EE 308

Exam 2

March 27, 2006

Name: ____

You may use any of the Motorola data books and EE 308 notes from the web. 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 9S12 with an 8 MHz crystal, and a 24 MHz bus clock.

Also, assume that `hcs12.h` has been included, so you can refer any register in the 9S12 by name rather than by its address in any C code you write.

1. The following questions concern writing C code.

   (a) Write some C code which will read the 16-bit signed integer at addresses `0x2000` and `0x2001`, and will store the one's complement of that number into memory locations `0x2010` and `0x2011`.

   ```
   *(short *)0x2010 = ~(*(short *)0x2000);
   ```

   (b) Write some C code which will set bits 3 and 4 of the byte at address `0x3000`, and leave all the other bits of that byte unchanged.

   ```
   #define VAR *(char *) 0x3000

   VAR = VAR | 0x18;
   ```

   (c) Write some C code which will do the following: If bits 3, 2, 1 and 0 of PORTB have the value 1001 (binary), write a `0xff` to PORTA. Otherwise, write a `0x00` to PORTA. (Assume that all bits of PORTA have been set up for output, and all bits of PORTB have been set up for input.)

   ```
   if ((PORTB & 0x0f) == 0x09)
   {
       PORTA = 0xff;
   }
   else
   {
       PORTA = 0x00;
   }
   ```

2. Below are the contents of the memory of an 9S12:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3BD0	10	23	3B	7C	10	04	86	C9	B7	10	25	3B	FC	10	18	F3
3BE0	12	50	FD	10	18	86	40	B7	10	23	3B	FC	10	12	DD	02
3BF0	86	02	B7	10	23	3B	7C	10	03	86	40	B7	10	25	3B	86
FFC0	CC	05	9F	CD	99	03	84	9C	01	9B	CC	90	66	FC	93	30
FFD0	7E	E3	4B	7E	E5	38	21	54	05	83	09	34	2A	38	3C	03
FFE0	41	38	66	F2	7C	13	37	0C	25	F2	0C	38	5F	1B	42	1A
FFF0	7A	26	21	13	6A	AA	20	1F	4B	38	33	38	45	38	80	00

The 9S12 just received a Timer Overflow Interrupt, and is starting to execute the Timer Overflow interrupt service routine — i.e., the Program Counter is set to the address of the first instruction of the Timer Overflow interrupt service routine. The stack pointer at this time has a value of `0x3BE0`.

(a) What value is in the Program Counter – i.e., what is the address of the Timer Overflow interrupt service routine?

The Timer Overflow Interrupt Vector is at address 0xFFDE and 0xFFDF. This contains a 0x3C03. That is the address of the Timer Overflow ISR.

(b) What is the return address — i.e., what is the address of the instruction the 9S12 will return to when it exits the ISR with the RTI instruction?

The stack pointer is 0x3BE0 *after* everything was stacked and the processor started on the first instruction of the ISR. After being stacked, the stack pointer points to the last thing pushed (CCR), above that are B (address 0x3BE1), A (address 0x3BE2), X (address 0x3BE3), Y (address 0x3BE5), Return Address (address 0x3BE7). Therefore, the Return Address is 0xB710.

(c) What value was in the 9S12 X register when it received the interrupt?

As above, the X register was 0x1028.

(d) What value was in the 9S12 condition code register when it received the interrupt?

As above, the X register was 0x12.

(e) What is the address of the first instruction the 9S12 will run after coming out of reset? The reset vector is at 0xFFFE and 0xFFFF, so the reset vector is 0x8000.

3. The following questions pertain to the 9S12 pulse width modulation subsystem.

   (a) You are asked to generate a 4 kHz PWM signal on Port P3 of the 9S12. Explain how you will set up the 9S12 to do this, and write some C code to implement it.

   Channel 3 uses PCKB and PWMSCLB.

   $\frac{24 \times 10^6}{4 \times 10^3} = 6000$, so you need

   Clock mode 0:

$$6000 = \text{PWMPER3} \times 2^{\text{PCKB}}$$

   or Clock mode 1:

$$6000 = \text{PWMPER3} \times 2^{\text{PCKB+1}} \times \text{PWMSCLB}$$

   We can get this with PWMPER3 = 200, PCKB = 0, PWMSCLB = 15 (and many other ways).

```
PWMCTL = 0x00;              /* 8-bit mode */
PWMPOL = 0xFF;              /* High polarity */
PWMCAE = 0x00;              /* Left aligned */
PWMCLK = PWMCLK | 0x08;     /* Clock mode 1 for Channel 3 */
PWMPRCK = PWMPRCK & ~0x70;  /* PCKB = 0;
PWMSCLB = 15;
PWMPER3 = 200;
PWME = PWME | 0x08;         /* Enable Channel 3 */
```

   (b) Using the values above, write some C code to set the duty cycle to 40%.

   $0.4 \times 200 = 80$, so

```
PWMDTY3 = 80;
```

   (c) An 9S12 has the following in its PWM registers:

PWME	PWPOL	PWMCLK	PWMPRCLK	PWMCAE	PWMCTL	PWMSCLA	PWMSCLB	PWMPER0	PWMDTY0
0F	FF	05	45	00	00	28	00	C8	28

   i. What is the period (in seconds) of the PWM signal of Channel 0? For partial credit (in case you get the final answer wrong) be sure to explain how you arrived at you answer.

```
PWMCLK = 0x05 = 0000101 => PCLK0 = 1 => uses clock mode 1
PWMPRCLK = 0x45 => PCKA = 5
PWMSCLA = 0x28 = 40
PWMPER0 = 0xC8 = 200
```

   With Clock Mode 1:

$$\text{cycles} = \text{PWMPER3} \times 2^{\text{PCKB+1}} \times \text{PWMSCLB} = 200 \times 2^{5+1} \times 40 = 512,000$$

$$\text{Period} = \frac{512,000 \text{ cycles}}{24 \times 10^6 \text{ cycles/sec}} = 0.0213 \text{ s}$$

   ii. What is the duty cycle (in percent) of the PWM signal on Channel 0? Explain.

$$\text{Duty cycle} = \frac{\text{PWMDTY0}}{\text{PWMPER0}} \times 100\% = \frac{\text{0x28}}{\text{0xC8}} \times 100\% = \frac{40}{200} \times 100\% = 20\%$$

4. The Starship Enterprise is on a mission monitoring a Klingon warship. Under normal conditions the warship emits bursts of radiation at random times, but at least once every 50 ms. If the time between bursts ever exceeds 50 ms, it means that the Klingons are preparing to attack. Mr. Spock decides to use the features of an 9S12 to determine if the time between bursts ever exceeds 50 ms. Mr. Spock builds a circuit so that a burst of radiation from the Klingon warship produces a rising edge on Pin 0 of Port T. The 9S12 should be programmed so that if the time between bursts ever exeeds 50 ms the 9S12 will bring Pin 5 of Port T high, which will energize the Enterprise's shields. Mr. Spock asks you to write the 9S12 program to do this.

   (a) How do you enable the timer subsystem on the 9S12? Write some C code to do this.
   Set TEN bit (bit 7) of TSCR1:

   ```
   TSCR1 = 0x80;
   ```

   (b) What value would you write to the prescaler to be able to measure time differences of at least 50 ms? Write some C code to do this.
   Prescaler of 5 gives overflow period of 87 ms, so set prescaler in TSCR2 to 5:

   ```
   TSCR2 = 0x05;
   ```

   (c) For this value of the prescaler, how many timer ticks will 50 ms take?
   Prescaler of 5 divides the bus clock by $2^5 = 32$, so the Timer clock frequency is $\frac{2\times10^6}{32} = 750,000$. Number of timer ticks is:

   $$50 \times 10^{-3} \text{ sec} \times 750,000 \text{ ticks/second} = 37,500 \text{ ticks}$$

   (d) How do you set up the 9S12 to capture the time and generate an interrupt when the radiation burst produces a rising edge on Pin 0 of Port T? Write some C code to do this. Be sure your C code does not affect the function of any of the other timer channels. Also, be sure to do **all** the setup necessary in order to use interrupts. (You do not need to write the interrupt service routine.)
   Set up Channel 0 for Input Capture; capture rising edge; clear flag; set interrupt vector; enable specific interrup; enable interrupts in general:

   ```
   TIOS = TIOS & ~0x01;                /* Channel 1 Input Capture */
   TCTL4 = (TCTL4 | 0x01) & ~0x02;     /* Capture Rising Edge on Ch 1 */
   TFLG1 = 0x01;
   UserTimerCh0 = (short) &tic0_isr;   /* Set interrupt vector */
   TIE = TIE | 0x01;                   /* Enable int on Ch 0 */
   asm(" cli");                        /* Enable interrupts in general */
   ```

   (e) How do you set up the 9S12 to bring Bit 5 of Port T high when TCNT has a value 50 ms after the time of the rising edge on Pin 0 ? Write some C code to do this. Be sure your C code does not affect the function of any of the other timer channels.
   From the setup above, the time of a rising edge on Ch 0 will be stored in TC0. To bring Ch 5 high 50 ms later, set up Ch 5 for Output Compare, go high, and put TC0 + 50 ms into TC5:

   ```
   TIOS = TIOS | 0x20;       /* Ch5 to output compare */
   TCTL1 = TCTL1 | 0x0C;     /* Set output pin on compare */
   TC5 = TC0 + 37500;        /* Bring Ch5 high 50 ms after Ch0 goes high */
   ```

   (f) The TFLG1 register has a value of 0x15.

      i. For which timer channels are the timer flags set?
      0x15 = 00010101_2, so flags are set of Channels 0, 2, and 4.

      ii. Write some C code which will clear the Time Channel 0 flag and **not** clear any of the other timer flags.

      ```
      TFLG1 = 0x01;
      ```