EE 308

Exam 3

April 30, 2010

Name: ____

You may use any of the class handouts and one page of notes. 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 MC9S12DG256 with an 8 MHz oscillator clock and a 24 MHz bus clock.

Also, assume that `derivative.h` has been included, so you can refer any register in the MC9S12 by name rather than by its address in any C code you write.

1. Two sensors are connected to the A/D converter of an MC9S12. V_{RL} is connected to 0 V and V_{RH} is connected to +5 V. The sensor connected to PAD4 measures the temperature of an oven. A temperature of 0°C gives a voltage of 0 V. A temperature of 500°C gives a voltage of 5 V. The sensor is linear from 0 to 500°C. The sensor connected to PAD5 measures the amount of oxygen in the oven. 0% oxygen gives a voltage of 0 V. 100% oxygen gives a voltage of 5 V. The sensor is linear from 0 to 100%.

   (a) Write some C code to set up the A/D converter to operate in 10 bit mode, right-justified, unsigned, and to convert all 8 A/D channels one time, then stop.

```
ATD0CTL2 = 0x80;    // Power up A/D, no interrupts
ATD0CTL3 = 0x00;    // 8 conversions
ATD0CTL4 = 0x05;    // 10 bit, maximum speed
ATD0CTL5 = 0x90;    /* 1 0 0 1 0 0 0 0
                       | | | |   \___/
                       | | | |     |
                       | | | |     \__ Start on Channel 0
                       | | | \________ MULT = 1 => multiple channels
                       | | \__________ Scan = 0 => Convert once
                       | \____________ DSGN = 0 => unsigned
                       \______________ DJM  = 1 => right justified
                     */
```

   (b) Write some C code to wait until the above set of 8 conversion has finished.

```
while ((ATD1STAT0 & 0x80) == 0 ) ;    /* Wait for sequence to finish */
```

(c) The manufacturing process requires the oven temperature to remain between 250°C and 252°C. What is the output of the A/D converter when the temperature is 250°C? What is the output of the A/D converter when the temperature is 252°C?

500°C = 5.00 V, so 250°C = 2.50 V and 252°C = 2.52 V.

250°C: 2.50 V $\times$ $(2^{10}$-1)/5 V = 512_{10} = 0x200.
252°C: 2.52 V $\times$ $(2^{10}$-1)/5 V = 516_{10} = 0x204.

For 250°C, the output will be 0x0200 in ATD0DR4.
For 252°C, the output will be 0x0204 in ATD0DR4.

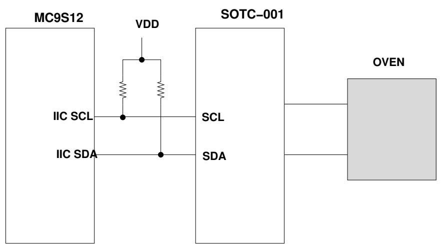
(d) After the set of eight conversions, the eight 16-bit result registers are as follows:

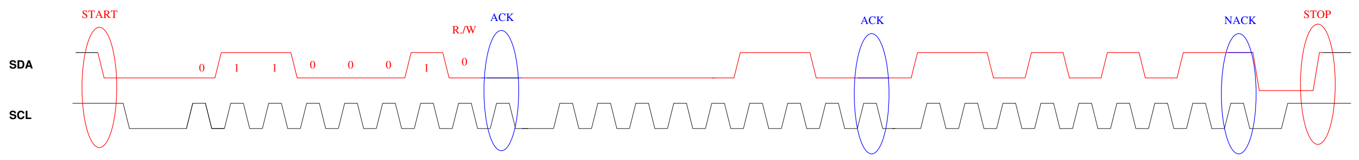
ADR0	ADR1	ADR2	ADR3	ADR4	ADR5	ADR6	ADR7
0x0124	0x0234	0x034C	0x0358	0x0160	0x0278	0x0388	0x01AC

What is the percentage of oxygen in the oven?

The oxygen sensor is connected to PAD5 so the result is in Register ADR5. 0x0278 = 632_{10}.
V = 5 V*632/$(2^{10}$-1) = 3.089 V.
3.086 V/5 V $\times$ 100% = 61.8% oxygen.

2. The figure below shows an SOTC-001 Super Oven Temperature Controller connected to a MC9S12 over its IIC bus. The MC9S12 can write to the SOTC-001, and can read from it. When the MC9S12 writes to the SOTC-001, it writes a 16-bit number (in two bytes), which tells the SOTC-001 what temperature to set the oven at. The value of the 16-bit number, which must be between 0 and 2500, is the termperature to the the oven at in Farenheit. When the MC9S12 reads from the SOTC-001, the SOTC-001 returns a 16-bit number (in two 8-bit bytes), which tells the MC9S12 the actual temperature of the oven (between 0°F and 2500°F). The figure shows some activity on the IIC bus.





(a) The spec sheet for the SOTC-001 show that the maximum SCL frequency is 200 kHz, the low period of the SCL clock is a minumum of 2.3 μs, and the high period of the SCL clock is a minumum of 2.0 μs. How would you set up the MC9S12 IIC hardware to have as high of a IIC speed as possible which meets these values?

A 200 kHz square wave has a period of 5 μs, so it will be low for 2.5 μs and high for 2.5 μs. You just need to make sure the frequency is 200 kHz or lower. 24 MHz/200 kHz = 120, so the clock divider needs to be 120 or larger. You can get a clock divider of 120 by writing an 0x85 to the IBFD register: `IBFD = 0x85;`

(b) On the data transfer diagram above, is the MC9S12 writing to the SOTC-001 or is it reading from the SOTC-001? Explain.

The last bit of the first byte sent is low, so the MC9S12 is writing to the SOTC-001.

(c) From the data transfer diagram above, what is the 7-bit address of the SOTC-001?

The 7-bit address is the first 7 bits of the first byte, or 0110001_2 = 0x31.

(d) On the data transfer diagram above, indicate the following:

i. The Start condition. The Start condition is when SDA goes high-to-low when SCL is high.

ii. The Stop condition. The Stop condition is when SDA goes low-to-high when SCL is high.

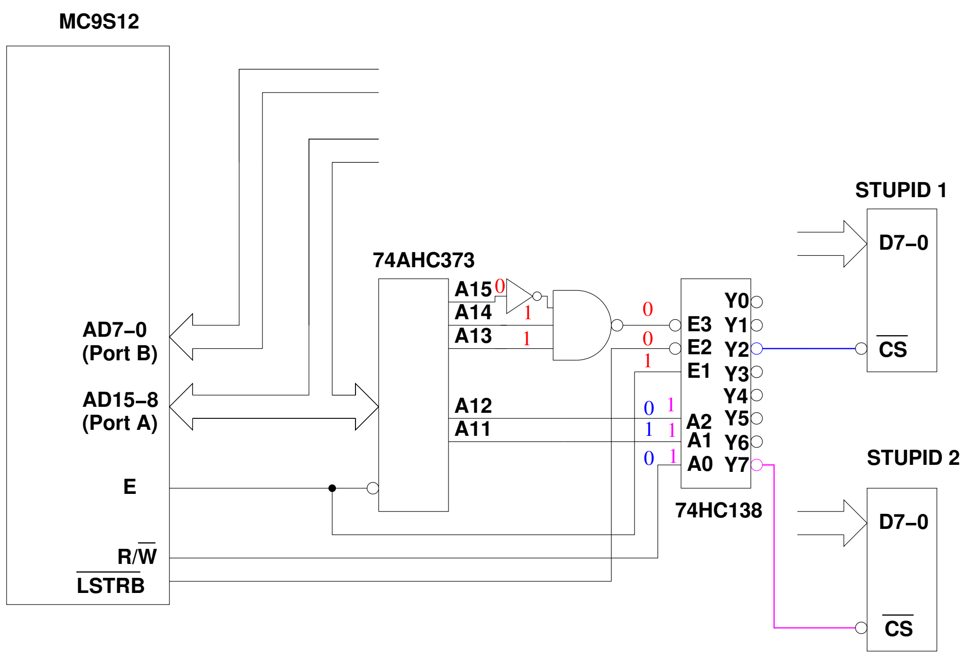
iii. Any Acknowledge (ACK) signal. ACK is when the acknowledging device holds SDA low on the 9th clock cycle of a data transfer.

iv. Any Not Acknowledge (NACK) signal.NACK is when the acknowledging device holds SDA high on the 9th clock cycle of a data transfer.

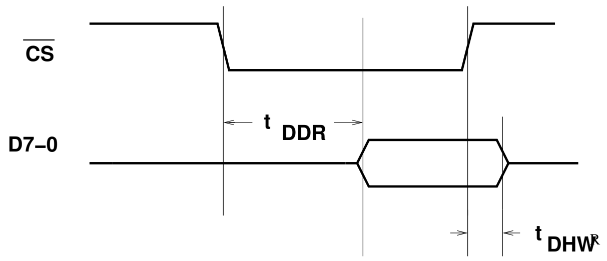
v. Indicate which device is controlling the data bus.

These are indicated on the timing diagram – red for master and blue for slave. The only time the slave controls the data bus when the master writes to it is when it provides the ACK and NACK signals.

3. A new startup company designs a series of peripheral chips they call the Super Terrific Universal Peripheral Interface Device (STUPID). The figure below shows two STUPID chips connected to an MC9S12. One is an input chip, the other is an output chip.



STUPID READ BUS TIMING



t_{DDR} =	40 ns Min	Data Delay Time, Read
t_{DHR} =	5 ns Min	Data Hold Time, Read

(a) What range of addresses will select STUPID 1 chip? Is it an input chip or an output chip.

STUPID 1 is selected when A2-A0 of the 74AHC138 chip is 2 (010) and the 74AHC138 is selected. For this to happen, need A15=0, A14=1, A13=1, A12=0, A11=1 and R/W = 0, LSTRB = 0 and E = 1. The 16-bit address will be:

0110 1XXX XXXX XXXX

The range will be 0x6800 through 0x6FFF with LSTRB low. This will occur on any 8-bit write to an odd address in this range and any 16-bit write to an even address in this range. Because R/W must be low, this is an output chip.

(b) Should `D7-0` of STUPID 1 be connected to Port A or Port B? Why?

LSTRB must be low, so this is the low byte – STUPID 1 should be connected to AD7-0, or Port B.

(c) What range of addresses will select STUPID 2 chip? Is it an input chip or an output chip.

STUPID 2 is selected when A2-A0 of the 74AHC138 chip is 7 (111) and the 74AHC138 is selected. For this to happen, need A15=0, A14=1, A13=1, A12=1, A11=1 and R/W = 1, LSTRB = 0 and E = 1. The 16-bit address will be:

0111 1XXX XXXX XXXX

The range will be 0x7800 through 0x7FFF with LSTRB low. This will occur on any 8-bit read from an odd address in this range and any 16-bit read from an even address in this range. Because R/W must be high, this is an input chip.

(d) Should `D7-0` of STUPID 2 be connected to Port A or Port B? Why?

LSTRB must be low, so this is the low byte – STUPID 1 should be connected to AD7-0, or Port B.

(e) Is $\mathtt{t}_{\mathtt{DDR}}$ for the input chip compatible with the MC9S12? Why or why not? What is the corresponding time for the 24 MHz MC9S12?

Zero clock stretch: E goes high, 2 ns later CS goes low, 40 ns later, STUPID 2 drives data on the bus, so 42 ns after E goes high, the data is available. The MC9S12 needs the data 13 ns before E goes low. With zero clock stretch E goes low 21 ns after E goes high, so the MC9S12 needs the data on the bus 21 ns-13 ns=8 ns after E goes high. The data isn't there until 42 ns after E goes high, so this will not work.

With one clock stretch, E goes low 63 ns after E goes high (clock stretch increases time E is high by one clock period, or 42 ns), so the MC9S12 needs the data on the bus 63 ns-13 ns=50 ns after E goes high. The data is there 42 ns after E goes high, so this will work.

The corresponding time is $\mathtt{t}_{\mathtt{DSR}}$, the `Read data setup time` on the MC9S12 data sheet, circled number 10.

(f) Is $\mathtt{t}_{\mathtt{DHR}}$ for the input chip compatible with the MC9S12? Why or why not? What is the corresponding time for the 24 MHz MC9S12?

E goes low, 2 ns later CS goes high, 5 ns later, STUPID 2 removes data from the bus, so the data is removed from the bus 7 ns after E goes low. After E goes low, the MC9S12 needs the data to be held on the bus for 0 ns ($\mathtt{t}_{\mathtt{DHR}}$ on the MC9S12 data sheet), so the data is held on the bus long enough. This will work.

The corresponding time is $\mathtt{t}_{\mathtt{DHR}}$, the `Read data hold time` on the MC9S12 data sheet, circled number 11.

(g) Explain in words what $\mathtt{t}_{\mathtt{DDR}}$ means.

$\mathtt{t}_{\mathtt{DDR}}$ is the data delay time on a read – it is the amount of time it take sfor the STUPID 2 chip to drive its data on the bus after its chip select line goes low.

(h) Write some C code to write an 0x55 to the output STUPID chip.

```
* (unsigned char *) 0x6801 = 0x55;
```