EE 308 – LAB 11

Port Expansion for the HCS12

It is sometimes necessary to add additional memory and/or hardware to a microprocessor or microcontroller. Interfaces such as the SPI allow you to add some hardware, it is often necessary to interface directly to the address/data bus. For a microprocessor, which does not have built-in interfaces, the address/data bus is the only way to add additional memory or hardware. In this lab you will will add an external output port to your HC12.

Figure 1 shows a block diagram for adding an external output port to the HCS19. We will implement the port in an Altera 7064 PLD. Note that you will have to connect the 16-bit multiplexed address/data bus and three control lines from your HCS12 to you Altera chip. You will also have to connect the eight bits of your output port to your LEDs to verify that the port is working.

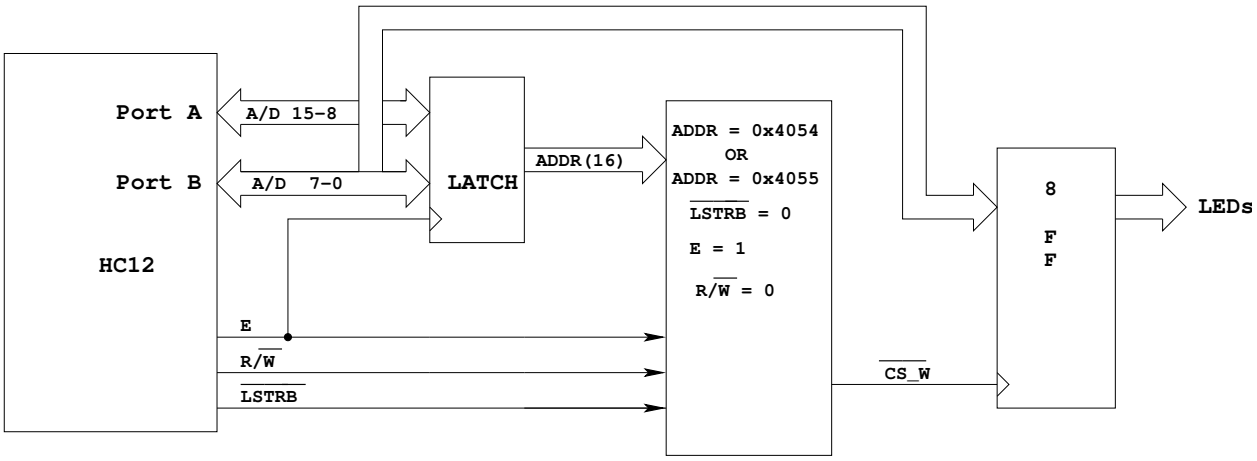

Figure 1. Block diagram of HCS12 output port at address 0x4055.

1. Write an Altera program to implement the output port.
2. Using the pinout diagram from the `.rpt` file, wire the Altera chip to your HCS12. Note that there will be a lot of wires to run, so it is essential that you are neat in your wiring.
3. Check the functioning of your port using DBug12. When you start your HCS12 using DBug12, the HCS12 is in single chip mode. In this mode you can manipulate `AD15-0`, `E`, `R/`$\overline{\texttt{W}}$ and $\overline{\texttt{LSTRB}}$ as general purpose I/O lines. You can use the `MM` command of DBug-12 to write data to the output port by changing `AD15-0` (PORTA and PORTB), `E`, `R/`$\overline{\texttt{W}}$ and $\overline{\texttt{LSTRB}}$ in the same sequence that the HCS12 would if it were in expanded wide mode. Look at Chapter 12 of the HCS12 Core Users Guide for more information.
   (a) Use the DDRE register to make `E`, `R/`$\overline{\texttt{W}}$ and $\overline{\texttt{LSTRB}}$ output pins. (Note: `E` is bit 4 of PORTE, `R/`$\overline{\texttt{W}}$ is bit 3 of PORTE, and $\overline{\texttt{LSTRB}}$ is bit 2 of PORTE.)
   (b) Bring `E` low by writing to PORTE.
   (c) Put 0x4055 on PORTA and PORTB.
   (d) Bring `R/`$\overline{\texttt{W}}$ and $\overline{\texttt{LSTRB}}$ low.

(e) Bring `E` high.

(f) Put the data you want to write to the port on PORTB.

(g) Bring `E` low.

When you bring `E` low in the final step, the data should appear on the LEDs. If this happens, your Altera code and your wiring is correct.

4. Now you will switch the HCS12 into expanded wide mode, so the HCS12 can write directly to the expanded output port using its address/data bus. Unfortunately, DBug-12 does not work when the chip is put into expanded wide mode. To verify that the expanded port works, you will have to put your program into EEPROM, along with the instructions needed to switch the HCS12 into expanded wide mode. Then by having the HCS12 execute EEPROM code after reset, you will be able to see that the expanded port works properly. Also, our HCS12 has no region of free memory in which to map our new port. We can use the MISC register to turn off the Flash EEPROM from 0x4000 to 0x7fff, which will give us a region in which to put our port. The following program will do all this:

```
PEAR:   equ $000a
MODE:   equ $000b
EBICTL: equ $000e
INITRM: equ $0010
MISC:   equ $0013
SYNR:   equ $0034
REFDV:  equ $0035
CRGFLG: equ $0037
CLKSEL: equ $0039
COPCTL: equ $003c
ARMCOP: equ $003f

CODE:  section .text
       org $400

       ldaa    #$55     ; Reset COP
       staa    ARMCOP
       coma
       staa    ARMCOP
       clr     COPCTL   ; Turn off COP
       ldab    #$11     ; Map RAM into proper location
       nop
       stab    INITRM
       ldab    #$00     ; Set clock reference divider to 0
       stab    REFDV
       ldab    #$03     ; Set PLL to multiply oscillator clock by 4
       stab    SYNR
       nop              ; wait
       nop
       nop
       nop
```

```
l1:     brclr   CRGFLG,#$08,l1  ; Wait for PLL to lock
        bset    CLKSEL,#$80     ; Switch to PLL clock

        ldab    #$e4     ; Expanded wide mode, internal visibility on
        stab    MODE
        ldab    #$0c     ; Turn on R/W, LSTRB
        stab    PEAR
        ldab    #$01     ; Use E-clock to control external bus
        stab    EBICTL
        ldab    #$03     ; No E-clock strech, disable ROM from 4000-7fff
        stab    MISC
```

Add to the above program code which will increment the external port, with a delay between incrementing. Implement the delay using a software loop as you did in Lab 4.

Note that you cannot increment your port through a simple command such as

```
        inc   $4055
```

because you would have to be able to read from your port as well as write to your port for this to work. Instead, keep track of the value to write in a memory location, increment this variable, and write it to address 0x4055.

5. A HCS12 with a functioning expansion port will be available at one of the logic analyzers during lab this week. Writing to address 0x4055 will write data to the port. Reading from address 0x4054 will read data from the port. The HCS12 is running the following loop:

```
loop:   ldaa    $4054
        inca
        staa    $4055
        bra     loop
```

The label `loop` is at address 0x0480.

(a) Hand-assemble this program to determine the op codes and op code addresses.

(b) Use the logic analyzer to grab data from the HCS12 address/data bus. Identify the memory cycle which reads data from address 0x4054, and the memory cycle which writes data to address 0x4055. Note that the logic analyzer has only 16 data lines. The HCS12 address/data bus uses 19 lines — `AD15-0` and the three control line `E`, `R/`$\overline{\texttt{W}}$, and $\overline{\texttt{LSTRB}}$. The HCS12 will either be fetching instructions from EEPROM (address 0x0400-0xfff), or accessing the external port (0x4054 or 0x4055). Thus, adress bits `D15`, `D13` and `D12` will always be zero.

Figure A-9 of the MC9S12DP256B Device Users Guide shows the external bus timing. As best you can, measure the following times. The numbers in parentheses are the labeled numbers on Figure A-9 and Table A-20. Compare the numbers to the values listed in Table A-20.

i. Cycle time (2)

ii. Pulse width, E low (3)
iii. Pulse width, E high (4)
iv. Address dealy time (5)
v. Muxed address hold time (7)
vi. Write data hold time (13)
vii. Read/write delay time (24)
viii. Read/write hold time (26)
ix. Low strobe delay time (27)
x. Low strobe hold time (29)

Pre-Lab

1. Write a preliminary Altera program to implement the expanded ports.
2. Hand assemble the instructions of Part 5.