HW #5 Yash Shah

Problem 4.16

4.16.1)

slowest datapath stage = ID = 350 ps ⇒ clock cycle time for pipelined = 350 ps

added datapaths = 250 + 350 + 150 + 300 + 200 = 1250 ps ⇒ clock cycle time for non-pipelined = 1250 ps

4.16.2)

lw = 20% = $\frac{1}{5}$ ⇒ total latency (pipelined) = 5 · clock cycle of pipelined

= 5 · 350 ps

= 1750 ps

latency pipelined = 1750 ps

total latency (non-pipelined) = clock cycle of non-pipelined

= 1250 ps

latency non-pipelined = 1250 ps

4.16.3)

Split ID because it has the longest cycle time. That means each stage now has 175 ps cycle times in each. That makes the new clock cycle time the next largest stage, which is now the MEM stage, that is, 300 ps.

Split ID, new cycle time = 300 ps

4.16.4)

Utilization of data memory = lw + sw

= 20% + 15%

= 35% data memory utilization

4.16.5)

Utilization of the write-register port = ALU + jump

= 45% + 20%

= 65% utilization

= 1 - data memory utilization

Problem 4.27

4.27.1)

```
add  $s3, $s1, $s0
nop
nop
lw   $2, 4($s3)
lw   $s1, 0($s2)
nop
or   $s2, $s3, $2
nop
nop
sw   $s2, 0($s3)
```

4.27.2) The instructions are interleaved so we cannot remove any NOP's

4.27.3) The processor returns false — the processor is unable to forward the data and thus returns false values.

4.27.4)

ADD	FP	FD	FX	M	W								
LW			FP	FD	FX	M	W						
LW					FP	FD	FX	M	W				
OR							FP	FD	FX	M	W		
SW									FP	FD	FX	M	W

4.27.5) No additional outputs are needed

4.27.6) Asserts

cycles 1, 2, 3 — IF/ID write = 1, control mux = 0

cycles 4, 5 — IF/ID write = 0, control mux = 1

Problem 4.28

4.28.1) Branch outcomes determined in EX $\Rightarrow$ ID, IF, and EX contribute to stall cycles due to mis predicted branches

$\Rightarrow 1+1+1 = 3$ stall cycles

Increase in CPI = # stall cycles · beq · (1 − always taken)

$= 3 \cdot 25\% \cdot (1-40\%)$

$= 3 \cdot .25 \cdot .55$

$= 0.413$

extra CPI = 0.413

4.28.2) extra CPI $= 3 \cdot 25\% \cdot (1 - \text{always not taken})$

$= 3 \cdot .25 \cdot (1-55\%)$

$= 3 \cdot .25 \cdot .45$

extra CPI $= 0.338$

4.28.3) extra CPI $= 3 \cdot 25\% \cdot (1 - \text{2bit predictor})$

$= 3 \cdot .25 \cdot (1-85\%)$

extra CPI $= 0.113$

4.28.4) convert branch instruction to some ALU instructions $\Rightarrow$ beq $\cdot \frac{1}{2} = 25\% \cdot \frac{1}{2} = .125$

extra CPI $= 3 \cdot (1-85\%) \cdot .125$

$= 0.056$

$$\text{speedup} = \frac{1+0.113}{1+0.056}$$

speedup = 1.054

4.28.5)

current CPI = 1.113

New CPI $= 1 + (1+3(0.15))(0.25)(0.5) = 1.181$

$$\text{speedup} = \frac{\text{current CPI}}{\text{new CPI}} = \frac{1.113}{1.181}$$

$= 0.94$

94% speedup

4.28.6)

80% of all executed branch instructions are easy-to-predict

$$\text{accuracy} = \frac{\#\text{ instr.} \cdot (85\% - 80\%)}{\#\text{ instr.} \cdot (100\% - 80\%)}$$

$$= \frac{0.05}{0.20}$$

accuracy = 0.25

accuracy = 25%