MOTOROLA
Semiconductor Products Inc.

AN897
Application Note

MC68008 MINIMUM CONFIGURATION SYSTEM

Prepared by
Geoffrey Brown and Kyle Harper
Advanced Microcomputer Applications Engineering
Microprocessor Division
Motorola Inc.
Austin, Texas

INTRODUCTION

This application note demonstrates the design of a simple high-performance MC68008 system that uses the MC68681 Dual Universal Asynchronous Receiver Transmitter (DUART) to interface with external devices. The MC68008 is an excellent low-cost alternative to the MC68000 and features an 8-bit data bus while maintaining software compatibility with the rest of the M68000 Family. The MC68681 DUART is an M68000 Family data communications chip that features:

Two independent asynchronous serial channels,
A programmable 16-bit counter/timer,
A 6-bit parallel input port, and
An 8-bit parallel output port.

Emphasis in this design concept is placed upon performance, expandability, and low chip count.

The M68000 system design principles demonstrated in this application note include:

Interrupt hardware,
Peripheral interfacing,
Memory interface techniques,
Memory refresh arbitration in an M68000 system, and
Efficient serial I/O software.

The system, described in this design concept, features the following hardware:

An 8 MHz MC68000 microprocessor,
16K bytes of ROM,
64K bytes of dynamic RAM with no wait states, and
An MC68681 DUART.

The following paragraphs describe the hardware required for a high-performance, expandable, low chip count MC68008 system followed by a description of the software necessary to initialize and drive the MC68681 DUART.

HARDWARE REQUIREMENTS

The MC68008 has an asynchronous bus structure in which bus cycles are initiated by the assertion of address strobe ($\overline{AS}$) by the processor and are terminated by the assertion of data transfer acknowledge ($\overline{DTACK}$) by the peripheral or memory device being addressed. Figures 1-4 show the minimum hardware necessary for an MC68008 system consisting of:

Address decode logic,
$\overline{DTACK}$ generation logic,
Reset logic,
Bus error generation logic,
System memory,
Interrupt handling logic, and
An MC68681 interface.

The following paragraphs detail the required hardware as applied to the design concept described in this application note.

Address Decode Logic

The only tricky part of address decoding for an MC68008 system is that the system ROM must be mapped to address $00000 at reset. It would be impractical to fix the ROM at the bottom of the address map, as this would not allow for dynamic programming of interrupt vectors. To provide dynamic mapping of these interrupt vectors, an SN74LS164 shift register (U28) is used to generate a signal, $\overline{MAP}$, which is low for the first eight memory cycles after reset (the number of cycles necessry to fetch the reset vector and stack pointer). U28 is reset along with the processor and is clocked by the rising edge of $\overline{AS}$. The $\overline{MAP}$ signal generated by U28 is used by the address decoding circuitry to force selection of ROM when $\overline{MAP}$ is low and to allow normal memory decoding when $\overline{MAP}$ is high.

In the design given in this application note, address decoding is accomplished by a PAL16L8 (U22). This PAL is programmed to generate eight chip-select signals from ten input signals. The inputs to the PAL are the upper eight address lines (A12-A19), $\overline{IACK}$ (the NAND of the MC68008 function code lines, FC0-FC2), and the $\overline{MAP}$ signal. Four of the PAL-generated chip-select lines are used in this design to locate RAM at the address $00000, ROM at $A0000, and the MC68681 at $F0000. The four remaining chip-select lines are available for future system expansion.

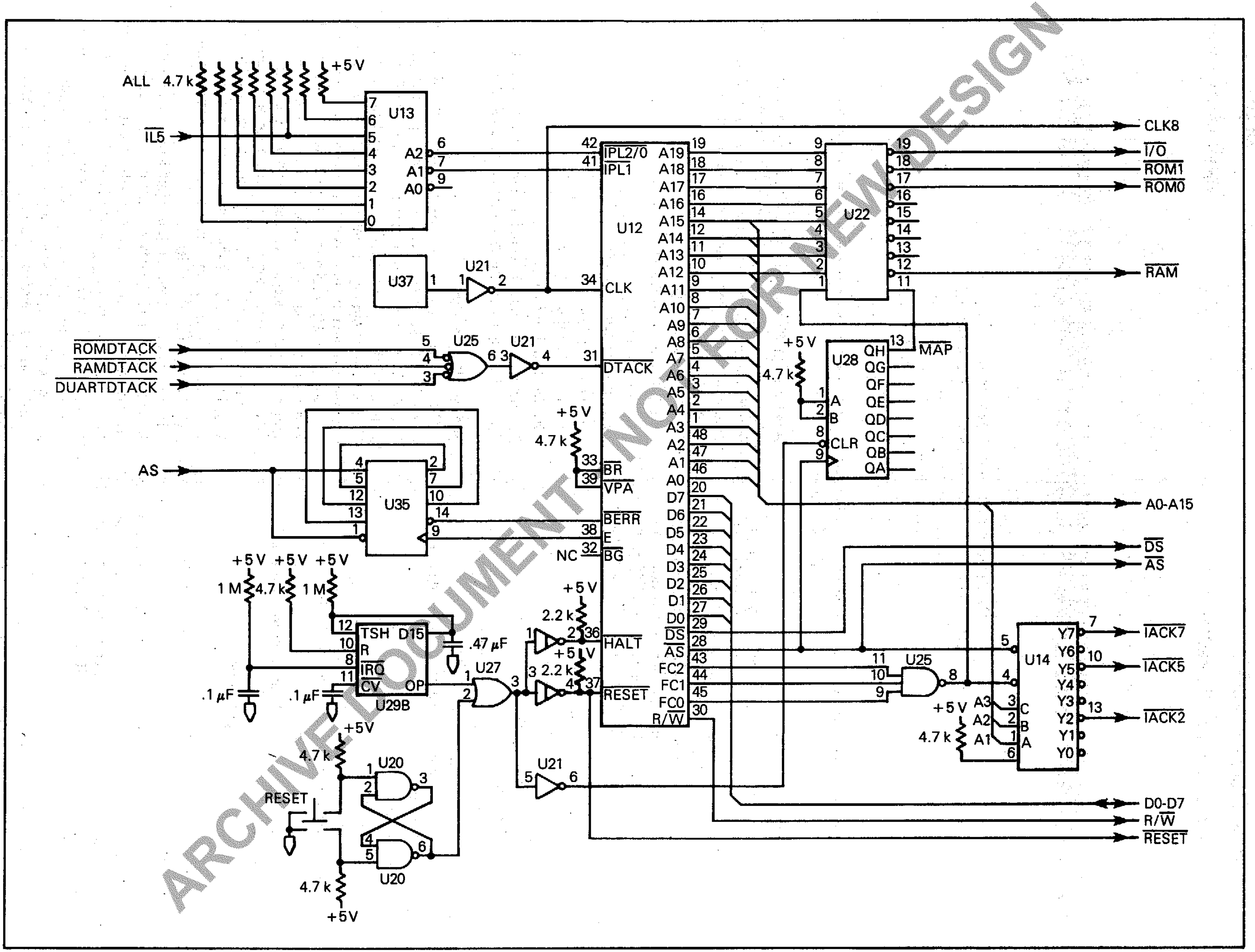


FIGURE 1 — MC68008 and Interrupt Hardware

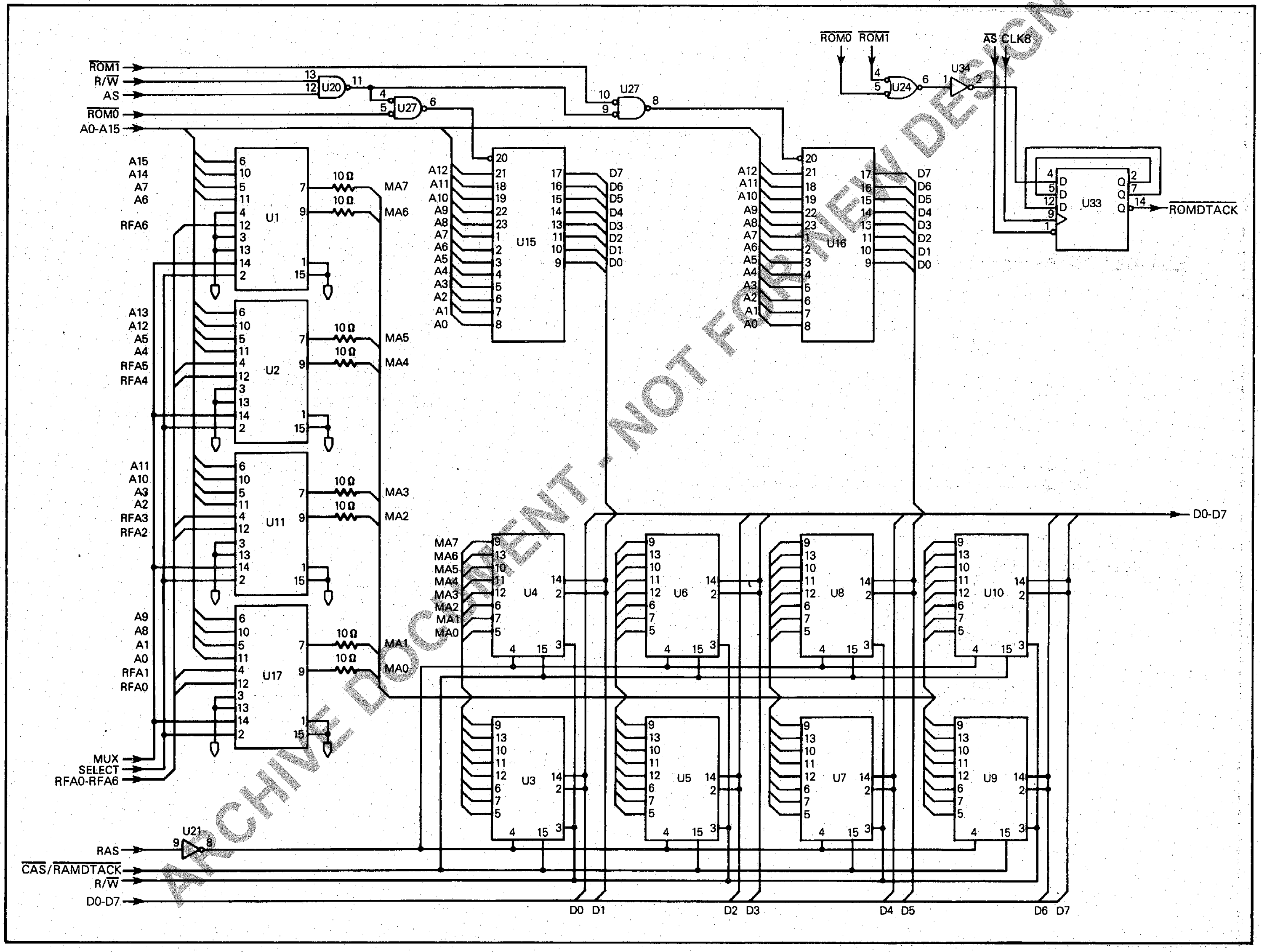


FIGURE 2 — RAM and ROM

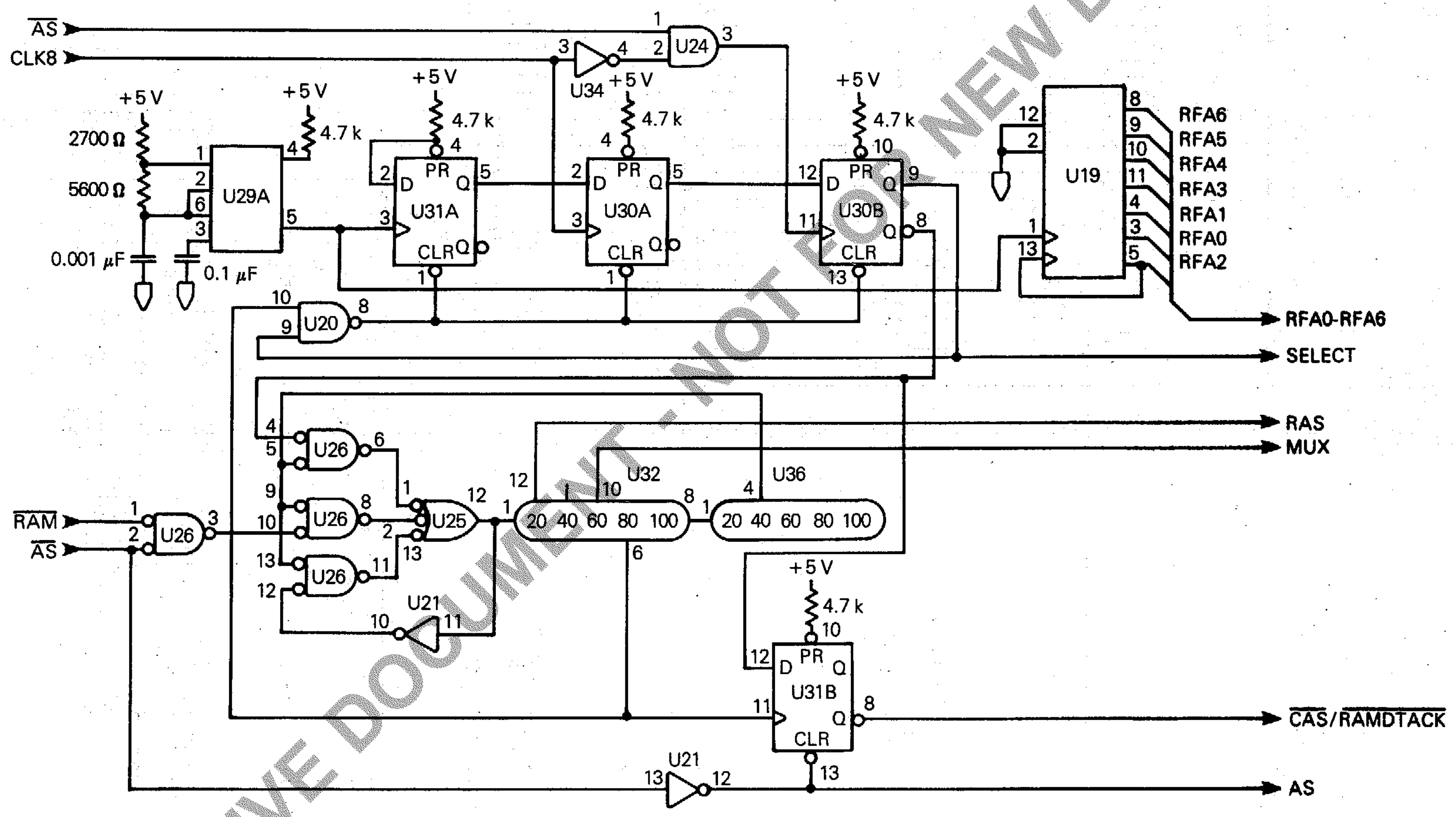
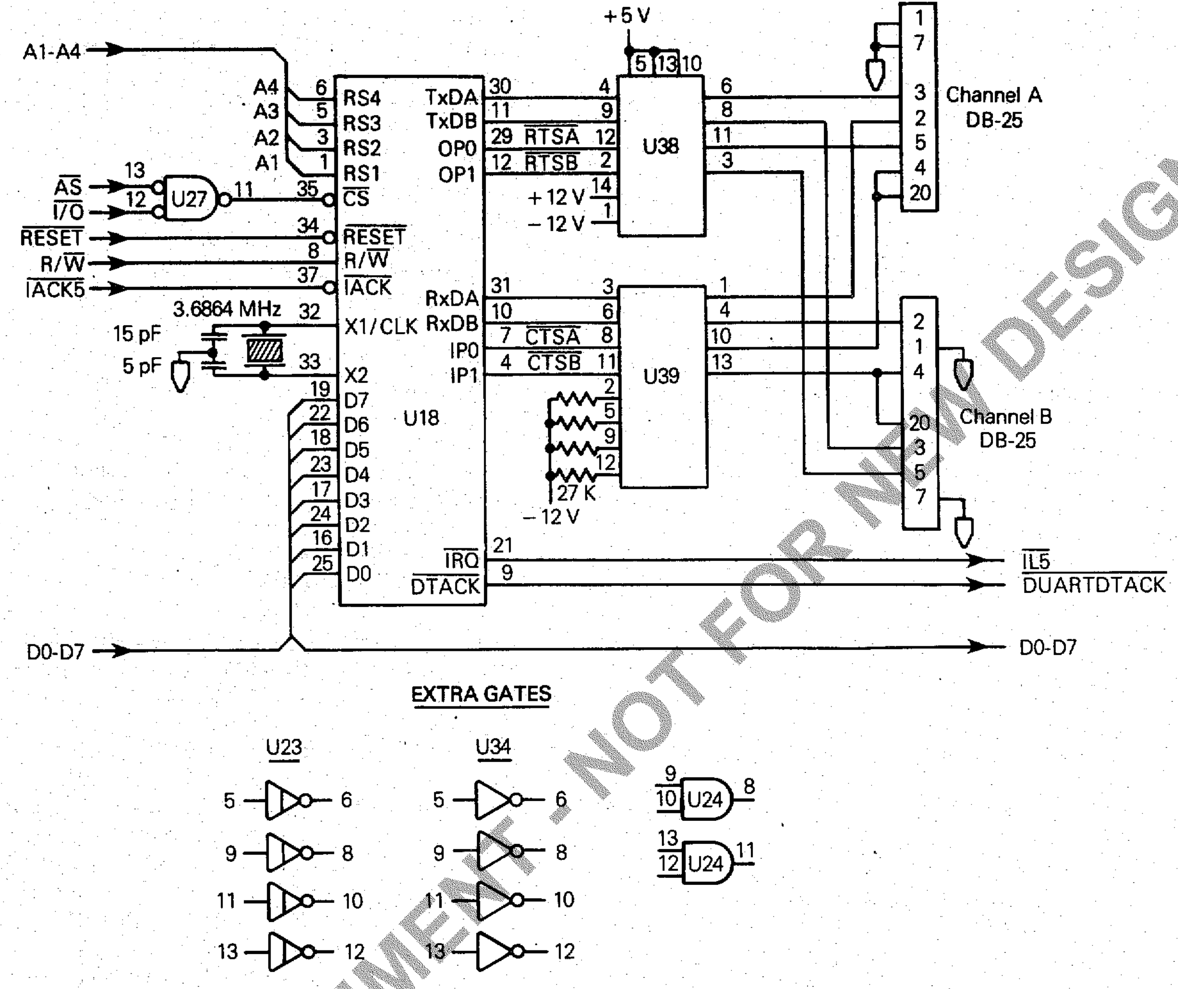


FIGURE 3 — Dynamic RAM Controller



Parts

U1, U2, U11, U17	SN74S153
U3-U10	MCM6665-15
U12	MC68008L8
U13	SN74LS148
U14	SN74LS138
U15, U16	MCM68766
U18	MC68681
U19	SN74LS393
U20	SN74LS00
U21, U34	SN74LS04
U22	PAL16L8A
U23	SN74LS05
U24	SN74LS08
U25	SN74S10
U26, U27	SN74532
U28	SN74LS164
U29	MC3456
U30, U31	SN74F74
U32, U36	DDU-4-5100
U33, U35	SN74LS175
U37	K1114A, 8 MHz OSC
U38	MC1488
U39	MC1489

FIGURE 4 — MC68681 DUART

DTACK Generation Logic

There are three possible sources of $\overline{\text{DTACK}}$:

DUART $\overline{\text{DTACK}}$,
RAM $\overline{\text{DTACK}}$, and
ROM $\overline{\text{DTACK}}$.

The DUART generates its own $\overline{\text{DTACK}}$, $\overline{\text{DTACK}}$ for RAM is generated by the RAM control circuitry, and $\overline{\text{DTACK}}$ for ROM is generated by an SN74LS175 quad flip-flop (U33). These three $\overline{\text{DTACK}}$ sources are NANDed together by U25 and U21 to generate one processor $\overline{\text{DTACK}}$.

Reset Logic

There are two sources of system reset:

Power-up reset, and
Pushbutton reset.

Power-up reset is generated by the timer (U29B) which produces an active high pulse of approximately one-half second duration. The pushbutton reset, which allows the user to reset the system without powering down, is generated by a debounced switch. These two reset signals drive $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ through SN74LS05 open-collector drivers (U23).

Bus Error Generation Logic

The bus error signal, $\overline{\text{BERR}}$, is generated by an SN74LS175 quad flip-flop (U35). U35 counts clock pulses that occur after $\overline{\text{AS}}$ becomes asserted. If $\overline{\text{AS}}$ is still asserted after four rising edges of the E clock (between 5 and 6.5 microseconds), U35 will generate $\overline{\text{BERR}}$.

System Memory

The MC68008 system presented here has a system memory that consists of 16K bytes of ROM and 64K bytes of dynamic RAM (see Figure 2). Because system performance is critical in this design, a fairly complicated, but fast, dynamic RAM control circuit has been designed (see Figure 3). This circuit uses two delay lines to sequence RAS and to address the MUX, $\overline{\text{CAS}}$, and $\overline{\text{DTACK}}$ signals. Delay lines are necessary in order to optimize memory cycle times and make it possible to design the memory controller such that the system can operate without wait states.

A description of RAM refresh request synchronization and arbitration is given in the following paragraphs. Note that, for now, a signal called SELECT is assumed which initiates refresh cycles. The principle requirements of this signal are that it occurs periodically and that it becomes asserted only while $\overline{\text{AS}}$ is negated. In addition, the RAM decode signal is qualified with $\overline{\text{AS}}$ in order to create a RAM request signal. Either the SELECT signal or the RAM request signal may initiate a RAM cycle.

The front end of the RAM controller consists of three OR gates (U26) followed by a three input NAND gate (U25) which in turn feeds into the first delay line (U32). Each of the three OR gates has as one of its inputs a signal from the second delay line (U36) which changes state in the middle of the memory cycle. The other inputs to these OR gates consist of SELECT, RAM request, and an inverted feedback path from the output of the three input NAND. The initiation of a RAM cycle via either SELECT or RAM request causes the output of the NAND gate to go high. The output of the NAND gate is then held high by the inverted feedback path until the feedback from the second delay line forces it low. The purpose of the feedback path from the second delay line is to guarantee that the delay lines will be cleared and ready to begin another RAM cycle at the end of a cycle. The outputs of the first delay line generate the RAS, MUX, and $\overline{\text{CAS}}$ signals. Both the RAS and address multiplex signals are asserted during both types of RAM cycles (normal and refresh). $\overline{\text{CAS}}$ is generated only during normal cycles and must be held asserted until the processor removes $\overline{\text{AS}}$. In order to accomplish this, the appropriate delay tap (80 nanoseconds) is used to clock the SELECT signal through a flip-flop (U31B). This flip-flop is cleared when $\overline{\text{AS}}$ is negated. The output of this flip-flop is used for both the $\overline{\text{CAS}}$ signal and for the RAM $\overline{\text{DTACK}}$. The 8 MHz MC68008 allows the $\overline{\text{DTACK}}$ to be asserted up to 90 nanoseconds before data from memory is valid on a read cycle. The specifications for MCM6665L15 dynamic memories guarantee that data is valid 75 nanoseconds after $\overline{\text{CAS}}$.

The memory refresh controller operates on the principle of cycle stealing. Refresh requests may only occur between MPU bus cycles. If an MPU RAM cycle request occurs during a refresh cycle, it will not be started until the refresh cycle is finished. At periodic intervals, a free-running clock (U29A) clocks a flip-flop (U31A) to generate a refresh request. This refresh request is synchronized with the MPU clock by two flip-flops (U30).

The MC68008 ac electrical specifications guarantee one falling clock edge during the $\overline{\text{AS}}$ high time and that there will be at least a one-half clock period of $\overline{\text{AS}}$ high time following that clock edge. Arbitration between MPU and refresh requests occurs during this one-half clock period. The refresh request synchronizer consists of two SN74F74 flip-flops (U30). The first flip-flop (U30A) has as its input the refresh request signal from the refresh request flip-flop (U31A) and is clocked by the MPU clock. The second flip-flop (U30B) has as its input the output of the first synchronizer flip-flop (U30A) and is clocked by the MPU clock qualified by $\overline{\text{AS}}$ high. This two-level synchronizer is used to ensure that there will be no risk of the first synchronizer flip-flop (U30A) entering a metastable state due to a missed setup time. The output of the synchronizer flip-flop (U30B) is the SELECT signal. All three flip-flops of the refresh circuitry are cleared after the $\overline{\text{CAS}}/\overline{\text{RAMDTACK}}$ flip-flop (U31B) has been clocked during the refresh cycle. Address multiplexing for the RAM is done by four SN74S153 multiplexers (U1, U2, U11, and U17) with the appropriate addresses routed to the RAMs by SELECT and MUX. Refresh addresses are generated by an SN74LS393 dual 4-bit counter (U19) which is clocked by the refresh clock.

Interrupt Handling Logic

The interrupt handling logic must prioritize incoming interrupt requests and generate interrupt acknowledge signals back to the interrupt sources. Interrupt prioritization is accomplished with an SN74LS148 8-to-3 priority encoder (U13). The MC68008 supports three of the M68000 interrupt levels (interrupt levels two, five, and seven); therefore, only two of the outputs of U13 are connected to the MC68008. An SN74LS138 3-to-8 demultiplexer (U14) is used to generate $\overline{\text{IACK}}$ signals for interrupting devices. The SN74LS138 is enabled when $\overline{\text{AS}}$ is asserted and FC0-FC2 are all high (indicating an interrupt acknowledge cycle). Because the MC68681 uses only one of the interrupt levels (interrupt level five), the remaining two levels are available for future system expansion.

The MC68681 Interface

With these logic circuits in place, interfacing the MC68681 to the MC68008 is trivial (see Figure 4). The $\overline{\text{RESET}}$, R/$\overline{\text{W}}$, and data bus lines (D0-D7) are connected directly between the MC68681 and the MC68008. The I/O chip-select line generated by the address decode logic ($\overline{\text{I/O}}$) is connected to the MC68681 chip-select ($\overline{\text{CS}}$) pin. These address lines are used instead of A0-A3 in order to maintain hardware design consistency with the other M68000 Family microprocessors (which do not have address line A0). The MC68681 interrupt ($\overline{\text{IRQ}}$) and interrupt acknowledge ($\overline{\text{IACK}}$) pins are tied to the $\overline{\text{IL5}}$ and $\overline{\text{IACK5}}$ lines of the interrupt handling logic, respectively, thus assigning the MC68681 interrupt a level 5 priority. Finally, a 3.6864 MHz crystal is connected between the MC68681 X1/CLK and X2 pins. The crystal is required for the on-chip baud-rate generator. 15 pF and 5 pF shunt capacitors must also be connected between the crystal and ground as shown to ensure proper operation of the oscillator.

The MC68681 serial channels are connected to external devices via RS-232 drivers and DB-25 connectors. The MC68681 OP0, IP0, OP1, and IP1 pins are used as the $\overline{\text{RTSA}}$, $\overline{\text{CTSA}}$, $\overline{\text{RTSB}}$, and $\overline{\text{CTSB}}$ handshake lines, respectively; therefore, they too are routed via the RS-232 drivers to their respective connectors.

THE DUART SOFTWARE

This design will use both of the channels and the $\overline{\text{RTS}}/\overline{\text{CTS}}$ handshake capabilities of the DUART. The interface software required for this design is flowcharted in Figure 5 and is listed at the end of this document. There are three routines: DINIT, INCH, and OUTCH.

DINIT is the DUART initialization routine and is executed upon system power-up. After DINIT initializes the DUART channels, it enables channel A and channel B in normal operation mode. INCH is the input character routine. Upon entry, INCH requires the channel base address in address register A0. Upon return, the lower byte of data register D0 will contain the received character. OUTCH is the output character routine. Upon entry, OUTCH requires the channel base address in address register A0 and the character to be transmitted in the lower byte of data register D0.

SUMMARY

The system presented in this design concept is a low-cost, high-performance minimal chip count system. With the MC68681 DUART alone, this system offers a high degree of interface flexibility. It provides two serial I/O channels, a parallel input port, a parallel output port, a counter/timer, and versatile interrupt capabilities. However, if more peripheral chips are required, they can be interfaced easily. Also, if chip count is of higher importance than performance and/or expandability, the design presented here can be reduced even further by simplifying the RAM controller logic and the interrupt handler logic.

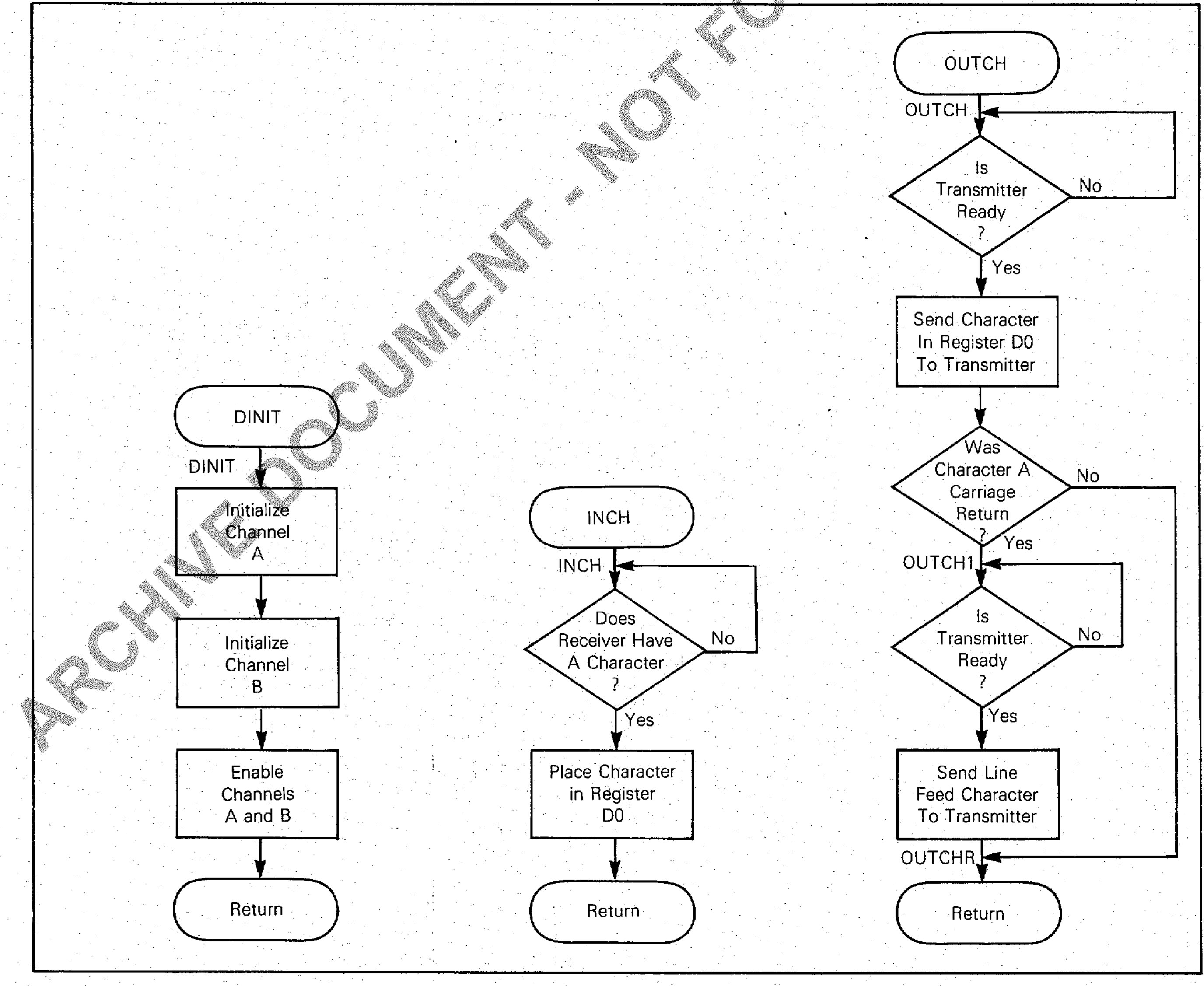


FIGURE 5 — MC68008 Minimum System Software Flowcharts

```
                      OPT      FRS,PCS,BRS
             *
             * MINSYS   - ROUTINES REQUIRED FOR A 68008-BASED SYSTEM TO INITIALIZE
             *            AND DRIVE A 68681 DUART:
             *
             *                  DINIT - SUBROUTINE TO INITIALIZE DUART
             *                   INCH - SUBROUTINE TO INPUT A CHARACTER
             *                  OUTCH - SUBROUTINE TO OUTPUT A CHARACTER
             *
             *            AUTHOR   - KYLE HARPER
             *            DATE     - DECEMBER 22, 1983
             *            VERSION  - 2
             *
             *

             *
             * SYSTEM ADDRESSES
             *

000F0001     DUART    EQU      $0F0001               BASE ADDRESS OF 68681 DUART

000F0001     CHANA    EQU      DUART+0               CHANNEL A BASE ADDRESS
000F0001     MR1A     EQU      DUART+0               MODE REGISTER 1A
000F0001     MR2A     EQU      DUART+0               MODE REGISTER 2A
000F0003     SRA      EQU      DUART+2               STATUS REGISTER A
000F0003     CSRA     EQU      DUART+2               CLOCK-SELECT REGISTER A
000F0005     CRA      EQU      DUART+4               COMMAND REGISTER A
000F0007     RBA      EQU      DUART+6               RECEIVER BUFFER A
000F0007     TBA      EQU      DUART+6               TRANSMITTER BUFFER A

000F0009     IPCR     EQU      DUART+8               INPUT PORT CHANGE REGISTER
000F0009     ACR      EQU      DUART+8               AUXILIARY CONTROL REGISTER
000F000B     ISR      EQU      DUART+10              INTERRUPT STATUS REGISTER
000F000B     IMR      EQU      DUART+10              INTERRUPT MASK REGISTER
000F000D     CMSB     EQU      DUART+12              CURRENT COUNTER/TIMER MOST SIGNIFICANT BYTE
000F000D     CTUR     EQU      DUART+12              COUNTER/TIMER UPPER REGISTER
000F000F     CLSB     EQU      DUART+14              CURRENT COUNTER/TIMER LEAST SIGNIFICANT BYTE
000F000F     CTLR     EQU      DUART+14              COUNTER/TIMER LOWER REGISTER

000F0011     CHANB    EQU      DUART+16              CHANNEL B BASE ADDRESS
000F0011     MR1B     EQU      DUART+16              MODE REGISTER 1B
000F0011     MR2B     EQU      DUART+16              MODE REGISTER 2B
000F0013     SRB      EQU      DUART+18              STATUS REGISTER B
000F0013     CSRB     EQU      DUART+18              CLOCK-SELECT REGISTER B
000F0015     CRB      EQU      DUART+20              COMMAND REGISTER B
000F0017     RBB      EQU      DUART+22              RECEIVER BUFFER B
000F0017     TBB      EQU      DUART+22              TRANSMITTER BUFFER B

000F0019     IVR      EQU      DUART+24              INTERRUPT VECTOR REGISTER
000F001B     IP       EQU      DUART+26              INPUT PORT (UNLATCHED)
000F001B     OPCR     EQU      DUART+26              OUTPUT PORT CONFIGURATION REGISTER
000F001D     STRC     EQU      DUART+28              START-COUNTER COMMAND
000F001D     BTST     EQU      DUART+28              OUTPUT PORT REGISTER BIT SET COMMAND
000F001F     STPC     EQU      DUART+30              STOP-COUNTER COMMAND
000F001F     BTRST    EQU      DUART+30              OUTPUT PORT REGISTER BIT RESET COMMAND

             *
             * CONSTANTS
```

```
                                   *

              0000000D    CR       EQU      $0D                     ASCII CARRIAGE RETURN
              0000000A    LF       EQU      $0A                     ASCII LINE FEED

              00002000             ORG      $002000

                                   *
                                   * DINIT - DUART INITIALIZATION ROUTINE.
                                   *         AFTER INITIALIZING THE DUART CHANNELS FOR
                                   *         OPERATION, DINIT ENABLES CHANNEL A AND CHANNEL B.
                                   *
                                   *         ENTRY CONDITIONS:
                                   *
                                   *                  (NONE)
                                   *
                                   *         EXIT CONDITIONS:
                                   *
                                   *                  CHANNEL A'S RX & TX ARE ENABLED.
                                   *                  CHANNEL B'S RX & TX ARE ENABLED.
                                   *                  ALL REGISTERS ARE UNALTERED.
                                   *
                                   *

    00002000 13FC0000000F DINIT    MOVE.B   #$00,ACR                USE BAUD RATE GENERATOR SET 1
             0009
    00002008 13FC00BB000F          MOVE.B   #$BB,CSRA               A: RX & TX AT 9600 BAUD
             0003
    00002010 13FC0082000F          MOVE.B   #$82,MR1A               RX-RTS,NO RX-IRQ,CHAR ER,EVN PRTY,7 CHAR
             0001
    00002018 13FC001F000F          MOVE.B   #$1F,MR2A               NORMAL,NO TX-RTS,CTS-TX,2 STOPS
             0001
    00002020 13FC00BB000F          MOVE.B   #$BB,CSRB               B: RX & TX AT 9600 BAUD
             0013
    00002028 13FC0082000F          MOVE.B   #$82,MR1B               RX-RTS,NO RX-IRQ,CHAR ER,EVN PRTY,7 CHAR
             0011
    00002030 13FC001F000F          MOVE.B   #$1F,MR2B               NORMAL,NO TX-RTS,CTS-TX,2 STOPS
             0011
    00002038 13FC0005000F          MOVE.B   #$05,CRA                ENABLE A'S RX & TX
             0005
    00002040 13FC0005000F          MOVE.B   #$05,CRB                ENABLE B'S RX & TX
             0015
    00002048 4E75                  RTS

                                   *
                                   * INCH - DUART CHANNEL INPUT CHARACTER ROUTINE.
                                   *        GETS CHARACTER FROM A DUART CHANNEL AND PLACES IT IN D0.
                                   *
                                   *        ENTRY CONDITIONS:
                                   *
                                   *                 CHANNEL BASE ADDRESS IN A0.
                                   *                 CHANNEL RX ENABLED.
                                   *
                                   *        EXIT CONDITIONS:
                                   *
                                   *                 RECEIVED CHARACTER PLACED IN D0.
                                   *                 ALL OTHER REGISTERS UNALTERED.
```

```
MOTOROLA M68000 ASM VERSION  1.30SYS :   5.        .MINSYSS .SA 01/20/84 13:44:18
MINSYSS

                              *
                              *

0000204A 082800000002 INCH      BTST.B    #0,2(A0)              WAIT FOR CHANNEL'S RX TO GET A CHAR
00002050 67F8                   BEQ       INCH
00002052 10280006               MOVE.B    6(A0),D0              GET CHARACTER FROM RECEIVER
00002056 4E75                   RTS

                              *
                              * OUTCH - DUART CHANNEL OUTPUT CHARACTER ROUTINE.
                              *         SENDS CHARACTER IN D0 TO A DUART CHANNEL.
                              *         IF CHARACTER IN D0 IS A CARRIAGE RETURN, OUTCH WILL
                              *         OUTPUT BOTH A CARRIAGE RETURN & LINE FEED CHARACTER.
                              *
                              *         ENTRY CONDITIONS:
                              *
                              *                  CHANNEL BASE ADDRESS IN A0.
                              *                  CHARACTER TO BE TRANSMITTED IN D0.
                              *                  CHANNEL TRANSMITTER ENABLED.
                              *
                              *         EXIT CONDITIONS:
                              *
                              *                  CHARACTER SENT TO CHANNEL'S TX.
                              *                  ALL REGISTERS UNALTERED.
                              *
                              *

00002058 082800020002 OUTCH     BTST.B    #2,2(A0)              WAIT FOR CHANNEL'S TX TO BECOME READY
0000205E 67F8                   BEQ       OUTCH
00002060 11400006               MOVE.B    D0,6(A0)              SEND CHAR TO TRANSMITTER
00002064 0C00000D               CMP.B     #CR,D0                WAS IT A CARRIAGE RETURN?
00002068 660E                   BNE       OUTCHR                NO, SKIP NEXT PART
0000206A 082800020002 OUTCH1    BTST.B    #2,2(A0)              YES, WAIT FOR TX TO BECOME READY AGAIN
00002070 67F8                   BEQ       OUTCH1
00002072 117C000A0006           MOVE.B    #LF,6(A0)             SEND A LINE FEED
00002078 4E75         OUTCHR    RTS
                                END

****** TOTAL ERRORS       0--
****** TOTAL WARNINGS     0--
```

```
MOTOROLA M68000 ASM VERSION  1.30SYS :   5.        .MINSYSS .SA 01/20/84 13:44:18
MINSYSS

SYMBOL TABLE LISTING

SYMBOL NAME      SECT   VALUE        SYMBOL NAME      SECT   VALUE


ACR                    000F0009      ISR                    000F000B
BTRST                  000F001F      IVR                    000F0019
BTST                   000F001D      LF                     0000000A
CHANA                  000F0001      MR1A                   000F0001
CHANB                  000F0011      MR1B                   000F0011
CLSB                   000F000F      MR2A                   000F0001
CMSB                   000F000D      MR2B                   000F0011
CR                     0000000D      OPCR                   000F001B
CRA                    000F0005      OUTCH                  00002058
CRB                    000F0015      OUTCH1                 0000206A
CSRA                   000F0003      OUTCHR                 00002078
CSRB                   000F0013      RBA                    000F0007
CTLR                   000F000F      RBB                    000F0017
CTUR                   000F000D      SRA                    000F0003
DINIT                  00002000      SRB                    000F0013
DUART                  000F0001      STPC                   000F001F
IMR                    000F000B      STRC                   000F001D
INCH                   0000204A      TBA                    000F0007
IP                     000F001B      TBB                    000F0017
IPCR                   000F0009
```

This information has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Motorola reserves the right to make changes to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein. No license is conveyed under patent rights in any form. When this document contains information on a new product, specifications herein are subject to change without notice.

MOTOROLA *Semiconductor Products Inc.*
Colvilles Road, Kelvin Estate - East Kilbride/Glasgow - SCOTLAND
Printed in Switzerland