Order this document
by AN1015/D

MOTOROLA
SEMICONDUCTOR
APPLICATION NOTE

AN1015

MC68020 Minimum System Configuration

As described in this application note, Motorola's MC68020 32-bit microprocessor minimum system configuration can be used for many applications that were formerly in the realm of mainframe computers or microprocessors. These applications need the benefits of the complete 32-bit architecture but with a simpler address and data bus configuration.

DESCRIPTION

This application uses the MC68020 microprocessor in a system having minimum hardware interconnects. The system, which includes an 8-bit data bus, a 24-bit address bus, and as few devices as possible, can upgrade an existing MC68008 design or can be constrained for use in a space-limited environment.

The system uses inexpensive large-scale integration (LSI) devices. In addition to the MC68020, a single byte-wide electrically programmable read-only memory (EPROM) (similar to a 27512-170) and static random-access memory (SRAM) (the Motorola MCM6064P15) are used with the Motorola MC68901 multifunction peripheral (MFP) for system timing and serial communications. Also, medium-scale integration/small-scale integration (MSI/SSI) TTL devices are used for clock generation, high-speed gating, buffering $\overline{BERR}$ generation, and address decoding. Other common components are required for power supply decoupling, reset generation, and pullups. The schematic diagram is shown in Figure 1 (found at the back of this document), and the list on page 2 shows the inputs, outputs, and logic equations for device U05, a programmable array logic PAL16L8. Table 1 lists the parts for the minimum system configuration.

INPUT/OUTPUT

In this minimum system configuration, the only system I/O required is a serial interface to a terminal or some similar device. This interface uses the USART contained on the MC68901 MFP, and the MFP also generates baud rates for the onboard serial port. The XTAL1 and XTAL2 inputs are connected to a 2.4576-MHz crystal. The delay-only timers C and D in the MFP are configured for prescaling and delay generation for timing a 9600 baud asynchronous communication port. The RS-232-C interface-level generation is accomplished by using Motorola's MC145406, a single 16-pin device providing three RS-232-C line drivers and three RS-232-C line receivers. It provides a very efficient single-device solution for the vast majority of RS-232-C interfacing requirements. Of the standard RS-232 handshake lines, only RTS is controlled via software, DTR is strapped in the active-high state, and all others are ignored.

Unused inputs are important considerations for any MC68020 system, regardless of configuration. All inputs must be driven to a known level. Several inputs to the MC68020 were not used in this application — signals such as $\overline{CDIS}$, $\overline{BR}$, $\overline{BGACK}$, $\overline{AVEC}$, and $\overline{HALT}$, all of which are active in a low state. These inputs were pulled to a high level to avoid conflict with functions on the devices that were used.

SYSTEM TIMING GENERATION

The MC68020RC12 microprocessor operates at a clock speed of 12.5 MHz. The simplest way to obtain a clean, symmetrical clock signal is to use the buffered output of a 12.5-MHz oscillator to drive a pair of F04 inverter/buffers, eliminating the use of expensive delay lines or complex timing functions. In addition, critical parameters for worst-case performance can be determined for a guaranteed functional design over worst-case timing constraints. These parameters include clock skew, setup and hold-time conformance, and worst-case signal propagation.

The basic bus cycle of the MC68020 is asynchronous and occurs in three clock periods. Using memory devices listed previously can provide for zero wait-state operation at a 12.5-MHz clock frequency. However, an MC68020 system using an 8-bit data bus would usually not have zero wait-state performance as a major system requirement. Thus, the extra gating required to allow zero wait-state access to the SRAM and EPROM is inconsistent with the minimum system configuration. A simpler approach is to allow a single wait state for memory accesses, using inexpensive 150–170-ns devices. This approach allows

INPUTS, OUTPUTS, AND LOGIC EQUATIONS FOR U05, PAL16L8

Inputs

Pin 1 = A00; Address Bus Bit 0
Pin 2 = A01; Address Bus Bit 1
Pin 3 = A02; Address Bus Bit 2
Pin 4 = A16; Address Bus Bit 16
Pin 5 = A17; Address Bus Bit 17
Pin 6 = A18; Address Bus Bit 18
Pin 7 = A19; Address Bus Bit 19
Pin 8 = A20; Address Bus Bit 20
Pin 9 = A21; Address Bus Bit 21
Pin 11 = A22; Address Bus Bit 22
Pin 13 = A23; Address Bus Bit 23
Pin 14 = FC0; Function Code Bit 0
Pin 15 = FC1; Function Code Bit 1
Pin 16 = FC2; Function Code Bit 2

Outputs

Pin 12 = !IACK; IACK cycle output
Pin 17 = !MFP; MFP select
Pin 18 = !ROM; ROM select
Pin 19 = !RAM; RAM select

Logic Equations

```
IACK = FC0 & FC1 & FC2 & A02 & !AO1 & !AOO ;

MFP  = FC0 & !FC1 & A16 & A17 & A18 & A19 & !A20 & A21 & A22 & A23 #
       !FC0 & FC1 & A16 & A17 & A18 & A19 & !A20 & A21 & A22 & A23 ;

ROM  = FC0 & !FC1 & !A16 & !A17 & !A18& !A19 & !A20 & A21& !A22 & !A23 #
       !FC0 & FC1 & !A16 & !A17 & !A18& !A19 & !A20 & !A21& !A22 & !A23 #

RAM  = FC0 & !FC1 & !A16 & !A17 & !A18 & !A19 & A20 & A21&.A22 & A23 #
       !FC0 & FC1 & !A16 & !A17 & !A18 & !A19 & A20 & A21 & A22 & A23 ;
```

Table 1. MC68020 Minimum System Configuration Parts List

Reference	Part Number	Description	Manufacturer
U01	MC68020RC12	MPU	Motorola
U02	MCM6064P15	8K × 8 SRAM	Motorola
U03	27512-170	64K × 8 EPROM	Various
U04	MC68901P	MFP	Motorola
U05	PAL16L8	PAL	Various
U06	MC74F32	Quad 2-in NOR	Motorola
U07	MC74F00	Quad 2-in NAND	Motorola
U08	MC74F161	4-Bit Sync Counter	Motorola
U09	MC145406	Hex RS232 Tx/Rx	Motorola
U10	MC74F74	Dual D-Flip-Flop	Motorola
U11	MC74F74	Dual D-Flip-Flop	Motorola
U12	MC74LS14	Hex Schmitt Inverter	Motorola
U13	MC74F04	Hex Inverter	Motorola
Y01	Oscillator	12.5 MHz	Various

NOTE: Capacitors, resistors, a crystal, a diode, and a switch are also required.

$\overline{\text{DSACK}}$ generation using a single MC74F74 and two of the four gates in an MC74F00.

The timing for generation of DSACK0 in this manner is simple. In all bus cycles, the address bus is guaranteed stable within 40 ns of the rising edge of the first clock of the bus cycle, and $\overline{\text{AS}}$ is guaranteed asserted within 40 ns of the falling edge of the same clock. In all read cycles, data is required to meet a 10-ns setup time with respect to the falling edge of the last clock in the cycle, regardless of any wait states. In write cycles, data is guaranteed stable well in advance of the same edge.

Figure 2 shows a RAM/ROM read cycle followed by a RAM write cycle in the minimum system configuration. DSACK generation begins on the falling edge of the second clock of the cycle, and $\overline{\text{DSACK0}}$ is asserted after the rising edge of the third clock cycle. The cycle completes after the falling edge of the fourth clock cycle.

Address strobe gates the output of the PAL16L8 at U05 to select access to the ROM, RAM, or MFP during an iACK cycle. If the MFP is selected, DSACK is generated by the MFP's DTACK output. If ROM or RAM access is selected, the dual F74 DSACK generation circuit is used per the diagram shown in Figure 3.

Using the previous timing constraints, a simple formula determines the number of wait states, the speed of memory devices required, or the time allowable for decode logic in any 12.5-MHz system:

110 ns + 80 ns(# wait states) = system access time

or

110 ns + 80 ns(# wait states) = device access + decode time

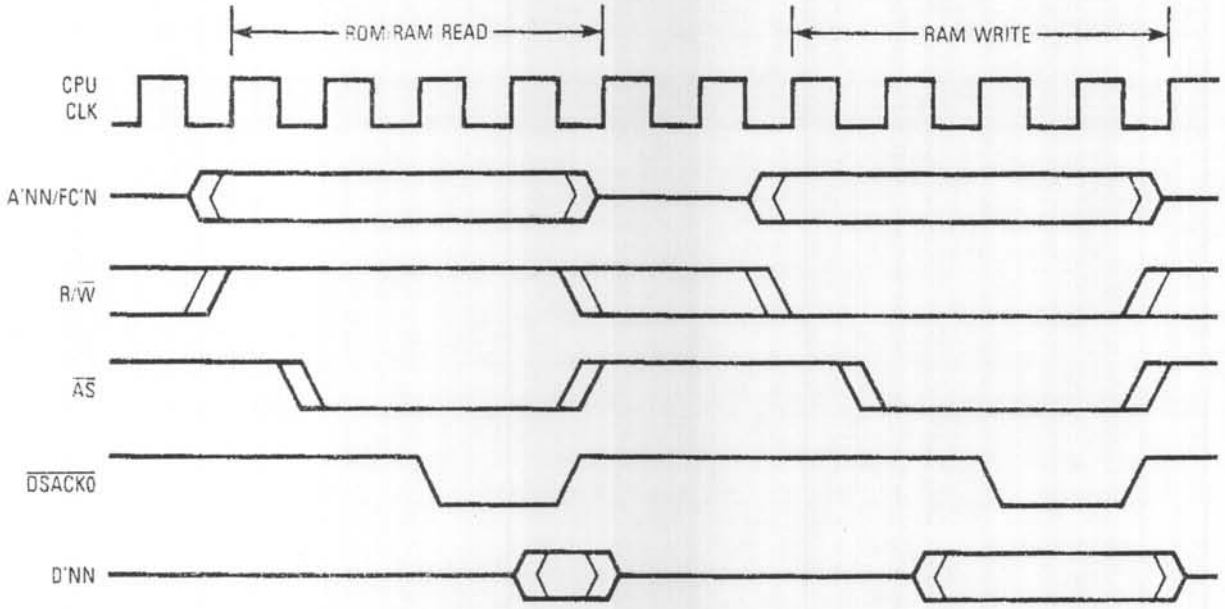


Figure 2. RAM/ROM Read Cycle

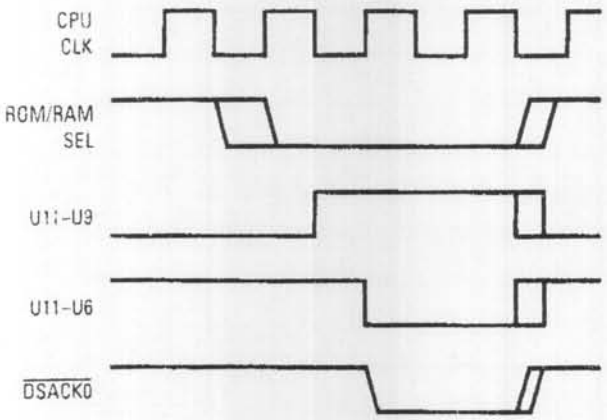


Figure 3. DSACK Timing Diagram

In the MC68020, minimum system configuration, one-wait-state operation yields a total system access time of 190 ns. Using 150–170-ns devices, at least 20 ns is available for decoding, well within the performance capability of an MC74F32. The MC74F32 is used to gate address decode with $\overline{AS}$ to develop chip selects for the memory devices or for higher speed B and D series PALs.

The MC68901 MFP operates at any clock frequency from 1–4 MHz. For simplicity, the 12.5-MHz clock used to operate the MC68020RC12 was divided by four, yielding a 3.125-MHz frequency for operating the MFP. This frequency is also suitable for operating an MC74F161 4-bit counter used to generate $\overline{BERR}$ as the result of an incomplete bus cycle. The MFP generates $\overline{DTACK}$ after it has been accessed and adheres to a basic four MFP-clock bus cycle. As such, all read/write accesses to the device complete within 1.28 μs, well within the 5–12-μs nominal timeout of the $\overline{BERR}$ watchdog.

BASE ADDRESS DECODING

Decoding of base addresses for the directly accessible devices is accomplished using a single 16L8 PAL. The PAL speed required for the minimum system configuration is not critical; only the address lines and functional codes are decoded. Depending on the cycle in process, an active-low output is logically ANDed with $\overline{AS}$ (also active low) in an MC74F32 to enable the ROM, RAM, MFP, or MFP IACK. It would be possible to eliminate the MC74F32 if a high-speed PAL similar to D-series devices is used.

SOFTWARE

The following software listing (see Figure 4) describes minimum system initialization and routines used to verify the prototype hardware developed in this application note. The routines include a simple memory exerciser program, which first performs a cursory test of RAM memory and then initializes RAM, including the interrupt vector table, with appropriate information. Also included is a minimum initialization of the MC68901 MFP. Actual application software can be added as needed. The software listing in Figure 4 can be used as minimum routines for any system of similar configuration.

SYSTEM EXPANSION

This minimum system configuration can be expanded to a 32-bit data bus configuration by adding three more SRAM and/or EPROM devices. Also add chip select connections to the memory devices, connection to the appropriate address and data bus lines, and expanded hardware in support of 8- and 16-bit accesses over the 32-bit data bus.

For an expanded system, additional I/O requirements can be handled with the unused portions of the MFP. With their highly functional programmability, the six unused ports in the general-purpose I/O can be used for external inputs to allow edge detection, pulse generation, or similar I/O functions. Added circuitry can be limited to external inputs to the device. For other functions, additional address decode logic and the particular I/O are needed.

The required hardware is described in the *MC68020 User's Manual*.

CONCLUSION

The minimum system configuration can be expanded to larger data paths and can be adapted to many applications requiring the performance of Motorola's MC68020 32-bit microprocessor.

```
*                    Equates section

ROMBAS      EQU      0              ROM BASE ADDRESS
RAMBAS      EQU      $F00000        RAM BASE ADDRESS
STACK       EQU      $F003FF        INITIAL STACK POINTER
MFPBAS      EQU      $EF0000        MFP BASE ADDRESS
MFPVCT      EQU      $40            VECTOR FOR MFP SOURCED INTERRUPT
NOP         EQU      $4E71          STANDARD 68000 NOP INSTRUCTION
*                    MC68901 MFP Registers
MFPGPIP     EQU      MFPBAS+$1      GPIP DATA
MFPAER      EQU      MFPBAS+$3      ACTIVE EDGE
MFPDDR      EQU      MFPBAS+$5      DATA DIRECTION
MFPIERA     EQU      MFPBAS+$7      INTERRUPT ENABLE A
MFPIERB     EQU      MFPBAS+$9      INTERRUPT ENABLE B
MFPIPRA     EQU      MFPBAS+$B      INTERRUPT PENDING A
MFPIPRB     EQU      MFPBAS+$D      INTERRUPT PENDING B
MFPISRA     EQU      MFPBAS+$F      INTERRUPT IN-SERVICE A
```

Figure 4. MC68020 Minimum System Configuration Startup Software (Sheet 1 of 5)

```
MFPISRB     EQU     MFPBAS+$11      INTERRUPT IN-SERVICE B
MFPIMRA     EQU     MFPBAS+$13      INTERRUPT MASK A
MFPIMRB     EQU     MFPBAS+$15      INTERRUPT MASK B
MFPVR       EQU     MFPBAS+$17      VECTOR
MFPTACR     EQU     MFPBAS+$19      TIMER A CONTROL
MFPTBCR     EQU     MFPBAS+$1B      TIMER B CONTROL
MFPTCDCR    EQU     MFPBAS+$1D      TIMER C/D CONTROL
MFPTADR     EQU     MFPBAS+$1F      TIMER A DATA
MFPTBDR     EQU     MFPBAS+$21      TIMER B DATA
MFPTCDR     EQU     MFPBAS+$23      TIMER C DATA
MFPTDDR     EQU     MFPBAS+$25      TIMER D DATA
MFPSCR      EQU     MFPBAS+$27      SYNCHRONOUS CHARACTER
MFPUCR      EQU     MFPBAS+$29      USART CONTROL
MFPRSR      EQU     MFPBAS+$2B      RECEIVER STATUS
MFPTSR      EQU     MFPBAS+$2D      TRANSMITTER STATUS
MFPUDR      EQU     MFPBAS+$2F      USART DATA

*
*       Program section
*       The ROM in this application is mapped to the variable
*       ROMBAS. All executable code is resident in ROM.

START       EQU     ROMBAS
            DC.L    STACK           INITIAL STACK POINTER
            DC.L    ROMSTART        INITIAL PROGRAM COUNTER

ROMBUF      DS.L    32              LEAVE A LITTLE SPACE HERE

MEMDAT      EQU     *               MEMORY EXERCISER DATA
*                                   THIS DATA IS USED TO CHECK MEMORY
            DC.B    $5
            DC.B    $A
            DC.B    $0
            DC.B    $F
            DS.L    $100            LEAVE MORE SPACE

ROMSTART    EQU     *               BEGINNING OF PROGRAM SECTION
            MOVE.L  #RAMBAS,D0      POINT AT BASE OF RAM
            MOVEC.L D0,VBR          AND INITIAL VBR TO POINT THERE

*
*                                   *** Memory exerciser ***
*       This routine performs a cursory check of memory prior to
*       proceeding. An error count is contained in D7 upon completion.

            CLR.L   D7              CLEAR ERROR COUNTER
            MOVE.L  #3,D3           INIT OUTER LOOP COUNTER
            LEA.L   RAMBAS,A0       POINT AT BASE OF RAM
            LEA.L   MEMDAT,A1       POINT AT MEMORY EXERCISER DATA

LOOP0       EQU     *
            MOVE.L  #$1FFF,D0       INIT INNER LOOP COUNTER
            MOVE.B  (A1,D3),D2      GET MEMORY DATA

LOOP1       EQU     *
            MOVE.B  D2,(A0,D0)      PUT DATA INTO MEMORY
            CMP.B   (A0,D0),D2      NOW COMPARE WITH STORED DATA
            BEQ.S   LOOP1_1         JUMP AROUND IF THE SAME
            ADDQ    #1,D7           ELSE INCREMENT ERROR COUNTER

LOOP1_1     EQU     *
            DBRA    D0,LOOP1        TEST ALL OF RAM MEMORY
            DBRA    D3,LOOP0        FOR ALL DATA TYPES (4 TESTS)
```

Figure 4. MC68020 Minimum System Configuration Startup Software (Sheet 2 of 5)

```
*
*
*       When done with test, memory is to be initialized with NOPs and vector table initialized with address of generic handler. Afte
*       initialization, D7 will contain the number of errors from the test section.

MEMINIT EQU     *
        LEA.L   RAMBAS,A0       POINT AT BASE OF RAM AGAIN
        MOVE.L  #$1FFE,D0       USE AS LOOP COUNTER FOR MEMINIT
        MOVE.L  #$3FE,D2        POINT AT BOTTOM OF VECTOR TABLE
        MOVE.W  #NOP,D1         FILL NON-VECTOR MEMORY WITH NOPs

LOOP2   EQU     *
        MOVE.W  D1,(A0,D0)      PUT DATA INTO MEMORY
        SUBQ    #2,D0           DECREMENT COUNTER
        CMP.L   D0,D2           NOW LOOK FOR BOTTOM OF VECTOR
*                               TABLE
        BNE.S   LOOP2           CONTINUE UNTIL THERE

        SUBQ    #2,D0           ELSE MOVE TO LONG-WORD INIT
        MOVE.L  #EXCHND,D1      AND PUT GENERIC EXCEPTION HANDLER
*                               IN REST OF VECTOR TABLE MEMORY

LOOP3   EQU     *
        MOVE.L  D1,(A0,D0)      PUT HANDLER ADDRESS THERE
        SUBQ    #4,D0           AND DECREMENT POINTER
        BGE.S   LOOP3           FILL REST OF MEMORY
*       Done with memory check/initialization
*
        Now init the MC68901 MFP

        JSR     MFPINIT         DO SO AS SUBROUTINE FOR ADDED USE
*                               AT LATER TIME

        TST     D7              NOW CHECK IF ANY ERRORS
        BEQ     NO_ERR          IF NONE OUTPUT OK MESSAGE
*                               ELSE OUTPUT ERROR MESSAGE
        LEA.L   ERRMSG,A0       POINT AT TOP OF MESSAGE
        LEA.L   ERMEND,A1       AND POINT AT END
        BRA.S   INITND          JUMP TO END OF INIT ROUTINE

NO_ERR  EQU     *               INIT OK!!!
        LEA.L   OKMSG,A0        POINT AT MESSAGE
        LEA.L   ERRMSG-1,A1     POINT AT END OF MESSAGE

INITND  EQU     *
        JSR     SEROUT          OUTPUT MESSAGE OVER SERIAL PORT
*                               THEN FALL THRU TO

POLL    EQU     *               POLL SERIAL PORT FOR INPUT
        BTST.B  #3,MFPRSR       CHECK FOR BREAK
        BNE.S   BREAK           IF PRESENT, JUMP TO PROCESS
        BTST.B  #7,MFPRSR       ELSE CHECK FOR CHARACTER
        BEQ.S   POLL            LOOP IF NO DATA PRESENT, ELSE
*       DATA PRESENT IN USART RECEIVER
*       USER INSERT INPUT CHARACTER
*       PROCESSING ROUTINE HERE


BREAK   EQU     *               BREAK DETECT ROUTINE
*
                                USER INSERT BREAK HANDLER HERE
        JMP     POLL            AND RETURN WHEN COMPLETED
```

Figure 4. MC68020 Minimum System Configuration Startup Software (Sheet 3 of 5)

— See Page 9 for Sheet 4 —

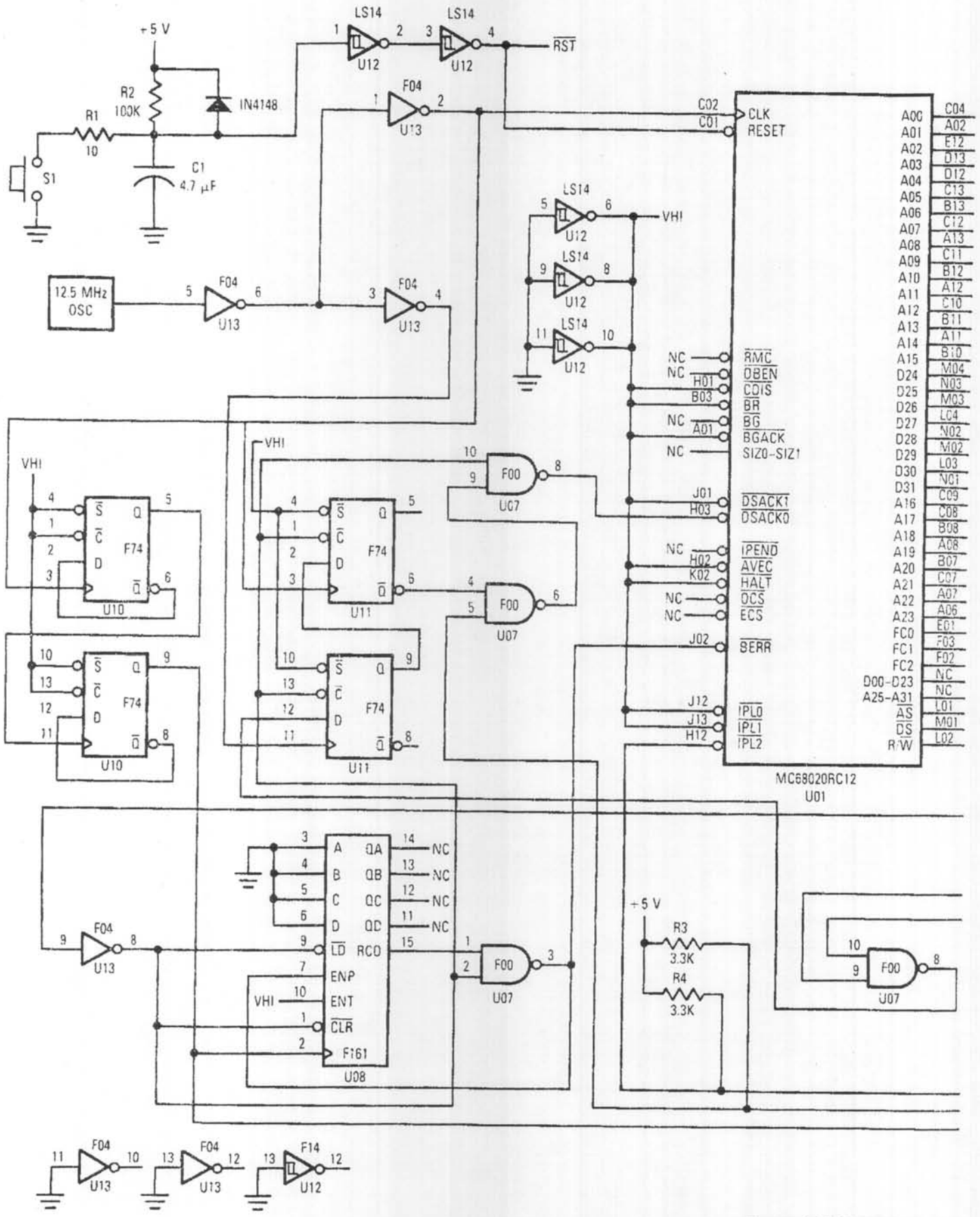
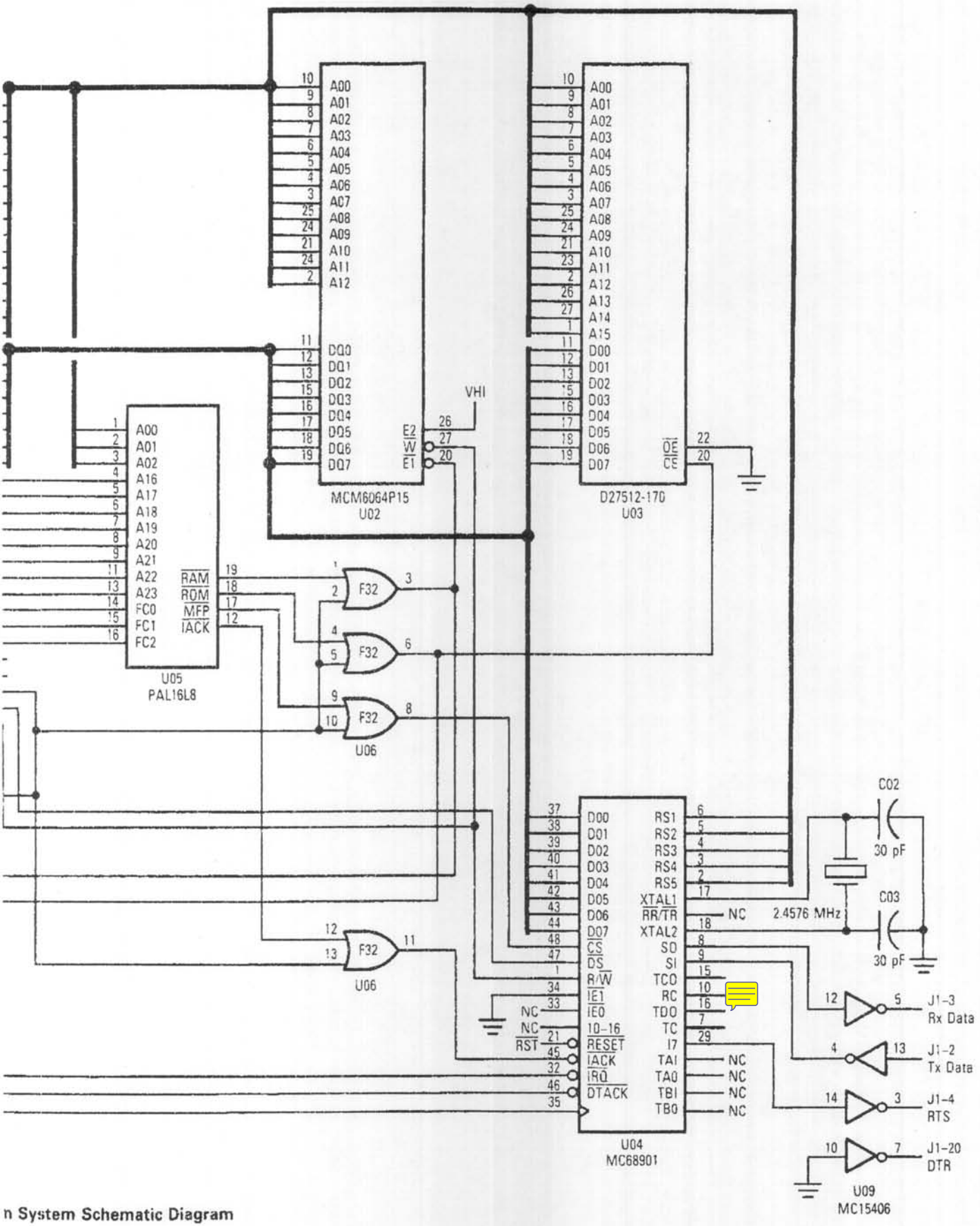


Figure 1. MC68020 Minimu



n System Schematic Diagram

```
SEROUT      EQU     *                   OUTPUT MESSAGE VIA MC68901 USART
*                                       BEGINNING OF MESSAGE POINTED AT BY
*                                       A0, END BY A1

            BTST.B  #7,MFPTSR           CHECK FOR BUFFER EMPTY
            BEQ.S   SEROUT              AND LOOP UNTIL SO

            MOVE.B  (A0)+,D0            GET DATA POINTED TO BY A0 INTO D0
            MOVE.B  D0,MFPUDR           AND PUT INTO USART DATA REGISTER
            CMPA    A1,A0               COMPARE CURRENT ADDRESS WITH END
            BGE.S   SEROUT              LOOP UNTIL DONE.

            RTS                         ELSE RETURN WHEN COMPLETED.

*           GENERIC EXCEPTION HANDLER

*
        *   THE GENERIC EXCEPTION HANDLER ACCOMMODATES EXCEPTIONS
        *   THAT OCCUR IN THE MINIMUM SYSTEM VIA RTE. ADD
        *   APPLICATION S/W HERE TO ACCOMMODATE EXCEPTIONS THAT
        *   ARE PROCESSED IN SPECIFIC APPLICATIONS

EXCHND      EQU     *                   GENERIC EXCEPTION HANDLER
*                                       USER INSERT HANDLER(S) AS NEEDED
            RTE                         AND RETURNS

MFPINIT     EQU     *                   MC68901 INITIALIZATION ROUTINE
            CLR.L   D0                  CLEAR D0
            SUBQ    #1,D0               THEN TURN INTO ALL 1's
            MOVE.B  D0,MFPDDR           ALL MFP I/O INIT'D TO OUTPUT
            ADDQ    #3,D0               NOW TURN D0 INTO 2
            MOVE.B  D0,MFPTCDR          SELECT 1/4 Tx CLOCK
            MOVE.B  D0,MFPTDDR          SELECT 1/4 Rx CLOCK
*                                       SELECT DIVIDE BY 4 IN C/D CNTRL REG
            MOVE.B  #$11,MFPTCDCR
            MOVE.B  #$88,MFPUCR         SELECT DIVIDE BY 16, 8-BIT,
*                                       NO PARITY IN USART CONTROL
*                                       REGISTER


*           INITIALIZE MFP VECTOR AND HANDLER

            MOVE.L  #MFPVCT,D0          GET VECTOR
            MOVE.B  D0,MFPVR            LOAD INTO MFP
            ASL.L   #2,D0               NOW SHIFT LEFT 2
            MOVE.L  D0,A0               PUT INTO ADDRESS REGISTER
            MOVE.L  #MFPEXC,(A0)        AND INIT APPROPRIATE VECTOR

*           NOW START Tx, Rx CLOCKS

            MOVE.B  #1,MFPRSR           START RECEIVER CLOCK
            MOVE.B  #5,MFPTSR           START TRANSMITTER CLOCK
            BSET.B  #7,MPFGPIP          NOW RAISE RTS

            RTS                         DONE!! RETURN FROM ROUTINE

*           MFP EXCEPTION HANDLER ROUTINE
MFPEXC      EQU     *
*                   USER INSERT EXCEPTION HANDLER HERE
            RTE

*           MESSAGES SECTION
```

Figure 4. MC68020 Minimum System Configuration Startup Software (Sheet 4 of 5)

```
OKMSG           EQU     *
                DC.B    'WELCOME TO MINSYS CONFIGURATION SYSTEM!>'

ERRMSG          EQU     *
                DC.B    'MEMORY ERRORS ENCOUNTERED!!!'
ERRMND          EQU     *
                DC.B    '>'

                END     START                   END OF PROGRAM
```

Figure 4. MC68020 Minimum System Configuration Startup Software (Sheet 5 of 5)

Literature Distribution Centers:

USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Center; 88 Tanners Drive, Blakelands Milton Keynes, MK145BP, England.
ASIA PACIFIC: Motorola Semiconductors H.K. Ltd.; P.O. Box 80300; Cheung Sha Wan Post Office; Kowloon Hong Kong.

MOTOROLA

AN1015 PRINTED IN USA 3-89 DIGITAL LITHO 057848 10,000 MPL