Solver: Yew Shao Jie

Email Address: YEWS0012@e.ntu.edu.sg

1)

a)

i) P1's Freq: 500MHz ⇔ Clock Period: 1/500 000 000 = 2* 10^-9 sec = 2ns
P2's Freq: 750MHz ⇔ Clock Period: 1/750 000 000 = 1.33* 10^-9 sec = 1.33ns
Class D has the highest CPI on P1 and on P2, thus minimum 4 CPI on both p1 and p2.
P1 peak performance = 4*2ns = 8ns per instruction
P2 peak performance = 4*1.33= 5.32ns per instruction

ii)

Class	CPI P1	CPI P2
A occur 2x more	1 * 2 = 4	2 * 2 =4
B	3	3.33
C	3	3.33
D	3	3.33

P1 Execution time = (2+3*3)*2ns = 22ns
P2 Execution time = (4+3.33*3)* 1.33ns = 18.6067ns
(1/18.6067)/(1/22) = 1.182 faster

b)

Class	CPI P1	CPI P2
A	1	2
B	2	2
C	3	4
D	4	4

P1's Freq: 5000MHz ⇔ Clock Period: 1/5000 000 000 = 2* 10^-10 sec = 0.2ns
P2's Freq: 7500MHz ⇔ Clock Period: 1/7500 000 000 = 1.33* 10^-10 sec = 0.133ns
P1 execution time = (1+2+3+4)*0.2ns*50% + (1+2+3+4)*2ns*50% = 11ns
P2 execution time = (2+2+4+4)*0.133ns*50%+(2+2+4+4)*1.33ns*50%= 8.778ns
Speed up = 11/8.778=1.253

i)

I1	F	D	E	M	W														
I2		F	D	E	M	W													
I3			Nop	Nop	F	D	E	M	W										
I4						Nop	Nop	F	D	E	M	W							
I5									Nop	Nop	F	D	E	M					
I6												F	D	E	M	W			
I7													Nop	Nop	F	D	E	M	W

I2,I3: $t2
I3,I4: $t0
I4,I5: $t0
I6,I7: $t2

ii)

I1	F	D	E	M	W										
I2		F	D	E	M	W									
I3			Nop	F	D	E	M	W							
I4				Nop	Nop	F	D	E	M	W					
I5							F	D	E	M					
I6								Nop	F	D	E	M	W		
I7										F	D	E	M	W	

4 NOP

iii) Path: I1,I2,I3,I4,I5,I6,I7,I3,I4,I5,I6,I7,I3...... I3,I4,I5,I6,I7
Total 48 loops
14 cycle per loop
Total cycle = 14*48
CPI = 14*48/10*48 = 1.4

2)

a)

i) ADD: 500+200+10+250+500+10+200 = 1670ps
LUI: 500+200+250+500+10+200 = 1660ps

SW: 500+200+250+500 = 1450ps
LW: 500+200+250+500+10+200=1660ps
BEQ: 500+200+10+250+10+100=1070ps

ii) 1/1670 ps= 598.8MHz

b) ADD $t1,$0,$a3
ADD $t2,$0,1
LW $a0, 0($a2)
Loop: ADD $a2,$a2,4
LW $a1, 0($a2)
JAL max-2
ADD $a0,$v0,$0
SUB $t1,$t1,1
BNE$t1,$t2, Loop
LUI $t3,0xF345
ORI $t3,0xA204
SW $v0, 0 ($t3)

3)

a)

i)

T T T N T
N T T T N
2/5*100= 40% accurate

ii)

T T T N T
N T T T T
3/5*100 = 60% accurate

b)

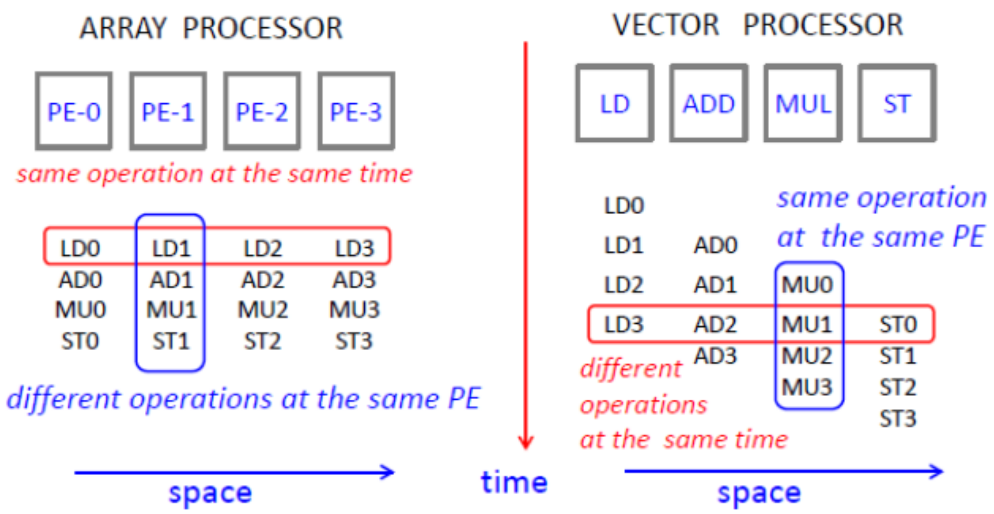
```
Loop:   lw $t0, 0($s1)
        addi $t0, $t0, 25
        sw $t0, 0($s1)
        addi $s1, $s1, 4
        addi $s2, $s2,–1
        bne $s2, $zero, Loop
```

CPI = 12/6=2 without reordering in a scalar processor
2-way super-scalar after instruction reordering CPI = 7/6

	Way-1	Way-2	Cycle
Loop	addi $s1, $s1, 4	lw $t0, 0($s1)	1
	addi $s2, $s2,–1	nop	2
	nop	nop	3
	addi $t0, $t0, 25	nop	4
	nop	nop	5
	nop	nop	6
	bne $s2, $zero, Loop	sw $t0, 0($s1)	7

c) Vector Processors, Pipelined execution of many data operations, Operations on multiple data elements are performed in consecutive time steps (clock cycles) in pipelined form.

SIMD array processors, Operations are performed on multiple data elements at the same time by multiple processing elements.



4)

a)

i)

offset: 2bit

number of set : 128B/8B = 16 [4bit]

76	01 1101 10	Miss
A	00 0010 10	Miss
36	00 1101 10	Miss
20	00 1000 00	Miss
74	01 1101 00	Hit
36	00 1101 10	Hit
4B	01 0010 11	Miss
9	00 0010 01	Hit
49	01 0010 01	Hit

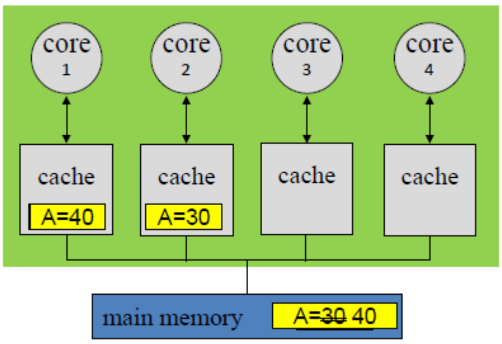
ii)

The hit rate will be the same because there wasn't any Miss due to replacement made in the cache in a 2 ways association, thus a fully association will only have initial miss

b)

Illustration of the coherence problem

- let core-1 reads a variable A from the main memory and caches that.
- let core-2 also reads the same value 30 of the variable A from the main memory and caches that.



- According to the program suppose core-1 modifies (30) of A to 40.
- Write-back policy – Then main memory value of A=30 will remain till the cache line has to be evicted
- Write-through policy – then memory value of A=40.

Core-2 assumes a different and incorrect value of A, while for core-3 and core-4 the value of A depends on cache update policy.

Solutions for cache coherence

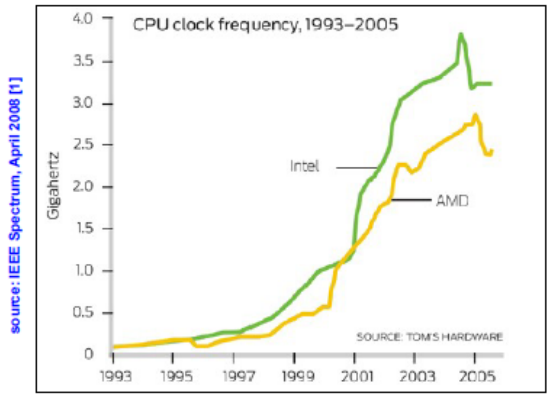
- Software-based approach: shared data are not be cached
  - makes the access of shared data too slow
- Invalidation approach: one core will have ownership of data and all others will invalidate on occurrence of write
  - will lead to lot of cache miss: loss of temporal locality
  - will demand higher memory bandwidth
- Update approach: writes are broadcast to all cores if they have a copy in the cache (i.e., if they share data) and write-through policy should be used to update the main memory
  - bus traffic will increase exponentially with the number of cores which share the data.

⇒ The memory bandwidth increases.

⇒ Hardware complexity of implementation of cache coherence also will increase with the number of cores

c)

The power wall



$P = \alpha \cdot C \cdot V_{dd}^{2} \cdot f$

V_{dd} increases with f

- Power consumption increases exponentially increasing with operating frequency.
- Around the beginning of 2003, the ever-increasing processor speed was checked due to very high power consumption.
- Intel Tejas dissipated 150W heat at 2.80GHz:The Tejas had been projected to run 7 GHz. The project was cancelled in May 2004.

Overcome power wall using multiple slow cores

- Cores running at lower clock frequency and lower voltage can still give the desired performance using less power
- Scale up the number of cores rather than frequency