National Sun Yat-Sen University
ASSEMBLY LANGUAGE AND MICROCOMPUTER
Midterm Exam
5:30-7:30PM May 4 2010

Name: _____

Note: Although there are total 110 points for this exam, the maximum score you can get is 100 points.

1. Translate the following C-codes into the assembly codes based on the simple MU0 instruction set. In addition, translate the assembly code into the binary code. You should describe the initial contents of the memory when your program start running. ***(12 pts)***

```
int a, b,c;

if (a>b) c=a-b;
else c=27;
```

Instruction	Opcode	Effect
LDA S	0000	ACC := mem_{16}[S]
STO S	0001	mem_{16}[S] := ACC
ADD S	0010	ACC := ACC + mem_{16}[S]
SUB S	0011	ACC := ACC - mem_{16}[S]
JMP S	0100	PC := S
JGE S	0101	if ACC >= 0 PC := S
JNE S	0110	if ACC !=0 PC := S
STP	0111	stop

2. Specify a single ARM instruction which implements the following equation: ***(9 pts)***
   (a) r0 = 8*r1-r2 (b) r2=mem[r1], r1=r1+4;
   (c) r1 = 25 ;

3. Suppose r1=0xF0000001, r2=0xF0000000 and C=1, N=0, Z=0, V=0, find out the resulting **r1** value of the following instructions. You should also provide the resulting conditional code value (**C N Z**). ***(20 pts)***
   (a) ADDC r1, r1, r2 (b) SBCS r1, r1, r2
   (c) MOVS r1, r2, LSL #1 (d) EOR r1, r2, r2.

4. For each of the following multiple register store instructions, write a short code to restore these register values by loading the data back from the memory. ***(8 pts)***
   (a) STMDA r9!, {r0, r5, r1}
   (b) STMED r9, {r5, r1, r0}

5. Explain the following two popular addressing modes: (1) Immediate (2) Base plus offset. You should provide an ARM instruction as an example for each mode. ***(8 pts)***

6. Answer for the following short questions: ***(16 pts)***
   (c) List two ARM exception types. Discuss how to invoke each of two listed exceptions. ***(8 pts)***
   (d) Which two of the 16 registers in ARM are used to be the program counter and link register. ***(4 pts)***
   (e) Explain what is so called “Little Endian” memory organization. ***(4 pts)***

7. Find out if the following instruction format is correct or not. If incorrect, point out the problem. ***(15 pts)***

   (a) STMIA r5!, {r5, r4, r9}
   (b) ADD r3, r7, #1023
   (c) SUB r12, r3, LSL #32
   (d) MOVS r4, r4, RRX #3
   (e) STRH r7, [r3, #14]

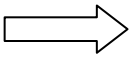
8. For the following C-expression: ***(12 pts)***

   if ((r0==r1) && (r2==r3)) r4++;

   (a) Write down the corresponding ARM code without using the conditional execution. (But you can still use conditional branches.)
   (b) Write down the corresponding ARM code using the minimum number of instructions.

9. Fill the missing three ARM instructions in the code shown in the right side of the following codes which represents a better implementation of "Jump Tables" than the code in the left. ***(10 pts)***

```
        BL    JUMPTAB
        ..
JUMPTAB CMP   r0, #0
        BEQ   SUB0
        CMP   r0, #1
        BEQ   SUB1
        CMP   r0, #2
        BEQ   SUB2
        ..
```



```
        BL      JUMPTAB
        ..
JUMPTAB
        B       ERROR
SUBTAB  DCD     SUB0
        DCD     SUB1
        DCD     SUB2
        ..
```