Team Ramsey.001

Kavon Ramsey

**List of Parts**

* Inputs
  + Clock 1-bit, feeds into the clock of the accumulator register
  + Input 1 (16-bits), the first binary integer used for input
  + Input 2 (16-bits), the second binary integer used for input
  + Opcode (4-bits), feeds into select of Multiplexor to select operation
* Outputs
* Output (16-bits), displays the results of the accumulator
* Interfaces
* Multiplexer (b) (16-bits), is the output of the multiplexer into the accumulator
* Multiplexer (S) (4-bits), is an input for the opcode to select multiplexer channels
* Multiplexor (CH0) (16-bits), is a feedback loop input for current accumulator
* Multiplexer (CH1) (16-bits), input for the 16-bit adder results
* Multiplexer (CH2) (16-bits), input for the 16-bit subtractor results
* Multiplexer (CH3) (16-bits), input for the 32-bit multiplier results
* Multiplexer (CH4) (16-bits), input for the 16-bit divider results
* Multiplexer (CH5) (16-bits), input for the 16-bit AND gate results
* Multiplexer (CH6) (16-bits), input for the 16-bit OR gate results
* Multiplexer (CH7) (16-bits), input for the 16-bit XOR gate results
* Multiplexer (CH8) (16-bits), input for the 16-bit NOT gate results
* Multiplexer (CH9) (16-bits), unused
* Multiplexer (CH10) (16-bits), unused
* Multiplexer (CH11) (16-bits), unused
* Multiplexer (CH12) (16-bits), unused
* Multiplexer (CH13) (16-bits), unused
* Multiplexer (CH14) (16-bits), unused
* Multiplexer (CH15) (16-bits), 16-bit 0s used to reset accumulator
* ACC (D) (16-bits), is the value being fed into the accumulator
* ACC (Q) (16-bits), is the current value held in the register
* Gates
* AND gate (16-bits), channels take Input 1 and Input 2 to AND them
* OR gate (16-bits), channels take Input 1 and Input 2 and OR them
* XOR gate (16-bits), channels take Input 1 and Input 2 and exclusive OR them
* NOT gate (16-bits), 1 channel takes Input 1 and returns the result of the NOT operation
* Combinational Logic
* Multiplexer (16-bit) 16 channels, used to select desired operation and result
* Adder (16-bit), arithmetic component to add 2 16-bit inputs
* Subtractor (16-bit), arithmetic component to subtract 2 16-bit inputs
* Multiplier (32-bit), arithmetic component to multiply 2 16-bit inputs (Need truncated to 16-bit)
* Divider (16-bit), arithmetic component to divide 2 16-bit inputs
* Sequential Logic
* ACC (16-bit), D register that contains current system value
* Modules
* ALU. This module is to be the breadboard for the ALU where all components and functions are set
* Testbench. This module runs the clock and stimulus

**OP-CODE**

|  |  |  |
| --- | --- | --- |
| **Command** | **Opcode** | **Description** |
| NO-OP | 0000 | No operation on this clock tick |
| ADD | 0001 | Feed adder into ACC |
| SUB | 0010 | Feed subtractor into ACC |
| MULT | 0011 | Feed multiplier into ACC |
| DIV | 0100 | Feed divider into ACC |
| AND | 0101 | Feed AND into ACC |
| OR | 0110 | Feed OR into ACC |
| XOR | 0111 | Feed XOR into ACC |
| NOT | 1000 | Feed NOT into ACC |
| RESET | 1111 | Reset ACC with 0s |

**MODE OF OPERATION**

|  |  |  |
| --- | --- | --- |
| **Mode** | **Code** | **Description** |
| Ready | 0000 | System is Idle |
| ADD | 0001 | Perform Addition |
| SUB | 0010 | Perform Subtraction |
| MULT | 0011 | Perform Multiplication |
| DIV | 0100 | Perform Division |
| AND | 0101 | Perform AND operation |
| OR | 0110 | Perform OR operation |
| XOR | 0111 | Perform XOR operation |
| NOT | 1000 | Perform NOT operation |
| RESET | 1111 | Reset ACC with 0s |