

Final Lab Specifications

Truth Table

R-Type Instructions					
	Add	Sub	Or	Mult	Div
RegDst	1	1	1	1	1
ALUsrc	0	0	0	0	0
MemToReg	0	0	0	0	0
RegWrite	1	1	1	1	1
MemWrite	0	0	0	0	0
PCSrc	0	0	0	0	0
ExtOp	X	X	0	X	X
ALUop	0000	0001	0010	0100	0101

I-Type Instructions					
	Addi	Ori	Lw	Sw	Beq
RegDst	0	0	0	X	X
ALUsrc	1	1	1	1	1
MemToReg	0	0	1	X	X
RegWrite	1	1	1	0	0
MemWrite	0	0	0	1	0
PCSrc	0	0	0	0	1
ExtOp	1	0	1	1	X
ALUop	0000	0010	0000	0000	0000

Half Adder Ports: I used two inputs one for the first operand of the adder and other one is the second operand of the adder. I also used carry which is the bit that is carried in from the next less significant stage.

Full Adder Ports: I used two inputs one for the first operand of the adder and other one is the second operand of the adder. I also used cin which is the bit that is carried in from the next less significant stage. I used cout for the output which is bit that is carried over from an addition when necessary and sum is the last bit of the result.

Instruction Memory: I used 2 cins, (cin1 and cin2) which are the first and second operand of the adder for the 16 bits. I also had op which is the bit that carried in from the less significant stage. For the outputs, I had neg which is the carry output that carried in from addition, sum (sum of the

two inputs, cin1 and cin2), overflow which checks for overflow errors and finally zero for two input addition.

Data Memory: I used two operands. The first operand of the adder and other one is the second operand of the adder. Then, I used obit which is the bit that carried from the less significant stage. I had carry for the bit output from addition, sum for the inputs addition, overflow which checks for overflow errors and finally zero for two input addition.

ALU: The inputs are 32-bit registers (RS, RT) to compute operations, 16-bit immediate to compute immediate operations, 32-bit MDR, clock and operation. The outputs are 32-bit register RD and 3 flags: zero, negative, overflow. I also used the other codes as a component to get the waveform.

Extender: I am using this to extend the most significant bit for IMM. It is used to extend the 16bit register data to 32bit register so I can use the 32-bit add/sub.