System Design

2.3.1.1 The List of Inputs

- "A" is a 16-bit integer that will represent the first integer passed to the functional operator.
- "B" is a 16-bit integer that will represent the second integer passed to the functional operator.
- "Op" is a 4-bit code that will determine the operation applied on the two values A and B.
- "Clock" is a signal that changes between a high and low state.

2.3.1.2 The List of Outputs

- "Result" is a 32-bit integer that will represent the output value of the selected operation on inputs A and B.
- "Error" is a 2-bit code that will determine the presence of an error and its type (overflow or division by 0).

2.3.1.3 The List of Interfaces

- "No-op" is a 32-bit line that contains the output value of the previous clock tick that feeds into multiplexor channel #0
- "Sum" is a 16-bit line that contains the output value from the adder-subtractor module for an addition or subtraction operation that will feed into multiplexor input channel #1 for addition and channel #2 for subtraction (determined by mode).
- "Product" is a 32-bit line that contains the output value from the multiplication module that will feed into multiplexor channel #3.
- "Quotient" is a 16-bit line that contains the output from the division module that will feed into multiplexor channel #4.
- "Remainder" is a 16-bit line that contains the output from the modulo module for a modulo operation that will feed into multiplexor channel #5.
- "Anded" is a 16-bit line that contains the output value of the AND module that will feed into multiplexor channel #6.
- "Nanded" is a 16-bit line that contains the output value of the NAND module that will feed into multiplexor channel #7.
- "Nored" is a 16-bit line that contains the output value of the NOR module that will feed into multiplexor channel #8.

- “Noted” is a 16-bit line that contains the output value of the NOT module that will feed into multiplexor channel #9.
- “Ored” is a 16-bit line that contains the output value of the OR module that will feed into multiplexor channel #10.
- “Xnored” is a 16-bit line that contains the output value of the XNOR module that will feed into multiplexor channel #11.
- “Xored” is a 16-bit line that contains the output value of the XOR module that will feed into multiplexor channel #12.
- “Reset” is a 32-bit line that is composed of all 0 bits that will feed into multiplexor channel #13.
- “Preset” is a 32-bit line that is composed of all 1 bits that will feed into multiplexor channel #14.
- “Mode” is a 1-bit line that determines whether the adder-subtractor does addition or subtraction.
- “Onehot” is a 16-bit one-hot line that contains the channel selection from the 4-bit operational code that will feed into the multiplexor input channels.
- “Mod0Err” is a 2-bit line that contains the error output for the modulus by 0 case which leads from the Modulo module into the Error-Handler module.
- “Div0Err” is a 2-bit line that contains the error output for the division by 0 case which leads from the Divisor module into the Error-Handler module.
- “OFErr” is a 2-bit line that contains the error output for overflow which leads from the Adder-Subtractor module to the Error-Handler module.

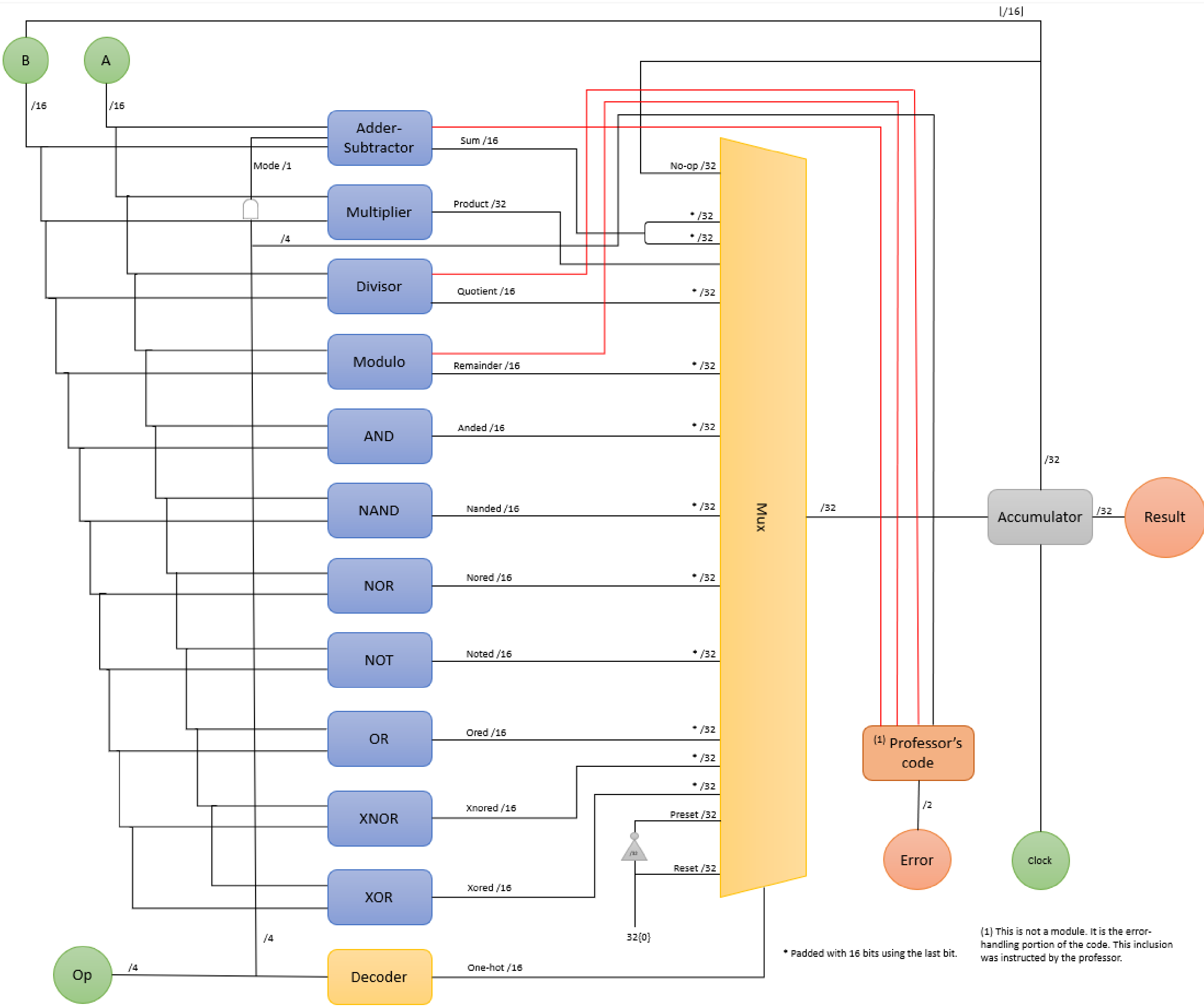
2.3.1.4 The List of Parts

- “AdderSubtractor” is the adder-subtractor component in the ALU that is capable of performing addition and subtraction of 16-bit binary numbers.
- “Multiplier” is the multiplier component in the ALU that is capable of performing multiplication on 16-bit binary numbers.
- “Divider” is the divisor component in the ALU that is capable of performing integer division on 16-bit binary numbers.
- “Modulus” is the modulo component in the ALU that is capable of performing the modulus operation on 16-bit binary numbers (finding the remainder).
- “Mux” is the multiplexor component used in the ALU that handles mutual exclusion among the channels of the ALU.
- “Decoder” is the decoder component used in the ALU that converts the operational code into a one-hot selection for use in the multiplexor.
- “And” is a 16-bit component in the ALU that is capable of performing AND operations on 16-bit binary numbers.

- “Nand” is a 16-bit component in the ALU that is capable of performing NAND operations on 16-bit binary numbers.
- “Nor” is a 16-bit component in the ALU that is capable of performing NOR operations on 16-bit binary numbers.
- “Not” is a 16-bit component in the ALU that is capable of performing NOT operations on 16-bit binary numbers.
- “Or” is a 16-bit component in the ALU that is capable of performing OR operations on 16-bit binary numbers.
- “Xnor” is a 16-bit component in the ALU that is capable of performing XNOR operations on 16-bit binary numbers.
- “Xor” is a 16-bit component in the ALU that is capable of performing XOR operations on 16-bit binary numbers.
- “Accumulator” is a memory register that is identified by a flip-flop with a multi-bit input and multi-bit output bus. It plays an important role in the execution of the instructions by storing an input or output operand.

2.4 The Top-Level Circuit Diagram

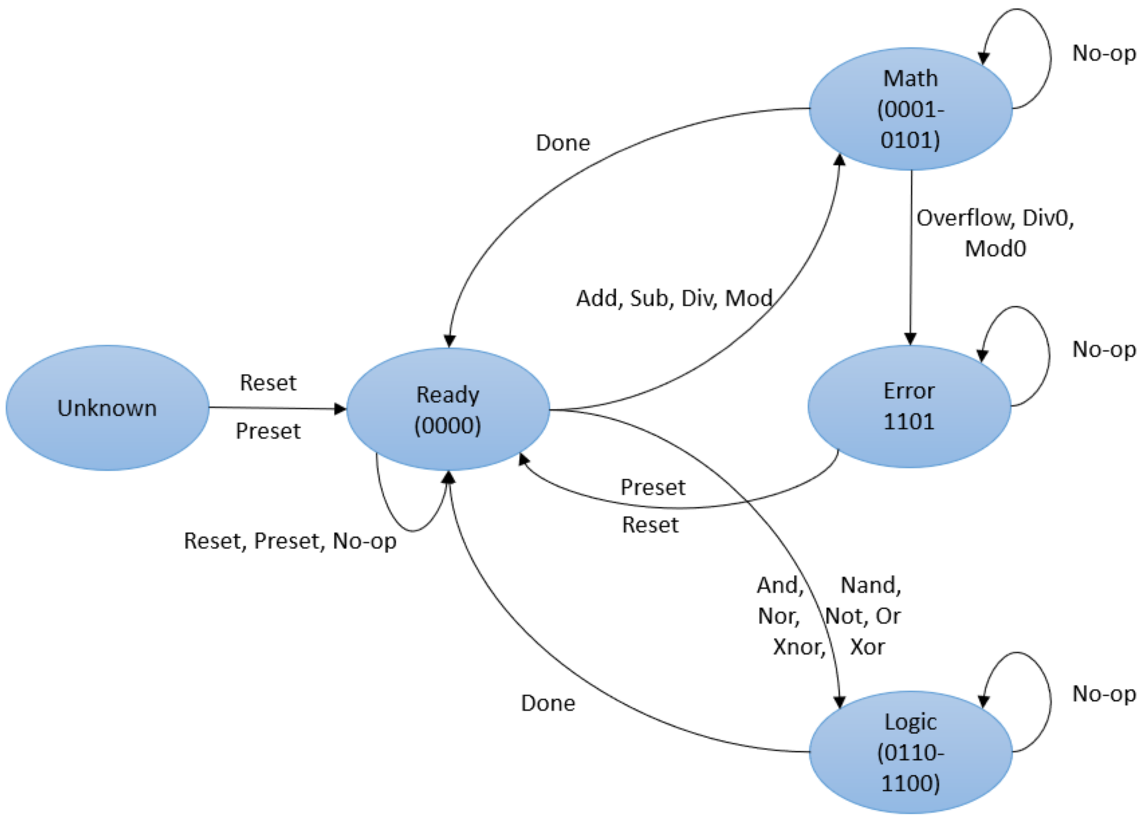
Note to grader: zoom in to see the image below in more detail. It will refine the image!



Opcode Table

0	No-op
1	Add
2	Sub
3	Mult
4	Div
5	Mod
6	And
7	Nand
8	Nor
9	Not
10	Or
11	Xnor
12	Xor
13	Reset
14	Preset

2.5 State Machine



State Table

Current State	Trigger	Next State
Unknown	Reset, Preset	Ready
Ready	Reset, Preset, No-op	Ready
Ready	Add, Sub, Div, Mod	Math
Ready	And, Nor, Xnor, Nand, Not, Or, Xor	Logic
Logic, Math	Done	Ready
Math	OFErr, Div0Err, Mod0Err	Error
Math	No-op	Math
Logic	No-op	Logic
Error	Reset, Preset	Ready
Error	Anything but Reset, Preset	Error

* If it receives a trigger other than those that take it out of its current state, it remains in its current state.