Cohort Name: Logical

Members: Vineet Goyal, Amogh Yatnatti, Muhammad Khan, Sanika Buche, Unnati Patel

System Design

Description

We are building an Arithmetic Logic Circuit (ALU) with 11 operations. The basic operations include RESET, PRESET, and NO-OP. The logical operations include 16-bit NOT, OR, AND, and XOR. The mathematical operations include 16-bit addition, subtraction, multiplication, and division. However, for the circuit diagram or the ALU, we are not taking overflow into consideration. We use a 16 by 32 multiplexor (MUX) to decide what operation we want to do. This means that our operations are working in parallel. Our input is 16 bits for all operations, but our output will always be 32 bits to account for the multiplication operation's output. For each operation, we have a corresponding opcode that we can input into the MUX. For operations where we have two inputs, we have a 16-bit input, and we use the lower 16-bits of the accumulator for the other input. For operations where we only have one input (i.e. NOT operation), we only use the lower bits of the accumulator. We store our 32-bit output from the MUX into the accumulator register by sending it to 32 1-bit, positive-edge trigger D Flip-Flops working synchronously. The ALU outputs the contents of the accumulator. The unused multiplexor channels in the ALU will be set to ground (0). All of the modules are written in behavioral-style.

Parts List

- **Wires**

  -> wire clk;

  -> wire rst;

  -> wire [15:0] A;

  -> wire [15:0] B;

  -> wire [3:0] opcode;

  -> wire [15:0][31:0]channels;

  -> wire [31:0] b;

  -> wire [31:0] outputADD;

  -> wire [31:0] outputAND;

  -> wire [31:0] outputXOR;

  -> wire [31:0] outputOR;

  -> wire [31:0] outputNOT;

  -> wire [31:0] outputSUB;

  -> wire [31:0] outputPRESET;

  -> wire [31:0] outputMULT;

  -> wire [31:0] outputDIV;

  -> wire [31:0] cur;

- **Gates**

  -> 16-Bit NOT

  -> 16-Bit XOR

  -> 16-Bit OR

  -> 16-Bit AND

- **Combinational Logic**

  -> 16-Bit DIVIDER

  -> 16-Bit MULTIPLIER

  -> 16-Bit ADDER

-> 16-Bit SUBTRACTOR

-> 16x32 Bit MUX

- **Sequential Logic**

  -> #32 1-Bit DFF -> Which acts like an Accumulator Register

- **OP Codes**

  -> No-op

  -> Preset

  -> Reset

  -> MULT

  -> ADD

  -> SUB

  -> DIV

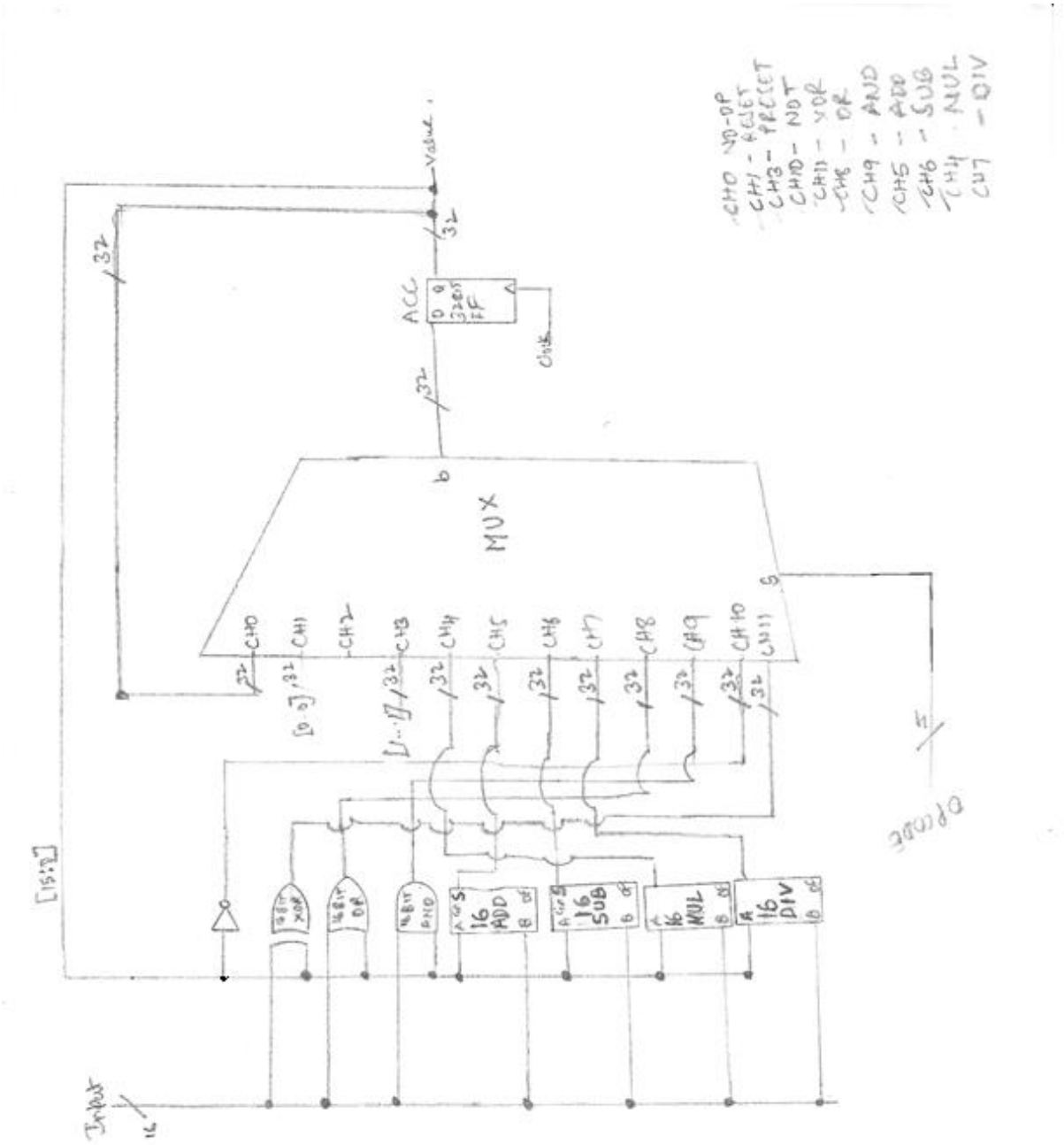
  -> OR

  -> AND

  -> NOT

  -> XOR

Circuit Diagram



Opcode Table

Command	Description	Opcode	Multiplexer Channel
No – Op	Refreshes Accumulator with feedback	0000	Channel 0
Reset	Sets Accumulator to 0	0001	Channel 1
Preset	Sets Accumulator to 1	0011	Channel 3
MULT	Multiply the input with lower 16 bits of accumulator	0100	Channel 4
ADD	Add the input with lower 16 bits of accumulator	0101	Channel 5
SUB	Subtract the input with lower 16 bits of accumulator	0110	Channel 6
DIV	Divide the input with lower 16 bits of accumulator	0111	Channel 7
OR	Does the OR operation on the input with lower 16 bits of accumulator	1000	Channel 8
AND	Does the AND operation on the input with lower 16 bits of accumulator	1001	Channel 9
NOT	Does the NOT operation on the input with lower 16 bits of accumulator	1010	Channel 10
XOR	Does the XOR operation on the input with lower 16 bits of accumulator	1011	Channel 11

State Machine