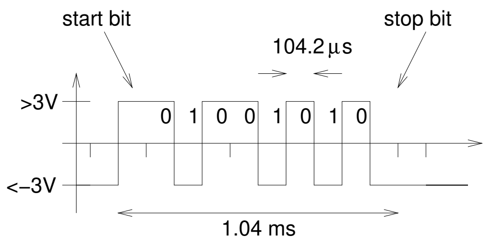
ELEC 464 : MICROCOMPUTER SYSTEM DESIGN
FALL 1995

Assignment 8 - Solutions

Question 1

The ASCII code for the character 'R' is hex $52 (binary 01010010). Bits are transmitted from LS to MS bit with a space-level start bit and a mark-level stop bit:



Question 2

If RTS is an output the first device must be wired up as a DTE. The second device is also a DTE because DTR is an output. The cause of the problem is probably that the two devices are wired as DTE's and thus can't be directly connected. A null modem could be used to connect the two devices.

Question 3

First we remove the '01111110' frame sync sequences from the start and end of the sequence. The remaining bits contain one sequence with five consecutive '1's. The '0' bit following these five '1's must also be removed. The remaining bits are 1011 1010 1111 and 1110 (hex B A F E).

Question 4

The number of clock cycles required to execute the instructions in the subroutine are shown in the following table:

instruction	clock cycles
MOVE.L	12
JSR	20
SUBQ.L	8
BNE.S	10/8
RTS	16

The first two instructions and the RTS (a total of 48 clock cycles) are only executed once. The SUBQ and BNE instructions are executed once per loop (DLY * 18) although the BNE instruction takes 2 fewer clock cycles on the last loop because the branch is not taken (-2).

The total delay in clock cycles is thus

$$\mathtt{DLY} \times 18 + 46$$

. At 8 MHz this is

$$\mathtt{DLY} \times 2.25 + 5.75$$

microseconds.

For a 52.1 microsecond (half a bit period) delay $\mathtt{DLY} = (52.1 - 5.75)/2.25 = 20.6$. DLY=21 would produce the delay closest to the required value. For DLY=21 the delay is 53 microseconds, the absolute error is $|52.1 - 53| = 0.9$ and the percentage error is $0.9/52.1 \approx 2\%$.

The error could be reduced by adding instructions before or after the loop. By adding instructions with an execution time of 4 clock periods (e.g. NOP) the total delay can be increased by multiples of 0.5 microseconds.

The flip-flop is loaded by a bus write cycle to an address of the form 111X XXXX XXXX XXXX XXXX. The write cycle would be caused by a MOVE instruction. The LS bit of the value that is MOVE'd to that address will be on the data bus during the write cycle and will be latched into the flip-flop. The 68000 code to generate the serial data waveform would be as follows:

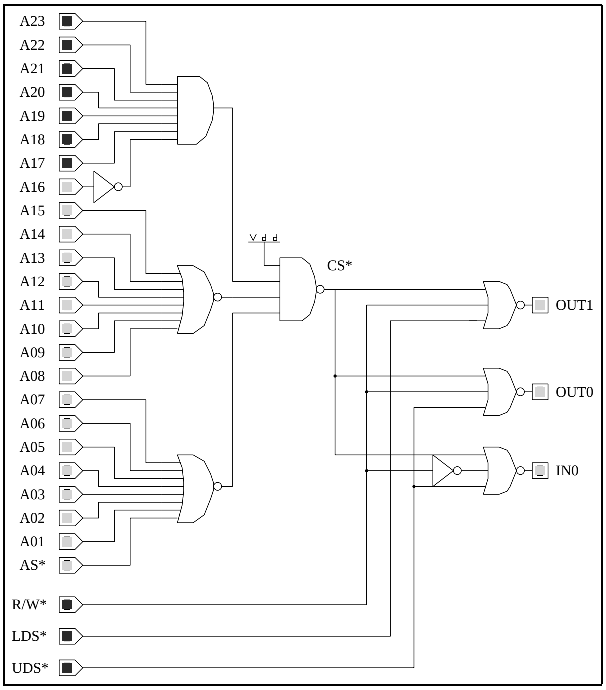
```
DELAY:  RTS                      ; dummy routine

SEROUT: MOVE.L  #$F00000,A0      ; address to output to port
        MOVE    #8,D2            ; number of bits to send out

        MOVE.B  #0,(A0)          ; write start bit to port
        JSR     DELAY            ; wait 1 bit period
        JSR     DELAY

LOOP:   MOVE.B  D1,(A0)          ; write LS bit to port
        JSR     DELAY
        JSR     DELAY
        LSR.B   #1,D1            ; shift in next LS bit
        SUBQ    #1,D2            ; decrement bit count
        BNE     LOOP             ; send next bit if not finished

        MOVE.B  #1,(A0)          ; write stop bit to port
        JSR     DELAY
        JSR     DELAY
        RTS                      ; return to calling routine
        END     SEROUT
```

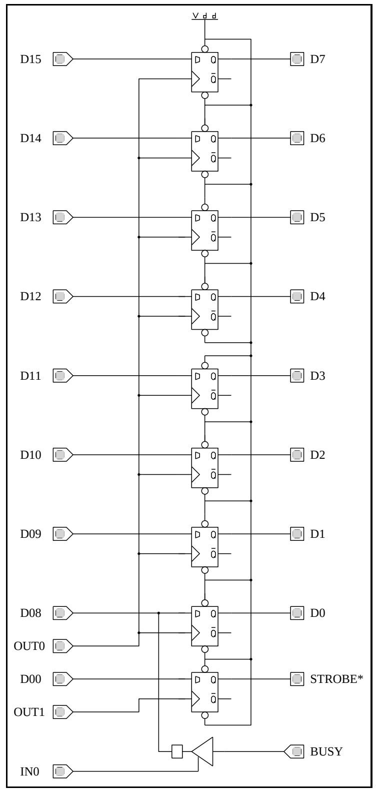
Question 5

Since the flip-flops (latches) for the 8-bit data output port should appear at an even address ($FE0000) they must be connected to D08 through D15. The STROBE* flip-flop which appears on the LS bit of $FE0000 is connected to D00. The flip-flop clock inputs are driven by strobes derived by AND'ing a chip select, R/W* and either UDS* or LDS*.

The input port is a tri-state buffer whose output is connected to D08 (since it appears on the LS bit of the byte at $FE0000) and whose output enable is driven by AND'ing the chip select, the complement of R/W* and UDS*.

The first schematic shows the decoding logic to enable the latch strobes (flip-flop clock inputs) and the tri-state output enable: The second schematic shows the latch details:





The code to output a character (assumed to be in register D1) is as follows:

```
*
* Routine to print a character on a parallel printer port.
* The character to be printed is assumed to be in register D1
* Strobe timing depends on a 68000 running at 8 MHz.
*
DATA    EQU     $FE0000         ; 8-bit parallel output register
STROBE  EQU     $FE0001         ; 1-bit (LS bit only) strobe output
STATUS  EQU     $FE0000         ; 1-bit (LS bit) printer busy status

PRTCHR:
        MOVE    #1,STROBE       ; STROBE* should normally be high

BSY:    MOVE    STATUS,D0       ; read printer status
        AND     #1,D0           ; mask in LS (BUSY) bit
        BNE     BSY             ; wait until printer ready

        MOVE.B  D1,DATA         ; put data on data lines port
        MOVE    #0,STROBE       ; bring STROBE* low for 2 microseconds
        NOP                     ;   assuming 8 MHz 68000
        NOP
        NOP
        NOP
        MOVE    #1,STROBE       ; bring STROBE* high again

        RTS
        END     PRTCHR
```