**EE 308**

**Final Exam**

**May 10, 2000**

Name: _____

You may use any of the Motorola data books, notes from the EE 308 web page, and you lab notebook. Show all work. Partial credit will be given. No credit will be given if an answer appears with no supporting work.

For all problems assume you are using a MC68HC912B32 with an 16-MHz crystal, resulting in an 8-MHz processor clock. Also, for C programming, assume the file hc12.h has been included, so you can refer to registers by name.

1. Do the following arithmetic:

   $A7_{16}$ = _____$_2$

   $A7_{16}$ = _____$_{10}$ (assuming $A7_{16}$ is unsigned)

   $A7_{16}$ = _____$_{10}$ (assuming $A7_{16}$ is signed)

   $80_{16}$ = _____$_2$

   $80_{16}$ = _____$_{10}$ (assuming $A7_{16}$ is unsigned)

   $80_{16}$ = _____$_{10}$ (assuming $A7_{16}$ is signed)

   Do the following operations on eight-bit numbers. Indicate the eight-bit answer, and whether a carry and/or overflow was generated:

| | 75<br>+ 9A | 8C<br>+ 74 | 90<br>+ 90 | 81<br>- 7F | 7F<br>- 80 |
|---|---|---|---|---|---|
| Answer | | | | | |
| C | | | | | |
| V | | | | | |
| N | | | | | |
| Z | | | | | |

2. Consider the following program fragment:

```
.area   CODE (ABS)
.org    0x0800
ldaa    #$23
staa    $23
anda    3,x
ldy     $0902
pshx
```

(a) Hand-assemble the code fragment. That is, show what hex numbers will be in what memory locations after this program has been assembled.

| Address | 0800 | 0801 | 0802 | 0803 | 0804 | 0805 | 0806 | 0807 |
|---|---|---|---|---|---|---|---|---|
| Byte | | | | | | | | |
| Address | 0808 | 0809 | 080A | 080B | 080C | 080D | 080E | 080F |
| Byte | | | | | | | | |

(b) How many bytes of memory does the code fragment occupy?

(c) How many processor clock cycles will it take to execute the five instructions of the code fragment?

(d) How many microseconds will it take the fragment to execute on an HC12 with an 8 MHz processor clock?

3. An HC12 has the following in its registers:

| Reg | - | - |
|---|---|---|
| CCR | S X H I N Z V C<br>1 0 0 0 1 0 0 1 | |
| A:B | A3 | 92 |
| X | 82F2 | |
| Y | 12F7 | |
| SP | 208D | |
| PC | 0800 | |

For the instructions below indicate the addressing mode and the effective address (i.e., address in memory the HC12 uses for data):

| Address of Instruction | Instruction | Addressing Mode | Effective Address |
|---|---|---|---|
| 0x0800 | `ldaa #$23` | | |
| | `staa $23` | | |
| | `anda 3,x` | | |
| | `ldy $0902` | | |
| | `pshx` | | |

4. The following problem concerns clearing flags.

   (a) How do you clear the Real Time Interrupt Flag (RTIF)? Write some C code which will do this.

   (b) How do you clear the Timer Channel 3 Flag (C3F)? Write some C code which will do this.

   (c) How do you clear the Serial Peripheral Interface Flag (SPIF)? Write some C code which will do this.

   (d) How do you clear the AD Sequence Complete Flag (SCF)? Write some C code which will do this.

5. The following problem concerns interrupts.

   Here are the contents of some memory locations of an HC12:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1000 | 10 | 23 | 3B | 7C | 10 | 04 | 86 | 80 | B7 | 10 | 25 | 3B | FC | 10 | 18 | F3 |
| 1010 | 12 | 50 | FD | 10 | 18 | 86 | 40 | B7 | 10 | 23 | 3B | FC | 10 | 12 | DD | 02 |
| 1020 | 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 | 25 | 1A |

   (a) What is the address of the first instruction the HC12 will execute after a reset?

   (b) How do you enable the Real Time Interrupt? Write some C code to enable a Real Time Interrupt with a 4.096 ms.

   (c) What is the address of the RTI interrupt service routine? I.e., what is the first instruction the HC12 will execute after receiving a Real Time Interrupt?

   (d) The HC12 has the following in its registers when a Real Time Interrupt occurs:

| Reg | - | - |
|---|---|---|
| | S X H I N Z V C | |
| CCR | 1 0 0 0 1 0 0 1 | |
| A:B | A3 | 92 |
| X | 82F2 | |
| Y | 12F7 | |
| SP | 09F8 | |
| PC | 31F4 | |

   Explain in detail what happens when the HC12 responds to the interrupt. Show what will be in the HC12 registers when it starts executing the first instruction of the interrupt service routine, and show what has happened to the stack.

| Reg | - | - | Stack Contents |
|---|---|---|---|
| | S X H I N Z V C | | |
| CCR | | | |
| A:B | | | |
| X | | | |
| Y | | | |
| SP | | | |
| PC | | | |

(e) What assembly language instruction should you use when you exit an interrupt service routine? Why?

6. Here are some questions about various sub-systems on the HC12.

(a) How do you enable the HC12 timer subsystem? Write some C code to do this.

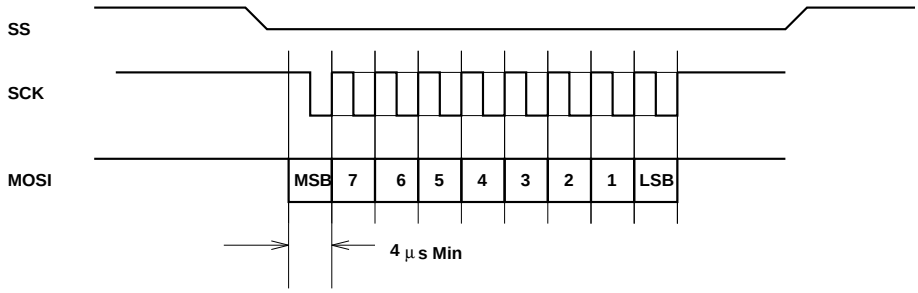
(b) What is the basic frequency of the timer subsystem clock?

(c) How do you change the frequency of the timer subsystem clock? Write some C code to set the frequency to 1 MHz.

(d) How do you set up the timer subsystem to capture the time of the rising edge on Port T3? Write some C code to do this.

(e) Below is a timing diagram for an SPI peripheral. What should you set `CPOL`, `CPHA`, and `SP0BR` to in order to use this SPI peripheral at the maximum possible speed? Explain your answers

`CPOL = ________  CPHA = _______  SP0BR = __________`




(f) Assume that the SPI has been properly set up to communicate with the above peripheral. Write some C code to send a 0x55 to the peripheral, and wait for the transfer to finish.

(g) How should you set up the A/D control registers to have the HC12 do the following: Perform eight conversions sequentially on pins PAD0-7, then stop. Write some C code to do this.

(h) An HC12 has $V_{RL} = 0V$ and $V_{RH} = 5V$. The input on Pin 3 of Port E is 2.4 V. What will be in the A/D result register ADR3H after the conversion initiated above?

(i) You want to use Channel 0 of the HC12 PWM subsystem to generate a 1 kHz PWM signal. What would you write to the PWM registers to do this?

(j) The HC12 has been set up to generate a 500 Hz PWM signal on Channel 3. `PWPER3` has been set to 199. What would you do to make the PWM signal have a 30% duty cycle?

(k) To what pin would you connect your motor controller to use PWM channel 3?

7. The following question concerns the HC12 in expanded mode.

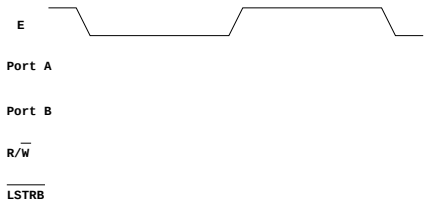
   (a) How does the HC12 know what mode to run in when it comes out of reset?

   (b) How can you change the operating mode of the HC12 using instructions? If the HC12 comes out of reset in Normal Single Chip mode, write a C instruction to put the HC12 into Normal Expanded Wide mode.

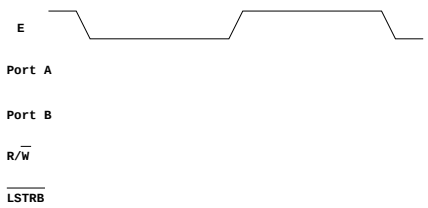
   (c) Explain the functions of Port A and B in expanded mode. (Be sure to discuss the E clock when you answer this question.)

   (d) Show what will be on the address/data bus and the control lines when the HC12 executes the following instructions:

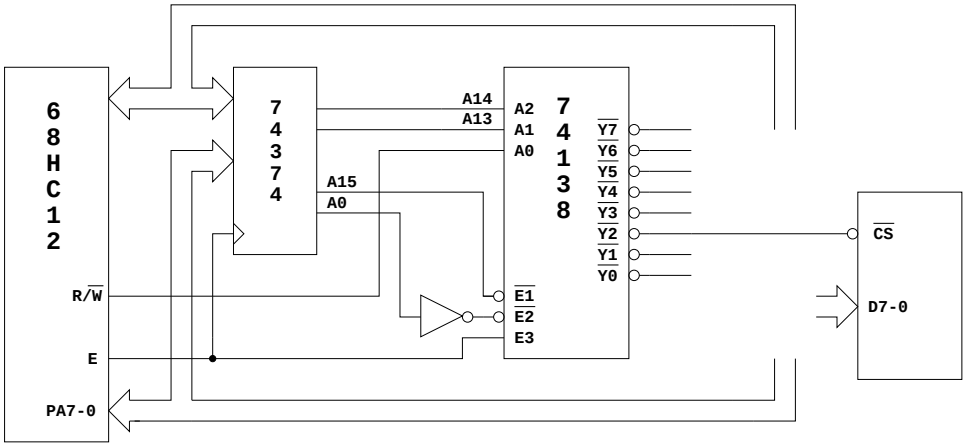
      i. Writes a 0xcc to address 0x1001.




      ii. Writes a 0x55aa to the two bytes at addresses 0x1002 and 0x1003.



8. The following shows the HC12 interfaced to a peripheral chip:




(a) Is the peripheral in input or output chip?

(b) What range of addresses will select the peripheral?

(c) Should you connect the data lines for the peripheral to Port A or Port B? Why?

(d) If the peripheral is an input, write some C code which will read a byte from it and save it in a variable called `data`. If the peripheral is an output, write the byte `0x55` to it.