CSE/ECE 511: Practice Questions

Q1. A program P is compiled for an ISA with 3 types of instructions, say A, B and C. Moreover, P's binary has 1,000,000 instructions. Each of these types of instructions has different execution latencies and relative frequencies associated with them. The latencies and the relative frequencies for these instructions in P are as follows:

Instruction Type	Relative Frequency	Execution Latency in cycles
Type A	20%	10 cycles
Type B	10%	8 cycles
Type C	70%	5 cycles

Using this information, compute
a.) The average cycles per instruction (CPI) for P running on this CPU.
b.) The minimum clock frequency (in cycles per second, i.e. Hz) at which the CPU must run to execute P in less than 5 seconds.

Ans.
a.) Total instructions in P = 1,000,000
Expected cycles per instruction in P = (0.2 * 10) + (0.1 * 8) + (0.7 * 5) = 2 + 0.8 + 3.5 = 6.3
b.) Assuming the frequency of the CPU to be f, the execution time = 1,000,000 * 6.3 * (1/f)
For execution time = 5 seconds,
5 = 6,300,000/f
=> f = 6,300,000/5
=> f = 1,260,000Hz = 1.26MHz

Q2. Various ISAs can be classified into two broad categories: fixed length and variable length. In fixed-length ISAs, each instruction occupies the same amount of memory. On the other hand, in variable-length ISAs, each instruction may occupy different amounts of memory. Suppose that you are given two CPUs, say CPU-A and CPU-B which support a fixed length ISA (ISA-A) and a variable length ISA (ISA-B) respectively. Now let's say that there is a program (P) which is compiled in both ISAs such that it can run on both CPUs.

- When P is compiled for ISA-A, the compiler generates 1024 instructions, each of length 8 bytes.
- When P is compiled for ISA-B, the compiler generates 1024 instructions, out of which 12.5% are of type-1, 37.5% are of type-2, and the remaining are of type-3. Type-1 instructions occupy 16 bytes in the memory, type-2 instructions occupy 8 bytes in the memory, and type-3 instructions occupy 4 bytes in the memory.

a.) Calculate the binary size for P when it is compiled for ISA-A and ISA-B.
b.) Based on your calculations done in part a.) of this question, predict which of the two CPUs will have a higher cache hit rate, assuming that both the CPUs have the same cache architecture. Justify your answer.
c.) Which of the two CPUs will have a simpler pipeline? Justify your answer.
d.) Based on your answers for the previous parts, which of the two CPUs would be faster? Justify your answer.

Ans.

a.) Total instruction footprint for ISA-A = Number of instructions * Size of instruction
= 1024 * 8
= 8192 bytes
= 8 kilobytes

Total instruction footprint for ISA-B = Number of instructions * Expected size of instructions
= 1024 * [(16 * 0.125) + (8 * 0.375) + (4 * 0.5)]
= 1024 * (2 + 3 + 2)
= 1024 * 7
= 7168 bytes
= 7 kilobytes

b.) Assuming that everything else is the same except the ISA, CPU-B, which runs ISA-B, will have a higher hit rate because the program size for it is smaller, which would result in a higher percentage of instructions sitting in the lower level and faster caches.

c.) CPU-A, which runs the fixed length ISA-A would have a simpler pipeline as the decode logic for it would be much simpler, as compared to CPU-B which runs the variable length ISA-B.

d.) It is impossible to determine which CPU would be faster based on the provided information alone. Although CPU-A would have a faster decoder, the instruction cache misses that it would incur can limit its performance. Similarly, although CPU-B will have a better hit rate, its decode latency can become the bottleneck.

Q3. Consider processor A with a clock frequency of 1 MHz. The processor executes one instruction per cycle.

a. Determine the clock period of processor A and the time needed to implement 197 instructions.
b. Processor A is now upgraded to a four-stage pipelined processor, say fetch [F], Decode [D], Execute [E], and Writeback [W], which have delays of 180 ns, 190 ns, 140 ns and 170 ns, respectively. The registers between the pipeline stages have a delay of 10 ns (**Note:** This time includes sequential overheads). Determine the new clock period of the upgraded processor.
c. Determine the time needed to implement 197 instructions after the upgradation of the processor. Assume no hazards are present.

Determine the speedup obtained by upgrading the processor [Hint: Take the ratio of time of 197 instructions obtained in parts a and c].

Ans.

a. Clock period = 1/1 MHz = 10^{-6} s = 1000 ns/ 1 us
As the processor executes one instruction in one clock cycle, time needed to implement 197 instructions = 197 us
b. Cycle time = max(stage delay) + register delay
= max (180, 190, 140, 170) + 10 = 190 + 10 = 200ns
c. Pipeline time to process 197 data items
= Time taken for 1st data item + Time taken for remaining 196 data items
= 1 x 4 clock cycles + 196 x 1 clock cycle
= 4 x cycle time + 196 x cycle time
= 200 x cycle time = 200 x 200 ns = 40 x 10^{-3} ns = 40 us

d. Both the answers are acceptable:
   Ans 1:
   As speed up is inversely proportional to time, speed up obtained = 197 us/ 40 us = 4.925 (Acceptable range = 4.8- 5.1)
   OR
   Ans 2:
   Ratio of time of processor B and A is 40/197 = 0.203 (Acceptable Range = 0.19 - 0.21)

Q4. Consider a four-stage pipeline with stages, Fetch [F], Decode [D], Execute [E] and Writeback [W]. The stages are described below:

1. Fetch - Fetches the instructions from instruction memory
2. Decode - Decodes the instructions and type of operations. It also reads the input operands. If the input operands are not present, the decode stage is stalled for one cycle to read the operands after they are made available.
   - The address of the unconditional branch is assigned to the PC at the end of the decode stage.
3. Execute - Executes the instruction in the ALU
4. Writeback - Updates the register value at the end of the stage.

Determine the number of cycles to execute the following program:

1. SUB R1, R2, R3
2. ADD R5, R2, R3
3. ADD R4, R5, R6
4. B 7
5. ADD R4, R2, R3
6. ADD R5, R4, R2
7. B 10
8. ADD R8, R9, R10
9. SUB R11, R12, R13
10. ADD R12, R13, R14

Ans. No. of cycles = 13

		1	2	3	4	5	6	7	8	9	10	11	12	13
1	SUB R1, R2, R3	F	D	E	W									
2	ADD R5, R2, R3		F	D	E	W								
3	ADD R4, R5, R6			F	D	D	D	E	W					
4	B 7				F	F	F	D	E	W				
5	ADD R4, R2, R3							F	-	-	-			
6	ADD R5, R4, R2								-	-	-	-		
7	B 10								F	D	E	W		
8	ADD R8, R9, R10									F	-	-	-	
9	SUB R11, R12, R13										-	-	-	-
10	ADD R12, R13, R14										F	D	E	W

Q5. Consider a 32-bit machine with a set-associative cache. The total cache size is 16KB, and it is a 4-way set-associative cache. The length of the memory location is 1 byte, and the block size is 16 Bytes.

a) Calculate the total number of sets present in the cache.

b) Draw the 32-bit memory address showing the tag, index, and offset bits.

c) From what memory locations are the data fetched if the 20ABCC88 memory location is accessed? Determine the set number where it is stored.

Solution

a) Number of sets = 16KB / 4*2^4 = 2^8 sets

b)

Tag (20 bits)	Index (8 bits)	Offset (4 bits)
32 12	11 4	3 0

c) Each cache line stores 4 words. When the address is fetched, it fetches 4 words of data from memory.

One memory location is 8 bits, and the word size of the processor is 32 bits. This implies that 4 memory locations are accessed for every processor word. If the first access was at location 0, for instance, then the next access would be at location 4, then 8 and so on. This implies that the first two bits are irrelevant and are always left as an offset.

The last byte of the address is 8 (1000), which means that the byte is stored at position 10 (binary)/ 3 (decimal). Thus, to fill the cache lines, memory locations 20ABCC80, 20ABCC84, 20ABCC88 and 20ABCCC are fetched.

The bits [11:4] represent the index bits and hence, the set number. The data is stored at set number C8 (hex)/ 200 (decimal).

Q6

Suppose you have a 32-bit processor with byte-addressable memory. This processor has a 512-byte fully-associative cache with 16-byte cache lines. The cache uses LRU (least recently used) replacement policy.

a) What is the total number of cache lines?
b) How many sets does this cache have?
c) How many cache-line in each set?
d) How many bits are required for the tag, index and cache-line offset?
e) Mark which address segments denote tag, index and cache-line offset.

Solution:

a) Total number of cache lines = Cache_size/size_of_cache_line = 512/16 = 32
b) Since this is a fully associative cache, it only has a single set.
c) All the cache lines belong to the one and only set, Therefore 32 lines in the set.
d) Cache_offset_bits = d

2^d= (size_of_cache_line/size_of_single_memory_address)

2^d= 16/1

$2^d = 2^4$

d = 4 bits

Index_bits = i
2^i= number of sets in cache
Number of sets = (cache_size/(cache_line_in_each_set * size_of_cache_line))
(number of sets in fully associative cache = 1)
2^d= 1
$2^d = 2^0$
d = 0 bits

Tag = address_size - Index_bits - Cache_offset_bits
Tag = 32- 0-4
Tag = 28 bits

e) Address will be broken down in the following manner:

Tag (28 bits)	Offset (4 bits)

32 4 3
0

Q7
Classify the following attributes of a machine as either a property of its microarchitecture or ISA. Clearly state your reason for each property.

Attribute	ISA	Microarchitecture
The latency of each pipeline stage in the machine.		
Addressing modes available for arithmetic operations		
A 5-bit immediate can be specified in an ADD instruction		
The machine has a multi-level cache hierarchy		
The width of general-purpose registers		
ALU bypasses its result to a 2-to-1 mux which feeds its output to one of the inputs of ALU		

Solution:

Attribute	ISA	Microarchitecture
The latency of each pipeline stage in the machine.		✓
Addressing modes available for arithmetic operations	✓	
A 5-bit immediate can be specified in an ADD instruction	✓	
The machine has a multi-level cache hierarchy		✓

The width of general-purpose registers	✔	
ALU bypasses its result to a 2-to-1 mux which feeds its output to one of the inputs of ALU		✔

Q8.

Consider a simple in-order superscalar pipeline with 2 pipes. The stages are Fetch, Decode, Execute, Memory and Writeback. The pipeline can fetch, decode and execute 2 instructions in a single cycle. All memory operations take 1 cycle to complete. Two instructions can write to the register file. Assume that instructions stall in the decode stage unless their operands are available.

ADDIU R1, R1, 1
ADDIU R3, R4, 1
ADDIU R5, R6, 1
ADDIU R7, R5, 1

Draw pipeline diagrams for the given sequence of instructions for the following cases.

a) In-order processor with no bypassing.

b) How does the pipeline sequence change with bypassing enabled?

Solution:

a. In-order processor with no bypassing

Instr	**C0**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**
1	F	D	X	M	W				
2	F	D	X	M	W				
3		F	D	X	M	W			
4		F	D	D	D	D	X	M	W

b. In-order processor, with bypassing

Instr	**C0**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**
1	F	D	X	M	W				
2	F	D	X	M	W				
3		F	D	X	M	W			
4		F	D	D	X	M	W		

Q9.

Assume I2OI architecture with in-order fetch, decode and issue; out-of-order execute and writeback; in-order commit. The pipeline has 4 functional units: ALU (1 cycle X0), Loads (2 cycles, L0 and L1), Store (1 cycle, S0) and multiply (4 cycles, Y0, Y1, Y2, Y3).

0: MUL R6, R7, R8
1: ADD R9, R6, R11
2: MUL R7, R1, R2
3: LW R10, R12

a) Draw the state of the scoreboard when instruction 3 is in the Issue stage of the pipeline.
b) Show the pipeline diagram.

Solution:

I2OI															
Instr	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**	**C13**	**C14**	**C15**
0	F	D	I	Y0	Y1	Y2	Y3	W	C						
1		F	D	I	I	I	I	X0	W	C					
2			F	D	D	D	D	I	Y0	Y1	Y2	Y3	W	C	
3				F	F	F	F	D	I	L0	L1	W	r		C

Contents of the scoreboard:

- R9 is pending in functional unit X and is marked as having ‘0’ more cycles until writeback
- R7 is pending in functional unit Y and is marked having ‘4’ more cycles until writeback
- The rest of the registers are not pending