CSE301 – Computer Organization

Tutorial 5

Q1 The 5 stages of the processor have the following latencies:The 5 stages of the processor have the following latencies:

	Fetch	Decode	Execute	Memory	Write back
a.	300ps	400ps	350ps	550ps	100ps
b.	200ps	150ps	100ps	190ps	140ps

Assume that when pipelining, each pipeline stage costs 20ps extra for the registers between pipeline stages.

1. Non-pipelined processor: what is the cycle time? What is the latency of an instruction? What is the throughput?
2. Pipelined processor: What is the cycle time? What is the latency of an instruction? What is the throughput?
3. If you could split one of the pipeline stages into 2 equal halves, which one would you choose? What is the new cycle time? What is the new latency? What is the new throughput?

Q2 Assume the distribution of instructions that run on the processor is:

```
50%: ALU
25%: BEQ
15%: LW
10%: SW
```

Assuming there are no stalls or hazards, what is the utilization of the data memory? What is the utilization of the register block's write port? (Utilization in percentage of clock cycles used)

Q3 You are given a non-pipelined processor design which has a cycle time of 10ns and average CPI of 1.4. Calculate the latency speedup in the following questions

1. What is the best speedup you can get by pipelining it into 5 stages?
2. If the 5 stages are 1ns, 1.5ns, 4ns, 3ns, and 0.5ns, what is the best speedup you can get compared to the original processor?
3. If each pipeline stage added also adds 20ps due to register setup delay, what is the best speedup you can get compared to the original processor?
4. The pipeline from 3 stalls 20% of the time for 1 cycle and 5% of the time for 2 cycles (these occurences are disjoint). What is the new CPI? What is the speedup compared to the original processor?

Q4 Sequence of instructions:

```
lw  $s2, 0($s1)
lw  $s1, 40($s3)
sub $s3, $s1, $s2
add $s3, $s2, $s2
or  $s4, $s3, $zero
sw  $s3, 50($s1)
```

1. List the Read-After-Write data dependencies.
2. Assume the 5-stage MIPS pipeline with no forwarding, and each stage takes 1 cycle. Instead of inserting nops, you let the processor stall on hazards. How many times does the processor stall? How long is each stall (in cycles)? What is the execution time (in cycles) for the whole program?
3. Assume the 5-stage MIPS pipeline with full forwarding. Write the program with nops to eliminate the hazards. (Hint: time travel is not possible!)

Q5 You'd like to add a branch predictor to your Baseline processor, and you're considering two options: PikaChooser and CharWizard. Evaluate the speedup of each relative to your Baseline if branches are 15% of all instructions. Assume normal CPI is 1, but the branch mispredict penalty is 2 extra stall cycles.

PikaChooser:

```
10% misprediction rate
Will increase the cycle time by 15%
```

CharWizard:

```
12% misprediction rate
Will increase the cycle time by 20%
```

1. Which predictor would you choose?
2. If branches are instead 25% of all instructions, which predictor would you choose?

Part 2: Questions with Answers

Q1

[1]

(a)
CT = 300 + 400 + 350 + 550 + 100 = 1700ps
Latency = 1700ps
Throughput = 1/1700 inst/ps

(b)
CT = 200 + 150 + 100 + 190 + 140 = 780ps
Latency = 780ps
Throughput = 1/780 inst/ps

[2]

(a)
CT = 550 + 20 = 570 ps
Latency = 5 * 570 = 2850ps
Throughput = 1/570 inst/ps

(b)
CT = 200 + 20 = 220 ps
Latency = 5 * 220 = 1100ps
Throughput = 1/220 inst/ps

[3]
(a)
CT = 400 + 20 = 420 ps
Latency = 6 * 420 = 2520 ps
Throughput = 1/420 inst/ps

(b)
CT = 190 + 20 = 210 ps
Latency = 6 * 210 = 1260 ps
Throughput = 1/210 inst/ps

Q2

LW and SW instructions use the data memory. As a result, the utilization of the data memory is 15% + 10% = 25%.

ALU and LW instructions use the register block's write port. As a result, the utilization of the register block's write port is 50% + 15% = 65%.

Q3

[1]

5x speedup

The new latency would be 10ns/5 = 2ns

[2]

The cycle time is limited by the slowest stage, so CT = 4ns.

Speedup = old CT / new CT = 10ns/4ns = 2.5x

[3]

Adding the register delay, the new CT = 4.02ns.

Speedup = 10ns/4.02ns = 2.488x

[4]

New CPI = 0.2(2.4) + 0.05(3.4) + 0.75(1.4) = 1.7

Old performance = old CT * old CPI = 10 * 1.4 = 14

New performance = new CT * new CPI = 4.02 * 1.7 = 6.834

Speedup = 14/6.834 = 2.049x

Q4

[1]

List the Read-After-Write data dependencies.
3 on 1 ($s2)
3 on 2 ($s1)
4 on 1 ($s2)
5 on 4 ($s3)
6 on 2 ($s1)
6 on 4 ($s3)

[2]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	F	D	E	M	W									
2		F	D	E	M	W								
3			F	-	-	D	E	M	W					
4						F	D	E	M	W				
5							F	-	-	D	E	M	W	
6										F	D	E	M	W

The processor stalls twice for instructions 3 and 5

[3]

	1	2	3	4	5	6	7	8	9	10	11
1	F	D	E	M	W						
2		F	D	E	M	W					
nop											
3				F	D	E	M	W			

4				F	D	E	M	W		
5					F	D	E	M	W	
6						F	D	E	M	W

Q5

$CT_{\text{PikaChooser}} = 1.15$

$CPI_{\text{PikaChooser}} = 0.15 \ast (0.1 \ast 3 + 0.9) + (1 - 0.15) = 0.15 \ast 1.2 + 0.85 = 1.03$

$ET_{\text{PikaChooser}} = CT_{\text{PikaChooser}} \ast CPI_{\text{PikaChooser}} = 1.15 \ast 1.03 = 1.185$

$CT_{\text{CharWizard}} = 1.2$

$CPI_{\text{CharWizard}} = 0.15 \ast (0.12 \ast 3 + 0.88) + (1 - 0.15) = 0.15 \ast 1.24 + 0.85 = 1.036$

$ET_{\text{CharWizard}} = CT_{\text{CharWizard}} \ast CPI_{\text{CharWizard}} = 1.2 \ast 1.036 = 1.243$

PikaChooser is the better branch predictor.