21st CSEC – Past Year Paper Solution (2018 – 2019 Semester 1)
CE/CZ 3001 – Advance Computer Architecture

1)

a.

	Category - 1	Category - 2	Category - 3	Category - 4
frequency	x	x	2x	x

5x = 100%
x = 20 %
T1 clock rate = 100 MHz
T2 clock rate = 250 MHz
T1 avg. CPI = (2 * 0.2) + (2 * 0.2) + (4 * 0.4) + (4 * 0.2) = 2.8
T2 avg. CPI = (1 * 0.2) + (2 * 0.2) + (3 * 0.4) + (4 * 0.2) = 2.6

Speedup of T2 over T1 = $\frac{Time\ taken\ for\ T1}{Time\ taken\ for\ T2} = \frac{2.8 * \frac{1}{100*10^6}}{2.6 * \frac{1}{250*10^6}} = 2.69$

b. Let execution time of x and y be t and 7t respectively
Execution time of X1 = $(\frac{3}{4}t) + (\frac{1}{4} * \frac{1}{2} * t) = \frac{7}{8}t$
Execution time of X2 = $(\frac{1}{2}t) + (\frac{1}{2} * \frac{1}{4} * 7t) = \frac{35}{8}t$
Speedup of X1 over Y1 = $\frac{\frac{35}{8}}{\frac{7}{8}} = 5$

c.

i. Raw dependencies:
   1. I2 & I3
   2. I3 & I4
   3. I4 & I6
   4. I7 & I8

   Number of iterations = 32

ii.

Cycle	1	2	3	4	5	6	7	8	9	11	12	13	14
I0	IF	ID	EX	M	WB								
I1		IF	ID	EX	M	WB							
I2			IF	ID	EX	M	WB						
I3				stall	IF	ID	EX	M	WB				
I4						IF	ID	EX	M	WB			
I5							IF	ID	EX	M	WB		
I6								IF	ID	EX	M	WB	
I7									IF	ID	EX	M	WB
I8										stall	IF	ID	EX

2 stall per cycle
Total number of stall = 2 * 32 = 64
Steady state CPI = $\frac{9+2}{9} = 1.22$

2)

a.

i. 0xA0000000. The content of the register is a signed number -1610612736 in decimal because the instruction addi is used for addition of signed number.
ii. 0xA0000000. The content of the register is an unsigned number 2684354560 in decimal because the instruction addiu is used for addition of unsigned number.

b.

i. R-type ALU = I-MEM + REG (R) + MUX + ALU + MUX + REG(W)
= 1000 + 50 + 25 + 1000 + 25 + 200

= 2300 ps

Load word = I-MEM + REG(R) + MUX + ALU + D-MEM + MUX + REG(W)
= 1000 + 50 + 25 + 1000 + 2000 + 25 + 200
= 4300 ps

Conditional branch = I-MEM + REG(R) + MUX + ALU + AND gate + MUX + PCin->PCout
= 1000 + 50 + 25 + 1000 + 10 + 25 + 1000
= 2210 ps

ii. Max period = 2300 ps
Clock frequency = $\frac{1}{2300 * 10^{-12}} = 434.78$ MHz

c.

Sequence	T	T	T	N	T	N
Predict	T	T	T	T	T	T
Next state	11	11	11	10	11	10

Accuracy = $\frac{4}{6} * 100\%$ = 66.67%

3)

a.

	Way - 1	Way – 2
Loop	LW $t0, 0($s0)	ADDI $t2, $t2, -2
	LW $t1, 4($s0)	ADDI $s0, $s0, 8
		XORI $t0, $t0, 0x0F0F
		XORI $t1, $t1, 0x0F0F
	SW $t0, -8($s0)	BNE $t2, $zero, Loop
	SW $t1, -4($s0)	

CPI = 8/9 = 0.889

b. For single core, write-through is expensive as it requires high bandwidth and memory becomes slow when its size increases. Cache coherence is not an issue because there is only one core accessing the memory.
However, for multi core, write-through is necessary for cache coherence to ensure that memory in lower-level hierarchy is also updated.

c. AMAT = Time for a hit + Miss rate * Miss penalty
$= \frac{1}{2*10^9} + (\frac{8}{100} * 25 * 10^{-9}) + (\frac{8}{100} * \frac{4}{100} * 100 * 10^{-9})$
= 2.82 ns

4)

a. There are three levels of parallelism.
   i. Instruction-level parallelism: Multiple independent instructions are identified and grouped to be executed concurrently indifferent functional units in a single processor. This may reduce CPI to less than one.

      Example: Superscalar and VLIW processors

   ii. Data-level parallelism: The same operation is performed on multiple data values concurrently in multiple processing units. This may reduce the instruction count.
   Example: Vector processors and array processors

iii. Thread/Task-level parallelism: More than one independent threads/tasks are executed simultaneously. This may reduce the total execution time of multiple tasks. Example: Multi-core and multi-processor systems

b. *Editor's note: was not covered for my batch*

c. In ascending order:

Performance: GPP, FPGA, ASIP, ASIC
Flexibility: ASIC, FPGA, ASIP, GPP
Time-to-market: GPP, FPGA, ASIP, ASIC

d. Memory wall describes implications of the processor/memory performance gap that has grown steadily over the last several decades. If memory latency and bandwidth become insufficient to provide processors with enough instructions and data to continue computation, processors will effectively always be stalled waiting on memory.
Multicore architecture allows memory parallelism, a derivation of the thread-level parallelism. Adding more cores provides a relatively narrower channel to shared memory resources.

Solver: CSEC Academic Committee