ASHESI
UNIVERSITY

# CS 331 COMPUTER ORGANIZATION AND ARCHITECTURE

**SPRING 2024**

**MID–SEMESTER EXAMINATION: SECTION B and C**

**Date: 14th March 2024.**

**Duration: 50 minutes**

**Total Points: 60**

**INDEX NUMBER: ____________________**

**INSTRUCTIONS:**

- Answer **ALL** Questions in Section B and ONE in Section C
- Each Question group should start on a new page. For instance, after answering all questions under Question 1, start answering Question 2 on a new page.
- Indicate ID and Question Number on each page of the answer booklet.

## PART B (33 pts)

**B1**

a) If we have an $n$-digit unsigned numeral $d_{n-1}d_{n-2}\ldots d_0$ in *radix* (or *base*) $r$, then the value of that numeral is $\sum_{i=0}^{n-1} r^i d_i$, which is just fancy notation to say that instead of a 10's or 100's place we have an $r$'s or $r^2$'s place. For the three radices binary, decimal, and hex, we just let $r$ be 2, 10, and 16, respectively.

Convert the following numbers from their initial radix into the other two common radices:

a) 0b10010011.11011

b) 63

c) 0x0123

**[9 pts]**

**SOLUTION**

**To convert the given numbers into the other two common radices, we'll perform the following conversions:**

**a) Convert 0b10010011.11011 from binary to decimal, then decimal to hexadecimal.**

**b) Convert 63 from decimal to binary and hexadecimal.**

**c) Convert 0x0123 from hexadecimal to binary and decimal.**

**Performing these conversions:**

**a) 0b10010011.11011:**

- **Binary to Decimal: 0b10010011.11011=1×27+0×26+0×25+1×24+0×23+0×22+1×21+1×20+1×2−1+1×2−2+0×2−3+1×2−4+1×2−5=147.84375**
- **Decimal to Hexadecimal: 147.84375 = 0x93.D8**

**b) 63:**

- **Decimal to Binary: 63=32+16+8+4+2+1=0b11111163=32+16+8+4+2+1=0b111111**
- **Decimal to Hexadecimal: 63=0x3F**

**c) 0x0123:**

- **Hexadecimal to Binary: 0x0123=0000 0001 0010 0011=0b00000001001000110x0123=0000 0001 0010 0011 =0b0000000100100011**
- **Hexadecimal to Decimal: 0x0123=1×16^2+2×16^1 +3×16^0 =256 + 32+ 3 = 291**

So, the conversions are:

a) 0b10010011.110110b10010011.11011 is 0x93.D8 in hexadecimal and 147.84375 in decimal.

b) 63is 0b111111 in binary and 0x3F in hexadecimal.

c) 0x0123 is 0b0000000100100011 in binary and 291 in decimal.

b) Compute the decimal result of the following arithmetic expressions involving 8-bit Two's Complement numbers as they would be calculated on a computer. Do any of these result in an overflow? Are all these operations possible?

a) 0b01011001 − 0b10000111

b) 0b10100011 + 0b10111010

c) 0x3B + 0x06

[9 pts]

SOLUTION

Let's compute the decimal result of the given arithmetic expressions involving 8-bit Two's Complement numbers:

a) 0b01011001 – 0b10000111

The decimal value of 0b01011001 is 89 (64+16+8+1).

The decimal value of 0b10000111 is -121 (-128+4+2+1).

Therefore, the computation is 89 – (-121) = 89+121, meaning that the result is a positive one. The result is 210 in decimal. This will not result in an overflow in an 8-bit calculator.

b) 0b10100011+0b10111010:

The decimal value of 0b10100011 is -93 (-128+32+2+1).

The decimal value of 0b10111010 is -70 (-128+32+16+8+2).

Therefore, we expect a negative result when we add two negative numbers. The addition of -93 to -70 results in -163. The result cannot be represented cannot be represented in an 8-bit computer as it does not fall in the range of -127 and 126.

c) 0x3B+0x06:

First, let's represent the numbers in decimal:

**0x3B represents 0b00111011 in binary, which is 59 in decimal (32+16+8+2+1).**

**0x06 represents 0b00000110 in binary, which is 6 in decimal (4+2).**

**Adding 59 and 6, we get 65. This will not result in an overflow in an 8-bit calculator.**

**So, 0x3B+0x06=65 in decimal.**

**In summary, all operations are possible without resulting in overflow for the given arithmetic expressions.**

**B2**

a) Given the IEEE 754 single-precision floating-point number representation: **1 01101101 11011000001000000000000, what is its equivalent number in decimal?** **[4 pts]**

b) Convert the decimal number **(-243.5625)** to IEEE 754 single-precision floating-point format. **[5 pts]**

c) Describe what each of the following parts of the computer does. **[6 pts]**
   a. Control Unit
   b. Memory unit
   c. ALU

SOLUTION

a) Given the IEEE 754 single-precision floating-point number representation: 1 01101101 11011000001000000000000, let's break it down:

1. Sign bit: The sign bit is 1, indicating a negative number.
2. Exponent: 01101101 represents the exponent in biased notation. Converting to decimal and subtracting the bias (127), we get $01101101_2 = 109_{10} - 127 = -18_{10}$
3. Mantissa: 11011000001000000000000 represents the mantissa (also known as significand or fraction).

Putting it all together, the number represented in decimal is calculated as follows: $(-1)^1 \times 2^{(-18)} \times (1+0.11011000001)$

$= -1 \times 2^{-18} \times (1.1101100001)$

$= -1.11011000001 \text{ x } 2^{-18}$

$= -0.0000070352107$

So, the equivalent decimal representation of the given IEEE 754 single-precision floating-point number is approximately -0.0000070352107.

b) To convert the decimal number −243.5625 to IEEE 754 single-precision floating-point format, we need to represent it in binary scientific notation and then normalize it. However, to perform this conversion, we first need to represent the integer part and the fractional part of the number separately.

$-243.5625=-(243+0.5625)$

1. Converting the integer part (243) to binary = 11110011
2. Converting the fractional part (0.5625) to binary:

   $0.5625\times2=1.125 \implies 1$

   $0.125\times2=0.25 \implies 0$

   $0.25\times2=0.5 \implies 0$

   $0.5\times2=1.0 \implies 1$

So, the binary representation of the fractional part is $0.1001_2$.

Now, we combine the integer part and the fractional part:

$-243.5625=-11110011.1001_2$

Next, we normalize the binary number:

$-11110011.1001_2=-1.11100111001_2\times2^7$

Now, we represent this number in IEEE 754 single-precision floating-point format.

The sign bit is 1 since it's negative.

The exponent is 7+127=134, represented in binary as 1000011010000110.

The mantissa is 11100111001000000000000.

So, the IEEE 754 single-precision floating-point representation of −243.5625 is: 11000011011100111001000000000000.

c) Description of each part of the computer:

a. Control Unit: The control unit is responsible for controlling the operation of the computer. It interprets program instructions, directs the flow of data between the CPU and other parts of the computer, and coordinates the activities of the other hardware components. It generates control signals to manage the execution of instructions, including fetching instructions from memory, decoding them, and executing them.

b. Memory unit: The memory unit is responsible for storing data and instructions that are currently being used by the CPU. It consists of different types of memory, such as RAM (Random Access Memory) and cache memory. RAM is used for temporary storage of data and instructions that are actively being processed by the CPU, while cache memory stores frequently accessed data to speed up access times. Memory units provide fast access to data and instructions for the CPU and are essential for the execution of programs.

c. ALU (Arithmetic Logic Unit): The ALU is a fundamental component of the CPU responsible for performing arithmetic and logical operations on data. It performs operations such as addition, subtraction, multiplication, division, bitwise AND, bitwise OR, and bitwise XOR. The ALU receives input data from registers, performs the specified operation according to the control signals generated by the control unit, and produces the result, which is then stored back in registers or sent to other parts of the CPU for further processing. The ALU is the computational engine of the CPU and plays a critical role in executing program instructions.

## PART C – Choose ONLY ONE (27 pts)

**C1:**

The following equations specify a sequential circuit with two D flip-flops:

$$A(t+1)=xy'+xB$$
$$B(t+1)=xA+xB'$$
$$z=A$$

Analyze the sequential circuit by answering the following:

a) Draw the sequential circuit. **[6 pts]**

b) Draw the state table. **[10 pts]**

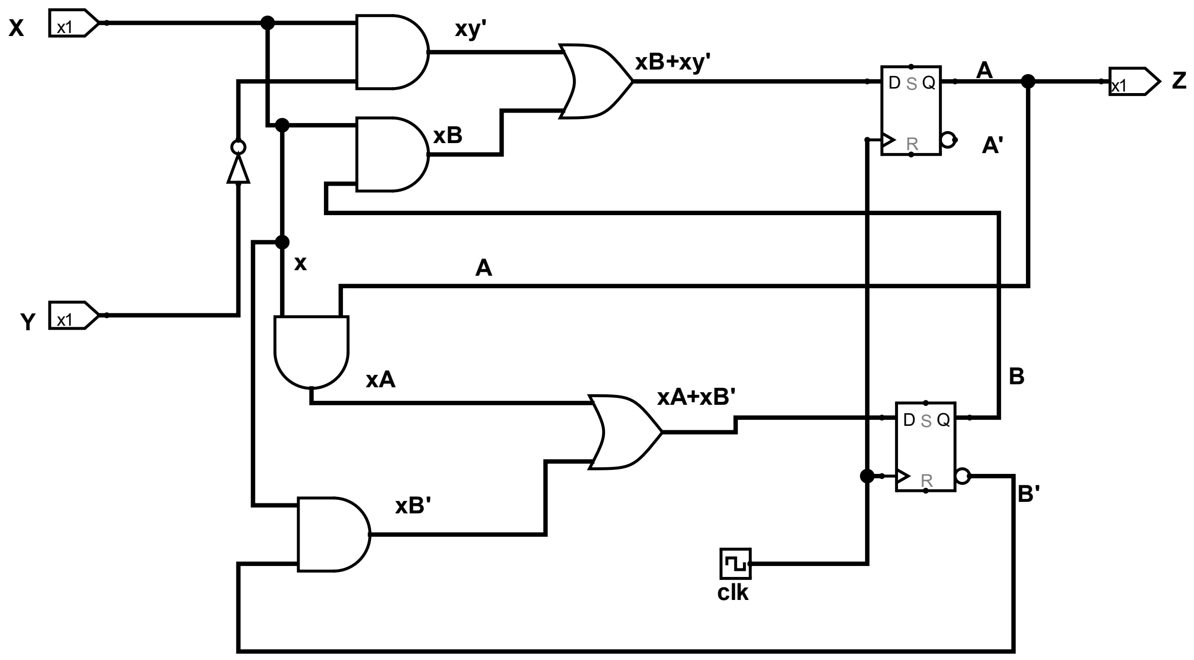
c) Draw the state diagram. **[5 pts]**

d) Is the circuit a Mealy or Moore Machine? What is your reason? **[3 pts]**

e) What is the difference between synchronous and asynchronous modes in sequential circuit operation? **[3 pts]**

**SOLUTION**

**a)**



**[6 pts]**

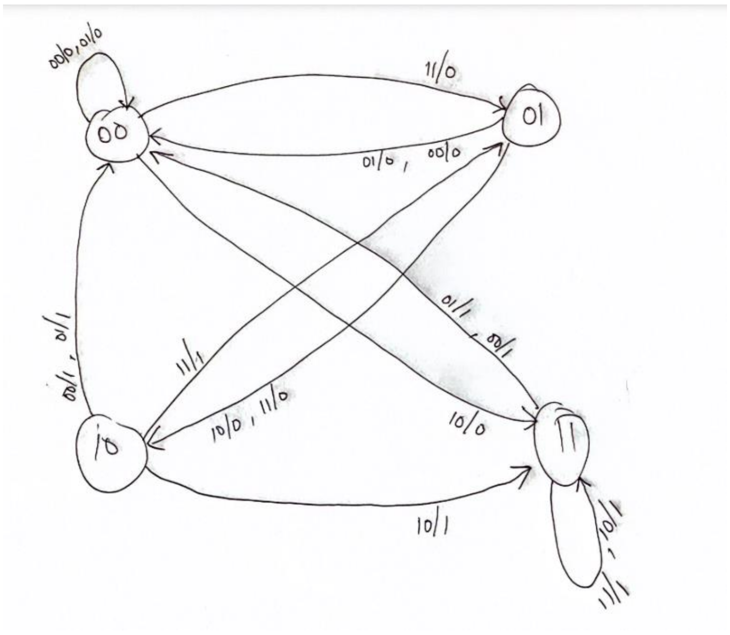
a) Draw the state table. **[10 pts]**

## State Table for the above State equations.

| A | B | x | y | xy' | xB | xA | xB' | A(t+1)= $xy'+xB$ | B(t+1)= $xA+xB'$ | z |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |

b) Draw the state diagram. **[5 pts]**

**SOLUTION**



c) Is the circuit a Mealy or Moore Machine? What is your reason? **[3 pts]**

**SOLUTION**

**The circuit is a Mealy machine. The output, Y depends on both the inputs and current states.**

e) Synchronous and asynchronous modes are two approaches to sequential circuit operation:

**Synchronous: Changes in state are synchronized to a global clock signal, ensuring predictable timing. This simplifies design and analysis, making it suitable for systems requiring precise timing like microprocessors and communication systems.**

**Asynchronous: State changes can happen at any time in response to inputs, without relying on a clock signal. This mode is more complex to design due to timing hazards but can offer advantages such as lower power consumption and potentially faster operation for specific applications like certain control systems.**

**Detailed solution: Synchronous and asynchronous modes are two different ways in which sequential circuits operate. The main difference lies in how they handle the timing of inputs and outputs.**

1. **Synchronous Sequential Circuits:**
    - **In synchronous circuits, the timing of all state changes (transitions from one state to another) is controlled by a clock signal.**
    - **All flip-flops within the circuit change state simultaneously in response to each clock pulse.**
    - **The circuit's behavior is synchronized to the clock signal, ensuring that all changes occur at specific points in time.**
    - **This synchronization simplifies the design process and analysis of the circuit, as it eliminates problems related to timing hazards.**
    - **Synchronous circuits are commonly used in digital systems where timing precision and predictability are crucial, such as in microprocessors, memory units, and communication systems.**
2. **Asynchronous Sequential Circuits:**
    - **In asynchronous circuits, the timing of state changes is not controlled by a global clock signal.**
    - **Instead, changes in the circuit's state can occur at any time in response to changes in input signals.**
    - **Asynchronous circuits are typically more complex and harder to design compared to synchronous circuits because the designer must consider timing hazards and ensure correct operation under all possible input sequences.**
    - **However, asynchronous circuits can offer advantages in certain situations, such as reduced power consumption and potentially faster operation for specific applications.**
    - **Asynchronous circuits are used in scenarios where strict timing constraints are not feasible or where specific performance requirements demand asynchronous operation, such as in some types of control systems or low-power applications.**

**In summary, synchronous circuits rely on a clock signal to coordinate state changes, while asynchronous circuits allow state changes to occur independently of a clock signal. Each mode has its own advantages and disadvantages, and the choice between them depends on the specific requirements of the application.**

**C2**

$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer} \times \text{Yield}}$$

$$\text{Dies per wafer} \approx \text{Wafer area}/\text{Die area}$$

$$\text{Yield} = \frac{1}{(1 + (\text{Defects per area} \times \text{Die area}/2))^2}$$

a) Assume a **15 cm** diameter wafer has a cost of **GHC 26**.0 and contains 84 dies and has a **0.20** defects per square centimetre. Find the cost per die of the wafer. **[6 pts]**

SOLUTION

Radius = 15/2 = 7.5 cm

Wafer area = pi * radius °2 = 22/7 * 7.5 * 7.5 = 176.7857

Die area = Wafer area / Dies per wafer = 176.7857/84 = 2.1046

Yield = 1 / (1 + (0.20 * 2.1046/2)) ^2 = 0.6825

Cost per die = 26 / (84 * 0.6825) = GHC 0.4535

b) The table below shows some measurement metrics of compiled code sequences using instructions in breadth-first search, depth-first search and A-star algorithms.

| **Algorithms** | **Breadth-first** | **Depth-first** | **A-star** |
|---|---|---|---|
| CPI for Algorithm | 2 | 3 | 1 |
| IC in Sequence 1 | 3 | 2 | 2 |
| IC in Sequence 2 | 1 | 1 | 4 |

I. Calculate the weighted average CPI for each sequence. **[4 pts]**

SOLUTION

Sequence 1

Total IC = 3 + 2 + 2 = 7

Clock cycles = (3*2) + (2*3) + (2*1) = 6 + 6 + 2 = 14

Average CPI = 14/7 = 2

Sequence 2

Total IC = 1 + 1 + 4 = 6

Clock cycles = (1*2) + (1*3) + (4*1) = 2 + 3 + 4 = 9

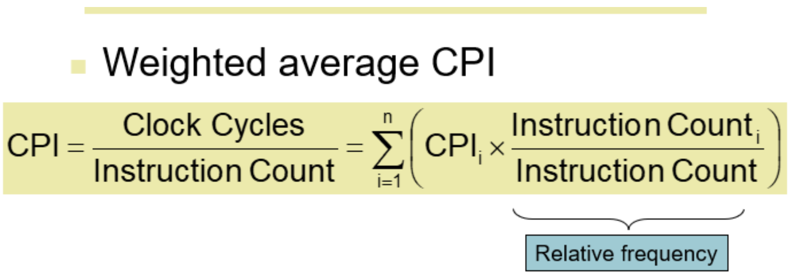
Average CPU = 9/6 = 1.5

II. Which code sequence will you prefer? What is your reason? **[2 pts]**

SOLUTION

Sequence 2 will be preferred. The average clock cycles needed to run the same algorithms in sequence 2 is smaller than that of sequence 1. Fewer clock cycles per instruction is always better.

Related Formulae:



- Weighted average CPI

$$\text{CPI} = \frac{\text{Clock Cycles}}{\text{Instruction Count}} = \sum_{i=1}^{n}\left(\text{CPI}_i \times \underbrace{\frac{\text{Instruction Count}_i}{\text{Instruction Count}}}_{\text{Relative frequency}}\right)$$

IC – Instruction Count

CPI – Cycles per instruction

c) You were recently hired as an engineer in a company that designs alarm systems custom-made to meet the customer's specifications. You are asked to design a system that uses the inputs of three sensors A, B, and C. The alarm should go off (activated) when the following criteria are met:

a. When A is off, or

b. When B is on and C is off, or

c. When both A and C are on.

i. Write the truth table for the function. **[5 pts]**

SOLUTION

| A | B | C | A is off. | B is on and C is off | Both A and C are on | Output |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |

ii. Write the Boolean expression in the Product of Sums (POS) and the Sum of Products (SOP) forms. **[5 pts]**

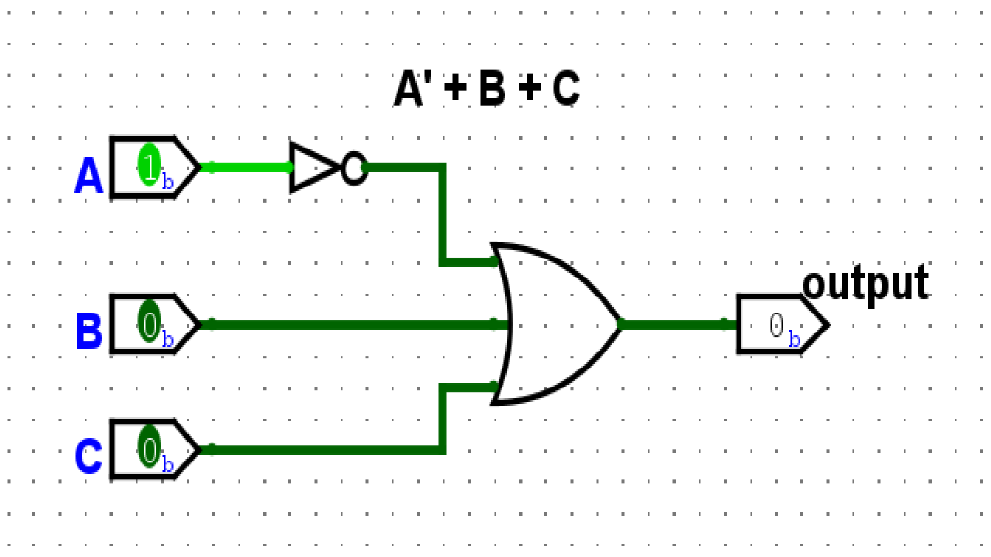
SOLUTION

POS = (A’ + B + C)

SOP = A’B’C’ + A’B’C + A’BC’ + A’BC + AB’C + ABC’ + ABC

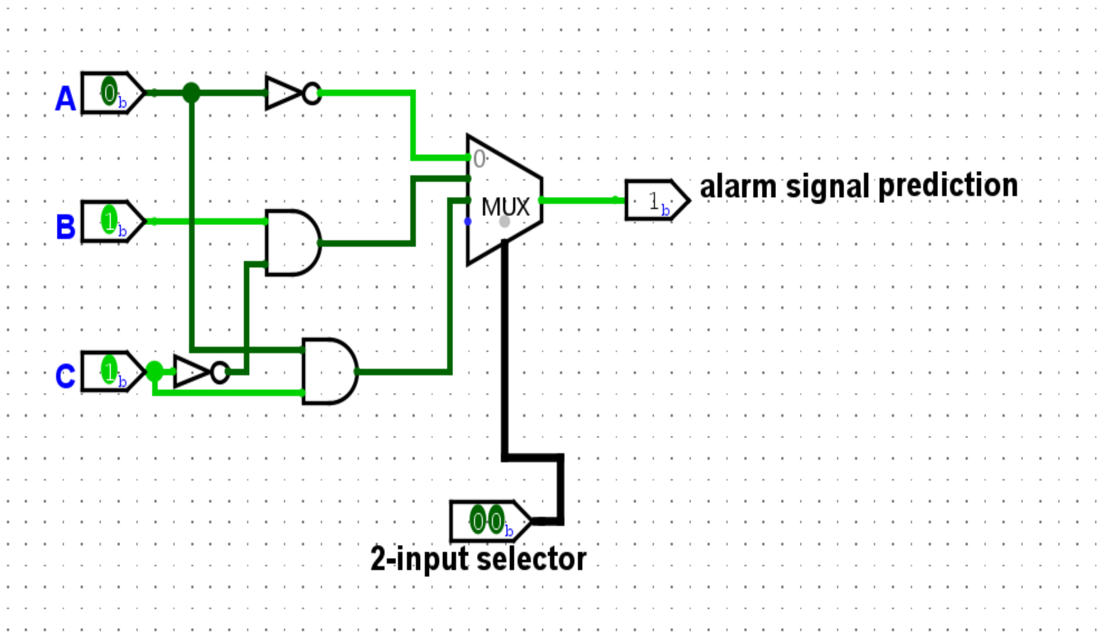
iii. Draw the circuit diagram and implement the same functionality using 2-input selector multiplexers. **[5 pts]**

SOLUTION



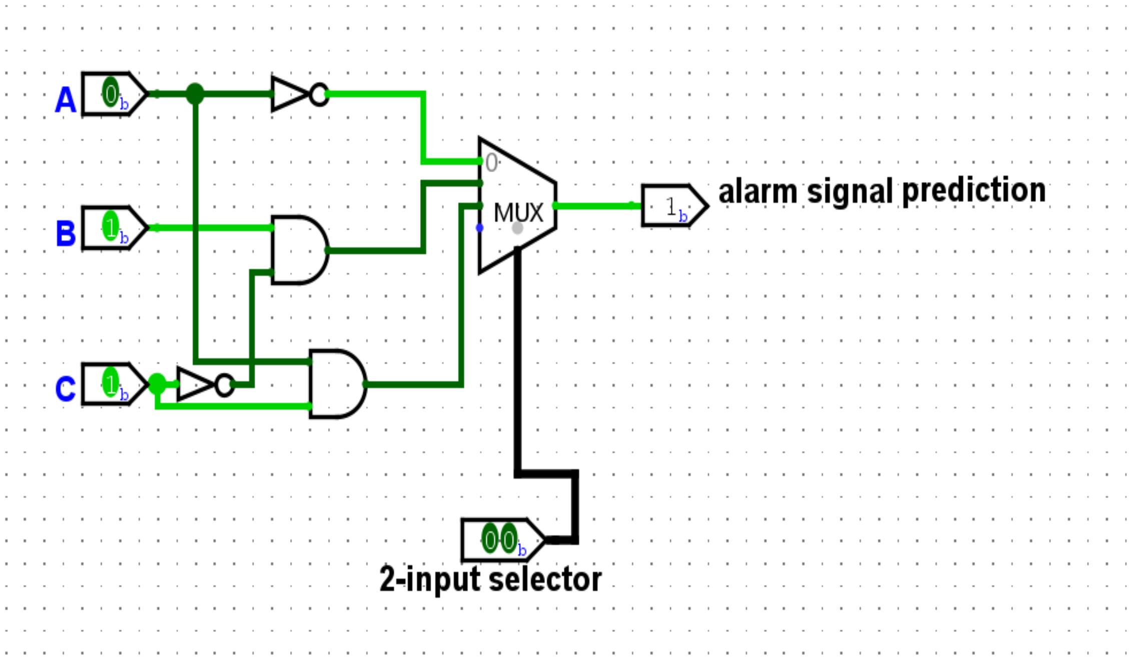
**<u>Using 2-input selector multiplexer</u>**

Since we need only one condition to be true for the alarm to be activated, the inputs to the mux are the three conditions, where the mux outputs a 1 (signalling that the alarm is activated) if a condition is met.



**Justification:**

- **When A is off: The picture below shows that this condition is true using selectors "00", hence we can use this to deduce that the alarm will be activated even if the other conditions are not met.**


A
0b
B
1b
C
1b
0
MUX
1b
alarm signal prediction
00b
2-input selector