**Description** (copied from instructions – is this ok?)

An Arithmetic Logic Circuit contains a 32-bit Accumulator register, that holds the current value. The 32- bit accumulator registers is made up of 32 one-bit D Flip-flops working synchronously. On each instruction, the ALU will take a 16-bit integer input and the lower 16-bits of the accumulator register, and send them as inputs to 8 different modules. These modules will be a 16-bit adder, a 16-bit subtractor, a 16-bit multiplier, a 16-bit divider, a 16-bit AND, a 16-bit OR, a 16-bit NOT, and a 16-bit XOR. Each of these modules operate in parallel, and each of their outputs goes to a different input channel of a 16x32 bit multiplexer. (The output of a 16-bit multiplier can go up to 32 bits) The output of the ALU will be the current contents of the Accumulator register. The Subtractor will have an Overflow check, which will be an additional output. Also, the multiplexer has to handle the “No Operation” when no results of any module is used. And also, the D-Flip Flops must be able to be either preset to 1 or reset to 0. No-op, preset, and reset, will be handled by unused multiplexer channels

**Parts List**

Wires

* 16-bit input
* 32-bit output (goes into accumulator register)
* 9-bit Opcode ???

Gates and Combination Logic Components

* 16x32 bit MUX
* 32-bit Accumulator (made up of 32 1 bit D Flip-Flops)
* 9-bit(???) Decoder (…idk, for selecting the operation)
* 16-bit Adder
* 16-bit Subtractor
* 16-bit Multiplier
* 16-bit Divisor
* 16-bit AND
* 16-bit OR
* 16-bit NOT
* 16-bit XOR

**Circuit Diagram**

**OpCode Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Description** | **Op-Code** | **Multiplexer Channel** |
| No-Op | Refreshes accumulator with feedback | 000000000 | Channel 0 |
| Reset | Sets accumulator to all 0s | 000000001 | Channel 1 |
| Add | Add the input to the lower 16 bits of the accumulator | 000000010 | Channel 2 |
| Subtract | Subtract the input from the lower 16 bits of the accumulator | 000000100 | Channel 3 |
| Multiply | Multiply the input with the lower 16 bits of the accumulator | 000001000 | Channel 4 |
| Divide | Divides the input with the lower 16 bits of the accumulator | 000010000 | Channel 5 |
| AND | ANDs the input with the lower 16 bits of the accumulator | 000100000 | Channel 6 |
| OR | ORs the input with the lower 16 bits of the accumulator | 001000000 | Channel 7 |
| NOT | NOTs the input… idk  Or does it NOT the lower 16 bits of the accumulator? | 010000000 | Channel 8 |
| XOR | XORs the input with the lower 16 bits of the accumulator | 100000000 | Channel 9 |

**State Machine**