MAKE A COPY OF THIS SHEET	Instruction Metadata				ROM Input [1]		Control Signals								ROM Output [2]
							addi is provided as an example								
Instruction	Type	Opcode	Funct3	Funct7	Binary	Decimal	**RegWEn**	**ImmSel**	**BrUn**	**ASel**	**BSel**	**ALUSel**	**MemRW**	**WBSel**	Hex
							1 binary digit	3 binary digits	1 binary digit	1 binary digit	1 binary digit	4 binary digits	1 binary digit	2 binary digits	
add rd, rs1, rs2	R	0b0110011	0b000	0b0000000	0b000000	0	1	111	0	0	0	0000	0	01	100F
mul rd, rs1, rs2			0b000	0b0000001	0b000001	1	1	111	0	0	0	1000	0	01	140F
sub rd, rs1, rs2			0b000	0b0100000	0b000010	2	1	111	0	0	0	1100	0	01	160F
sll rd, rs1, rs2			0b001	0b0000000	0b000011	3	1	111	0	0	0	0001	0	01	108F
mulh rd, rs1, rs2			0b001	0b0000001	0b000100	4	1	111	0	0	0	1001	0	01	148F
mulhu rd, rs1, rs2			0b011	0b0000001	0b000101	5	1	111	0	0	0	1011	0	01	158F
slt rd, rs1, rs2			0b010	0b0000000	0b000110	6	1	111	0	0	0	0010	0	01	110F
xor rd, rs1, rs2			0b100	0b0000000	0b000111	7	1	111	0	0	0	0100	0	01	120F
srl rd, rs1, rs2			0b101	0b0000000	0b001000	8	1	111	0	0	0	0101	0	01	128F
sra rd, rs1, rs2			0b101	0b0100000	0b001001	9	1	111	0	0	0	1101	0	01	168F
or rd, rs1, rs2			0b110	0b0000000	0b001010	10	1	111	0	0	0	0110	0	01	130F
and rd, rs1, rs2			0b111	0b0000000	0b001011	11	1	111	0	0	0	0111	0	01	138F
lb rd, offset(rs1)	I	0b0000011	0b000		0b001100	12	1	000	0	0	1	0000	0	00	0041
lh rd, offset(rs1)			0b001		0b001101	13	1	000	0	0	1	0000	0	00	0041
lw rd, offset(rs1)			0b010		0b001110	14	1	000	0	0	1	0000	0	00	0041
addi rd, rs1, imm		0b0010011	0b000		0b001111	15	1	000 [3]	0 [4]	0 [5]	1 [6]	0000	0	01 [7]	1041
slli rd, rs1, imm			0b001	0b0000000	0b010000	16	1	000	0	0	1	0001	0	01	10C1
slti rd, rs1, imm			0b010		0b010001	17	1	000	0	0	1	0010	0	01	1141
xori rd, rs1, imm			0b100		0b010010	18	1	000	0	0	1	0100	0	01	1241
srli rd, rs1, imm			0b101	0b0000000	0b010011	19	1	000	0	0	1	0101	0	01	12C1
srai rd, rs1, imm			0b101	0b0100000	0b010100	20	1	000	0	0	1	1101	0	01	16C1
ori rd, rs1, imm			0b110		0b010101	21	1	000	0	0	1	0110	0	01	1341
andi rd, rs1, imm			0b111		0b010110	22	1	000	0	0	1	0111	0	01	13C1
sb rs2, offset(rs1)	S	0b0100011	0b000		0b010111	23	0	001	0	0	1	0000	1	00	0842
sh rs2, offset(rs1)			0b001		0b011000	24	0	001	0	0	1	0000	1	00	0842
sw rs2, offset(rs1)			0b010		0b011001	25	0	001	0	0	1	0000	1	00	0842
beq rs1, rs2, offset	B	0b1100011	0b000		0b011010	26	0	010	0	1	1	0000	0	00	0064
bne rs1, rs2, offset			0b001		0b011011	27	0	010	0	1	1	0000	0	00	0064
blt rs1, rs2, offset			0b100		0b011100	28	0	010	0	1	1	0000	0	00	0064
bge rs1, rs2, offset			0b101		0b011101	29	0	010	0	1	1	0000	0	00	0064
bltu rs1, rs2, offset			0b110		0b011110	30	0	010	1	1	1	0000	0	00	0074
bgeu rs1, rs2, offset			0b111		0b011111	31	0	010	1	1	1	0000	0	00	0074
auipc rd, offset	U	0b0010111			0b100000	32	1	011	0	1	1	0000	0	01	1067
lui rd, offset		0b0110111			0b100001	33	1	011	0	1	1	1111	0	01	17E7
jal rd, imm	J	0b1101111			0b100010	34	1	100	0	1	1	0000	0	10	2069
jalr rd, rs1, imm	I	0b1100111	0b000		0b100011	35	1	000	0	0	1	0000	0	10	2041

[1] This is the value that will be passed into the ROM

[2] This is the value that will be outputted from the ROM. It's all the control signals concatenated together.

[3] This value is provided as an example. Based on your design for the immediate generator, you may need to modify this value to generate the correct immediate value

[4] This value actually doesn't matter because the addi instruction never uses the branch comparator. However, you must fill out every cell so the control bits line up properly

[5] This value is provided as an example. Based on your design for the A MUX, you may need to modify this value to generate the correct immediate value

[6] This value is provided as an example. Based on your design for the B MUX, you may need to modify this value to generate the correct immediate value

[7] This value is provided as an example. Based on your design for the Writeback MUX, you may need to modify this value to generate the correct immediate value