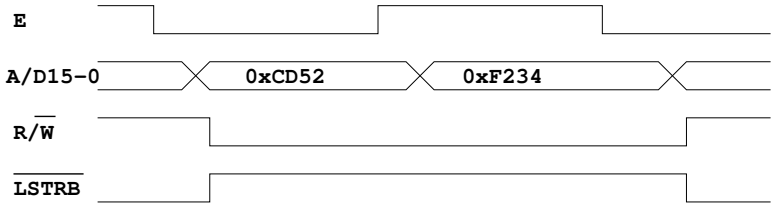
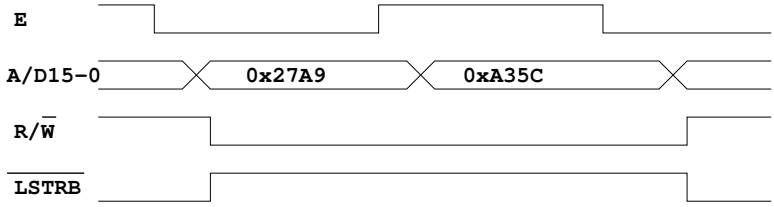
EE 308 – Homework 11

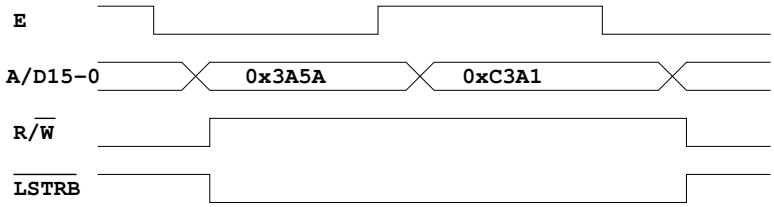
Due Apr. 12, 2010

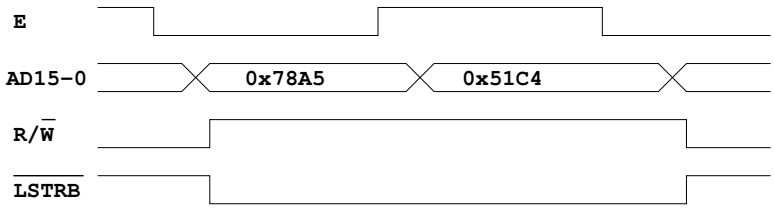
For all problems below assume your are using a MCS12DP256 chip with a 24 MHz bus clock and a 8 MHz oscillator clock.

1. The figures below show some things which might be on the MC9S12 bus in normal expanded wide mode. For each figure, indicate if that combinations of signals can occur. If so, explain what the memory cycle does — read or write, 8-bit or 16-bit access, what data is read from or written to, what memory address(es) are accessed. If the combination of signals cannot occur, explain why not.







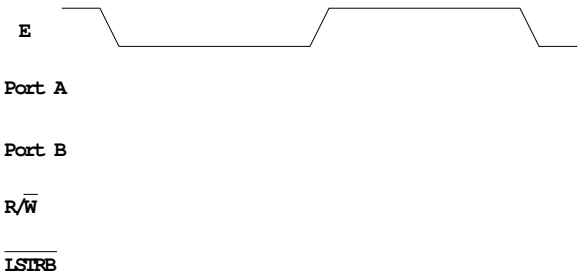


2. The following table shows some values in the MC9S12 memory:

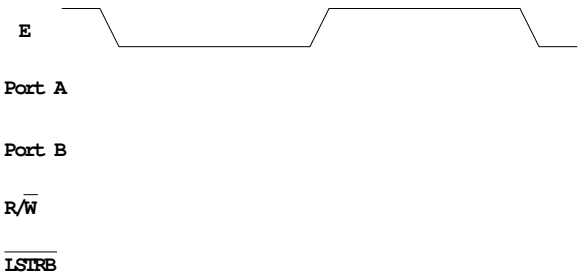
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4000	A2	5C	4A	9F	5B	86	C6	03	5B	8D	C6	FF	5B	02	4C	80

Show what will be on the address/data bus and the control lines when the MC9S12 does the following:

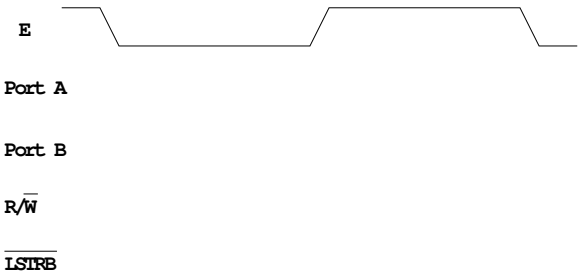
(a) Writes a 0x55 to address 0x4000.



(b) Writes a 0xABCD to the two bytes at addresses 0x4002 and 0x4003.



(c) Reads a single byte from addresses 0x4009.



3. Immediately upon coming out of reset, an MC9S12 is operating in Normal Expanded Wide mode. How did the MC9S12 know it should run in this mode — i.e. what pins did it check, and what was the state of those pins?

4. Immediately upon coming out of reset, an MC9S12 is operating in Normal Single Chip mode. How did the MC9S12 know it should run in this mode — i.e. what pins did it check, and what was the state of those pins?

5. Immediately upon coming out of reset, an MC9S12 is operating in Normal Single Chip mode. How can you switch the chip into Normal Expanded Wide mode? Write some code to do this.

6. Write a C instruction which remaps the EEPROM to start at address 0x5800.

7. You want to use the EE 231 FPGA board with its Altera EPM2210F324C3 chip to implement an 8-bit input port at address 0x4000, and an 8-bit output port at address 0x4001. Write a Verilog file which does the following:

   (a) Demultiplexes the address from the data. It generates the output lines A15-0. The inputs are AD15-0 and the E clock.

   (b) Does the address decoding:

   - Generates an output `cs_r_4000` which goes low when the MC9S12 does a read from the byte at memory address 0x4000.
   - Generates an output `cs_w_4001` which goes low when the MC9S12 does a write to the byte at memory address 0x4001.
   - Generates an output `cs_r_4001` which goes low when the MC9S12 does a read from the byte at memory address 0x4001.

   (c) Latches the appropriate eight bits of data into a set of eight D flip-flops when on the falling edge of E when `cs_w_4001` is low.

   (d) Drives the outputs of the eight D flip-flops onto the appropriate eight bits of data when `cs_r_4001` is low.

   (e) Drives the outputs of four input pins (to be connected to the four switches on the EE 231 board) and four zeros onto the appropriate eight bits of data when `cs_r_4000` is low.

   The inputs to the device should be address/data lines AD[15:] (the Port A and Port B lines when the MC9S12 is in single chip mode), the E, R/$\overline{\text{W}}$, and $\overline{\text{LSTRB}}$ control lines, and the four signals connected to the four switches. The outputs should be the outputs of the eight D flip-flops, to be connected to the eight LEDs on the EE 231 board.

   Here is the start of the Verilog code:

```
module port_exp (
   input e, rw, lstrb,clk,      /* Control lines from MC9S12 */
   inout [15:0] mux_ad,         /* Multiplexed address/data lines from MC9S12 */
   input [7:0] my_in_port,      /* 8-bit input port at address 0x4000 */
   output reg [7:0] my_out_port /* 8-bit output port at address 0x4001 */
   output cs_r_4000;            /* Goes low when MC9S12 reads from 0x4000 */
   output cs_r_4001;            /* Goes low when MC9S12 reads from 0x4001 */
   output cs_w_4001;            /* Goes low when MC9S12 writes to 0x4001 */
);

reg [15:0] address;

...

endmodule
```