**Description of all inputs**

clk: Clock signal. Input to memory register (flip flop). Drives the flip flops.

inputA: 16-bit integer that feeds into modules

opcode: 4-bit operational code that determines which operation will be performed on the operands

- Opcode Definitions:

0000 – No-op (Feedback output from memory register)

0001 – Reset (Set all 0)

0010 – AND

0011 – OR

0100 – Addition (AddSub)

0101 – Subtraction (AddSub)

0110 – Multiplication (Mult)

0111 – Division (Div)

1000 – Modulo (Mod)

1001 – NOT

1010 – XOR

1011 – NAND

1100 – NOR

1101 – XNOR

1110 – Unknown

1111 – Preset (Set all 1)

**Description of all outputs**

outputNum:

32-bit integer output of the memory register.

err:

2-bit error code. Most significant bit represents divide by zero error and least significant bit represents overflow error.

**Description of all interfaces**

**Sequential Logic**

feedback\_input:

Feedback of memory register (flip flop). NOTE: 32 bit, but the lower 16 bits are fed into all modules except for NOT, which takes 32 bits. Also feeds 32 bits into No-Op channel of Multiplexor.

**Logic**

AndToMux:

32-bit. Connects And module to multiplexor.

OrToMux:

32-bit. Connects Or module to multiplexor.

NotToMux:

32-bit. Connects Not module to multiplexor.

XorToMux:

32-bit. Connects Xor module to multiplexor.

NandToMux:

32-bit. Connects Nand module to multiplexor.

NorToMux:

32-bit. Connects Nor module to multiplexor.

XnorToMux:

32-bit. Connects Xnor module to multiplexor.

**Arithmetic**

AddSubToMux:

32-bit. Connects Adder-subtractor to multiplexor addition and subtraction channels.

MultToMux:

32-bit. Connects Multiplier to multiplexor multiplier channel.

DivToMux:

32-bit. Connects Divisor to multiplexor divisor channel.

ModToMux:

32-bit. Connects Modulo to multiplexor modulo channel.

DecToMux:

16-bit. Connects decoder to multiplexor select channel. Contains a one hot encoding of the opcode.

DecToMux is also used to change mode in AddSub module: DecToMux[5] (sub bit) is passed to AddSub.

DecToMux is also used to check if AddSub, Div, or Mod were enabled when setting error bits.

OverflowFlag:

1-bit. Indicates that there is an overflow as a result of an adder-subtractor module operation.

AND’d with the result of DecToMux[4] (add) OR’d with DecToMux[5] (sub) to produce Overflow bit for err.

DivZero:

1-bit. Output of Div module indicating there was an attempt to divide by zero. OR’d together with ModZero to output DivByZeroFlag.

ModZero:

1-bit. Output of Mod module indicating there was an attempt to divide by zero. OR’d together with DivZero to output DivByZeroFlag.

DivByZeroFlag:

1-bit. Indicates whether there was an attempt to divide by zero.

AND’d with result of DecToMux[7] OR’d with DecToMux[8] to produce DivByZero bit for err.

Overflow:

1-bit. Set in least significant bit of err.

DivByZero:

1-bit. Set in most significant bit of err.

**Description of all parts**

**Sequential Logic**

ACC:

Accumulator. 32-bit Memory register (flip flop). Input is multiplexor output and clock signal. Allows 32-bit output to be stored in memory. Output feeds back to input to allow subsequent operations to be performed on the stored value.

**Logic**

And:

Performs AND on two 16-bit inputs. Lower 16 bits of feedback\_input AND inputA. 32-bit padded output feeds into multiplexor.

Or:

Performs OR on two 16-bit inputs. Lower 16 bits of feedback\_input OR inputA. 32-bit padded output feeds into multiplexor.

Not:

Performs NOT on all 32 bits of feedback\_input. 32-bit output feeds into multiplexor.

Xor:

Performs XOR on two 16-bit inputs. Lower 16 bits of feedback\_input XOR inputA. 32-bit padded output feeds into multiplexor.

Nand:

Performs NAND on two 16-bit inputs. Lower 16 bits of feedback\_input NAND inputA. 32-bit padded output feeds into multiplexor.

Nor:

Performs NOR on two 16-bit inputs. Lower 16 bits of feedback\_input NOR inputA. 32-bit padded output feeds into multiplexor.

Xnor:

Performs XNOR on two 16-bit inputs. Lower 16 bits of feedback\_input XNOR inputA. 32-bit padded output feeds into multiplexor.

**Arithmetic**

AddSub:

adder-subtractor. Performs addition or subtraction and detects overflow. Sum feeds into multiplexor. Error feeds into the least significant bit of error code. Lower 16 bits of feedback\_input +/- inputA.

Mult:

multiplier. Performs multiplication. Product feeds into multiplexor. Lower 16 bits of feedback\_input \* inputA.

Div:

divisor. Performs division and detects divide by 0. Quotient feeds into the multiplexor. Divide by 0 is OR’d with Modulo module’s divide by 0 and feeds into the most significant bit of error code. Lower 16 bits of feedback\_input / inputA.

Mod:

modulo. Performs modulo operation and detects divide by 0. Remainder feeds into a multiplexor. Divide by 0 is OR’d with the division module’s divide by 0 and feeds into the most significant bit of error code. Lower 16 bits of feedback\_input % inputA.

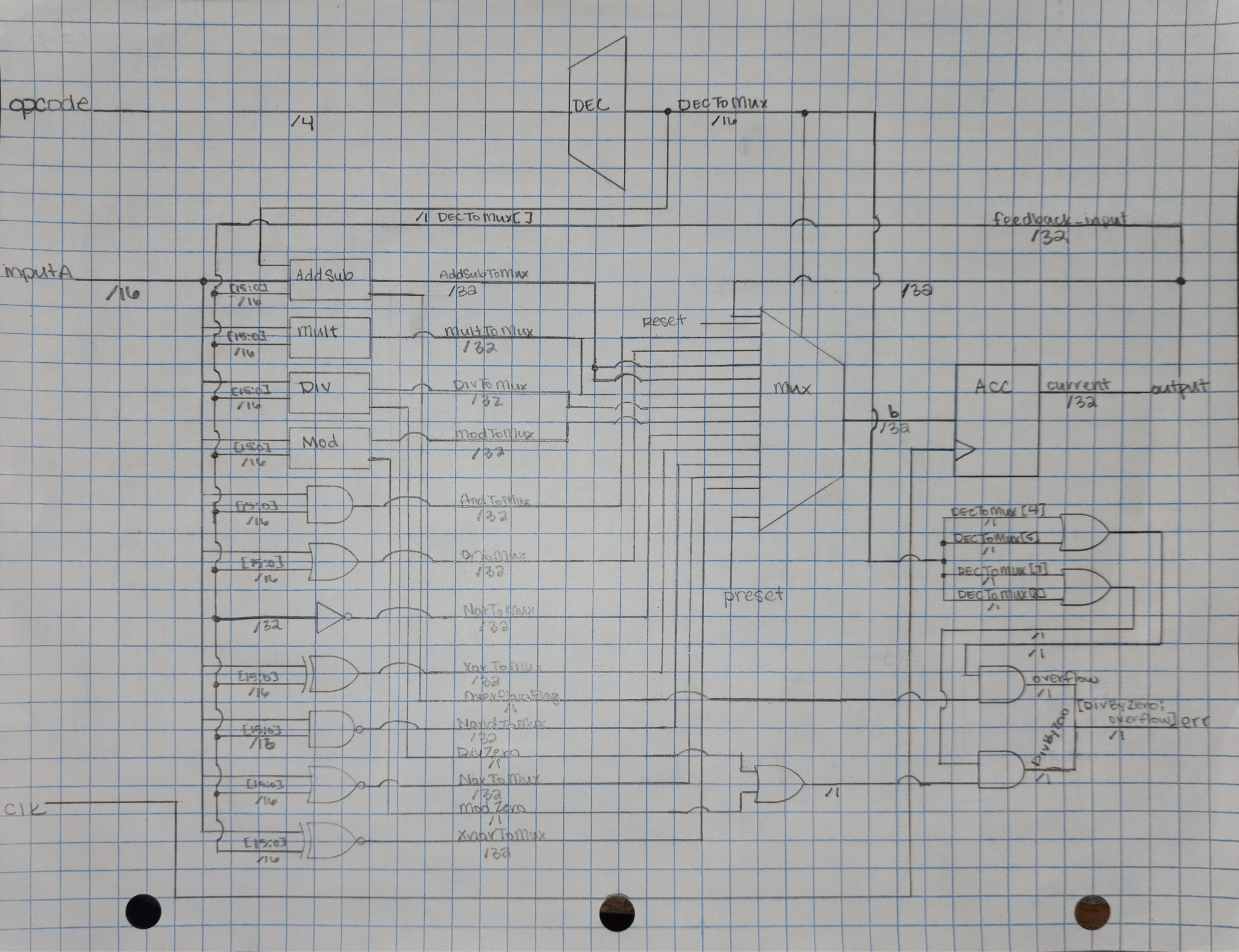
Mux:

16-channel, 32-bit Multiplexor. Outputs the result of the operation selected by the one-hot from the decoder.

Dec:

4-to-16 Decoder. Converts opcode to one-hot to feed into multiplexor which determines the operation.

**Top Level Circuit Diagram**

****

**State Diagram**

