MOTOROLA
Semiconductor Products Inc.

AN-881
APPLICATION NOTE

DUAL-PORTED RAM FOR THE MC68000 MICROPROCESSOR

Prepared by
Trey West
Microprocessor Applications Engineer
Motorola Microprocessor Division
Austin, Texas

INTRODUCTION

Dual-ported RAM provides a means for multiprocessor systems to exchange data without directly interfering with each other. In most systems this data exchange involves a master MPU passing information to a slave MPU. For example, a host MPU may need to transfer information to a graphic processing circuit to direct the display operation. Redundant processing schemes may require a "checking processor" to compare the results of several MPUs operating simultaneously. Whatever the application, some form of communication between processors is required.

A dual-ported RAM has, as it's name implies, two independent ports or address/data/control buses. This scheme simply allows two processors to access the same memory contents without interfering with each other. Thus, depending on the amount of dual-ported RAM that is available, messages, instructions, data, etc. may be transfered from one processor to the other.

Access to the dual-ported RAM is controlled by one or more semaphore registers. A semaphore register is simply a memory location set aside as a flag to indicate whether or not a dual-ported RAM is currently in use. If the semaphore bit is set, one of the two processors is currently using the dual-ported RAM space and the other processor is not allowed access. Other semaphore registers could be defined to indicate messages available, contents changed, etc.

The MC68000 TAS (test and set) instruction supports this semaphore register concept. The TAS instruction reads the semaphore register and determines if the most significant bit (MSB) of the semaphore register is set. If the MSB is clear the TAS instruction sets it. If the MSB of the semaphore register is already set, the TAS instruction simply reports (via the condition code register) that the resource is allocated to another processor at this time. Besides replacing several instructions with one, the TAS instruction executes the entire read/modify/write cycle in one indivisible bus cycle. This alleviates the possibility of both processors reading the semaphore flag bit (MSB) as clear and both assuming they have the resources after setting the semaphore flag bit (MSB).

The cycle timing considerations for the dual-processor system discussed in this application note were calculated using two MC68000L8 (8-MHz clock) microprocessors. In addition, a separate 12.5 MHz arbitration clock was used to clock the dual-ported RAM arbitration circuit. The same system could be used with any other MC68000 microprocessors and arbitration clock speeds. However, the arbitration clock speed should be sufficiently high to ensure that the cycle timing is commensurate with the system speed.

HARDWARE CONSIDERATIONS

In a multiprocessor system, the operation of each processor may be asynchronous with respect to the other MPUs in the system. With asynchronous operation it is impossible to predict when an MPU will request access to the contents of the dual-ported RAM (hereafter referred to as DPR); therefore, a DPR arbitration circuit is a necessity. As shown in the circuit diagram of Figure 1, four D latches (U3A, U3B, U4A, U4B) form the basis of the arbitration circuit. The first two latches (U3A and U3B) are clocked on opposite phases of a 12.5 MHz arbitration clock. Initially after reset, the state of the four D latches is such that the U4A and U4B $\overline{Q}$ input to OR gates U2A and U2B are low. If one of these two OR gates receives a low chip select input (CS1, CS2) it will cause a change in one set of the D latches (U3A, U3B). Since both D latches of the first pair (U3A, U3B) are clocked on opposite phases of the 12.5 MHz arbitration clock, only one will change state, even if the falling edge of both CS1 and CS2 signals occur simultaneously. The second set of latches (U4A, U4B) provides a debounce latch for the first set. The debounce latch is required since, if a rising 12.5 MHz arbitration clock edge and the D input both change state at the same

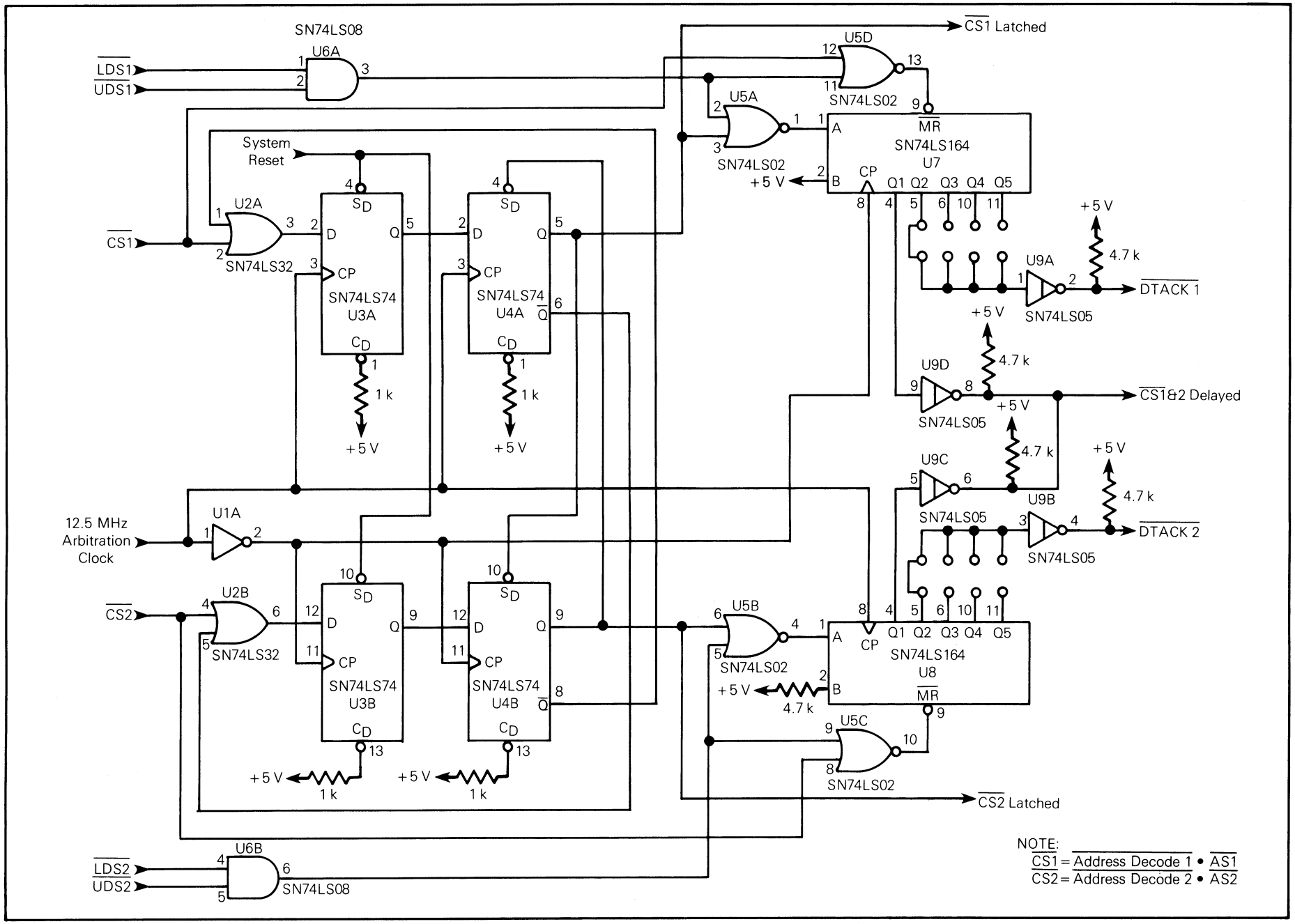

FIGURE 1 — Dual-Ported RAM Bus Arbitration Circuit, Schematic Diagram

time, the corresponding Q output could become unstable for up to 75 nanoseconds. The U4A-U4B pair of latches are also clocked by the 12.5 MHz arbitration clock, and approximately 80 nanoseconds later the latched low chip select signal appears at the Q output (CS1 at U4A, CS2 at U4B). The latched low chip select signal also presets the other D latch; similarly, the high $\overline{Q}$ output is cross-coupled to the OR gate input of the first D latch. This feedback holds off the access of the other processor until the first processor has finished its access and releases its chip select ($\overline{CS1}$, $\overline{CS2}$) signal.

Once the arbitration circuit has selected the processor which is allowed access to the DPR, the other processor is locked out and cannot gain access to the DPR until the first processor has completed its bus cycle. Locking out the other processor is accomplished by holding off its data transfer acknowledge ($\overline{DTACK}$) signal. Because holding off $\overline{DTACK}$ locks out the other processor, the $\overline{BERR}$ timer signal of the locked out processor should be longer than any $\overline{DTACK}$ delay caused by DPR accesses.

NOTE

The longest $\overline{BERR}$ timeout required would result from a lockout at the start of a TAS instruction. In the case of an 8-MHz MC68000 running with this DPR hardware, the worst case time would be approximately two microseconds. The suggested $\overline{BERR}$ timeout in this case is 10 microseconds.

The latched $\overline{CS1}$ or $\overline{CS2}$ signal (referred to as CSx) is presented to an SN74LS164 shift register provided that one or both of the data strobes ($\overline{LDS}$, $\overline{UDS}$) is asserted. Once the signal is present at the serial input of the shift register, it begins propagating through on positive transitions of the 12.5 MHz clock. The shift register outputs become $\overline{CSx}$ Delayed and $\overline{DTACK}$ for either processor. The delay before $\overline{DTACK}$ can be adjusted by using a switch or jumper to tap off the appropriate delayed output of the SN74LS164. The cycle ends when the $\overline{CSx}$ signal or both data strobes are negated, clearing the shift register.

The random logic gates attached to the SN74LS164s allow the use of the TAS instruction. The TAS instruction simplifies support of the dual port RAM semaphore registers. This instruction is a special case because two bus cycles take place during one $\overline{AS}$ asserted time (see Figure 2).

The first bus cycle starts in the same manner as all MC68000 cycles with $\overline{AS}$ and $\overline{LDS}$ or $\overline{UDS}$ asserted (TAS is a byte operation only). At the end of this read cycle the only indication of a complete transfer is the negation of the data strobes. Therefore, negated data strobes must clear the shift register to remove $\overline{CSx}$ Delayed and $\overline{DTACK}$. The next cycle (write) starts by asserting the data strobes ($\overline{LDS}$ or $\overline{UDS}$). An asserted data strobe releases the shift register clear input ($\overline{CSx}$ is also low) and allows application of the $\overline{CSx}$ Latched input to the SN74LS164 serial input. The cycle continues as a normal write.

The schematic diagram in Figure 3 shows the necessary MC68B10 static RAMs and buffers required to allow two processors to access the same devices. Octal bi-directional buffers, U10, U11, U16, U17 (SN74LS245s) provide buffering for the data bus. These devices are controlled by the $\overline{CSx}$ Data input ($\overline{CSx}$ Data signals are the $\overline{CSx}$ Delayed outputs of the SN74LS164 shift registers delayed again by passing through an SN74LS244). Only one set of data buffers is on at any time as controlled by the arbitration circuitry. Data direction is determined by the state of the corresponding R/$\overline{W}$ line. The address lines are buffered in a similar manner with the $\overline{CSx}$ Latched signal providing the enable input for the two SN74LS244s. By using the $\overline{CSx}$ Latched signal (instead of $\overline{CSx}$ Delayed), the address setup time (t_{as}) required by the MC68B10s is met prior to the $\overline{CSx}$ Delayed signal being presented to the chip select inputs of the MC68B10s (this address setup time is not required by many other memories). An SN74LS244 (U13) is used as a multiplexer for the R/$\overline{W}$, $\overline{LDS}$, and $\overline{UDS}$ control signals from the two processors. The $\overline{CSx}$ Latched signal from the arbitration circuit determines which set of control signals are presented to the MC68B10 RAMs.

The above described scheme could allow expansion to 256 words of dual-ported RAM simply by using the available A8 line on the SN74LS244s (U12 and U18 shown in Figure 3), plus larger memories. Expansion to larger dual-ported RAM memory space would also require additional address buffers and larger memories.

TIMING CONSIDERATIONS

The timing signals related to the arbitration circuit are shown in Figure 4. This diagram illustrates the special case of chip select signals from two different processors requesting

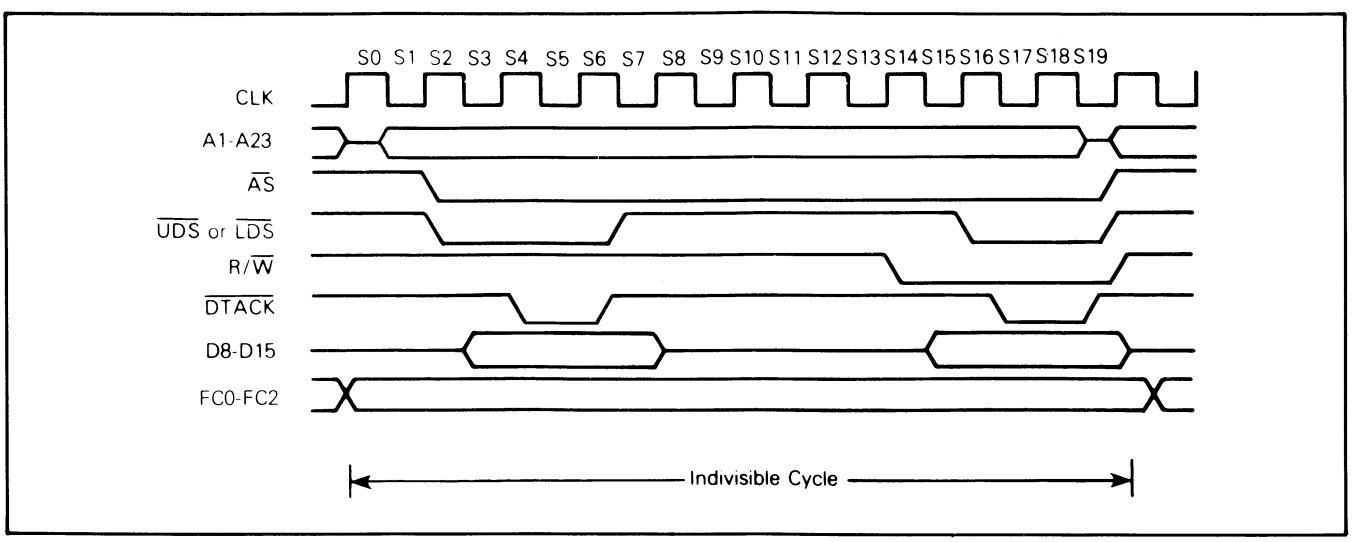

FIGURE 2 — Read-Modify-Write Cycle Timing Diagram

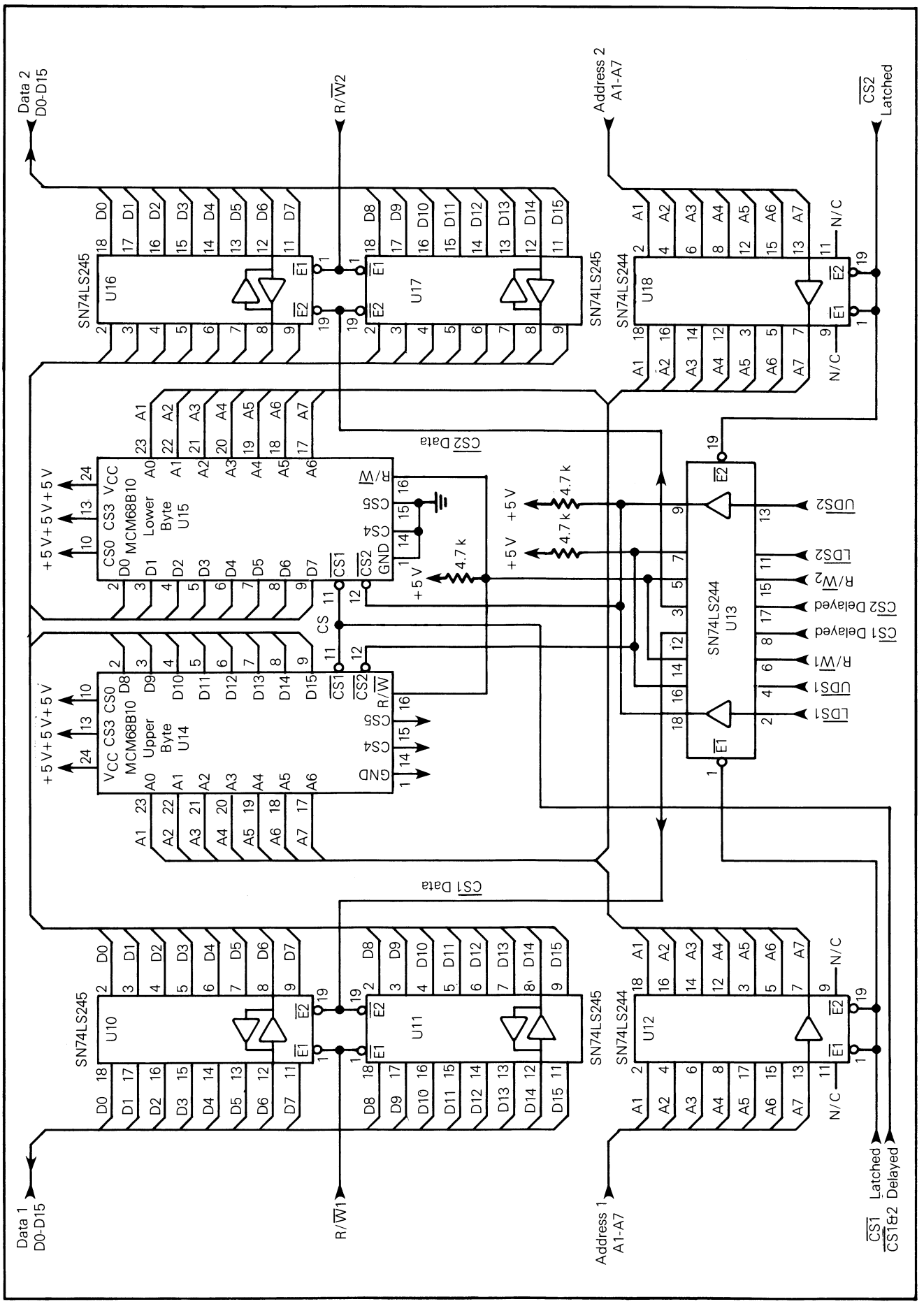

FIGURE 3 — DPR RAM and Buffers Schematic Diagram

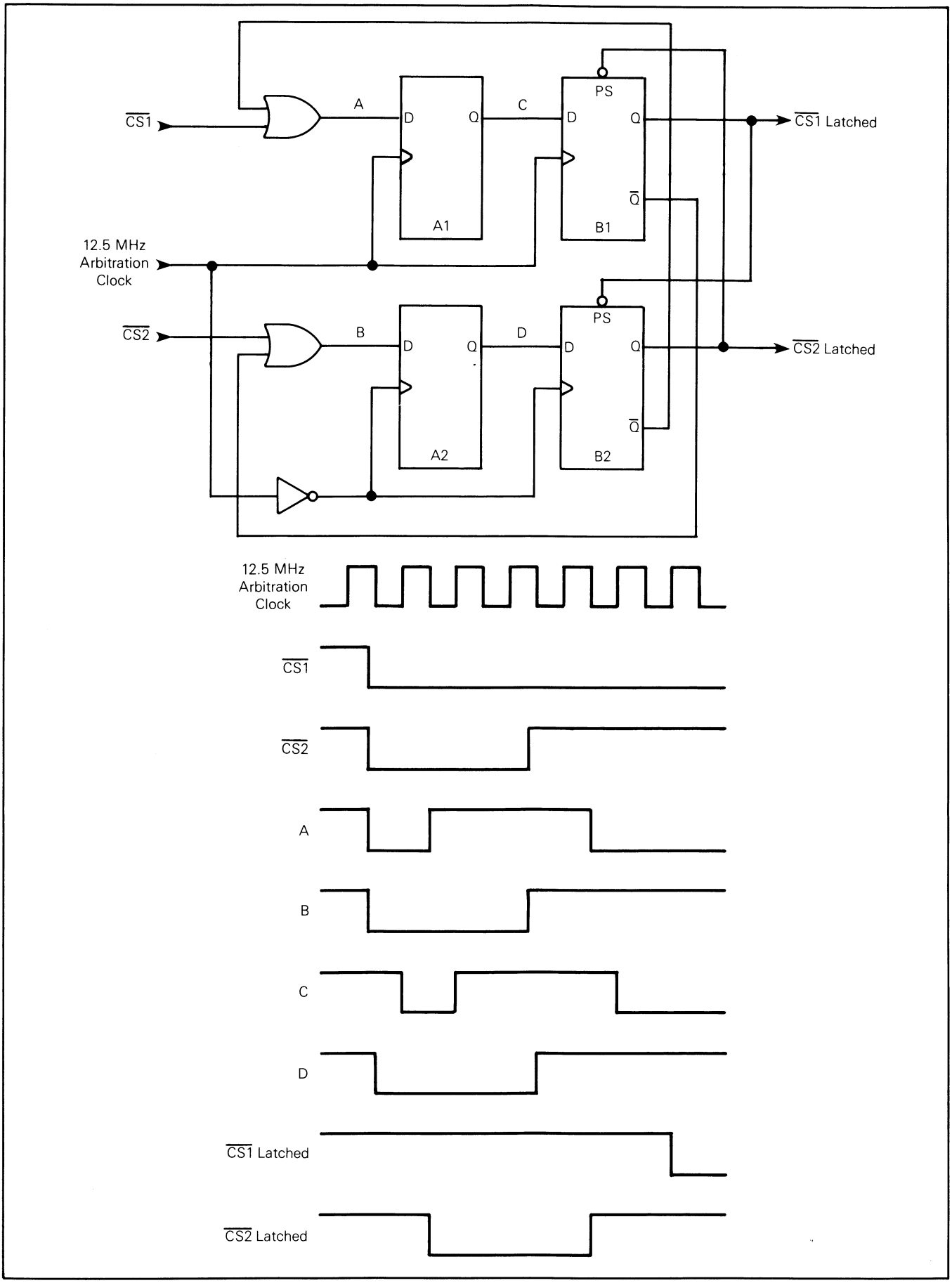

FIGURE 4 — CS Arbitration Timing Diagram

simultaneous access. As will be described, when the simultaneous requests are made, only the $\overline{CS}$ signal of one processor will be allowed to propagate through to form the $\overline{CSx}$ ($\overline{CS1}$, $\overline{CS2}$) Latched signal. In the diagram of Figure 4, the chip select requests are both already asserted before the falling edge of the 12.5 MHz arbitration clock. Since the $\overline{CS2}$ signal is clocked on the falling edge in Figure 4, it will be allowed to propagate through. Note that the Q output for $\overline{CS1}$ (point C) does fall but the signal is not allowed to be clocked into the debounce latch (B1) because it is now held in preset by the $\overline{CS2}$ Latched signal. The complement of the $\overline{CS2}$ Latched signal is also presented to the OR gate at the input of the first latch for $\overline{CS1}$ causing the D input to go high, and on the next positive edge clock, the waveform at C goes high. When $\overline{CS2}$ is removed, the input latch (waveform D) and the debounce latch for $\overline{CS2}$ are driven high on consecutive negative clock edges. This action removes preset from B1 and allows $\overline{CS1}$ to propagate through on positive clock edges to form the $\overline{CS1}$ Latched signal.

As shown in Figure 3, the $\overline{CS1}$ or $\overline{CS2}$ Latched signal enables the address buffer for one of the MCM8B10s. The $\overline{CS1}$ or $\overline{CS2}$ latched signal is also presented to the input of one of the SN74LS164 shift registers, and a clock period later the $\overline{CS1}$ or $\overline{CS2}$ Delayed signal is presented to the $\overline{CS}$ input of both MCM68B10s. This delay ($\overline{CS1}$ or $\overline{CS2}$ Latched to $\overline{CS1}$ or $\overline{CS2}$ Delayed) is necessary to accommodate the address setup time for the memories.

The read and write cycle timing is shown in Figure 5 (starting with the $\overline{CSx}$ Latched signal). For a read cycle, data is valid 250 nanoseconds after addresses are valid. The $\overline{DTACK}$ signal for the processor (8-MHz MC68000) is asserted a clock period after the $\overline{CSx}$ Delayed signal goes low. Data is then valid approximately 50 nanoseconds later (this satisfies the $\overline{DTACK}$ low to data setup time for the MC68000L8 specification). For a write cycle, data is valid before $\overline{CSx}$ Delayed; therefore, the only specifications that must be met are t_{cyc} for addresses and t_{cs} for the $\overline{CSx}$ Delayed signal. Asserting $\overline{DTACK}$ at the same time as a read cycle assures that both these specifications are met. The last specification to be met is the data hold time for the memories. This 10 nanosecond hold time is met by passing the $\overline{CSx}$ Delayed signal through control multiplexer U13 to form the $\overline{CSx}$ Data signal. The $\overline{CSx}$ Data signal will negate approximately the same time as the $\overline{CS}$ signal at the memories; however the buffer delays (U10, U11, U16 and U17) will provide the required data hold time.

TEST SOFTWARE

Once both processors are able to read and write to all DPR (dual-ported RAM) locations, it is necessary to check the arbitration circuitry during real time processing. Two test programs are included. The first program (listing 1 in Figure 6) causes both processors to execute a loop in which they request access to the DPR. Once access is allowed, the processor sets all DPR memory locations to a known value, and then checks to see if all locations retained this value. Completion of the loop results in a message (error or successful completion) and the processor executes a short delay loop to allow the other processor to complete a TAS instruction. The other processor then executes the same routine with the exception of writing a different value to the DPR locations.

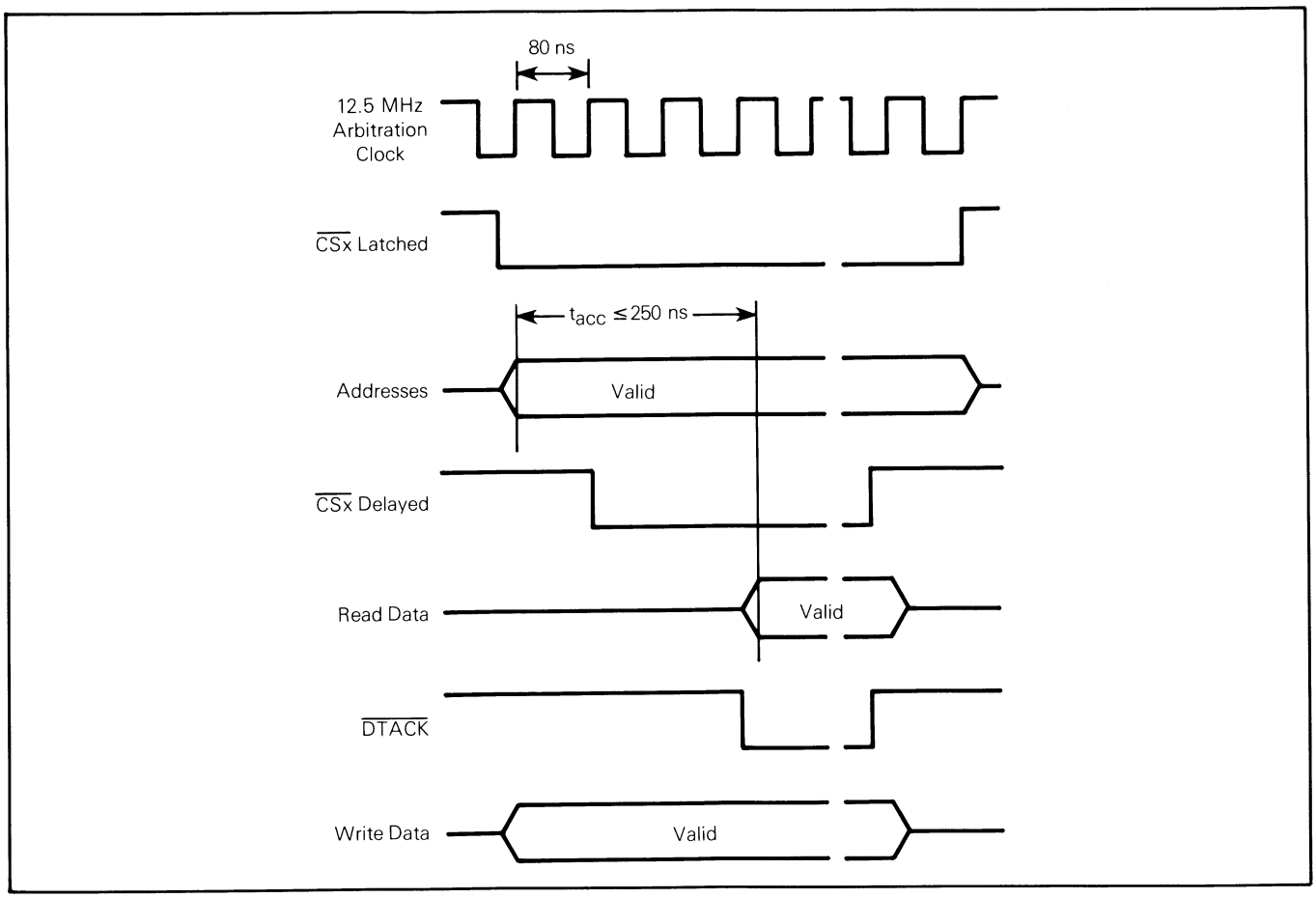

FIGURE 5 — Read/Write Cycle Timing Diagram

Two other programs (listing 2 and listing 3) provide examples of messages passing from a master to a slave processor and vice-versa. These listings are shown in Figure 7 (listing 2 for the master routine) and Figure 8 (listing 3 for the slave routine). The master processor takes characters sent from a terminal and stores them in a buffer until a carriage return is input. The master MPU then sets the semaphore flag and transfers the message from the buffer to the DPR. Upon exiting the DPR, the master MPU sets a message flag (in the low byte of the semaphore register) to indicate a message is ready for the slave. The slave continually checks this message flag to determine if a message is present, and if so prints the message out on its terminal and resets the message flag.

```
MOTOROLA M68000 ASM VERSION  1.30SYS :  12.COLOR   .DPRTEST .SA 06/02/82 13:52:24
DUAL PORT RAM TEST

                        ********************************************************************************
                        *                                                                              *
                        *  THIS ROUTINE TESTS THE DUAL PORT RAM ARBITRATION CIRCUIT.                   *
                        *  THE PROGRAM IS LOADED INTO BOTH BOARDS. THE ADDRESS OF THE DPR              *
                        *  SHOULD BE INITIALIZED AS WELL AS THE VALUE USED FOR THE CHECK FLAG.         *
                        *                                                                              *
                        ********************************************************************************
                        *
            00000900            ORG       $900
                        *
00000900 00000004       DPRBASE  DS.L      1                    ADDRESS OF THE DUAL PORT RAM. THIS VALUE SHOULD
                        *                                       BE INITIALIZED BEFORE THE PROGRAM IS
                        *                                       STARTED.
                        *
                        *
00000904 0D0A0A0A       STRSTART DC.B      $0D,$0A,$0A,$0A
00000908 5441534B2043           DC.B      'TASK COMPLETED.'
00000917 2E             STREND   DC.B      '.'
            0000007E    RAMLEN   EQU       126                  LOCATIONS IN RAM (-1 ONE WORD FOR SEMAPHORE)
                        *                          TAS IS BYTE ONLY; THEREFORE, TWO FLAGS COULD BE USED IN
                        *                          THE WORD SPACE ALLOCATED FOR THE SEMAPHORE.
00000918 00000002       FLAG     DS.W      1                    THIS LOCATION SHOULD BE SET TO THE CHECK VALUE.
                        *                                    THE VALUE SHOULD BE DIFFERENT FOR EACH BOARD
                        *        INITIALIZATION
                        *
0000091A 41F80900       START    LEA.L     DPRBASE,A0           GET ADDRESS OF DPR IN A0
0000091E 2248                    MOVE.L    A0,A1                SET MEMORY POINTER
00000920 5449                    ADD.W     #2,A1                INC FOR SEMAPHORE
00000922 303B0918                MOVE.W    FLAG,D0              GET FLAG
                        *
                        *        MAIN PROGRAM
                        *
00000926 4AD0           ALLOC    TAS       (A0)                 DPR AVAILABLE?
00000928 6BFC                    BMI.S     ALLOC                NO, TRY AGAIN
0000092A 323C007E                MOVE.W    #RAMLEN,D1           COUNTER
0000092E 2449                    MOVE.L    A1,A2                SET POINTER
00000930 34C0           WLOOP    MOVE.W    D0,(A2)+             WRITE A WORD
00000932 51C9FFFC                DBRA      D1,WLOOP             ALL DONE?
00000936 323C007E                MOVE.W    #RAMLEN,D1           RESTORE COUNT
0000093A 2449                    MOVE.L    A1,A2                RESET INDEX
0000093C B05A           CLOOP    CMP.W     (A2)+,D0             CHECK MEMORY
0000093E 56C9FFFC                DBNE      D1,CLOOP             ALL DONE?
00000942 6618                    BNE.S     ERROR                REPORT ERROR
00000944 4250                    CLR       (A0)                 RESET SEMAPHORE
00000946 2A7C00000904            MOVE.L    #STRSTART,A5         SETUP MESSAGE FOR PRINTOUT/ 'TASK
0000094C 2C7C00000917            MOVE.L    #STREND,A6           SETUP MESSAGE FOR PRINTOUT/  COMPLETED'
00000952 1E3C00F3                MOVE.B    #243,D7              'TUTOR' MONITOR (TM) I/O CALL
00000956 4E4E                    TRAP      #14                  'TUTOR' MONITOR (TM) I/O CALL
00000958 6104                    BSR.S     DELAY                DELAY
0000095A 60CA                    BRA.S     ALLOC
0000095C 4E40           ERROR    TRAP      #0
                        *
                        *    SUBROUTINE DELAY
                        *
0000095E 363C0005       DELAY    MOVE.W    #$5,D3               SET COUNT
00000962 4E71           DLOOP    NOP
00000964 51CBFFFC                DBRA      D3,DLOOP             LOOP UNTIL TIME OUT
00000968 4E75                    RTS
```

FIGURE 6 — Listing 1, Dual-Ported RAM Test Program

```
DUAL PORT RAM TEST

                        *
                        *
                        *       THIS PROGRAM SENDS A MESSAGE FROM THE MASTER TO THE SLAVE
                        *       PROCESSOR.  CHARACTERS ARE ENTERED ON THE MASTER CRT UNTIL A
                        *       CARRIAGE RETURN IS DETECTED. AFTER THE CR THE MASTER ROUTINE
                        *       TRANSFERS THE MESSAGE FROM A BUFFER TO THE DPR AND SETS A FLAG.
                        *       THE SLAVE CONTINUALLY READS THE FLAG UNTIL IT IS SET THEN THE
                        *       SLAVE READS THE MESSAGE FROM THE DPR AND DISPLAYS IT ON THE
                        *       SLAVE CRT.
                        *
                        ********************************************************************************
                        *                                                                              *
                        *                       THIS IS THE MASTER ROUTINE.                            *
                        *                                                                              *
                        ********************************************************************************
                        *
            00000850                    ORG       $850
00000850    00000002    COUNT           DS.W      1               NUMBER OF CHARACTERS TRANSFERED
00000852    000000C8    CHARBUFF        DS.W      100             TEMPORARY STORAGE FOR TRANSFERED CHARCTERS
                        *
            0000000D    CR              EQU       $0D
            00020000    SEMAPHRE        EQU       $20000          BASE LOCATION OF THE MASTER'S DPR
            00020001    MSGFLAG         EQU       SEMAPHRE+1      MESSAGE FLAG IS THE LOWER BYTE OF THE SEMAPHORE WORD.
                        *
                        *
0000091A    42780850    START           CLR.W     COUNT           INITIALIZE COUNTER
0000091E    267C00000852                MOVE.L    #CHARBUFF,A3    SET POINTER
00000924    1E3C00F7    INLOOP          MOVE.B    #247,D7         SET UP D7 FOR TRAP CALL TO INPUT ROUTINE
00000928    4E4E                        TRAP      #14             CALL INPUT ROUTINE  ("TUTOR" MONITOR - TM)
0000092A    1E3C00F8                    MOVE.B    #248,D7         SET FOR ECHO CHAR BACK TO TERMINAL
0000092E    4E4E                        TRAP      #14             CALL OUTPUT ROUTINE ("TUTOR" MONITOR - TM)
00000930    0C00000D                    CMP.B     #CR,D0          END OF STRING?
00000934    6708                        BEQ.S     ENDSTRG         GOT STRING
00000936    16C0                        MOVE.B    D0,(A3)+        SAVE CHARACTER
00000938    52780850                    ADDQ      #1,COUNT        INC COUNT
0000093C    60E6                        BRA.S     INLOOP          GET NEXT CHARACTER
0000093E    103C000A    ENDSTRG         MOVE.B    #$0A,D0         LINE FEED
00000942    1E3C00F8                    MOVE.B    #248,D7         OUTPUT LF
00000946    4E4E                        TRAP      #14             CALL OUTPUT    "TUTOR"
00000948    4AF900020000 TASLOOP        TAS.B     SEMAPHRE        DPR AVAILABLE?
0000094E    6BF8                        BMI.S     TASLOOP         NOT YET
00000950    183900020001                MOVE.B    MSGFLAG,D4      GET FLAG
00000956    0C3900FF0002                CMP.B     #$FF,MSGFLAG    ALREADY SET?
            0001
0000095E    660A                        BNE.S     NOTSET          NO
00000960    423900020000                CLR.B     SEMAPHRE        RESET SEMAPHORE
00000966    612C                        BSR.S     DELAY
00000968    60DE                        BRA.S     TASLOOP         TRY AGAIN
0000096A    13FC00FF0002 NOTSET         MOVE.B    #$FF,MSGFLAG    SET MESSAGE FLAG
            0001
00000972    227C00020002                MOVE.L    #SEMAPHRE+2,A1  SET POINTER TO DPR
00000978    32F80850                    MOVE.W    COUNT,(A1)+     STORE COUNT
0000097C    247C00000852                MOVE.L    #CHARBUFF,A2    POINTER TO BUFFER
00000982    12DA        TXLOOP          MOVE.B    (A2)+,(A1)+     MOVE DATA TO DPR
00000984    53780850                    SUBQ      #1,COUNT        DECREMENT COUNTER
00000988    66F8                        BNE.S     TXLOOP          FINISHED?
0000098A    423900020000                CLR.B     SEMAPHRE        RESET SEMAPHORE
00000990    6102                        BSR.S     DELAY           WAIT
00000992    6086                        BRA.S     START           START OVER
                        *
                        *
                        *
                        *       DELAY SUBROUTINE
                        *
00000994    3A3C0005    DELAY           MOVE.W    #$5,D5          SET COUNT
00000998    4E71        DLOOP           NOP
0000099A    51CDFFFC                    DBRA      D5,DLOOP        LOOP UNTIL FINISHED
0000099E    4E75                        RTS
                                        END
```

FIGURE 7 — Listing 2, Dual-Ported RAM Test Program

```
DUAL PORT RAM TEST ROUTINE

                                *
                                *
                                *
                                *       THIS PROGRAM SENDS A MESSAGE FROM THE MASTER TO THE SLAVE
                                *       PROCESSOR.
                                *
                                ********************************************************************************
                                *                                                                              *
                                *                          THIS IS THE SLAVE ROUTINE.                          *
                                *                                                                              *
                                ********************************************************************************
                    00000900            ORG     $900
                                *
                    00FD5400    SEMAPHRE EQU    $FD5400         BASE LOCATION OF SLAVE'S DPR
                    00FD5401    MSGFLAG EQU     SEMAPHRE+1      MESSAGE FLAG IS THE LOWER BYTE OF THE SEMAPHORE WORD
                    00FD5402    COUNT   EQU     SEMAPHRE+2      NUMBER OF BYTES TRANSFERED FROM THE MASTER
                                *
00000900 4AF900FD5400 START     TAS     SEMAPHRE        DPR AVAILABLE?
00000906 6BF8                   BMI.S   START           NOT YET
00000908 103900FD5401           MOVE.B  MSGFLAG,D0      GET FLAG
0000090E 0C0000FF               CMP.B   #$FF,D0         FLAG SET?
00000912 670A                   BEQ.S   MSGAVAIL        YES
00000914 423900FD5400           CLR.B   SEMAPHRE        NO, CLEAR SEMAPHORE
0000091A 612A                   BSR.S   DELAY           WAIT
0000091C 60E2                   BRA.S   START           AND TRY AGAIN
0000091E 423900FD5401 MSGAVAIL  CLR.B   MSGFLAG        RESET MESSAGE FLAG
00000924 4BF900FD5404           LEA     SEMAPHRE+4,A5   SETUP FOR OUTPUT ROUTINE
0000092A 4DF900FD5404           LEA     SEMAPHRE+4,A6   SETUP FOR OUTPUT ROUTINE
00000930 DCF900FD5402           ADDA    COUNT,A6        OFFSET
00000936 1E3C00E3               MOVE.B  #227,D7         FUNCTION CODE FOR OUTPUT ROUTINE
0000093A 4E4E                   TRAP    #14             CALL OUTPUT ROUTINE "TUTOR" MONITOR
0000093C 423900FD5400           CLR.B   SEMAPHRE        CLEAR SEMAPHORE
00000942 6102                   BSR.S   DELAY           SHORT WAIT
00000944 60BA                   BRA.S   START           DO AGAIN
                                *
                                *
                                *       DELAY SUBROUTINE
                                *
00000946 3A3C0005   DELAY       MOVE.W  #$5,D5          COUNT
0000094A 4E71       DLOOP       NOP
0000094C 51CDFFFC               DBRA    D5,DLOOP        LOOP UNTIL FINISHED
00000950 4E75                   RTS
                                END
```

FIGURE 8 — Listing 3, Dual-Ported RAM Test Program

MOTOROLA *Semiconductor Products Inc.*

3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721 • A SUBSIDIARY OF MOTOROLA INC.

A16015-2 PRINTED IN USA 12-84 IMPERIAL LITHO C27122 10,000