Modules

ALU (Breadboard) – all parts connected together to for the ALU.

Input- 16-bit input A, 16-bit input B, 1-bit input CLK, 4-bit input opcode

The ALU takes advantage of behavioral style coding in some instances, but is mostly structural.

The ALU is made up of the following parts.

Testbench – Main module, runs clock and gives input to the ALU.

Each part listed below has the following properties:

i. The name of the component

ii. The inputs and their bit-size

iii. The outputs and their bit-size

iv. The interfaces and their bit-size

v. Any controls and their bit-size. (Such as Multiplexor Select)

vi. A one-sentence description if necessary.

i.) **AND**

ii.) inputs: i - 16 bit, j - 16 bit

iii.) output:(AND): f1 - 16 bit

iv.) none

v.) no controls

vi.) standard 16 bit AND gate.

i.) **DIVIDER**

ii.) inputs: a - 16 bit

b - 16 bit

iii.) outputs: (quotient): q - 16 bit

(remainder): r - 16 bit

iv.) none

v.) no controls

vi.) Divides a by b and gives remainder

i.) **ADDER_SUBTRACTOR**

ii.) inputs: a - 16 bit, b - 16 bit

iii.) output: s - 16 bit, cout - 1 bit, overflow – 1 bit

iv.) none

v.) Controls: Cin - 1 bit

vi.) Standard 16 bit adder subtractor that is controlled by a Cin

value and returns in 2s complement form during subtraction.

Overflow value will be set to 1 when overflow occurs.

i.) **MULTIPLEXOR**

ii.) input:

Ch0 – Ch14 32 bits

iii.) output: b - 32 bits

iv.) none

v.) s - 16 bit one hot select

vi.) 16 to 1 multiplexor with 32 bit channels. Selects output channel based on value of s.

i.) **MULTIPLIER**

ii.) input: a - 16 bits, b - 16 bits

iii.) output: plow -16 bits phigh -16 bits

iv.) none

v.) no controls

vi.) Multiplies a and b returns in the form of two 16 bit values

corresponding to the highest 16 and lowest 16 bits

i.) **NAND**

ii.) inputs: i - 16 bit

j - 16 bit

iii.) output:(NAND): f1 - 16 bit

iv.) none

v.) no controls

vi.) standard 16 bit NAND gate.

i.) **NOT**

ii.) inputs: a - 16 bit

iii.) output: r – 16 bit

iv.) none

v.) no controls

vi.) standard 16 bit NOT gate.

i.) **NOR**

ii.) inputs: i - 16 bit

j - 16 bit

iii.) output:(NOR): f1 - 16 bit

iv.) none

v.) no controls

vi.) standard 16 bit NOR gate.

i.) **OR**

ii.) inputs: i - 16 bit

j - 16 bit

iii.) output:(OR): f1 - 16 bit

iv.) none

v.) no controls

vi.) standard 16 bit OR gate.

i.) **XNOR**

ii.) inputs: i - 16 bit

j - 16 bit

iii.) output:(XNOR): f1 - 16 bit

iv.) none

v.) no controls

vi.) standard 16 bit XNOR gate.

i.) **XOR**

ii.) inputs: a - 16 bit

b - 16 bit

iii.) output:(XOR) r - 16 bits

iv.) n/a

v.) n/a

vi.) standard 16 bit XOR gate

i.) **DECODER**

ii.) inputs: a- 4-bit binary

iii.) output: b- 16-bit one-hot

iv.) n/a

v.) n/a

vi.) converts a 4-bit binary number to a 16-bit one-hot