)730-4074
TAIWAN, Taipei ..... 886(2)717-7089
THAILAND, Bangkok ..... 66(2)254-4910
UNITED KINGDOM,
Aylesbury ..... 44(296)395-252

FULL LINE REPRESENTATIVES

CALIFORNIA, Loomis
Galena Technology Group .. (916)652-0268
NEVADA, Reno
Galena Tech. Group ..... (702)746-0642
NEW MEXICO, Albuquerque
S&S Technologies, Inc. ... (505)298-7177
UTAH, Salt Lake City
Utah Comp. Sales, Inc. .... (801)561-5099
WASHINGTON, Spokane
Doug Kenley ..... (509)924-2322

HYBRID/MCM COMPONENT SUPPLIERS

Chip Supply ..... (407)298-7100
Elmo Semiconductor ..... (818)768-7400
Minco Technology Labs Inc. . (512)834-2022
Semi Dice Inc. ..... (310)594-4631



Field Programmable Gate Arrays

Literature Distribution Centers:
USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Centre; 88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan.
ASIA PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Center, No. 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.

3PHX32225–0 PRINTED IN USA 10/94 IMPERIAL LITHO 99879 10,000 LOGIC YDAAAA

DL201/D