

Instructions: Time **70** minutes. Open book and notes exam. No electronics. Please answer all problems in the space provided and limit your answer to the space provided. There are six problems and each problem is for 5 marks. **No questions are allowed.**

<Good Luck>

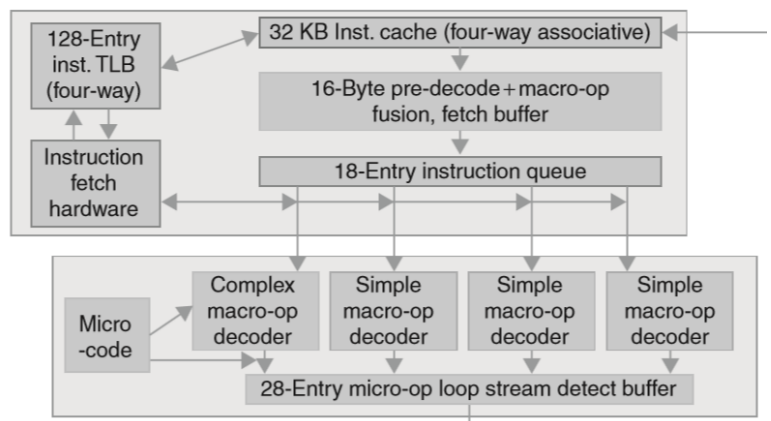
P1. A processor has three instruction classes that has the CPIs shown in the following table. For a 10-billion-instruction program that has the instruction frequencies shown in the table, what is the processor's clock rate if executing this program takes 5 seconds?

Class	CPI	Frequency
A	1	50%
B	2	30%
C	3	20%

Solution:

$$\begin{aligned}
 f &= IC \times (\sum_i CPI_i \times Freq_i) / \text{Time} \\
 &= 10 \times 10^9 \times (1 \times 0.5 + 2 \times 0.3 + 3 \times 0.2) / 5 \\
 &= 2 \times (0.5 + 0.6 + 0.6) \times 10^9 \\
 &= 2 \times 1.7 \times 10^9 \\
 &= 3.4 \text{ GHz}
 \end{aligned}$$

P2. Consider the following figure that shows parts of the Intel Core i7 pipeline. What is the main benefit of the shown Complex macro-op decoder?



Solution: It converts the complex macro-ops to a sequence of simple micro-ops that can be more easily pipelined.

P3. Schedule the following code sequence in the table below for a static multiple issue processor similar to the one described in the class but is three-issue wide instead of two. Assume that the issue packet can have one ALU instruction, one branch instruction, and one memory instruction as shown in the table below. Note that you need to perform register renaming in order to have an efficient schedule.

```

Loop: lw    $t0, 0($s1)
      addu  $t0, $t0, $s2
      sw    $t0, 0($s1)

      lw    $t0, -4($s1)
      addu  $t0, $t0, $s2
      sw    $t0, -4($s1)

      lw    $t0, -8($s1)
      addu  $t0, $t0, $s2
      sw    $t0, -8($s1)

      addi  $s1, $s1, -12
      bne  $s1, $zero, Loop
  
```

Packet	ALU instruction	Branch instruction	Memory instruction
1			lw t0, 0(s1)
2	addi s1, s1, -12		lw t1, -4(s1)
3	addu t0, t0, s2		lw t2, 4(s1)
4	addu t1, t1, s2		sw t0, 12(s1)
5	addu t2, t2, s2		sw t1, 8(s1)
6		bne s1, zero, Loop	sw t2, 4(s1)
7			
8			

P4. Assume that the following code sequence is executed by a triple-issue speculative pipelined processor. This processor uses reservation stations, common data buses, and reorder buffer. The fetch stage takes one cycle and the issue stage takes one cycle. The integer latency is 1 cycle and the load latency is 2 cycles (1 cycle for address calculation and 1 cycle for data memory access). The processor has one address calculation unit, one memory access unit, one integer ALU unit, and one branch unit. Using the multi-cycle pipeline diagram below, specify the execution of these instructions in this processor pipeline.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
lw r2, 0(r1)	F	I	A	M	W	C									
lw r3, 4(r1)	F	I		A	M	W	C								
lw r4, 8(r1)	F	I			A	M	W	C							
add r5, r2, r3		F	I				E	W	C						
sub r6, r5, r4		F	I						E	W	C				
sw r6, 12(r1)		F	I			A					C				

P5. Assume that the following MIPS instructions are executed on a processor that has a branch prediction unit that uses a 2-bit branch history table. Also assume that the BHT is initialized to zeros and the initial prediction is not taken.

```

    addi    R1, R0, #0
    addi    R2, R0, #16
Loop: lw   R3, 0(R1)
    add    R4, R4, R3
    addi   R1, R1, #4
    bne   R1, R2, Loop
  
```

Note that the branch instruction in these instructions is executed four times. Complete the following table to specify for every round of execution of this branch instruction its prediction, actual direction, and whether the prediction is correct or not.

Round	BHT Contents	Branch Prediction	Actual Branch Direction	Correct Prediction (Yes or No)?
1	00	NT	T	No
2	01	NT	T	No
3	10	T	T	Yes
4	11	T	NT	No

P6. Draw a direct mapped cache with the following specifications: size = 32 KB, block size = 32 bytes, word size = 4 bytes, address width = 32 bits, and write back scheme.

Number of blocks = $32 \text{ KB} / 32 \text{ bytes} = 1 \text{ K blocks}$

<index> = $\lg_2(\text{No. of sets}) = \lg_2(1 \text{ K}) = 10 \text{ bits}$

<tag> = $32 - \text{<index>} - \text{<block offset>} = 32 - 10 - \lg_2(32) = 32 - 10 - 5 = 17 \text{ bits}$

