Name and Student ID: Rafat Momin 593070532 Lab Section: 7

Date: 12/7/23

Net ID: rafatmom

**Submission Instructions:**

**Prelab:**

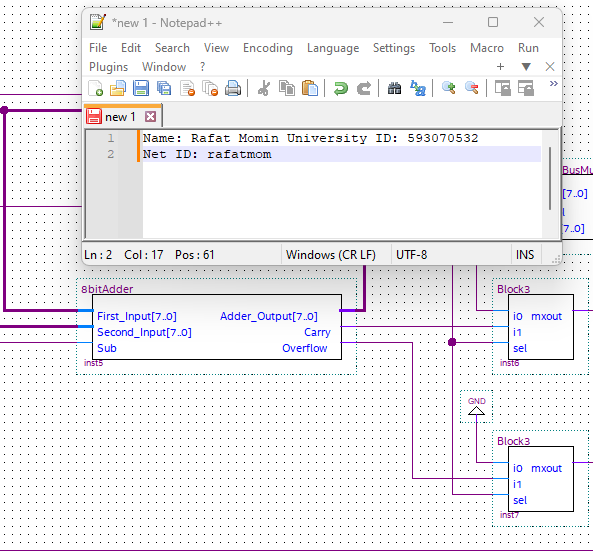
1. **No prelab**

**Lab:**

1. **Complete the scavenger hunt according to the instructions**
2. **Take screenshots of each component and include them in the document below (note: to receive points your NetID has to be visible in the screenshot, say in a command window that is in the background).**
3. **Complete this document and upload it to Canvas**

**Part 1: CPU Scavenger Hunt**

**Q1.** Find the **adder** inside the ALU and answer the following:

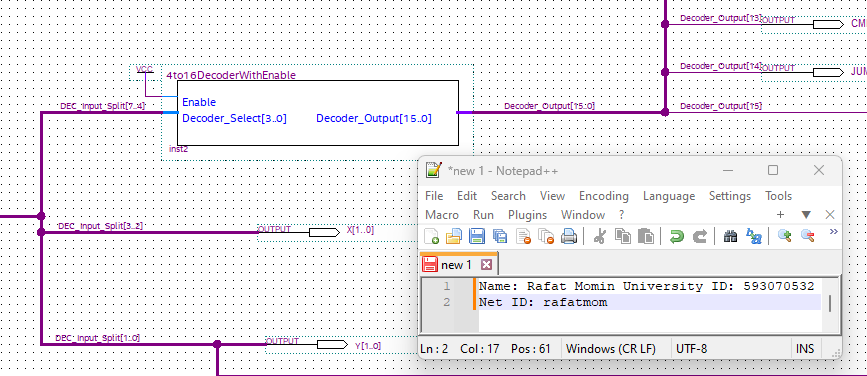
* What is the name of this component?
* **Answer:** 8-bit adder
* Is it a ripple-carry or carry lookahead adder?
* **Answer:** Ripple-carry
* Can it also do subtraction?
* **Answer:** Yes, it can. (Because of XOR)
* What is the size of its two operands in bits?
* **Answer:** 8 bits
* <<< screenshot of the adder symbol >>>
* 
* **Screenshot for Q1**

**Q2.** Find a 4-to-16 **decoder** and answer the following:

* What is the name of this component?
* **Answer:** 4to16DecoderWithEnable
* Does it have an enable input?

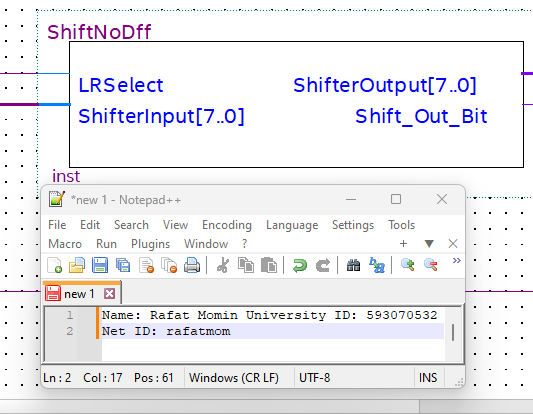
**Answer:** Yes

* What are the names of its outputs 4 and 6?
* **Answer:** 4 is Add, and 6 is Sub.
* Can you guess what is its function in this CPU?
* **Answer:** It helps to decide whether an addition or a subtraction will happen based on the output of the instruction memory.
* <<< screenshot of the decoder symbol >>>



**Screenshot for Q2**

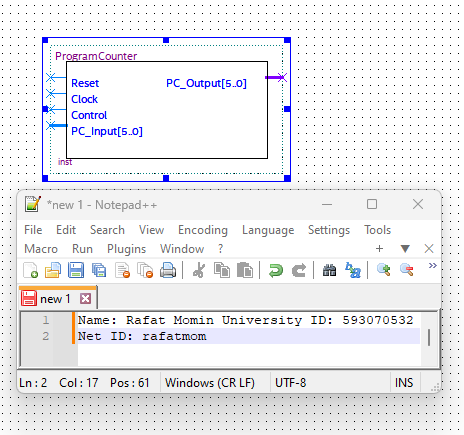
**Q3.** Find a **shifter circuit** and answer the following:

* What is the name of this component?
* **Answer:** ShiftNoDFF
* What is the size of the input in bits?
* **Answer:** 8
* What happens to the most significant bit on shift left?  
  **Answer:** It takes the most significant bit out of the circuit to the output.
* What happens to the least significant bit on shift right?
* **Answer:**  It takes the least significant bit out the circuit to the output.
* <<< screenshot of the shifter symbol >>>
* 
* **Screenshot for Q3**

**Q4.** Find the **program counter** and answer the following:

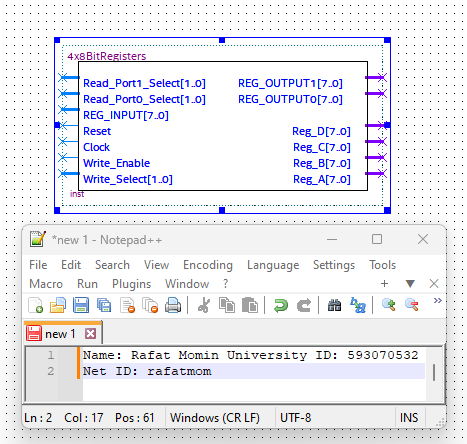
* What is the name of this component?
* **Answer:** ProgramCounter
* What is the size of the output bus in bits?
* **Answer:** 6
* How many control lines does it have?
* **Answer: 1**
* What type of high-level circuit does it implement?
* **Answer:** Parallel Load Resgister

* <<< screenshot of the program counter symbol >>>



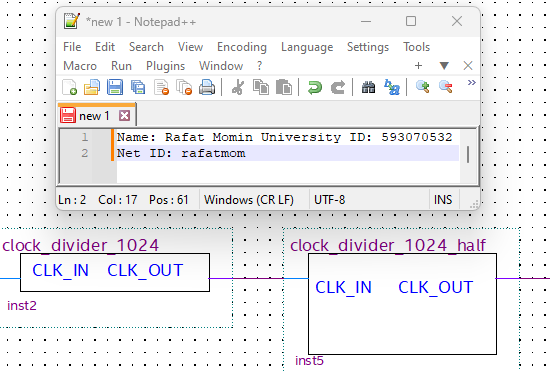
**Screenshot for Q4**

**Q5.** Find a **register file** with exactly 4 registers and answer the following:

* What is the name of this component?
* **Answer:** 4x8BitRegisters
* What is the size of each register in bits?
* **Answer:** 8 bits
* What type of Flip-Flops are used to construct each register?
* **Answer:** D flip-flop
* The contents of how many registers can be read at the same time?
* **Answer:** 2
* <<< screenshot of the register file symbol >>>
* 
* **Screenshot for Q5**

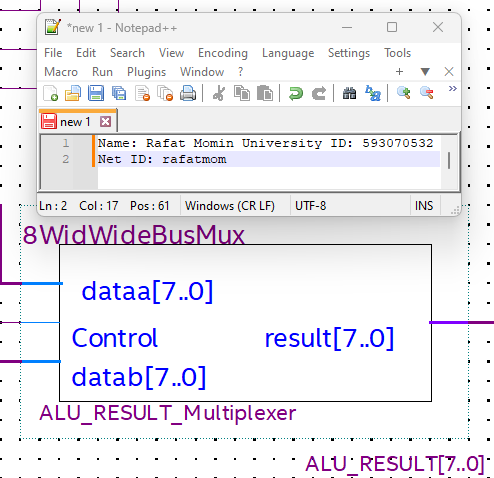
**Q6.** Find the two **clock dividers** for the **VideoGame\_Clock** and answer the following:

* How are they implemented?
* **Answer:** It has 10 flip-flops and the clock-divider is a counter, implemented with counters.
* They slow down the clock by a factor of X and Y. What are X and Y?
* **Answer:** X= 512, Y= 1024. The reason is : every total clock cycle takes about 29 for the counter to go through the cycle of the 8 bit combinations. Therefore, X = 512 (the first bit remains intact). And for the second factor, it could also reset the whole cycle of the 9 bits so it is 210 = 1024.
* <<< screenshot of the two clock dividers >>>

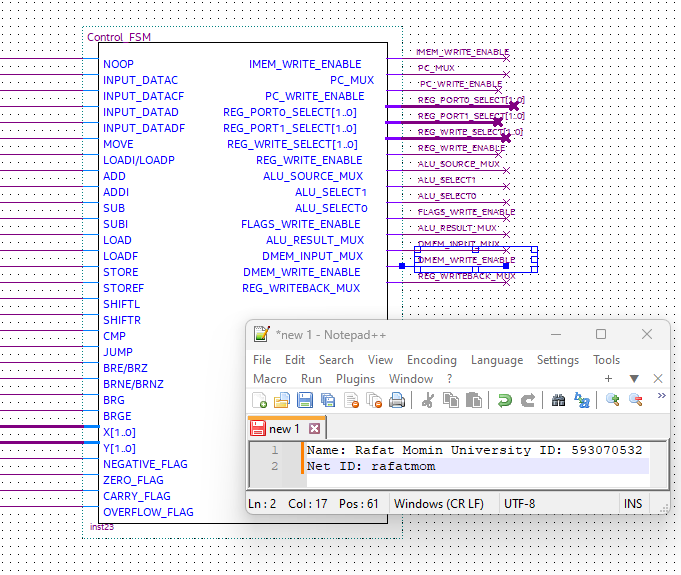


**Screenshot for Q6**

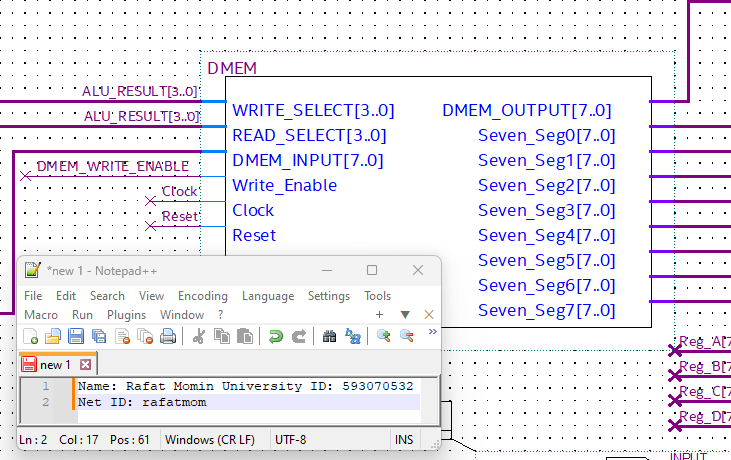
**Q7.** Find the **multiplexer** that sits after the ALU and takes the output of the ALU as one of its inputs. Then, answer the following:

* What is the name of this component?
* **Answer:** 8WidWidebusMux
* Where does the other input come from?
* **Answer:** From the box IMEM. (It is taking 8 bits out of the 16).
* What is the size of each input in bits?
* **Answer:** 8
* How many select lines does it have in bits?
* **Answer:** A single line.
* <<< screenshot of the multiplexer symbol >>>
* 
* **Screenshot for Q7**

**Q8.** Find the circuit that outputs the signal **DMEM\_WRITE\_ENABLE** and then answer the following:

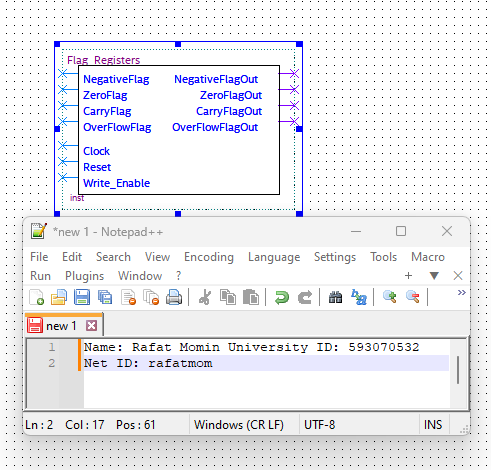
* In which block is this circuit located?
* **Answer:** Control FSM.
* What is the Boolean expression for this signal?  
  **Answer:** INPUT\_DATAD + INPUT\_DATAF + STOREF + STORE
* <<< screenshot of the block in which the circuit is located >>>
* 
* **Screenshot for Q8**

**Q9.** Examine the **DMEM** box and answer the following:

* What are the names of the control lines for this box?
* **Answer:** READ\_SELECT[3..0]
* WRITE\_SELECT[3..0]
* What the high-level component is used to store the data?  
  **Answer:** Register file
* What is the size of the data memory in bytes?  
  **Answer:** 16 bytes
* <<< screenshot of the high-level component symbol >>>
* 
* **Screenshot for Q9**

**Q10.** Find the **flags register** and answer the following:

* How many flags does it store?  
  **Answer:** 4
* What are the names of these flags?  
  **Answer:** Negative\_Flag, Zero\_Flag, Carry\_Flag, Overflow\_flag
* <<< screenshot >>>



**Screenshot for Q10**

**Part 2: PONG. Take a cellphone picture of the board as you are playing the game. Selfies are OK too as long as the game is visible in the background ☺**

* <<< picture >>>

  
**Screenshot for Part 2**