EE 308 – LAB 12

Port Expansion for the HC12

In this lab you will will add external ports to your HC12. These bi-directional expansion ports will replace Ports A and B which are no longer available when the HC12 is used in expanded mode. The ports will be added into the Altera EPM7128S chip on the memory expansion daughterboard. The daughterboard has two sets of eight-pin headers which will allow you to access these ports. We will call these new ports Expanded Port A and Expanded Port B.

A possible design for the expansion ports was discussed in class, and the GDF files for this design are in the lecture notes for Wednesday, April 10. (You could also do the entire project in an Altera TDF file.) In addition to the GDF files, you will need files to generate the signals which indicate that the HC12 is doing a read from or a write to addresses 0x0400, 0x0401 or 0x0402. Sample TDF files for 0x0400 are included in the lecture notes as well.

The schematic from Lab 11 shows how the Altera chip is connected to the two header blocks. You will need to assign pins in the top-level GDF file to correspond to the pinouts shown on the schematic.

Note: The IRQ line on the Altera chip should be set up as an output pin, and should be connected to VCC. This pin is included so a future design could use the Altera chip to interrupt the HC12.

Pre-Lab

For the pre-lab, complete the files for reading from and writing to addresses 0x0400, 0x0401, and 0x0402. Make sure you handle the A0 and LSTRBn lines correctly. Also, finish (or at least get a good start on) the other files needed for the port expansion.

The Lab

1. Complete the programming for the expansion ports. Download this into the Altera chip.
2. Connect wires from Expanded Port A (from now on called Port EA) to the LEDs on your breadboard. Set up Port EA for output by setting bit 0 of address 0x402. To do this, do the following in DBug-12: Change the value of address 0x0402 to 0x01, Then use Port EA to turn on and off the LEDs on the breadboard. Do this in DBug-12 by changing the value of address 0x0400.
3. Check out your Expanded Port B (Port EB) in the same way. To make Port EB an output, set bit 4 of address 0x0402. You should be able to use Port EB to turn on and off LEDs by writing to address 0x0401.
4. Verify that Port EA and Port EB can be written to with either 8-bit writes or with 16-bit writes. For example, the following instruction will do an eight-bit write to Port EA:

   ```
   movb   #$aa,$0400
   ```

   while this instruction will do a 16-bit write to both Port EA and Port EB:

```
movw   #$aa55,$0400
```

5. Check to make sure that Port EA works for input.

6. Modify your C program from Lab 6 to use Port EA rather than Port A. Include `#define`s in your C program so you can refer to Port EA, Port EB, and the expanded data direction register by name rather than by address. Run your program and verify that the proper LED pattern is displayed on Port EA.