CS224
Fall 2020
QUIZ NO. 3 Version 2 Solutions

November 5, 2020
~~**November 3, 2020**~~ Nov. 11, 2020

PLEASE READ: 1. Only handwritten answers are accepted. 2. Convert your handwritten answers to pdf and upload one pdf file to Moodle. Make sure that you answer the questions in the order they are given. Provide a neat work and make sure that your answers are numbered and in the order of questions and distinguishable from each other. Do not miss the deadline.
Note: Quiz questions indicate nothing about the difficulty of the exam questions.

1. Perform the addition of the following two decimal numbers which are given in normalized form.
1.052×10^{-2} and 1.001×10^{-5}
Note that you are allowed to keep three digits after the decimal point.

2. Write a simple high level language (Java, C++ etc.) program that demonstrates the possible inaccuracy in floating point arithmetic. Give the essential part of your program (again handwritten) and explain the inaccuracy.

3. Consider the hypothetical R type instruction called addinc ~~movefm- move from memory-~~ (funct 0X3F ~~0X7F~~). An example for this instruction is the following: addinc $t0, $t1

The RTL definition of addinc is as follows.

IM [PC]
RF [rd] <== RF [rs] + 1
PC <== PC + 4

rt is not used specify it as $0 in the instruction (remember 3R instruction).

a. Modify the single-cycle MIPS data path given on the next page to implement the addinc instruction. If needed you may modify the existing multiplexers or add new hardware, etc. You are expected to provide a simple easy to follow solution. In your drawing you do not need to draw the entire datapath you may just draw the relevant parts.

(1) $1.052 \times 10^{-2} + 1.001 \times 10^{-5}$

The exponent of the smaller no. must match the exponent of the larger number.

$1.001 \times 10^{-5} \rightarrow 0.1001 \times 10^{-4} \rightarrow 0.01001 \times 10^{-3} \rightarrow 0.001001 \times 10^{-2}$ (001 will be lost)

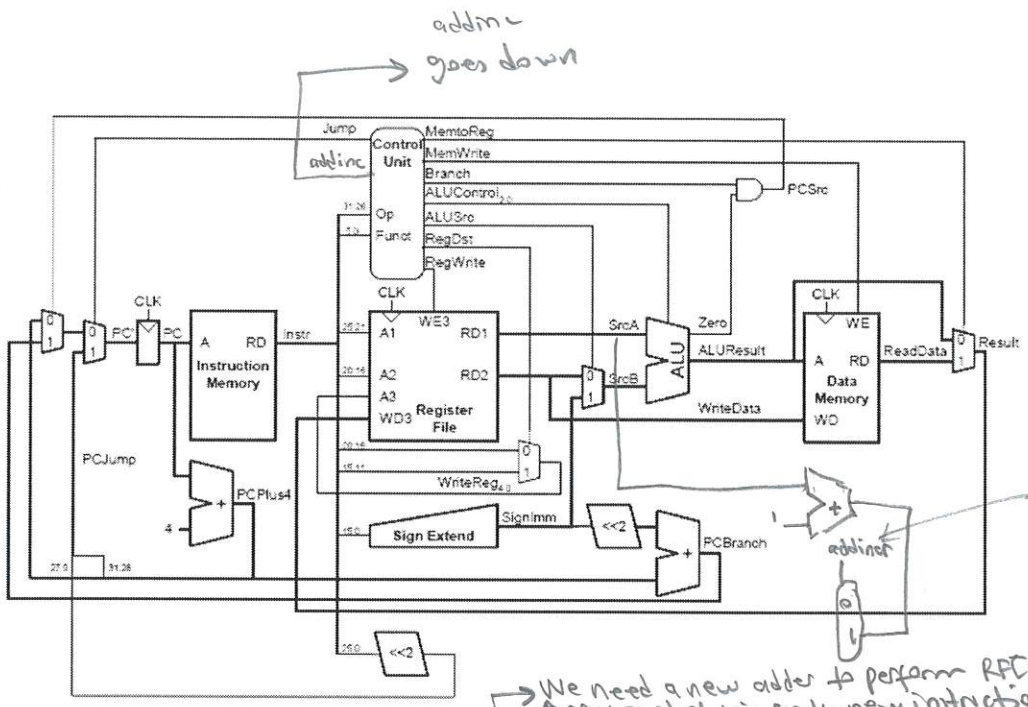
The second number becomes 0.001×10^{-2}

$$\begin{array}{r} 1.052 \times 10^{-2} \\ 0.001 \times 10^{-2} \\ \hline 1.053 \times 10^{-2} \end{array}$$

Not as precise as manual addition

0.

(2) Program: Your assignment.



b. What are the new components of the data path and what are their purposes? If no additional component is needed explain why.

c. State the values of the following control signals for the ~~movefm~~ addinc instruction. If you have new control signal(s) state their name(s), and value.

Jump: 0

addinc (new control line) = 1

MemtoReg: X

MemWrite: 0

Branch: 0

ALUSrc: X

RegDst: 1

RegWrite: 1

addinc $t0, $t1
R type
IM[PC]
RF[rd] ← RF[rs] + 1
PC ← PC + 4

d. If we follow your implementation does this instruction change the clock period, T_c, of the processor? Explain briefly. (The answer to this question gets credit if the implementation is correct.)

e. Generate the machine instruction for addinc $t0, $t1.

4. Consider the single-cycle MIPS structure. The time requirement of each stage except WB is given as follows, IF: 150, ID: 50, EX: 50, MEM: 200 (all nano seconds: ns). In ~~it~~ this environment we execute a program and execute 10^9 instructions. The total execution time is observed as 8 min 20 sec.

exec time given

8×60+20 = 500 sec
500×10^9 nsec

a. What is the time requirement of the WB stage?

b. What is clock rate (clock speed)? Express it in terms of GHz and MHz.

Incorrect it changes since we have a new multiplexer Result signal must go thru this multiplexer. It increases clock period!

d) It does not change T_c. As the modified single-cycle implementation shows its critical path is much shorter than lw instruction

e) Machine inst. opcode/rs/rt/rd/shamt/funct
Note rt is not used, specify it as $0 as in jr instruction
0 8 0 8 0 3F
00 0000 01001 00000 01000 00000 111111
0 1 2 0 4 0 3 F
2 ⇒ 0X01204 03F

4) Exec time = (No. of inst) × (150+50+50+200+t_{WB}) × CPI ← 1 it is single-cycle
500×10^9 ns = 10^9 × (450 + t_{WB}) ns ⇒ 50 nsec
a. t_{WB} = 50 ns

5) T_c = 500 ns/cycle → Clock Rate = 1/(500 ns) = $\frac{1}{500\times10^{-9}} = \frac{1000}{500}\times10^6 = 2\times10^6$ cycle/sec
= 2 MHz
⇒ 2×10^{-3} GHz

2 MHz = 2×10^{-3} GHz