ASYNCHRONOUS COMMUNICATIONS FOR THE MC68000 USING THE MC6850

Prepared by:
Charles Melear
Microprocessor Applications Engineer

Interfacing the MC6850 Asynchronous Communications Interface Adapter (ACIA) to the MC68000 is easy due to the fact that the MC68000 has a special cycle to handle M6800 peripherals. The ACIA data bus can be placed on either the upper or lower eight bits of the MC68000 data bus with equivalent results. Using the upper byte implies an even address and use of the upper data strobe ($\overline{UDS}$), and the lower byte implies an odd address and the use of the lower data strobe ($\overline{LDS}$). In this application, the ACIA is placed on the lower byte of the data bus.

INTERCONNECTIONS

Enable (E) and R/$\overline{W}$ are connected to the corresponding pins of the MC68000. Several signals are generated to form chip selects as shown in Figure 1. Valid memory address ($\overline{VMA}$) from the MC68000 is an active low signal (as opposed to active high for the MC6800) as well as $\overline{LDS}$. The NOR of the two signals is used to develop CS1. The address $F3FFXX is generated by address lines A8 through A23 to enable a SN74LS154 four-to-sixteen line selector. Address lines A4 through A7 are used to generate a low output at 02 of the SN74LS154 to be used for CS2 of the ACIA. Address line A1 is used for the register select (RS) pin of the ACIA. This puts the ACIA status register at address $F3FF21 and the control register at address $F3FF23. If the ACIA has been placed on the upper byte, the addresses would be $F3FF20 and $F3FF22, respectively. To complete the circuit, a signal called valid peripheral address ($\overline{VPA}$) must be generated and returned to the MC68000 to indicate that a M6800 cycle is being executed. The SN74LS154 has two active low chip enable lines which are driven by the gates that form address $F3FFXX from address lines A8 through A23. Since the SN74LS154 always picks M6800 peripherals, the two chip enable lines can be ORed to develop $\overline{VPA}$. Since more than 16 peripherals could exist, it is best to make the device actually driving the $\overline{VPA}$ line an open collector output so that several gates can be wire ORed.

OPERATION

Operating the ACIA is relatively easy as shown in the flow chart given in Figure 2. Once the control register is set up, the status register is monitored for receive data register full (RDRF) and transmit data register empty (TDRE) indications, as well as error signals and handshake lines. The handshake lines such as request to send ($\overline{RTS}$), clear to send (CTS), and data carrier detect ($\overline{DCD}$) indicate which conditions are present so that the MPU can ascertain when transmission can occur. Once all conditions are ready, transmission or reception or both can begin.

A sample program is given in Figure 3 that shows the MC68000 receiving a character from a terminal through the ACIA and then echoing that character back to the terminal. Essentially, the MC68000 checks to see that transmission and reception can occur. The status register is polled until a character is received. The character is read and then written back to the ACIA for transmission to the terminal as soon as the transmit data register is empty. Of course, any number of subroutines or additional code could be executed before looking for the next character from the ACIA.

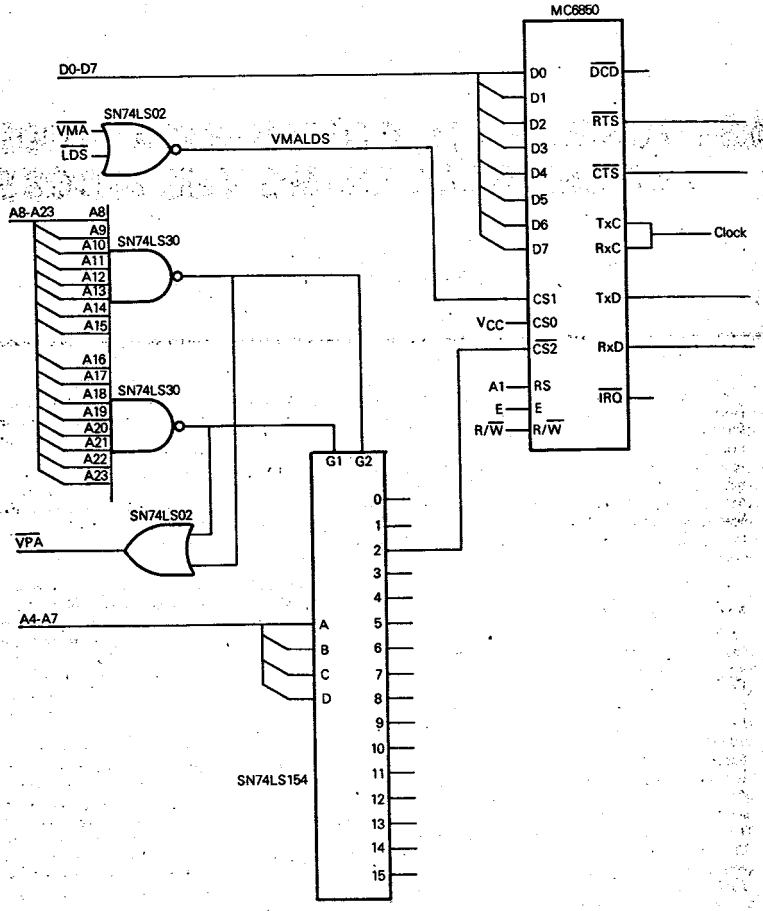

Figure 1. MC6800 to MC6850 Interconnections

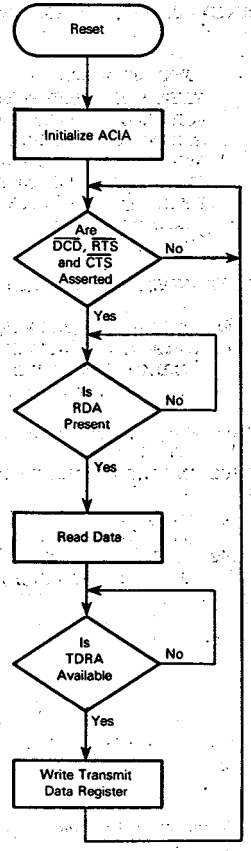

Figure 2. ACIA Operation — Flow Chart

```
 1          00000000      ORG $00000000
 2          00F3FF00      ACIASR EQU $00F3FF00
 3          00F3FF00      ACIACR EQU $00F3FF00
 4          00F3FF02      ACIADR EQU $00F3FF02
 5          00F3FF02      ACIATR EQU $00F3FF02
 6          00020000      SYSTACK EQU $00020000
 7          00000008      RESET EQU $00000008
 8 000000 00020000         DC.L SYSTACK
 9 000004 00000008         DC.L RESET
10 000008 13FC0003
                00F3FF00   MOVE.B #$03,ACIACR RESET ACIA
11 000010 13FC0051
                00F3FF00   MOVE.B #$51,ACIACR INITIALIZE ACIA
12 000018 103900F3FF00 ERROR MOVE.B ACIASR,D0 GET STATUS
13 00001E 0200007C         AND.B #$7C,D0 MASK IRQ,TDRA,RDA
14 000022 66F4             BNE ERROR ANY ERRORS?
15 000024 08390001
                00F3FF00 READS1 BTST #01,ACIASR
16 00002C 66F6             BNE READS1
17 00002E 103900F3FF02     MOVE.B ACIADR,D0 READ CHARACTER
18 000034 08390002
                00F3FF00 READS2 BTST #02;ACIASR IS TDRA SET?
19 00003C 66F6             BNE READS2 LOOP IF NO
20 00003E 13C000F3FF02     MOVE.B D0,ACIATR TRANSMIT CHARACTER
21 000044 60D2             BRA ERROR START OVER
22                         END

****** TOTAL ERRORS   0--   0


SYMBOL TABLE

ACIACR   F3FF00 ACIADR   F3FF02 ACIASR   F3FF00 ACIATR   F3FF02
ERROR    000018 READS1   000024 READS2   000034 RESET    000008
SYSTACK  020000
```

Figure 3. ACIA Operation — Sample Program