ECE/CS 5780/6780: Embedded System Design

Scott R. Little

Midterm 1 Solution
Exam grades will be final on 02/27/2008.
Please discuss any questions about the grading of a question with the person who graded that question.
Question 1: Anthony
Question 2: Steve
Question 3: Scott
Class average is 73.6.
Approximate grade breakdown

This will give you an idea of how you scored although 5780 and 6780 are graded on different curves.

<table>
<thead>
<tr>
<th>Score Range</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>91-100</td>
<td>A</td>
</tr>
<tr>
<td>82-90</td>
<td>A-</td>
</tr>
<tr>
<td>74-81</td>
<td>B+</td>
</tr>
<tr>
<td>73-66</td>
<td>B</td>
</tr>
<tr>
<td>65-58</td>
<td>B-</td>
</tr>
<tr>
<td>57-50</td>
<td>C+</td>
</tr>
<tr>
<td>49-40</td>
<td>C</td>
</tr>
<tr>
<td>39-0</td>
<td>C-</td>
</tr>
</tbody>
</table>
4 points.

- Points were awarded for each guideline and justification.
- Guidelines mentioned in lecture (4 & 5) were given full credit.
- Other reasonable guidelines were also given full credit.
Question 1b

6 points.

Two of the following needed for full points:

- Load the accumulator from different memory locations.
- Differ in object code length.
- Both load accumulator from a memory address.
- Addressing mode differences.
20 points.

Question 2a

- Idle
- Buzz=0
- Wait1Min
- Buzz=0
- Buzz=1
- alarmOff
- soundAlarm
- snooze
- soundAlarm
- Buzz
- Snooze
- Buzz=0
- Wait5Min
- Buzz=0
- Buzz=1
- alarmOff
- soundAlarm
- snooze
- soundAlarm
- Buzz
- Snooze
- Buzz=0
- Wait5Min
#define IDLE 0
#define BUZZ 1
#define SNOOZE 2
#define SOUNDALARM 1
#define ALARMOFF 2
#define SNOOZEIN 4

while(1) {
    int state = IDLE;
    unsigned char input;
    switch(state) {
    case IDLE:
        putOutput(0x00);
        waitMin(1);
        input = getInput();
        if(input == SOUNDALARM) {
            state = BUZZ;
        }
        break;
    }
case BUZZ:
    putOutput(0x01);
    input = getInput();
    if(input & ALARMOFF) {
        state = IDLE;
    }
    if(input & SNOOZEIN) {
        state = SNOOZE;
    }
    break;

case SNOOZE:
    putOutput(0x00);
    waitMin(5);
    input = getInput();
    if(input == ALARMOFF) {
        state = IDLE;
    } else {
        state = SNOOZE;
    }
    break;
Question 3a

- 6 points.

```
<table>
<thead>
<tr>
<th>PTT0</th>
<th>Row 0</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTT1</td>
<td>Row 1</td>
<td>Output</td>
</tr>
<tr>
<td>PTT2</td>
<td>Row 2</td>
<td>Input</td>
</tr>
<tr>
<td>PTT3</td>
<td>Row 3</td>
<td>Input</td>
</tr>
</tbody>
</table>
```

```plaintext
PTT0  Row 0  Output
PTT1  Row 1  Output
PTT2  Row 2  Input
PTT3  Row 3  Input
```
4 points.

DDRT = 0x03;
void Keypad_WaitPress() {
    unsigned char keyValue;
    do {
        keyValue = getKey();
    } while(keyValue == 255);
    Timer_Wait1ms(10);
}

void Keypad_WaitRelease() {
    unsigned char keyValue;
    do {
        keyValue = getKey();
    } while(keyValue != 255);
    Timer_Wait1ms(10);
}
Question 3d

- 20 points.

```c
unsigned char getKey() {
    unsigned char row, rawKey, keyValue = 255;
    for(row = 0; row < 4; row++) {
        PTT = row;
        rawKey = (PTT & 0x0C) >> 2;
        if(rawKey < 3) {
            break;
        }
    }
}
```
if(row < 3) {
    keyValue = (rawKey + 1) + (row * 3);
}
else {
    if(rawKey == 1) {
        keyValue = 10;
    }
    else if(rawKey == 2) {
        keyValue = 0;
    }
    else if(rawKey == 3) {
        keyValue = 11;
    }
}
return keyValue;