

CS/ECE 5780/6780
Spring 2007
Myers

Midterm Exam 1

- Fill in your name:
- This exam is open book and open notes.
- The exam is 80 minutes and worth 100 points.
- Show all your work.

Question	Score
1	
2	
3	
Total	

1. **Short Answer** (10 points)

(a) Computer systems often crash due to software that either does not live within its resource bounds or due to misuse of its resources. Such crashes can be catastrophic in embedded systems. To avoid these problems, name two things you should and/or should not do in embedded software. Be sure to explain your answers.

(b) In computer systems, we typically assume that the hardware is going to behave correctly. In embedded systems, it is usually not safe to make this assumption. Name two things that we do in embedded system design to catch hardware that is misbehaving. Be sure to explain your answers.

2. Assembly Code (30 points)

Consider the following assembly code.

```
A  movb #$80,$0037
   bsr  Fifo_Get
   tsta
   beq  B
   staa $0000
   bclr $0010,$01
   bset $0010,$01
   bra  C
B  movb #$00,$0038
C  rti
   org $FFF0
   fdb A
```

(a) What does this program do?

(b) Write a C subroutine that does the same thing.

3. Interrupts (60 points)

In lab 5, periodic polling was used for the interrupt interface to the matrix keypad. However, it is possible to use key wakeup interrupts. For this question, assume that you have access to all pins of port P, and that we will use these pins to interface to your keypad. You can detect that a key is pressed by writing all zeros to the row bits and waiting for a falling edge on a column bit. You can detect that a particular key is released by writing a zero to its row only and waiting for a rising edge on its particular column. Using these facts, design a keypad interface using key wakeup interrupts. Your code should support two key rollover, but you may assume that the keypad is hardware debounced. You may also assume that when a key is initially pressed that it will be the only key pressed and your software will have time to execute before another key is pressed. In other words, only when your software is waiting on a key to be released may there be more than one key pressed. You may return the value of the key using any encoding that is convenient for you as long as each key has a unique value. Finally, you can assume the existence of the functions `Fifo_Put`, `Fifo_Get`, and `Fifo_Init`.

(a) Draw a schematic for your design.

(b) Describe any global variables that your solution will use.

(c) Write the ritual that is called by the main program.

(d) Write the key wakeup interrupt handler.