CSE 120 Spring 2021
Homework Assignment 1

Jaden Liu
zliu259@ucsc.edu
University of California at Santa Cruz
Santa Cruz, CA 95064 USA

April 27, 2021

1 HW1

1. Two engineers solve the same problem, but each has a different solution (A and B). Using the default compiler options, the A program takes 6 seconds to run on processor P1 when compiled, and 5 seconds for the B program. This means that B is 1.2 times faster with the default options.

a) If we enable the gcc optimizations, we increase the CPI by 1.3 times and decrease the dynamic instruction count by 20% for program B (program A remains the same). What is the new speedup? (1 point)

b) If we also enable the compiler optimizations for the A program, we increase CPI by 70%. How much does the instruction count need to change (increase or decrease?) so that it matches the performance of both programs (both with optimizations)? (1 point)

Solution for a.

$$\begin{aligned}
speedup &= \frac{T_{oldA}}{oldB} \cdot \frac{T_{old}}{T_{new}} \\
&= \frac{T_{oldA}}{oldB} \cdot \frac{Instruction\ count_{old} \cdot CPI_{old} \cdot Clock\ Cycle\ Time_{old}}{Instruction\ count_{new} \cdot CPI_{new} \cdot Clock\ Cycle\ Time_{new}} \\
&= \frac{6}{5} \cdot \frac{1}{1.3 * 0.8} \\
&= \frac{6}{5} \cdot \frac{1}{1.04} \\
&\approx 1.1538
\end{aligned}$$

□

Solution for b. Assume new instruction count x times the old instruction count.

$$\begin{aligned}
speedup_b &= \frac{T_{oldB}}{T_{newB}} \\
&= \frac{Instruction\ count_{old} \cdot CPI_{old} \cdot Clock\ Cycle\ Time_{old}}{Instruction\ count_{new} \cdot CPI_{new} \cdot Clock\ Cycle\ Time_{new}} \\
&= \frac{1}{1.7 \cdot x} = speedup_a = \frac{1}{1.04}
\end{aligned}$$

Solving the equation we get:

$$\begin{aligned}
x &= \frac{52}{85} \\
&\approx 0.6117
\end{aligned}$$

Therefore, instruction count needs to decrease $(1 - 0.6117) * 100\% = 38.83\%$ so that it matches the performance of both programs □

2. The following table shows how many cycles each type of instruction takes and what the percentage in a given P processor

Instruction Type	Instruction Frequency	# of Cycles
Loads and Stores	25%	1
Arithmetic Operations	45%	2
Other	30%	3

a) Given the information in the table above, calculate the CPI and IPC. (1 point)

b) Now if some engineering effort is done on P to improve the arithmetic operations by 25%. Calculate the new CPI? (1 point)

c) What is the maximum speedup if the loads and stores are optimized? (1 point)

Solution for a.

$$\begin{aligned}
CPI &= 25\% \cdot 1 + 45\% \cdot 2 + 30\% \cdot 3 \\
&= 0.25 + 0.9 + 0.9 \\
&= 2.05 \\
IPC &= \frac{1}{CPI} \\
&= \frac{20}{41} \\
&\approx 0.4878
\end{aligned}$$

□

Solution for b.

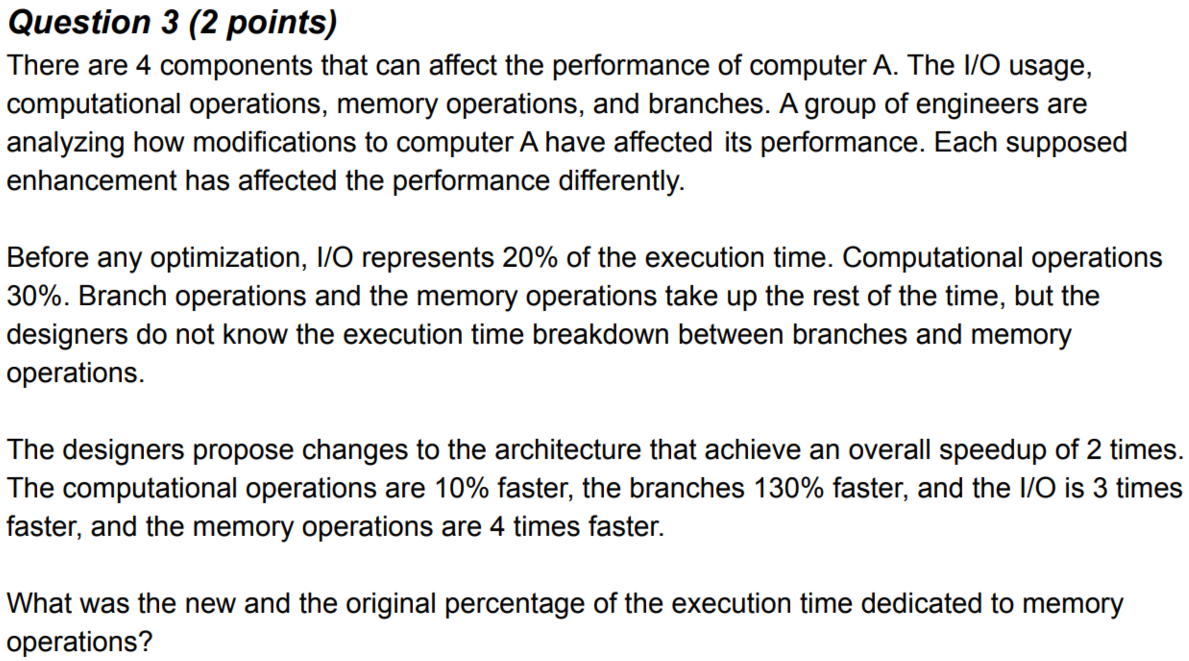
$$\begin{aligned} CPI &= 25\% \cdot 1 + 45\% \cdot 2 \cdot 0.75 + 30\% \cdot 3 \\ &= 0.25 + 0.675 + 0.9 \\ &= 1.825 \end{aligned}$$

□

Solution for c. By Amdahl's law, we can get maximum CPI is $0.25 + 0.9 = 1.15$

$$\begin{aligned} speedup &= \frac{T_{old}}{T_{new}} \\ &= \frac{Instruction\ count_{old} \cdot CPI_{old} \cdot Clock\ Cycle\ Time_{old}}{Instruction\ count_{new} \cdot CPI_{new} \cdot Clock\ Cycle\ Time_{new}} \\ &= \frac{2.05}{1.15} \\ &\approx 1.7826 \end{aligned}$$

□



Question 3 (2 points)

There are 4 components that can affect the performance of computer A. The I/O usage, computational operations, memory operations, and branches. A group of engineers are analyzing how modifications to computer A have affected its performance. Each supposed enhancement has affected the performance differently.

Before any optimization, I/O represents 20% of the execution time. Computational operations 30%. Branch operations and the memory operations take up the rest of the time, but the designers do not know the execution time breakdown between branches and memory operations.

The designers propose changes to the architecture that achieve an overall speedup of 2 times. The computational operations are 10% faster, the branches 130% faster, and the I/O is 3 times faster, and the memory operations are 4 times faster.

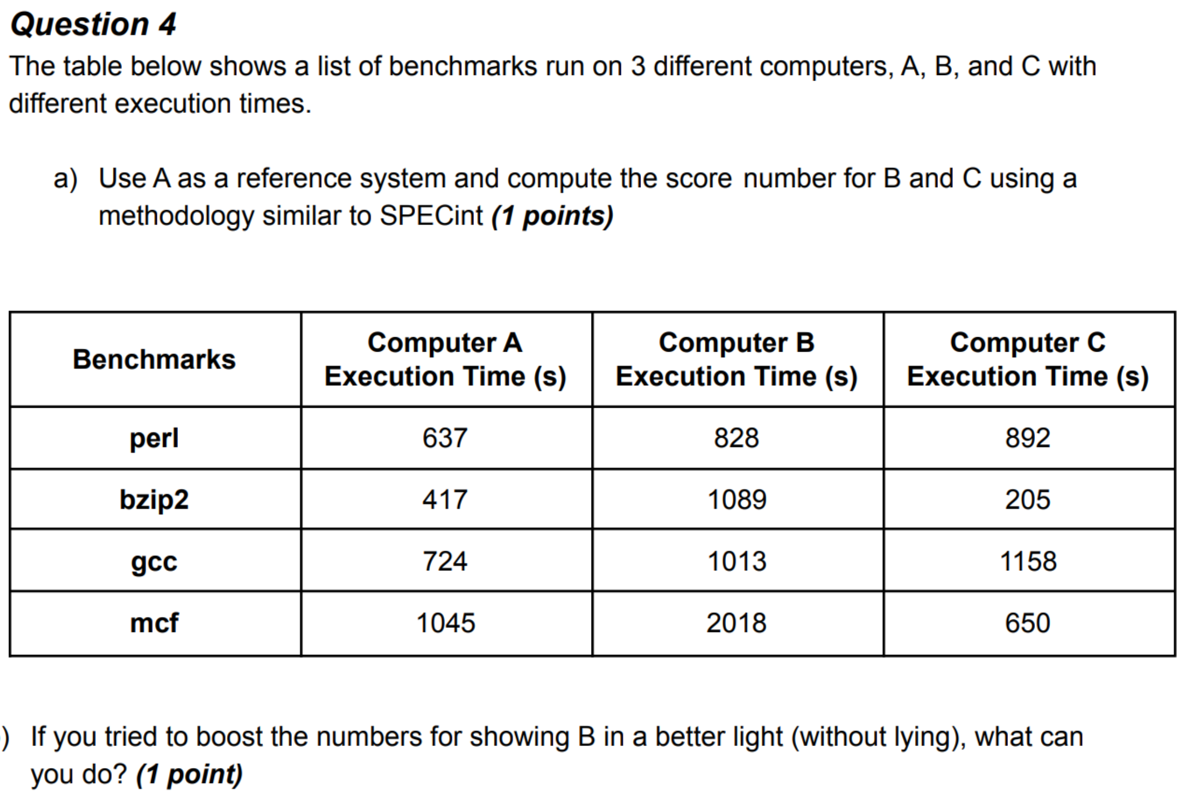
What was the new and the original percentage of the execution time dedicated to memory operations?

Solution. Let the branches and memory operation execution time be x, and y, then we have the equations

$$\begin{cases} \dfrac{0.3}{1.1} + \dfrac{0.2}{3} + \dfrac{x}{2.3} + \dfrac{y}{4} = 0.5 \\ x + y = 0.5 \end{cases} \tag{1}$$

$$\begin{cases} x \approx 0.192692 \\ y \approx 0.307308 \end{cases} \tag{2}$$

Thus, the branches take approximately 19.27% and memory operations take 30.73% in the total execution time. □



Question 4

The table below shows a list of benchmarks run on 3 different computers, A, B, and C with different execution times.

a) Use A as a reference system and compute the score number for B and C using a methodology similar to SPECint ***(1 points)***

Benchmarks	Computer A Execution Time (s)	Computer B Execution Time (s)	Computer C Execution Time (s)
perl	637	828	892
bzip2	417	1089	205
gcc	724	1013	1158
mcf	1045	2018	650

b) If you tried to boost the numbers for showing B in a better light (without lying), what can you do? ***(1 point)***

Solution for a. Using arithmetic mean to compute the speed up based on A

Benchmarks	A Execution Time	B Execution Time	C Execution Time
perl	1	1.2998	1.4003
bzip2	1	2.6115	0.4916
gcc	1	1.3992	1.5995
mcf	1	1.9311	0.622
arithmetic mean	1	1.8104	1.02835
speedup base A	1	0.552364118	0.972431565

□

Solution for b. Using geometric mean to compute the speed up based on A, so we can gain better speed up.

Benchmarks	A Execution Time	B Execution Time	C Execution Time
perl	1	1.2998	1.4003
bzip2	1	2.6115	0.4916
gcc	1	1.3992	1.5995
mcf	1	1.9311	0.622
geometric mean	1	1.740254561	0.909707757
speedup base A	1	0.574628576	1.09925412

□

5. Iron's Law helps us calculate the program run time and is the product of instruction per program, clock period, and cycles per instruction.
Program A is running on the processor P1 with a clock rate of 1.61 GHz. This program consists of 13 billion instructions. Program A has a floating-point multiplication instruction type that is executed 40% of the time and takes 5 cycles and the remaining instructions take 3 cycles on average to finish.

a) What is the execution time of program A? (1 point)

b) With some engineering investments, the machine compiler is enhanced to execute more instructions and allows a floating-point multiplication to take place three times as fast and rest of the instructions to finish in half the time. How long does it take program A to finish now? (1 point)

c) What is the gained speedup? (1 point)

Solution for a.

$$\begin{aligned}\text{Time per Program} &= \text{(instruction per program) * CPI * Clock Period}\\ &= 13*10^9\cdot(0.4*5+0.6*3)\cdot\frac{1}{1.61GHZ}\\ &= 30.6832\end{aligned}$$

□

Solution for b.

$$\begin{aligned}\text{Time per Program} &= \text{(instruction per program) * CPI * Clock Period}\\ &= 13*10^9\cdot(0.4*5/3+0.6*3/2)\cdot\frac{1}{1.61GHZ}\\ &= 12.6501\end{aligned}$$

□

Solution for c.

$$\begin{aligned}speedup &= \frac{T_{old}}{T_{new}}\\ &= \frac{Instruction\ count_{old}\cdot CPI_{old}\cdot Clock\ Cycle\ Time_{old}}{Instruction\ count_{new}\cdot CPI_{new}\cdot Clock\ Cycle\ Time_{new}}\\ &= \frac{30.6832}{12.6501}\\ &= 2.42553\end{aligned}$$

□

6. You need to write the following expression in 4 different ISAs using the high-level assembly used in class: $d = (b+a)*(b*a)+a$

a) For a generic accumulator-based architecture (1 point)

b) For a generic Register-Memory Architecture (1 point)

c) For a generic Load-Store Architecture (1 point)

d) For a RISC-V ISA (1 point)

Solution for a.

```

        ld(a)
        add b
        mult b
        mult a
        add a
        st (d)
```

□

Solution for b.

```

        ld,R1,(b)
        add,R2,R1,(a)
        mult,R3,R1,(a)
        add,R4,R2,R3
        st,R4,(d)
```

□

Solution for c.

```

        ld,R1,(b)
        ld,R2,(a)
        add,R3,R1,R2
        mult,R4,r1,r2
        add,R5,R3,R4
        st,R5,(d)
```

□

Solution for d.

```

        lw r1,0(b)
        lw r2,0(a)
        add r3,r1,r2
        mult r4,r1,r2
        add r5,r3,r4
        sw r5,0(d)
```

□

Question 7

a) Write the machine code for the following RISC-V programs (using an RV32G machine).

10238: ???????? beq a5,a4,10242

1023c: ???????? sd zero,128(a5)

101dc: ???????? lui s2,0xc07

b) Write the machine code for the following RISC-V programs. USE ONLY COMPRESS (RV32GC) if you can. Compress use 16bits, RV32 uses 32 bits per instruction.

101e4: ???? or ???????? addi a5,s2,-80

101e8: ???? or ???????? ld a5,8(a5)

10206: ???? or ???????? bnez a5,1020e

Solution for a.

1. 0x10242-0x10238=0xa=0b0000 0000 1010
   0b0 000000 01110 01111 000 0101 0 1100011
   0x00E78563

2. 0d128:0b10000000
   0b0000100 00000 01111 010 00000 0100011
   0x0807A023

3. 0b00000000110000000111 10010 0110111
   0x00C07937

□

Solution for b.

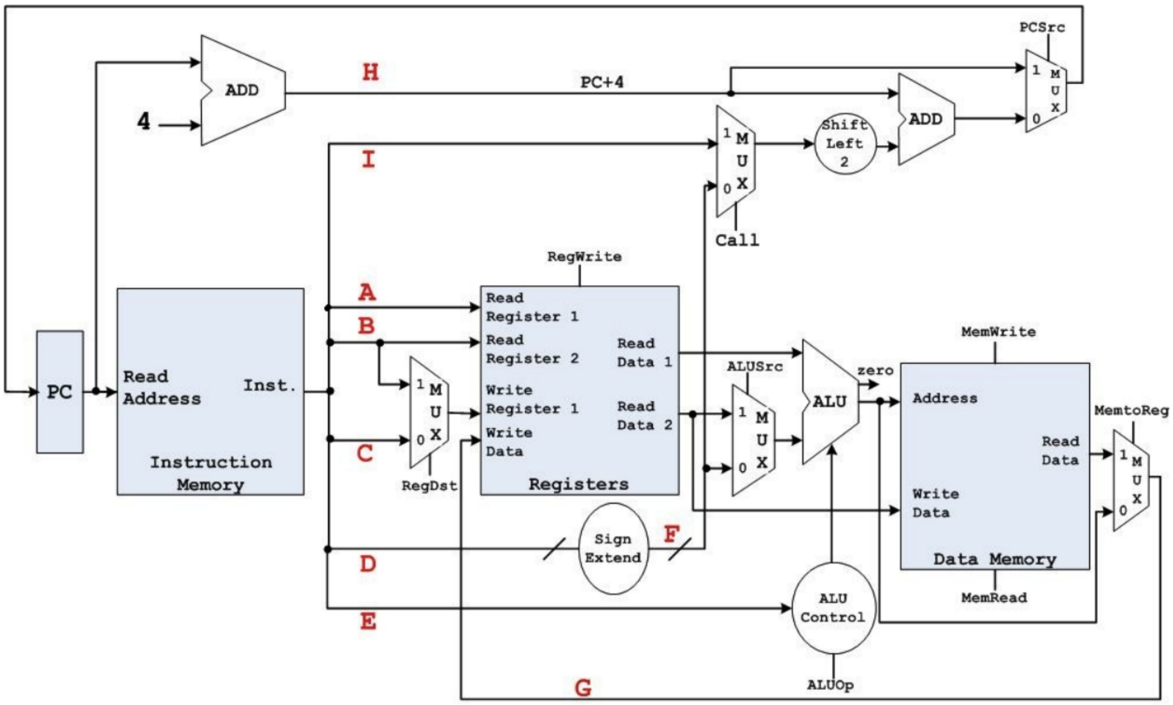
1. 0d-80:0b111110110000
   0b111110110000 10010 000 01111 0010011
   RV32G:0xFB090793
   can’t express in RV32C

2. 0d8:1000
   ob000000001000 01111 010 01111 0000011
   RV32G:0x0087A783
   0b 010 010 111 0 0 111 00
   RV32C:0x4B9C

3. 0x1020e-0x10206=0x8=0b 0000 0000 1000
   0b 0000000 00000 01111 001 10000 1100011
   RV32G:0x00079863
   0b111 0 10 111 00 00 0 01
   RV32C:0xEB81

□

Question 8



a)The previous datapath diagram can execute many instructions. If the CPU were to execute a "XORI R3,R5,-7" in RISC-V, what would be the values in each of the wires marked with a letter? The register file is initialized with r0=0x100,r1=0x101,r2=0x102,r3=0x103… the program counter for the instruction is 0x100. If a value is not relevant or known because of lack of instruction encoding, use X. ***(1 point)***

A	B	C	D	E	F	G	H	I

b) If the previous diagram were to execute a subset of RISC-V instructions. Set the control signals for the following LD, CALL, and SUB instructions. ***(1 point)***

```
ld   x7, 1(R3)
call 0xDEED
sub  x13, x17, 7
```

Solution for a.

a	b	c	d	e	f	g	h	i
R5	X	R3	-7	XORI	sign extend -7	f xor 0x105	0x104	X

□

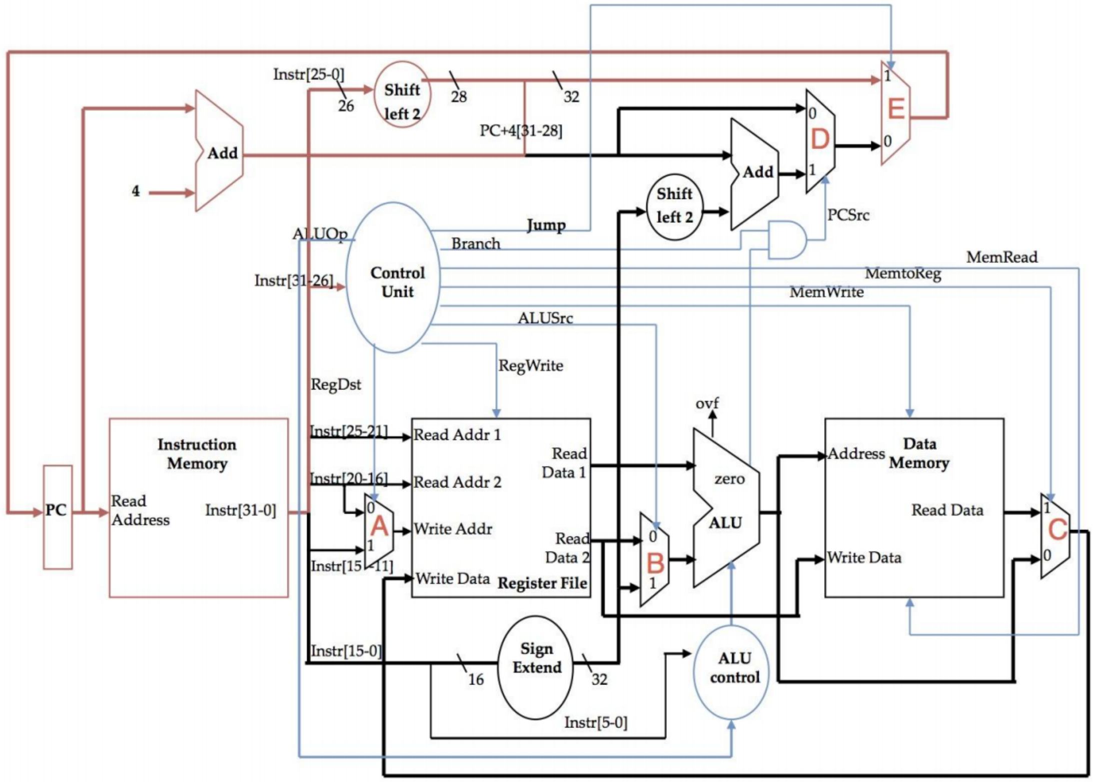
Solution for b.

	BR	JP	ALUsrc	ALUop	Dmwe	Rwe	Rdst	Rwd
lw x7, 1(r3)	X	0	0	lw	0	1	0	1
call 0xDEED (auipc)	X	0	0	auipc	0	1	0	1
call 0xDEED (jalr)	X	1	0	jalr	0	1	0	1
sub x13,x17,7	X	0	1	sub	0	1	0	1

□

Question 9

The single-cycle datapath below is taken from some textbook (slight variation).



For the different sets of control instructions write the control values for the multiplexor D and E (either 0 or 1)? ***(2 points)*** The (taken) (not taken) comment in the branches indicates if the branch was taken or not taken.

Solution.

Instruction	MUX D	MUX E
jr x30	X	1
bne x1,x0,foo(not taken)	0	0
jalr x0, 0xFF00	X	1
ret	X	1
jal 0xAAC0	X	1
beq x1,x0,bar(taken)	1	0

□