



Australian
National
University

School of Computing

Venue

Student Number

|_|_|_|_|_|_|_|_|_|

EXAMINATION

Semester 1 -End of Semester, 2025

COMP2300_COMP6300_ENGN2219 Semester 1

Computer Architecture

This paper is for ANU students.

Examination Duration: 180 minutes

Reading Time: 15 minutes

Exam Conditions:

No materials permitted

Materials Permitted In The Exam Venue:

(No electronic aids are permitted e.g. laptops, phones)

No materials permitted

Materials To Be Supplied To Students:

1 x 20 pages

Instructions To Students:

The exam is marked out of 100. There are seven sections. Each section has one or multiple questions. The breakdown of marks for each section is indicated in the table below. You should spend more time on questions with more weight.

Section 1	Multiple Choice	20					
Section 2	Digital Logic	7	3				
Section 3	Timing Issues	4	6				
Section 4	Finite State Machine	14					
Section 5	ARM Assembly	10					
Section 6	Input/Output	3	3				
Section 7	Microarchitecture	4	16	10			
Aggregate Marks		100					

- Write your student number and answers **clearly in good handwriting**.
- An ARM cheat sheet is provided at the end of the exam sheet.
- Any formula sheet is neither deemed necessary nor provided for this closed-book exam.
- All the arithmetic required to answer questions is simple as long as you are comfortable with the binary number system. The use of a calculator is not deemed necessary for this exam. Using a calculator or a mobile aid is strictly forbidden.
- This exam has seven sections and multiple questions per section. The first question consists of 20 multiple-choice and true/false questions on a variety of topics. The subject of each remaining question is indicated.
- Do not attach extra sheets. Write your answers in the space provided in the question sheet.
- We provide an extra sheet after the final exam question. If you use that extra sheet to provide answers to any questions, clearly mention page number 23 so we can mark your answers.
- In the upcoming question, **there is only one correct answer**. Write the correct selection in the grid provided. We will only mark the selected option in the grid for each question.
- Clearly read the marking scheme of the first question. There is negative marking (-1/2 for each wrong selection) in the first question only. Avoid random guesses as it might negatively affect your score.

Section 1: Multiple choice and true/false questions [1 for correct, -1/2 for wrong, 0 for no selection] Write your choice of answer in the table below.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Q1. The number of bits needed to represent -32 in 2's complement number system:

- A. 7
- B. 8
- C. 9
- D. 10
- E. None of the above

Q2. A logic gate with two inputs, A and B, outputs B when A is Logic 1, and the complement of B, when A is Logic 0. Which logic gate is it?

- A. OR
- B. AND
- C. XOR
- D. XNOR
- E. NAND

Q3. The expression $A + B$ is equivalent to:

- A. $A' \cdot B'$
- B. $(A \cdot B)'$
- C. $(A' \cdot B')'$
- D. $((A \cdot B)')'$
- E. None of the above

Q4. The expression $ABC + AB'C + ABC'$ is equivalent to:

- A. $ABC + BC$
- B. $A(B + C)$
- C. ABC
- D. $AB + BC$
- E. $ABC + B'C$

Q5. Which of the following correctly describes the behavior of SR latches built using cross-coupled NOR and NAND gates, when both inputs are set to Logic 1?

- A. In both NOR and NAND latches, the outputs retain their previous state
- B. In a NOR latch, both outputs go to Logic 0; in a NAND latch, both outputs go to Logic 1
- C. A NOR latch enters an invalid state; a NAND latch holds the previous state
- D. Both types of latches enter a valid and predictable state
- E. In a NOR latch, the outputs are complementary; in a NAND one, both outputs are Logic 0

Q6. A memory array that uses a 4:16 decoder to store 2-byte words has an address space (in bits) of:

- A. 16
- B. 32
- C. 64
- D. 128
- E. 256

Q7. The width of pipeline registers in a pipelined CPU is determined by:

- A. ISA
- B. Microarchitecture
- C. None of the above

Q8. The decision to increment the PC by a specific offset to fetch the next instruction is determined by:

- A. ISA
- B. Microarchitecture
- C. None of the above

Q9. In general, for the 32-bit ARM architecture, which of the following registers is not changed by a leaf function?

- A. PC
- B. SP
- C. LR
- D. R12
- E. None of the above

Q10. What is the clock frequency (in gigahertz) of a CPU if it executes 4 billion instructions with an average IPC of 0.5 in 2 seconds?

- A. 0.25
- B. 0.5
- C. 1
- D. 2
- E. 4

Q11. The execution time of a program with no memory accesses and I/O is 100 seconds on a CPU with a power rating of 100 watts. Which of the following options is the best for reducing the energy consumed by the program:

- A. Reduce supply voltage by 2
- B. Reduce clock frequency by 2
- C. Reduce transistor count by 2
- D. All three options above have the same effect
- E. None of the above

Q12. A program consists of 2 billion instructions of which 50% are ALU, 25% are memory, and 25% are branch instructions. On a 1 gigahertz multi-cycle CPU, an ALU instruction takes 6 clock cycles, each memory instruction takes 8 clock cycles, and each branch instruction takes 4 clock cycles. What is the execution time of the program in seconds?

- A. 4 seconds
- B. 6 seconds
- C. 8 seconds
- D. 12 seconds
- E. 14 seconds

Q13. An assembly programmer's skill in writing compact and efficient code does not impact:

- A. Instructions per second
- B. Cycles per instruction
- C. Stack size
- D. Peak power rating
- E. Energy-delay product

Q14. How many cycles does it take for an 11-stage pipeline to execute 100 independent instructions?

- A. 89
- B. 100
- C. 109
- D. 110
- E. 111

Q15. Which of the following is not impacted by data cache in modern CPUs?

- A. Performance
- B. Power
- C. Energy
- D. Correctness
- E. Throughput

Q16. Which of the following ARM assembly instruction sequences correctly implements the two C statements in program sequence, `A[i] = 0; i++;` assuming R1 holds the base address of array A, R0 holds the index i, and each element of A is one byte?

- A. `LDRB R2, [R1, R0]; ADD R2, R2, #1; STRB R2, [R1, R0]`
- B. `MOV R2, #0; STRB R2, [R1, R0]; ADD R0, R0, #1`
- C. `ADD R0, R0, #1; MOV R2, #0; STRB R2, [R1, R0]`
- D. `LDR R2, =0; STRB R2, [R1, R0]; ADD R2, R2, #1`
- E. None of the above

Q17. Register R2 is initially 0x80000003. After executing the instruction: `ROR R2, R2, #1`, what is the new value of R2?

- A. 0xC0000000
- B. 0x40000000
- C. 0xC0000000
- D. 0xC0000001
- E. 0x00000003

Q18. For an initial state of 01, and a misprediction due to a Taken branch outcome, the next state of a 2-bit branch prediction counter is:

- A. 00 (strongly not-taken)
- B. 01 (weakly not-taken)
- C. 10 (weakly taken)
- D. 11 (strongly taken)
- E. It depends on the branch target address

Q19. Which of the following CPUs does not exploit spatial parallelism:

- A. A single-core, 2-way superscalar, in-order, pipelined CPU
- B. An 8-core, 1-way superscalar, in-order, pipelined CPU
- C. A single-core multi-cycle CPU
- D. A 4-way out-of-order CPU with hardware speculation
- E. A 2-core single-cycle CPU

Q20. Each of the following instruction sequence pairs expose a hazard on an out-of-order CPU. Which sequence is no longer a hazard with register renaming?

- A. `ADD R0, R1, R2` *followed by* `ADD R1, R4, R5`
- B. `LDR R0, [R1]` *followed by* `STR R0, [R1]`
- C. `ADD R0, R1, R2` *followed by* `SUB R3, R4, #16`
- D. `ADD R0, R1, R2` *followed by* `ADD R6, R1, R7`
- E. `ADD R8, R6, R7` *followed by* `ADD R9, R8, R8`

Section 2: Digital Logic

Q1. An interrupt controller receives interrupt requests (IRQ0, IRQ1, and IRQ2) from three I/O devices. Each device has a different priority, with IRQ0 being the highest and IRQ2 the lowest. The controller has three outputs (OUT0, OUT1, and OUT2), where only one output is asserted (set to Logic 1) at any time, based on the highest active interrupt.

For example:

If IRQ0 = 1, IRQ1 = 1, and IRQ2 = 1, then OUT0 = 1 and OUT1 = OUT2 = 0, since IRQ0 has the highest priority. If only IRQ2 = 1, then OUT2 = 1. If all IRQs are Logic 0, then all outputs are 0.

Answer the following three parts. You may use the white space on the next page if needed.

- A. Draw the truth table for the combinational logic of the interrupt controller.
 - Create one version showing all possible input combinations (no simplifications).
 - Create a second version using don't-care conditions where lower-priority interrupts can be ignored when a higher-priority interrupt is active.

- B. Derive the optimized Boolean expressions for each output (OUT0, OUT1, OUT2) using the simplest sum-of-products (SOP) form. Show how each expression is obtained from the truth table.

- C. Using only basic logic gates (AND, OR, NOT), draw a logic diagram (schematic) of the interrupt controller. Clearly label all inputs and outputs.

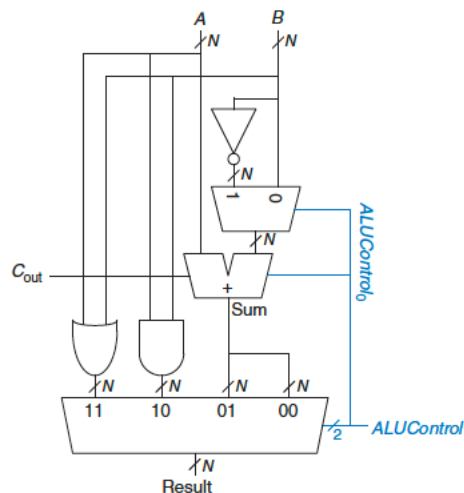
Q2. Implement the function shown in the following truth table using a 2:1 multiplexer and two other 2-input logic gates. A, B, and C are inputs, and Y is the output.

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Draw the schematic below.

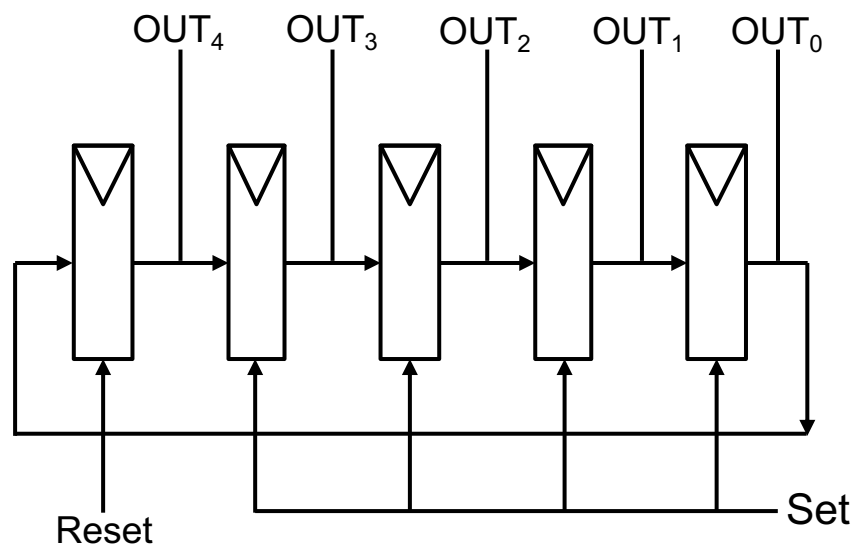
Section 3: Timing in Digital Circuits

Q1. Consider the schematic of an N-bit ALU for performing two's complement arithmetic below. The ALU contains an N-bit adder and N two-bit AND and OR gates. A 4:1 multiplexer chooses the desired function based on ALUControl. More specifically, if ALUControl is 00, the output multiplexer chooses $A+B$, $A - B$ if 01, $A \text{ AND } B$ if 10, and $A \text{ OR } B$ if 11. The inverter and multiplexer invert input B when ALUControl_0 is asserted (i.e., subtract). Recall that in two's complement arithmetic, $B' + 1 = -B$. Note that picoseconds = ps.



- A. If the N-bit inverter, the N-bit 2:1 mux, and N-bit adder each has a delay of 1.1 ps, and the N-bit AND/OR gates each have a delay of 3.4 ps, then the critical path consists of:
- N-bit inverter, 2:1 N-bit mux, N-bit adder, 4:1 N-bit mux
 - N-bit AND, 4:1 N-bit mux
 - 4:1 N-bit mux
 - 2:1 N-bit mux, N-bit adder, 4:1 N-bit mux
 - None of the above is correct
- B. If N is 32, and the N-bit adder is built using 32 1-bit full adders, where the delay of each one of the 1-bit full adders is 0.25 ps, and all other N-bit components each have a delay of 0.2 ps, then the minimum time to compute $A - B$ is:
- 16.4 ps
 - 8.2 ps
 - 16.2 ps
 - 8.4 ps
 - 8.6 ps
- C. If $N = 4$, $\text{ALUControl} = 01$, $A = 1111$, and $B = 0001$, then $A - B$ is:
- 1
 - 2
 - 1
 - 2
 - 0

Q2. The following synchronous sequential circuit consists of two types of (rising) edge triggered flip-flops. The left-most flip flop is synchronously resettable (when Reset is Logic 1, the state of the flip-flop is Logic 0 at the next rising edge of the clock; otherwise, input flows to the output at the rising edge). The remaining flip-flops are synchronously settable (when Set is Logic 1, the state of the flip-flop is Logic 1 at the next rising edge of the clock; otherwise, input flows to the output at the rising edge). All flip-flops receive the same clock (CLK) input (not shown). The circuit has a 5-bit output $OUT_{4:0}$.



Timing specification

- The clock-to-Q propagation delay (t_{pcq}) is equal to 0.3 nanoseconds.
- The setup time (t_{setup}) is equal to 0.2 nanoseconds.
- The hold time (t_{hold}) is zero.
- Assume wires have no delay.

Answer the questions on the next page.

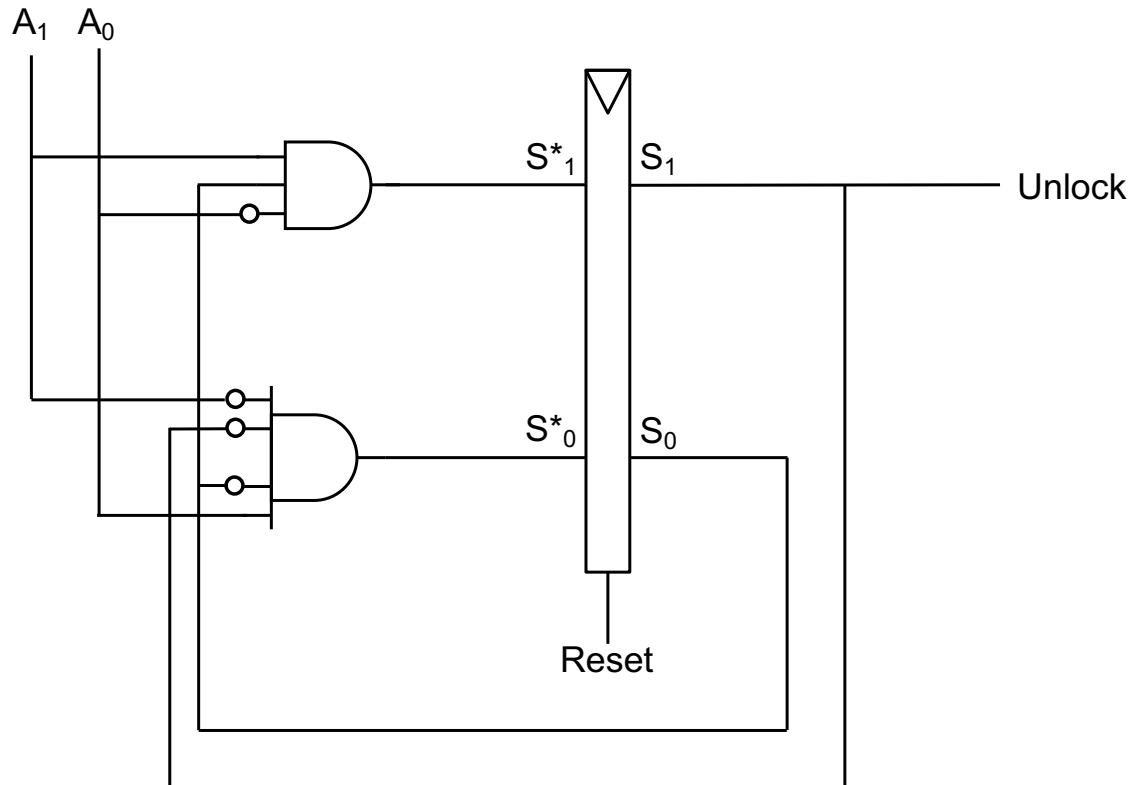
A. Briefly explain what does the above circuit do?

B. What is the maximum clock frequency at which a circuit designer can operate the circuit?

C. Assume set/reset are asserted just prior to the rising edge of the clock, respecting the setup constraint. What is the delay (in seconds) for $OUT_{4:0}$ to transition from 01111 to 11110 (beginning at the start of the cycle, i.e., just after the rising edge)?

Finite State Machines

Q1. The circuit below implements a Moore finite state machine (FSM). The 2-bit current state is given by bits S_0 and S_1 . The next state is given by bits S^*_0 and S^*_1 . The output of the circuit is Unlock, and the external inputs are A_0 and A_1 .



- Write the next state and output equations.
- Write down the encoded output and state transition tables. Use binary encoding for representing states. Use reduction using don't cares whenever possible.
- Write down the symbolic next state and output tables. Use any naming scheme in English for representing states.
- Draw the state machine diagram of the above circuit.
- In simple words, briefly explain the purpose of the above state machine.
- Suppose the FSM designer implements the output, Unlock, as the AND of the two state bits, S_0 and S_1 . Explain the behavior of Unlock in this case.

Write your answers on the next two pages.

ARM Assembly and ISA

Q1. Translate the following C functions into 32-bit ARM assembly code. The parameters of all functions are provided in `R0` and `R1`. The result should be returned in `R0`. Assume the base address of `array` is `0x0000FFFF`. Implement all three functions, and do not use inlining.

Your code should preserve registers using the ARM calling convention, if necessary.

```
// C code
void setArray(int num) {
    int i;
    int array[10];

    for (i = 0; i < 10; i++)
        array[i] = compare(num, i);
}

int compare(int a, int b) {
    if (sub(a,b) >= 0)
        return 1;
    else
        return 0;
}

int sub(int a, int b) {
    return a - b;
}
```

Write your code in the space provided on this page and the next page (if needed).

Input/Output

Q1. A 128 KB Flash memory-based storage device has a built-in cache for storing the most recently used data. The cache can be enabled or disabled by writing to a memory-mapped control register on the device. The device access delays are as follows:

- A hit costs one clock cycle.
- A miss costs 4 clock cycles.
- A device access if the cache is disabled costs 3 clock cycles.

Is it beneficial to enable the cache for the hit rates below? Show your analytical reasoning and write Yes or No below. A simple Yes or No without reasoning provided will lead to 0 marks.

A. 25%: _____

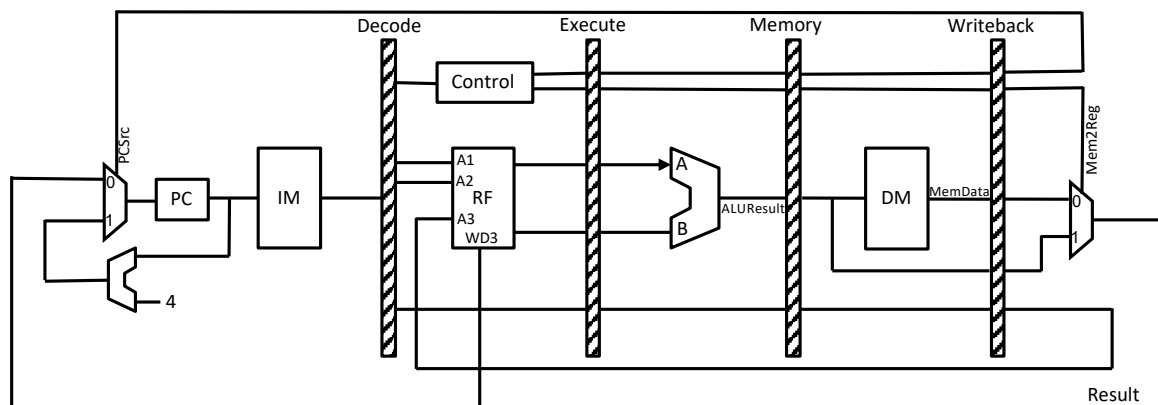
B. 50%: _____

Q2. An embedded system uses memory-mapped I/O to interact with peripheral devices. A sensor value can be read (either 0 or 1) from the memory-mapped address 0x40000000. An LED can be turned on by setting bit 8 of a control register at address 0x30000000.

Write 32-bit ARM assembly code to continuously poll the sensor until it reads Logic 1. Then, set bit 8 of the LED control register to turn the LED on. Exit the loop after setting the LED.

Section 7: Microarchitecture

Q1. Consider the *scalar* in-order pipelined CPU shown below. The five pipeline stages are Fetch, Decode, Execute, Memory, and Writeback. Each stage performs its work in one clock cycle. The pipeline registers are labeled with the stage they are in. PC is in fetch stage. The CPU lacks forwarding and hazard detection capabilities, including no ability to deal with incorrect branch prediction. ALU operations and memory accesses take one clock cycle. The address of a memory instruction is computed in the Execute stage. The register file is written during the first half of the clock cycle, and read during the second half of the clock cycle.



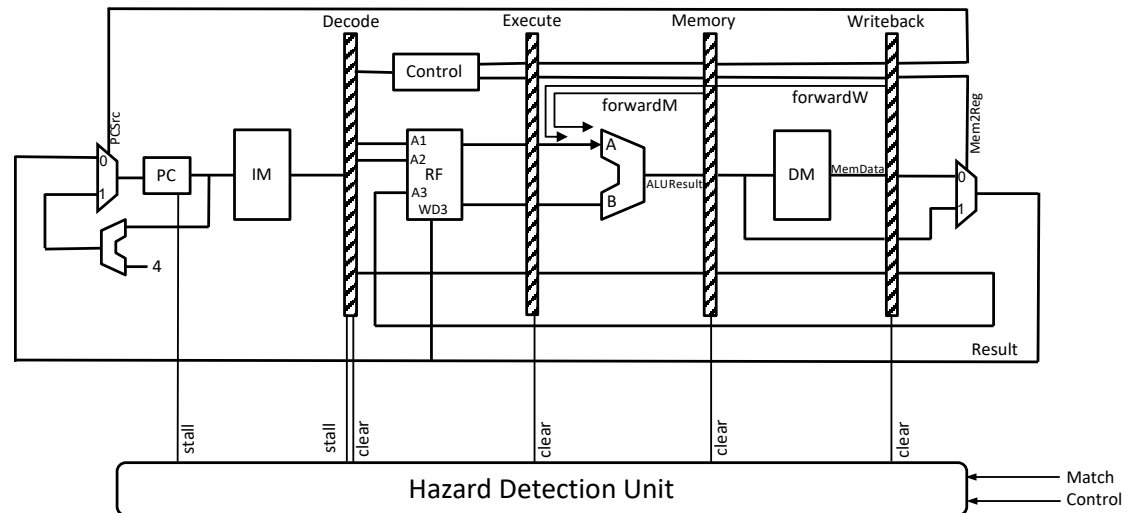
For the instruction sequence shown below, rewrite the assembly code by inserting NOPs to execute the code correctly on the pipelined CPU. Your answer is only valid if you use as few NOPS as possible. Do not write complete instructions. Use instruction labels, i1, i2, etc.

```
i1:  LDR R3, [R2, #4]
i2:  ADD R4, R3, R2
i3:  ADD R5, R4, #1
i4:  SUB R7, R5, R5
i5:  B  TARGET
i6:  ADD R0, R0, #1
i7:  ADD R1, R1, #1
    TARGET:
i8:  ADD R8, R0, #1
i9:  ADD R9, R1, R5
i10: STR R9, [R12]
```

Write your answer on the next page.

Write your answer in this box.

Q2. Consider the pipelined CPU from Q1, but now, with forwarding and hazard detection in hardware. Assume in the case of always-untaken prediction policy, the CPU is capable of flushing pipeline registers on misprediction. Also, in the case of always-taken branch prediction policy, CPU is capable of stalling the pipeline as soon as the branch is decoded (i.e., pipeline is stalled when the branch is in the Decode stage). The register file is written during the first half of the clock cycle, and read during the second half of the clock cycle. CMP sets flags in EX, and BEQ reads flags in EX, i.e., CMP-BEQ pair is not a hazard. PC is updated at the rising edge of the clock.



Now, carefully study the following instruction sequence. The address of each instruction is shown in hexadecimal on the left. Each instruction is 32 bits in size.

```

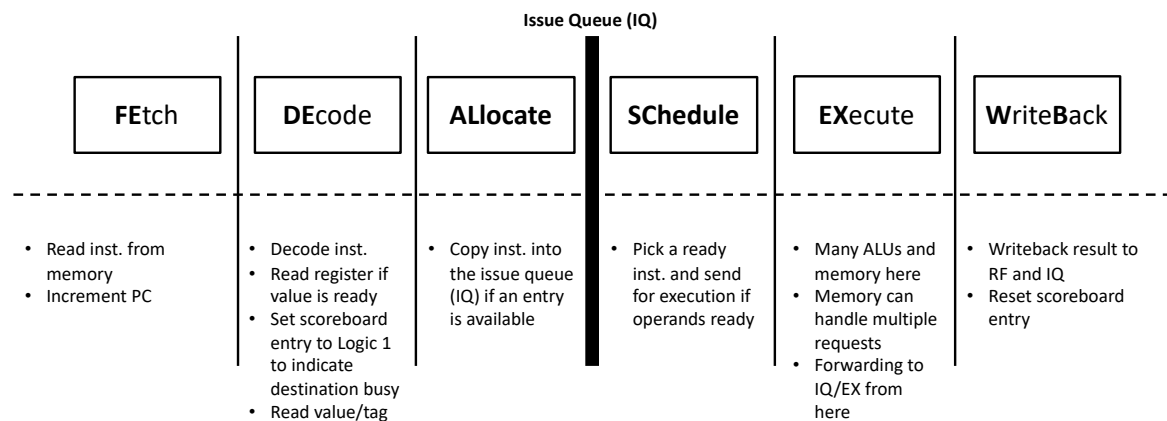
0x00:  MOV R0, #0
0x04:  MOV R1, #1
      LOOP:
0x08:  CMP R0, R1
0x0C:  BEQ DONE
0x10:  LDR R2, [R6, R7, LSL 2]
0x14:  ADD R2, R2, #4
0x18:  STR R2, [R6, R7, LSL 2]
0x1C:  ADD R0, R0, #1
0x20:  B LOOP
      DONE:
0x40:  NOP
0x44:  NOP
0x48:  NOP
0x4C:  NOP

```

For the above CPU and instruction sequence, answer questions on the next page.

- A. Assuming CPU uses the branch-always-taken policy for branch instructions, which instruction's address (hexadecimal) does PC contain in cycle # 8: _____
- B. Assuming CPU implements the branch-always-untaken policy for branch instructions, and branches resolve (i.e., update PC with correct target address) in the writeback stage, which instruction's address (hexadecimal) does PC contain in cycle # 8: _____
- C. Assuming CPU implements the branch-always-untaken policy for branch instructions, and branches resolve (i.e., update PC with correct target address) in the execute stage, which instruction's address (hexadecimal) does PC contain in cycle # 8: _____
- D. Which one of the four pipeline registers (see labels in schematic) forwards R2 to instruction at the address 0x14: _____
- E. In which cycle at the rising edge does the partially processed CMP instruction appears in the Memory pipeline register: _____
- F. Assuming CPU implements the branch-always-untaken policy for branch instructions, and branches resolve (i.e., update PC with correct target address) in the writeback stage, in which cycle does PC contains 0x40: _____
- G. Assuming CPU implements the branch-always-untaken policy for branch instructions, and branches resolve (i.e., update PC with correct target address) in the execute stage, how many cycles does it take to process all instructions (including NOPs): _____
- H. What is the output of the 2:1 multiplexer in the fetch stage at the end of cycle #3: _____

Q3. Consider a dynamically scheduled out-of-order CPU pipeline shown below. The six stages are FE, DE, AL, SC, EX, and WB. Carefully read what each stage does below. DE contains a scoreboard to indicate valid values and busy registers. AL copies instructions into an issue queue (IQ) or scheduler. SC issues (sends) ready instructions to the EX stage. EX contains many functional units as well as memory. The scoreboard has eight entries one corresponding to each register in the hypothetical ARM ISA. Before the execution of the first instruction, all scoreboard entries are set to 1 (i.e., values in RF are ready, and no register is busy). A busy register, i.e., value is being computed, and dependent instruction must wait, is indicated by Logic 0 in a scoreboard entry. The IQ is empty with 16 entries available for allocation.



- A. Complete the pipeline diagram below by indicating, for each cycle, the stage (e.g., FE) that each instruction is in. Assume each memory access takes four cycles to complete.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
i1: LDR R2, [R1, #0]																	
i2: LDR R4, [R3, #1]																	
i3: ADD R5, R2, #1																	
i4: SUB R6, R4, #1																	

- B. What are the contents of the following scoreboard entries during cycle # 9.

R1____
R2____
R3____
R4____

- C. List four issues in the out-of-order (OOO) CPU in B above that are addressed by an OOO CPU with register renaming and hardware speculation (one short bullet point per issue).

END OF EXAMINATION