

EE 308
Exam 3
April 26, 2002

Name: _____

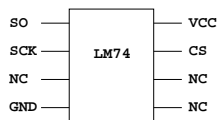
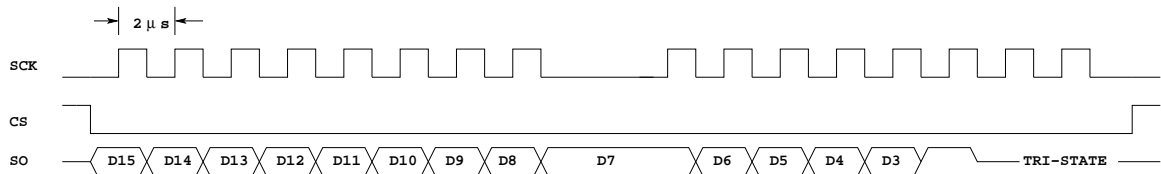
You may use any of the Motorola data books, the overheads posted on the Internet, your notes and you lab book. Show all work. Partial credit will be given. No credit will be given if an answer appears with no supporting work.

For all the problems in this exam, assume you are using an HC12 with a 16 MHz crystal, resulting in a 8 MHz processor clock.

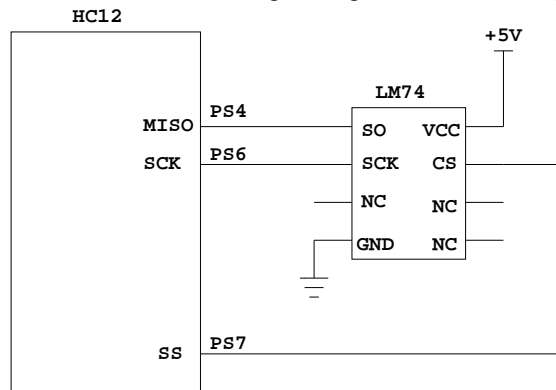
Also, assume that hc12b32.h has been included, so you can refer any register in the HC12 by name rather than by its address in any C code you write.

- The LM74 is a temperature sensor manufactured by National Semiconductor. It converts the temperature into a 13-bit two's complement number, where each bit represents 0.0625°C. The data is sent out over the SPI port left-justified, with the last three bits unused. Thus 25°C will be represented by the number 0x0C87 — shift this right by three bits to get 0x0190, or 400₁₀, for a temperature of 400 × 0.0625°C = 25°C. Similarly, -25°C will be represented by the number 0xF387. (Shift right 3 bits to get 0xFE70, or -400₁₀.)

The figure below shows the SPI data format for an LM74. SO is the serial data output, SCK is the serial data clock, and CS is the slave select for the LM74.



- Draw a simple diagram showing how you would connect the HC12 to the LM74 to use it. You can refer to the name of the HC12 signal (e.g., MOSI) or to the port pin (e.g., PS5) in your diagram.



- (b) Write some C code to set up the HC12 to communicate with the LM74. Be sure to explain how you set up the HC12 registers and why.

```

DDRS = DDRS | 0xE0;      /* SS, MOSI, SCK outputs */
PORTS = PORTS | 0x80;    /* Deselect slave */
SP0CR1 = 0x50;          /* 0 1 0 1 0 0 0 0
                        | | | | | | | |
                        | | | | | | | | _ MSB first
                        | | | | | | | | \_ Don't use auto SS
                        | | | | | | | | \_ CPHA = 0 (1st edge)
                        | | | | | | | | \_ CPOL = 0 (clock idle low)
                        | | | | | | | | \_ Master mode
                        | | | | | | | | \_ No wired OR more
                        | | | | | | | | \_ Enable SPI
                        | | | | | | | | \_ No interrupts on SPI
*/
SP0CR2 = 0;             /* Normal (not bidirectional) mode */
SP0BR = 3;              /* 500 kHz (2 us period) */

```

- (c) Write some C code to read the temperature from the LM74 and assign it to a 16-bit variable called temp. (You do not have to shift the data or calculate the temperature in °C, just write the raw 16-bit data to temp.)

Need to read two bytes from LM74, and put first byte in upper 8 bits of 16-bit number, second byte in lower 8 bits of 16-bit number

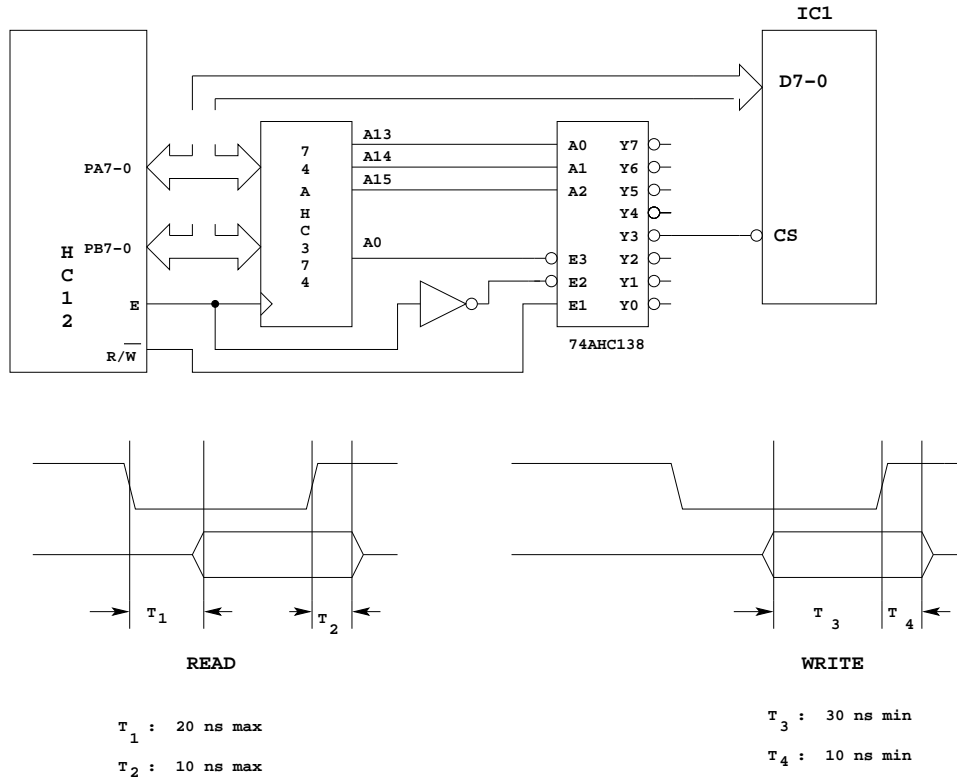
```

int temp;

PORTS = PORTS & ~0x80;    /* Select LM74 */
SPODR = 0;                /* junk byte to make SCK run */
while ((SP0SR & 0x80) == 0) ; /* Wait for transfer to finish */
temp = SP0DR;              /* Read 1st byte */
SPODR = 0;                /* junk byte to make SCK run */
while ((SP0SR & 0x80) == 0) ; /* Wait for transfer to finish */
temp = (temp << 8) | SP0DR; /* Put 1st byte in upper 8 bits, 2nd */
                             /* byte in lower 8 bits */
PORTS = PORTS | 0x80;    /* Deselect slave */

```

2. An engineer drew a quick sketch of an IC interfaced to the HC12. She accidentally spilled some coffee on the sketch, and some details were lost. On the same piece of paper she drew the timing diagram for an input IC and an output IC, but forgot to label which diagram corresponds to IC1 interfaced to the HC12. The figure below shows her sketch:



- (a) Is IC1 an input or an output port? Explain.

A bubble on the enable indicates input is active low; no bubble indicates input is active high. R/W is connected to an enable without a bubble, so the 74AHC138 chip will be enabled only if R/W is high. This means the HC12 is reading from IC1, so IC1 is an input device. The READ timing is the appropriate timing diagram.

- (b) Should the data lines of IC1 be connected to the Port A or the Port B pins? Explain.

PORTA ---
 A0 is connected to an active low enable of the 74AHC138, so the 74AHC138 is enabled only when A0 is low. This means IC1 is enabled on accesses to even addresses only. Even addresses are the high bytes (AD15-8) of a 16-bit data word; high bytes of data use PORT A

- (c) For what range of addresses will IC1 be selected? Explain.

IC1 chip select is connected to Y3 output of the 74AHC138. The Y3 output will be low when A2 A1 A0 of the 74AHC138 are 011 (decimal 3). This means A15 = 0, A14 = 1, A13 = 1. Also, A0 must be 0, as discussed above. Therefore IC1 is selected on reads from even addresses with the address bits:

A15	A14	A13	A12		A11	A10	A9	A8		A7	A6	A5	A4		A3	A2	A1	A0
0	1	1	X		X	X	X	X		X	X	X	X		X	X	X	0

If all the X's are 0's, the address is 0x6000

If all the X's are 1's, the address is 0x7FFE

The chip will be selected on reads from any even address in the range 0x6000 to 0x7FFE

- (d) If IC1 is an input port, write some C code to read a byte of data from IC1 and save it in a variable called temp. If IC1 is an output port, write some C code to write a 0x55 to IC1.

IC1 is an input port, so read a byte and assign to a variable called temp:

```
char temp;
```

```
temp = *(char *)0x6000;
```

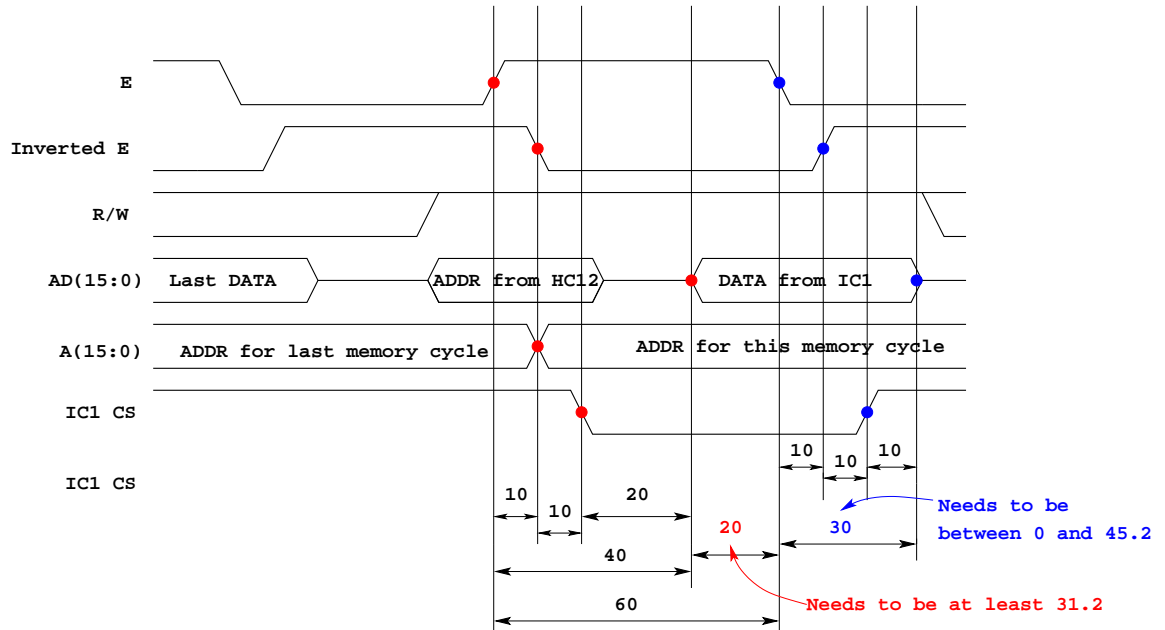
OR

```
#define IC1 (* (char *)0x6000)
```

```
char temp;
```

```
temp = IC1;
```

(e) Is the timing of IC1 compatible with an HC12 with an 8 MHz E-clock, and no E-clock stretches? Explain. (Assume the propagation delays through each glue logic chip is 10 ns.)



Data Setup Time

HC12 reads data from IC1

IC1 supplies data to HC12

Determine how long from time IC1 gets data on data bus until E goes low to meet HC12's data setup time

- 1) E goes high to Inverted E goes high: 10 ns
- 2) E goes high to New Addr on Addr Bus: 10 ns
- 3) Inverted E goes high and New Addr on Bus to CS (Y3 of 74AHC138) low: 10 ns
(Note: R/W changes before Inverted E, so we don't have to worry about it)
- 4) CS low to IC1 data on Data Bus: 20 ns
- 5) E high to E low: 60 ns (Circled number 3 on HC12 timing diagram)
- 6) Therefore, IC1 data on Data Bus to E low: 60 ns - 20 ns - 10 ns - 10 ns
IC1 data on Data Bus to E low: 20 ns

HC12 needs data on data bus at least 31.2 ns before E goes low (circled number 11 on HC12 timing diagram)

Data from IC1 on data bus only 20 ns before E goes low

Not soon enough: Doesn't work

IC1 plus glue logic is NOT compatible with HC12 at 8 MHz with 0 clock stretch

Data Hold Time

HC12 reads data from IC1

IC1 supplies data to HC12

Determine how long from time E goes low until IC1 removes data from data bus to meet HC12's Data Hold time

- 7) E low to Inverted E low: 10 ns
- 8) Inverted E low to CS high: 10 ns
- 8) CS high to IC1 removes data from data bus: 10 ns
- 10) E low to IC1 removes data from data bus: $10 \text{ ns} + 10 \text{ ns} + 10 \text{ ns} = 30 \text{ ns}$

HC12 needs data on data bus at least 0 ns after E goes low
(Circled number 12 on HC12 timing diagram)

HC12 needs data to be removed from data bus no longer than 45.2 ns after E goes low (Circled number 10 on HC12 timing diagram)

E low to data off bus must be between 0 ns and 45.2 ns

Actual hold time is 30 ns, so Data Hold Time meets specs

(f) Is the timing of IC1 compatible with an HC12 with an 8 MHz E-clock, and one E-clock stretch? Explain.

Data Setup Time:

- 1) Same as Part (e)
- 2) Same as Part (e)
- 3) Same as Part (e)
- 4) Same as Part (e)
- 5) E high to E low: $60 \text{ ns} + 125 \text{ ns} = 185 \text{ ns}$ (1 clock stretch)
- 6) Therefore, IC1 data on Data Bus to E low: $185 \text{ ns} - 20 \text{ ns} - 10 \text{ ns} - 10 \text{ ns}$
IC1 data on Data Bus to E low: 145 ns

HC12 needs data on data bus at least 31.2 ns before E goes low
(circled number 11 on HC12 timing diagram)

Data from IC1 on data bus 145 ns before E goes low

Soon enough: Does work

Data Hold Time:

Everything is same as Part (e), so this works as well

IC1 plus glue logic is compatible with HC12 at 8 MHz with 1 clock stretch