Scott Strathman
525.612 SP24
Dr. Beser
Assignment #1
1/27/24

1.)

| | $P_1$ | $P_2$ | $P_3$ |
|---|---|---|---|
| Clock Rate | 3 GHz | 2.5 GHz | 4 GHz |
| CPI (Cycles per Instruction) | 1.5 | 1.0 | 2.2 |

1a.) $P_1$/ $\frac{1}{\text{Clock Rate}} = \frac{1}{3e9} = 333\,e{-12}\ \text{sec/cycle}$

$$P_1^{-1} = \frac{\text{sec}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} = \left(333e{-12}\ \frac{\text{sec}}{\text{cycle}}\right)\left(1.5\ \frac{\text{cycles}}{\text{instruction}}\right) 500\,e{-12}\ \text{sec/instruction}$$

$$\left\{P_1 = \frac{1}{P_1^{-1}} = 2e9\ \text{instructions/sec}\right\}$$

$$P_2^{-1} = \left(\frac{1}{2.5e9}\right)\left(1.0\ \frac{\text{cycle}}{\text{instruction}}\right) = 400\,e{-12}\ \text{sec/instruction}$$

$$\left\{P_2 = 2.5e9\ \text{instructions/sec}\right\}$$

$$P_3 = \frac{1}{\left(\frac{1}{4e9}\right)(2.2)} \quad \left\{P_3 = 1.81\,e9\ \text{instructions/sec}\right\}$$

$\boxed{P_2 \text{ has the highest performance}}$

1b.) Program in 10 seconds

Find: • number of cycles
• number of instructions

$P_1$: $3e9 \frac{cycles}{\cancel{sec}} \times 10 \cancel{sec} = \boxed{3 \times 10^{10} \text{ cycles}}$ $P_1$

$2e9 \frac{instructions}{sec} \times 10 sec = \boxed{2 \times 10^{10} \text{ instructions}}$

$P_2$: $2.5e9 \frac{cycles}{sec} \times 10 sec = \boxed{2.5 \times 10^{10} \text{ cycles}}$ $P_2$

$2.5e9 \frac{instructions}{sec} \times 10 sec = \boxed{2.5 \times 10^{10} \text{ instructions}}$

$P_3$: $4e9 \frac{cycles}{sec} \times 10 sec = \boxed{4 \times 10^{10} \text{ cycles}}$ $P_3$

$1.81e9 \frac{instructions}{sec} \times 10 sec = \boxed{1.81 \times 10^{10} \text{ instructions}}$

1c.) $$\text{CPU Time} = \left(\frac{\text{Instructions}}{\text{Program}}\right)\left(\frac{\text{Clock cycles}}{\text{Instruction}}\right)\left(\frac{\text{seconds}}{\text{Clock cycle}}\right) \leftarrow \frac{1}{\text{Clock rate}}$$

$$P_1: \; 0.7\left(\frac{10\text{ sec}}{\cancel{\text{Program}}}\right) = \left(\frac{2\times10^{10}\ \cancel{\text{Instructions}}}{\cancel{\text{Program}}}\right)\left(\frac{1.2 \times 1.5 \text{ cycles}}{\cancel{\text{Instruction}}}\right)\left(\frac{1}{\text{Clock rate}}\right)$$

$$P_{1,new} = \frac{(2e10)(1.2)(1.5)}{7}$$

$$\boxed{P_{1,new} = 5.143 \text{ GHz}}$$

$$P_2: \; 0.7\left(\frac{10\text{ sec}}{\cancel{\text{program}}}\right) = \left(\frac{2.5\times10^{10}\ \cancel{\text{instructions}}}{\cancel{\text{program}}}\right)\left(\frac{1.2 \times 1.0 \text{ cycles}}{\cancel{\text{instruction}}}\right)\left(\frac{1}{\text{Clock rate}}\right)$$

$$P_{2,new} = \frac{(2.5\times10^{10})(1.2\times1.0)}{7}$$

$$\boxed{P_{2,new} = 4.286 \text{ GHz}}$$

$$P_3: \; 0.7\left(\frac{10\text{ sec}}{\cancel{\text{program}}}\right) = \left(\frac{1.81\times10^{10}\ \cancel{\text{instructions}}}{\cancel{\text{program}}}\right)\left(\frac{1.2 \times 2.2 \text{ cycles}}{\cancel{\text{instruction}}}\right)\left(\frac{1}{\text{Clock rate}}\right)$$

$$\text{Clock rate} = \frac{(1.81\times10^{10})(1.2\times2.2)}{7}$$

$$\boxed{P_{3,new} = 6.826 \text{ GHz}}$$

2.) A B C D

$P_1$ 2.5 GHz

| | A | B | C | D |
|---|---|---|---|---|
| CPIs | 1 | 2 | 3 | 3 |

10% class A

20% class B

50% class C

20% class D

$P_2$ 3 GHz

| | A | B | C | D |
|---|---|---|---|---|
| CPI | 2 | 2 | 2 | 2 |

Global CPI's:

2a) $P_1$: $P_{1,CPI} = (1)(0.1) + (2)(0.2) + (3)(0.5) + (3)(0.2)$

$$\boxed{P_{1,CPI} = 2.6}$$

$P_2$: $P_{2,CPI} = (2)(0.1) + (2)(0.2) + (2)(0.5) + (2)(0.2)$

$$\boxed{P_{2,CPI} = 2}$$

2b.) Given 1e6 instructions $P_{1_{clock}} = 2.5e9 \frac{cycles}{second}$ $P_{2_{clock}} = 3e9 \frac{cycles}{second}$

$$P_1 = \left(\frac{1}{2.5e9 \frac{cycles}{second}}\right)\left(2.6 \frac{cycles}{instruction}\right)\left(1e6\ instructions\right) = P_{1_{time}}$$

$$P_{1_{time}} = \left(400e{-12} \frac{seconds}{cycle}\right)\left(2.6 \frac{cycles}{instruction}\right)\left(1e6\ instructions\right)$$

$$P_{1_{time}} = 1.04e{-3}\ sec = \underline{1.04\ ms}$$

$$P_{1\,cycles} = \left(2.5e9 \frac{cycles}{second}\right)\left(1.04e{-3} \frac{seconds}{program}\right)$$

$$\boxed{P_1\ cycles = 2.6 \times 10^6\ cycles}$$

$P_2$ is Faster

$$P_2: P_{2_{time}} = \left(\frac{1}{3e9 \frac{cycles}{second}}\right)\left(2 \frac{cycles}{instruction}\right)\left(1e6\ instructions\right) = 666.7 \times 10^{-6}$$

$$P_{2_{time}} = \underline{666.7\,\mu s}$$

$$P_2\ cycles = \left(3e9 \frac{cycles}{second}\right)\left(666.7e{-6} \frac{seconds}{program}\right)$$

$$\boxed{P_2\ cycles = 2 \times 10^6\ cycles}$$

3.) compiler A : 1.0e9 instructions @ 1.1 sec execution time

compiler B : 1.2 e9 instructions @ 1.5 sec execution time

3a.) $P_{\text{cycle time}} = 1\text{ns} \rightarrow$ Clock Rate $= 1$ GHz

$$\text{CPU Time} = \frac{\text{Instructions}}{\text{program}} \times \frac{\text{clock cycles}}{\text{Instruction}} \times \frac{\text{seconds}}{\text{clock cycle}}$$

$\downarrow$

$$\text{CPI} = \frac{\text{CPU Time}}{\left(\frac{\text{Instructions}}{\text{program}}\right)\left(\frac{\text{seconds}}{\text{clock cycle}}\right)}$$ ✓

$$\text{CPI}_A = \frac{1.1 \text{ sec}}{(1\text{e}9 \text{ instructions})\left(1\text{e}{-9} \frac{\text{sec}}{\text{cycle}}\right)} \rightarrow \boxed{\text{CPI}_A = 1.1 \frac{\text{cycles}}{\text{instruction}}}$$

$$\text{CPI}_B = \frac{1.5 \text{ sec}}{(1.2\text{e}9 \text{ instructions})\left(1\text{e}{-9} \frac{\text{sec}}{\text{cycle}}\right)} \rightarrow \boxed{\text{CPI}_B = 1.25 \frac{\text{cycles}}{\text{instruction}}}$$

3b.)

CPU time is equal

$$\frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{1}{\text{Clock}_A} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{1}{\text{Clock}_B}$$

$$\frac{1e9 \text{ instructions}}{\text{Program}} \times \frac{1.1 \text{ cycles}}{\text{instruction}} \times \frac{1}{\text{Clock } A \frac{\text{cycles}}{\text{sec}}} = \frac{1.2e9 \text{ instructions}}{\text{Program}} \times \frac{1.25 \text{ cycles}}{\text{instruction}} \times \frac{1}{\text{Clock}_B}$$

$$\frac{(1e9)(1.1)}{(1.2e9)(1.25)} = \boxed{\frac{\text{Clock } A}{\text{Clock } B} = 0.733}$$

clock A is not faster... Clock B is

3c.) Compiler C: 6e8 instruction, $CPI_C = 1.1 \frac{\text{cycles}}{\text{Instruction}}$

$$(6e8 \text{ instructions})\left(1.1 \frac{\text{cycles}}{\text{instruction}}\right) = 660e6 \text{ cycles}$$ ✓

$$\text{Clock rate} = 1e9 \frac{\text{cycles}}{\text{sec}}$$

$$C_{\text{execution time}} = \frac{660e6 \text{ cycles}}{1e9 \frac{\text{cycles}}{\text{sec}}} = \boxed{0.660 \text{ sec} : C_{\text{execution time}}}$$

CPI's:

4.) arithmetic: $1 \frac{\text{cycle}}{\text{instruction}}$ — load/store: $12 \frac{\text{cycles}}{\text{instruction}}$ — branch: $5 \frac{\text{cycles}}{\text{instruction}}$

2 GHz clock

$\downarrow$

2.56e9 instructions — 1.28e9 instructions — 256e6 instructions

$\uparrow$

$/0.7 \times p$ — $/0.7 \times p$

☐ 1, 2, 4, 8 processors

☐ show relative speedup for each

$$\text{CPU Time } (t) = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

4a.)

1 processor

$$\frac{\text{cycles}}{\text{program}} = (2.56e9 \text{ instruction})\left(\frac{1 \text{ cycle}}{\text{instruction}}\right) + (1.28e9 \text{ instructions})\left(\frac{12 \text{ cycles}}{\text{instruction}}\right) + (256e6 \text{ instructions})\left(\frac{5 \text{ cycles}}{\text{instruction}}\right)$$

$$\frac{\text{cycles}}{\text{program}} = 2.56e9 \text{ cycles} + 15.36e9 \text{ cycles} + 1.28e9 = 19.2e9 \frac{\text{cycles}}{\text{program}} \checkmark$$

$$t_{1 \text{ processor}} = \left(19.2e9 \frac{\text{cycles}}{\text{program}}\right)\left(\frac{\text{second}}{2e9 \text{ cycles}}\right)$$

$$\boxed{t_{1 \text{ processor}} = 9.6 \text{ seconds}}$$

4a cont.)

2processors

$$\frac{\text{cycles}}{\text{program}} = \left(\frac{2.56e9 \text{ instructions}}{0.7 \times 2 \text{ processors}}\right)\left(\frac{1 \text{ cycle}}{\text{instruction}}\right) + \left(\frac{1.28e9 \text{ instructions}}{0.7 \times 2 \text{processors}}\right)\left(\frac{12 \text{ cycles}}{\text{instruction}}\right) + \left(256e6 \text{ instructions}\right)\left(\frac{5 \text{ cycles}}{\text{instruction}}\right)$$

$$\frac{\text{cycles}}{\text{program}} = 1.82857e9 \text{ cycles} + 10.9714e9 \text{ cycles} + 1.28e9 \text{ cycles}$$

$$\frac{\text{cycles}}{\text{program}} = \underline{14.08e9 \text{ cycles}} \checkmark$$

$$t_{2\text{ processor}} = \left(14.08e9 \frac{\text{cycles}}{\text{program}}\right)\left(\frac{\text{second}}{2e9 \text{ cycles}}\right)$$

$$\boxed{t_{2\text{ processor}} = 7.04 \text{ seconds}} \leftarrow \frac{t_2}{t_1} = \frac{7.04}{9.6} = 0.733$$

36% less time than using 1 processor

4 processors

$$\frac{\text{cycles}}{\text{program}} = \left(\frac{2.56e9 \text{ instructions}}{0.7 \times 4 \text{ processors}}\right)\left(\frac{1 \text{cycle}}{\text{instruction}}\right) + \left(\frac{1.28e9 \text{ instructions}}{0.7 \times 4 \text{ processors}}\right)\left(\frac{12 \text{ cycles}}{\text{instruction}}\right) + \left(256e6 \text{ instructions}\right)\left(\frac{5 \text{ cycles}}{\text{instruction}}\right)$$

$$914.286e6 \text{ cycles} + 5.48571e9 \text{ cycles} + 1.28e9 \text{ cycles} = \underline{7.68e9 \text{ cycles}} \checkmark$$

$$t_{4\text{ processor}} = \left(7.68e9 \frac{\text{cycles}}{\text{program}}\right)\left(\frac{\text{second}}{2e9 \text{ cycles}}\right)$$

$$\boxed{t_{4\text{ processor}} = 3.84 \text{ seconds}} \leftarrow \frac{t_4}{t_1} = \frac{3.84}{9.6} = 0.4$$

60% less time than using 1 processor



Scanned by CamScanner

4a cont.)

8 processors

$$\frac{\text{cycles}}{\text{program}} = \left(\frac{2.56\text{e}9 \text{ instructions}}{0.7 \times 8 \text{ processors}}\right)\left(\frac{1 \text{ cycle}}{\text{instruction}}\right) + \left(\frac{1.28\text{e}9 \text{ instructions}}{0.7 \times 8 \text{ processors}}\right)\left(\frac{12 \text{ cycles}}{\text{instruction}}\right) + \left(256\text{e}6\right)\left(5\right)$$

$$\frac{\text{cycles}}{\text{Instruction}} = 457.143\text{e}6 \text{ cycles} + 2.74286\text{e}9 \text{ cycles} + 1.28\text{e}9 \text{ cycles} = 4.48\text{e}9 \text{ cycles}$$

$$t_{8\,\text{processor}} = \left(4.48\text{e}9 \frac{\text{cycles}}{\text{program}}\right)\left(\frac{\text{second}}{2\text{e}9 \text{ cycles}}\right)$$

$$\boxed{t_{4\,\text{processor}} = 2.24 \text{ seconds}} \leftarrow \frac{t_8}{t_1} = \frac{2.24}{9.6} = 0.233$$

77% less time than using 1 processor

4b.) CPI of arithmetic instructions is now doubled

| arithmetic | load/store | branch |
| --- | --- | --- |
| $\underline{\underline{2}}$ $\frac{\text{cycles}}{\text{instruction}}$ | 12 $\frac{\text{cycles}}{\text{instruction}}$ | 5 $\frac{\text{cycles}}{\text{instruction}}$ |
| 2.56 e9 instructions | 1.28 instructions | 256 e6 instructions |

1 processor

arithmetic: $(2.56e9 \text{ instructions})\left(\frac{2 \text{ cycles}}{\text{inst}}\right) = 5.12\,e9$ instructions

$\frac{\text{cycles}}{\text{program}} = 5.12\,e9 + 15.36\,e9 + 1.28\,e9 = 21.76\,e9$ cycles/program

$t_{1\,processor} = \left(21.76\,e9 \frac{\text{cycles}}{\text{program}}\right)\left(\frac{\text{second}}{2e9 \text{ cycles}}\right)$

$\boxed{t_{1\,processor} = 10.88 \text{ seconds}}$ ← compared to 9.6 seconds

2 processors

arithmetic: $\left(\frac{2.56\,e9 \text{ instructions}}{0.7 \times 2 \text{ processors}}\right)\left(\frac{2 \text{ cycles}}{\text{instruction}}\right) = 3.657\,e9 \frac{\text{cycles}}{\text{program}}$

$\frac{\text{cycles}}{\text{program}} = 3.657e9 + 10.9714e9 + 1.28\,e9 = 15.9085e9$ cycles ✓

$\boxed{t_{2\,processor} = \frac{(15.9085e9)}{2e9} = 7.95 \text{ seconds}}$ ← compared to 7.04 seconds

4 processors

$(914.286e6 \text{ cycles})(2) + 5.48571e9 \text{ cycles} + 1.28e9 \text{ cycles} = 8.59428\,e9$ cycles ✓

$\boxed{t_{4\,processor} = \frac{8.59428e9}{2e9} = 4.297 \text{ seconds}}$ ← compared to 3.84 seconds

8 processors

$(457.143e6 \text{ cycles})(2) + 2.74286e9 \text{ cycles} + 1.28e9 \text{ cycles} = 4.93715e9$ ✓

$\boxed{t_{8\,processor} = \frac{4.93715e9}{2e9} = 2.469 \text{ seconds}}$ ← compared to 2.24 seconds

4c.) Goal: reduce CPI of load/store so $t_1$ matches $t_4$

| arithmetic | load/store | branch |
|---|---|---|
| $1 \frac{\text{cycle}}{\text{instruction}}$ | $\underline{\underline{X}} \frac{\text{cycles}}{\text{instruction}}$ | $5 \frac{\text{cycles}}{\text{instruction}}$ |
| ↓ | ↓ | ↓ |
| 2.56e9 instructions | 1.28e9 instructions | 256 e6 instructions |

CPU Time = $t_{4\,processor} = 3.84$ seconds

$$\left[(2.56e9)(1) + (1.28e9)X + (256e6)(5)\right]\left(\frac{\text{second}}{2e9\text{ cycles}}\right) = 3.84$$

$$2.56e9 + (1.28e9)X + 1.28e9 = 7.68e9$$

$$(1.28e9)X = 3.84e9$$

$$\boxed{X = 3 \text{ cycles/instruction}}$$

load/store needs to be reduced from 12 CPI to 3 CPI

5.)

\$s0 f int f, g, h, i, j;

\$s1 g \$s6 A[0]

\$s2 h \$s7 B[0]

\$s3 i

\$s4 j

number of words (32 bit slots) C code

```
1  sll $t0, $s0, 2        # t0 = f * 4              (f x 4)
2  add $t0, $s6, $t0      # t0 = &A[f x 4]          &A[f]
3  sll $t1, $s1, 2        # t1 = g * 4              (g x 4)
4  add $t1, $s7, $t1      # t1 = &B[g]              &B[g] ←
5  lw $s0, 0($t0)         # f = A[f]   ← the value @ A[f] is assigned to f
6  addi $t2, $t0, 4       # t2 = &A[f]+4            &A[f+1] ?
7  lw $t0, 0($t2)         # t0 = &A[f+4]  ← the value @ A[f+1] is assigned to t0
8  add $t0, $t0, $s0      t0 = &A[f+4] + f          A[f+1] + A[f]
                                        ↑
                                       A[f]
9  sw $t0, 0($t1)         &B[g] = $t0
                                   ↑
                               &A[f+4] + f
```

the value at &B[g]

5 cont.)

$\Downarrow$

$t\emptyset = (f \times 4)$

$\&A[f \times 4]$

1 and 2

$f = f \times 4$

3 and 4

$g = g \times 4$

$\downarrow$

$\&A[0 + (f \times 4)]$ $\&B[0 + (g \times 4)]$

$\hookrightarrow \&A[f]$ $\hookrightarrow \&B[g]$

$f = A[f]$

5) f gets the value at A[f]

6) $t2 = \&A[f+1]$

7) $t\emptyset = A[f+1]$

8) store $t\emptyset$ in B[g], which is A[f+1] + A[f]

$$\boxed{B[g] = A[f] + A[f+1];}$$

6.)

$t1 i

$s2 result

$s0 &MemArray[0]

bne ← while loop

```
      add $t1, $t0, 0        ← i = 0;
LOOP: lw $s1, 0($s0)         ← load MemArray[0] into s1
      add $s2, $s2, $s1      ← result = result + MemArray[0] [next time, MemArray[i]]
      add $s0, $s0, 4        ← increment &MemArray by 1 index
      addi $t1, $t1, 1       ← i = i + 1
      slti $t2, $t1, 100     ← if i is less than 100, set $t2 to 1
      bne $t2, $t0, LOOP     ← if $t2 is set, then LOOP
      nop                    ← delay
```

→ MemArray index is same value as i in C

```
i = 0;
while(i < 100)
{
    result = result + MemArray[i];
    i++;
}
```

7.) ☑ → Generate an assembled version of the program

☐ → Provide the equations to solve for the LOOP field

From MARS simulator:

| Address | Code | Assembly |
|---|---|---|
| 0x00000000 | 0x20090000 | addi $t1, $0, 0 |
| → 0x00000004 | 0x8e110000 | LOOP: lw $s1, 0($s0) |
| 0x00000008 | 0x02519020 | add $s2, $s2, $s1 |
| 0x0000000c | 0x22100004 | addi $s0, $s0, 4 |
| 0x00000010 | 0x21290001 | addi $t1, $t1, 1 |
| 0x00000014 | 0x292a0064 | slti $t2, $t1, 100 |
| → 0x00000018 | 0x1540fffa | bne $t2, $0, LOOP |
| 0x0000001c | 0x00000000 | nop |

bne equation if(R[rs] != R[rt])

PC = PC + 4 + BranchAddr

7 cont.)

bne equation: PC = PC + 4 + Branch Addr

bne equation works as follows:

1.) PC is the current address (in this case, 0x00000018)

2.) Adding 4 gives us the next address (in this case, 0x0000001C)

3.) The Branch Addr is listed as 0xfffa in the immediate field. Its decimal value is -6, which is the number of words that the PC should move if the condition is met.

This is so a larger address space can be accessed

Thus, -6 words × 4 bytes/word = -24 bytes (in decimal)

and 0x1C minus 24 bytes is 0x04, which is where the LOOP label is located.