CENG3420 Homework 2

Due: Mar. 13, 2022

All solutions should be submitted to the blackboard in the format of **PDF/MS Word**.

Q1 (20%)

1. What is 5ED4 - 07A4 when these values represent unsigned 16-bit hexadecimal numbers? The result should be written in hexadecimal. Show your work.
2. What is 5ED4 - 07A4 when these values represent signed 16-bit hexadecimal numbers stored in sign-magnitude format? The result should be written in hexadecimal. Show your work.

Q2 (30%) Read through the multiplication/devision algorithm:

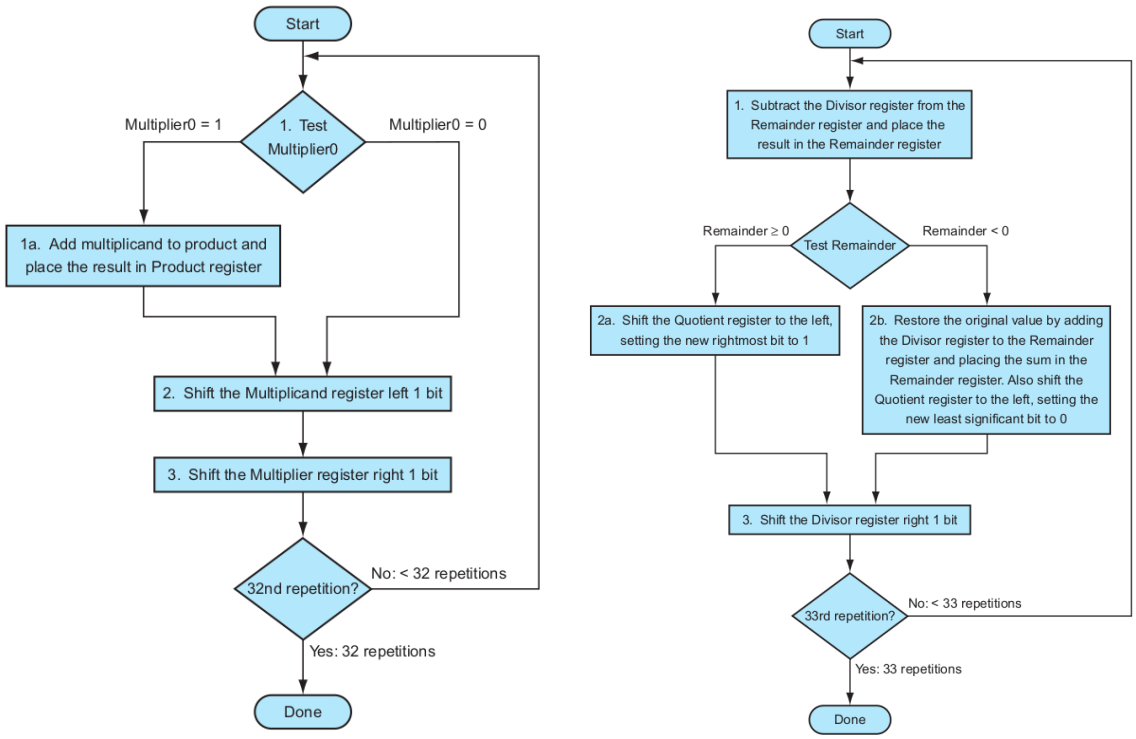

Figure 1: The left figure is the multiplication algorithm for reference. The right figure is the division algorithm.

1. Write down the step by step procedure to calculate 7×3 or 0111×0011. Use Multiplier0 to indicate the least significant bit of the multiplier.
2. Write down the step by step procedure to calculate $7 \div 2$ or $0111 \div 0010$.

Q3 (10%)

EEE 754-2008 contains a half precision that is only 16 bits wide. The leftmost bit is still the sign bit, the exponent is 5 bits wide and has a bias of 15, and the mantissa is 10 bits

long. A hidden 1 is assumed. Write down the bit pattern to represent $-1.5625 \times 10 - 1$ assuming a version of this format, which uses an excess-16 format to store the exponent.

Q4 (20%)

Calculate the sum of 2.6125×10^1 and $4.150390625 \times 10^{-1}$ by hand, assuming A and B are stored in the 16-bit half precision described in Q3. Assume 1 guard, 1 round bit, and 1 sticky bit, and round to the nearest even. Show all the steps. (To perform addition of 2 number, firstly shift mantissa right/left to make sure the exponent are matched)

Q5 (10%)

Consider the following code in C:

```
a = b + e;
c = b + f;
```

Here is the generated RISC-V code for this segment, assuming all variables are in memory and are addressable as offsets from x31:

```
ld x1, 0(x31) // Load b
ld x2, 8(x31) // Load e
add x3,x1,x2 // b + e
sd x3, 24(x31) // Store a
ld x4, 16(x31) // Load f
add x5,x1,x4 // b + f
sd x5, 32(x31) // Store c
```

Find the hazards in the preceding code segment and reorder the instructions to avoid any pipeline stalls.

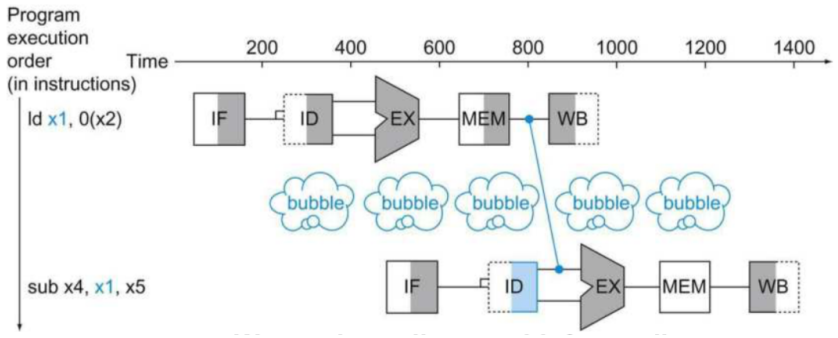

Figure 2: We need a stall even with forwarding when an R-format instruction following a load tries to use the data

Q6 (10%)

Consider the following instruction:

Instruction: `and rd, rs1, rs2`
Interpretation: `Reg[rd] = Reg[rs1] AND Reg[rs2]`

1. What are the values of control signals generated by the control in figure 3 for this instruction?

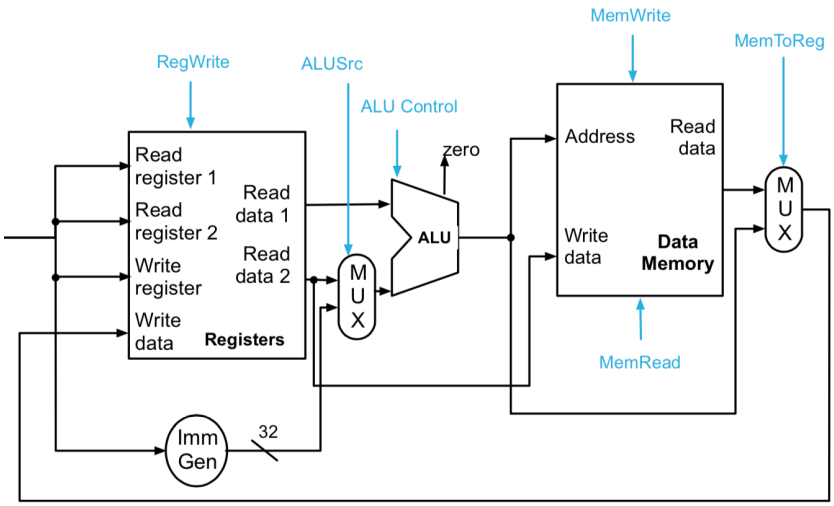

Figure 3: The datapath for the memory instructions and the R-type instructions.

2. Which resources (blocks) produce no output for this instruction? Which resources produce output that is not used?