

Please put your name on the outside of the paper also

Name \_\_\_\_\_KEY\_\_\_\_\_

1. List three methods to accomplish I/O synchronization. (6 points)

**Delay loops****Polling (spin loops, handshaking)****Interrupts**

2. For the M68HC12 what instruction (3 points)

Masks interrupts?

**SEI**

Enables the timer overflow interrupt?

**BSET TMASK2,TOI where****TMASK2 EQU \$8D****TOI EQU %10000000**

Captures the current TCNT register contents?

**LDD TCNT where****TCNT EQU \$84**

3. The TCNT register is receiving an 8 MHz clock. (4 points)

- a. How many clock cycles will constitute a delay of 6.2 milliseconds?

$$6.2 \times 10^{-3} \text{ sec} / 125 \times 10^{-9} \text{ sec/cycle} = 49,600 \text{ cycles}$$

- b. If you are using the output compare with interrupts to delay 6.2 milliseconds, can this be done without multiple interrupts?

**Yes, the number of cycles is <65536**

4. Show a segment of HC12 code used to reset the timer overflow flag. (2 points)

**TFLG2 EQU \$8F****TOF EQU %10000000****ldaa #TOF****staa TFLG2**

5. A 12FC68 microcontroller has an 8-bit input port F, and 8-bit output port G and an 8-bit bidirectional port H. There are appropriate control registers within the microcontroller to control any I/O. There are no external address, data or control buses. There are no interrupts.

You are to interface an 8-bit A/D converter which has START\_CONVERT and END\_OF\_CONVERT control signals to the microcontroller.

- a. Show the hardware interface which will accomplish this task. (5 points)

**Connect Port F to A/D binary output. Connect PortH-bit0 to A/D START\_CONVERT and PortH-Bit1 to A/D END\_OF\_CONVERT. A bit on Port G could be used for the START\_CONVERT signal.**

- b. Based on the hardware you have designed, give a pseudo-code design for the software that will initialize any hardware that needs it and which will retrieve an 8-bit digital value from the A/D. (5 points)

```
; Initialize PortH-bit 0 as an output
; Initialize PortH-bit 1 as an input (if needed)
;
; Get a value from A/D
; Set START_CONVERT high
; Set START_CONVERT low
; Wait for END_OF_CONVERT to be asserted
; Read the value from the A/D
```