Homework Assignment 1

(Due 2:10pm, Sep. 18, scan (or take a photo) and upload it in Canvas)

You can use the following instructions only.

R# is a register. (# = 0 ~ 12)

<table>
<tr><th>Instruction</th><th>Meaning</th></tr>
<tr><td>MVN Rd, Ra</td><td>Bitwise inversion. (Rd = Bitwise-NOT Ra)
<table>
<tr><td>Before</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>After</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>AND Rd, Ra, Rb
AND Rd, Ra, #imm</td><td>Bitwise AND. (Rd = Ra AND Rb), (Rd = Ra AND #imm)
<table>
<tr><td>Ra</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>Rb</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Rd</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>ORR Rd, Ra, Rb
ORR Rd, Ra, #imm</td><td>Bitwise OR. (Rd = Ra OR Rb), (Rd = Ra OR #imm)
<table>
<tr><td>Ra</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>Rb</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Rd</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
</table></td></tr>
<tr><td>EOR Rd, Ra, Rb
EOR Rd, Ra, #imm</td><td>Bitwise exclusive-OR. (Rd = Ra ⊕ Rb), (Rd = Ra ⊕ #imm)
<table>
<tr><td>Ra</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>Rb</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Rd</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>MOV Rd, Ra, LSR #imm</td><td>Logical shift right by (#imm) bits. (Rd = Ra >> #imm)
Ex) #imm = 3
<table>
<tr><td>Before</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>After</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table></td></tr>
<tr><td>MOV Rd, Ra, LSL #imm</td><td>Logical shift left by (#imm) bits. (Rd = Ra << #imm)
Ex) #imm = 3
<table>
<tr><td>Before</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>After</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>MOV Rd, Ra
MOV Rd, #imm</td><td>(Rd = Ra)
(Rd = #imm)</td></tr>
<tr><td>ADD Rd, Ra, Rb
ADD Rd, Ra, #imm</td><td>(Rd = Ra + Rb)
(Rd = Ra + #imm)</td></tr>
<tr><td>SUB Rd, Ra, Rb
SUB Rd, Ra, #imm</td><td>(Rd = Ra - Rb)
(Rd = Ra - #imm)</td></tr>
</table>

1. (20 points) The following shows the values in the registers R0-R3.

R0: 1

R1: 2

R2: 3

R3: 4

Show the values of R0-R3 after the following code is executed.

Code:

```
ADD R0, R0, R1        // R0 = 1 + 2 = 3
SUB R1, R3, R2        // R1 = 4 – 3 = 1
ADD R2, R0, R1        // R2 = 3 + 1 = 4
AND R3, R1, R0        // R3 = 1
EOR R3, R3, R2        // R3 = (0001) ^ (0100) = 0101
```

Answer:

R0: 3

R1: 1

R2: 4

R3: 5

2. (20 points) Assume all the number systems are unsigned number systems.

1) Represent 85 using the binary number system. 1010101
2) Represent 85 using the radix-3 number system. 10011
3) Represent 85 using the radix-16 number system. 55
4) What is the max. value that can be represented by the 4-digit hex number system? $2^{16} - 1 = 65{,}535$

3. (20 points) $R0 = a_7 a_6 \ldots a_0$ and $R1 = b_7 b_6 \ldots b_0$. Generate $R2$ from $R0$ and $R1$ (show the instructions). Try to minimize the # instructions.

$$R2 = a_7 b_6 a_5 b_4 a_3 b_2 a_1 b_0$$

```
AND R3, R0, #10101010b

AND R4, R1, #01010101b

ORR R2, R3, R4
```

4. (40 points)

1) Use the 8-bit binary number system. Represent 0, 1, 2, …, 20 using the 8-bit binary number system.

0000 0000

0000 0001

…

0001 0100

2) Find the remainders of $\frac{0}{4}, \frac{1}{4}, \ldots, \frac{20}{4}$. Represent them in the 8-bit binary number system.

0

1

2

3

0

1

…

3

0

3) $R0 = a_7 a_6 \ldots a_0$ is given. When does the remainder of $\frac{R0}{4}$ become 2? Show the condition for that.

$$a_1 a_0 = 10$$