National Sun Yat-Sen University
ASSEMBLY LANGUAGE AND MICROCOMPUTER
Midterm Exam
2:20-4:20PM Nov 21 2013

Name: _____

Note: Although there are more than 100 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 (in binary) when your program start running. ***(12 pts)***

```
int a, b,c;

if (a==b) c=a+b;
else c=a-8;
```

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 = -0.5*r1 (r1 is an even number) (b) r2=mem_{32}[r1], r1=r1-8;
   (c) r1 = $r2_{[7:0]}$;

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) CMP r1, r2 (b) SBCS r1, r1, r2
   (c) MOVS r1, r2, LSR #2 (d) EOR r1, r2, r2.

4. Suppose r0=0x80010, r1=0x80018, r2=6, r3=7, r4=8, and the following table shows some part of the memory contents (Big-Endian memory organization is used). ***(12 pts)***
   (a) What is the result of r5 after executing the following instruction: ***(4 pts)***
   LDRB r5, [r1, #3]
   (b) Find out the value of r0, r1, r2, r3, and r4 after executing the following two instructions: ***(8 pts)***
   STMIB r0!, {r2, r3, r4}
   LDMEA r1, {r2, r4, r3}

Memory address	Data
0x00080020	0x00000005
0x0008001c	0x00000004
0x00080018	0x00000003
0x00080014	0x00000002
0x00080010	0x00000001
0x0008000c	0x00000000

5. Answer for the following short questions: ***(8pts)***
   (a) Explain what exceptional handler is. ***(4 pts)***
   (b) What is ***spsr*** register and its usage? ***(4 pts)***

6. For the following ARM assembly code: ***(15 pts)***
   (a) Explain its function using a short C-code expression. ***(5 pts)***
   (b) Explain what **ADR** instruction is. ***(3 pts)***
   (c) Try to modify the following code to reduce the number of instructions being executed. ***(7 pts)***

```
        MOV r1,#0
        ADR r2, a[0]
        MOV r0, #0
LOOP    CMP r0, #20
        BGE EXIT
        STR r1, [r2, r0, LSL #2]
        ADD r0, r0, #1
        B LOOP
EXIT
```

7. Find out if the following instruction format is correct or not. If incorrect, point out the problem. ***(12 pts)***
   (a) STMIB r4!, {r5, r4, r9}
   (b) ADD r3, r7, #1023
   (c) SUB r12, r3, LSL #32
   (d) STR r7, [r3, #14]

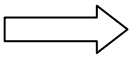
8. For the following C-expression: ***(12 pts)***

```
if ((r0==2*r1) && (r2==r3)) r4++;
else 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
        ..
```