Review 2

Pipelined Processor

1) Assume that individual stages of the MIPS processor datapath have the following latencies

IF	ID	EX	MEM	WB
250ps	350ps	150ps	300ps	200ps

Also, assume that instructions executed by the processor are broken down as follows:

ALU	beq	lw	sw
45%	20%	20%	15%

a) What is the clock cycle time in a pipelined and non-pipelined processor?
b) What is the total latency of an LW instruction in a pipelined and non-pipelined processor?
c) If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor?
d) Assuming there are no stalls or hazards, what is the utilization of the data memory?
e) Assuming there are no stalls or hazards, what is the utilization of the write-register port of the "Registers" unit?
f) Assuming there are no stalls or hazards, what is the utilization of the sign extender?

A) Pipelined 350 ps, which represents the longest stage delay
Single-cycle 1250 ps, which represents the delay of the critical path (LW instruction)

B) Pipelined **1750 ps**, which represent the clock cycle multiplied by the number of stages
Single-cycle 1250 ps, which represents the total delay of any instruction

C) Stage to split is ID [longest delay] –> New clock cycle time 300 ps

D) 35% (LW SW)

E) 65% (ALU+LW)

F) 55% (beq lw sw)

Direct Mapped Cache

1) For a direct-mapped cache design with a 32-bit address, the following bits of the address are used to access the cache.

Tag	Index	Offset
31-10	9-5	4-0

a) What is the cache block size (in words)?
b) How many entries does the cache have?
c) What is the ratio between total bits required for such a cache implementation over the data storage bits?

d) Starting from power on (empty cache), the following byte-addressed cache references are recorded.
0, 4, 16, 132, 232, 160, 1024, 30, 140, 3100, 180, 2180
How many blocks are replaced?

e) What is the hit ratio?

f) List the final state of the cache, with each valid entry represented as a record of <index, tag, data>

a) block offset is determined using 5 bits [0-4]
block size = 2^5 =32 bytes = 8 words

b) cache block index is addressed using 5 bits [5-9]
number of entries = number of blocks = 2^5= 32

c) total cache size to data size = $\frac{total\,Cache\,Size}{Cache\,Data\,Size}$

$$= \frac{32(blocks)*(32[bytes/block]+(22\,[tag\,bits/Blk]+1\,[dirty\,bit]\,+1\,[valid\,bit])/8(bits\rightarrow bytes))}{32(blocks)*(32[data\,bytes/block])}$$

= 1.09375

d) 4

	0	4	16	132	232	160	1024	30	140	3100	180	2180
Memory Blk	0	0	0	4	7	5	32	0	4	96	5	68
Cache Blk	0	0	0	4	7	5	0	0	4	0	5	4
replaced							x	x		x		x
Hit/miss	m	h	h	m	m	m	m	m	h	m	h	m
Index	00000	00000	00000	00100	00111	00101	00000	00000	00100	00000	00101	00100
Tag	0	0	0	0	0	0	0..01	0	0	0..011	0	0..010
MEM	0	0	0	128	**224**	160	1024	0	128	**3072**	**160**	**2176**

Notes:

- Blocks get replaced when the cache has a different block at the same Index. You are trying to add ablock that has the same index but a different TAG.
- the tag is the most significant 22 bits of the address
- MEM is calculated as (Memory Block ID x 32), where 32 is the block size.

e) hit ratio = 4 /12 = 0.33

f)

Index	Tag	Data
0	0...011	mem[3072]
4	0....010	mem[2176]
5	0	mem[160]
7	0	mem[224]

Notes:

- Using Associative cache would reduce the miss ratio because we reduce the competition on the individual block locations in the cache.