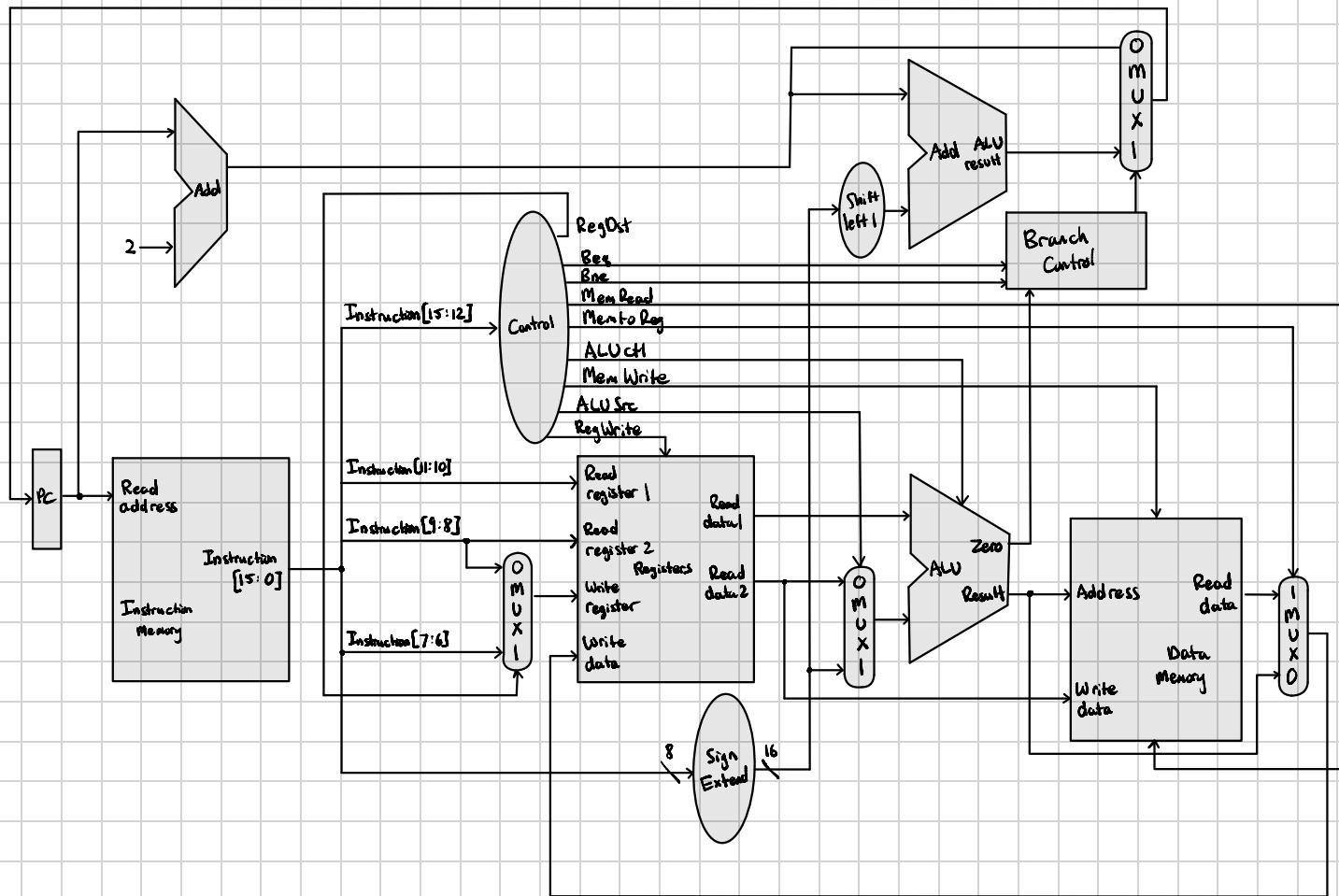


Objective: Building complete single-cycle datapath for implementing all instructions (16-bit MIPS).



Reg Array Implementation:

- PC
- Instruction memory
- Data memory

Behavioral:

- Main Control
- Register File
- Sign Extend/Adder

Gate Level:

- ALU
- Multiplexers
- Branch Control

* Logic diagrams for half-adders & full-adders are not shown.

Instruction Set Architecture/Control Unit Truth Table:

Instr	Opcode	Reg Dat	ALUsrc	MemtoReg	RegWrite	MemWrite	Beg	Bne	ALUct1
add	0000	1	0	0	1	0	0	0	0010
sub	0001	1	0	0	1	0	0	0	0110
and	0010	1	0	0	1	0	0	0	0000
or	0011	1	0	0	1	0	0	0	0001
nor	0100	1	0	0	1	0	0	0	1100
nand	0101	1	0	0	1	0	0	0	1101
slt	0110	1	0	0	1	0	0	0	0111
addi	0111	0	1	0	1	0	0	0	0010
lw	1000	0	1	1	1	0	0	0	0010
sw	1001	0	1	0	0	1	0	0	0010
beq	1010	0	0	0	0	0	1	0	0110
bne	1011	0	0	0	0	0	0	1	0110
j	1100	0	0	0	0	0	1	0	0110

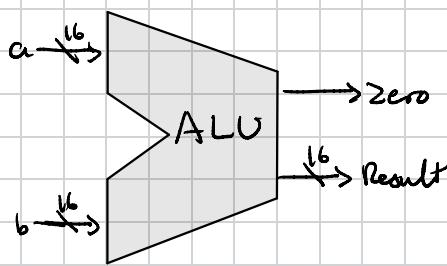
Instruction Formats:

R-format: add, sub, and, or, nor, nand, slt

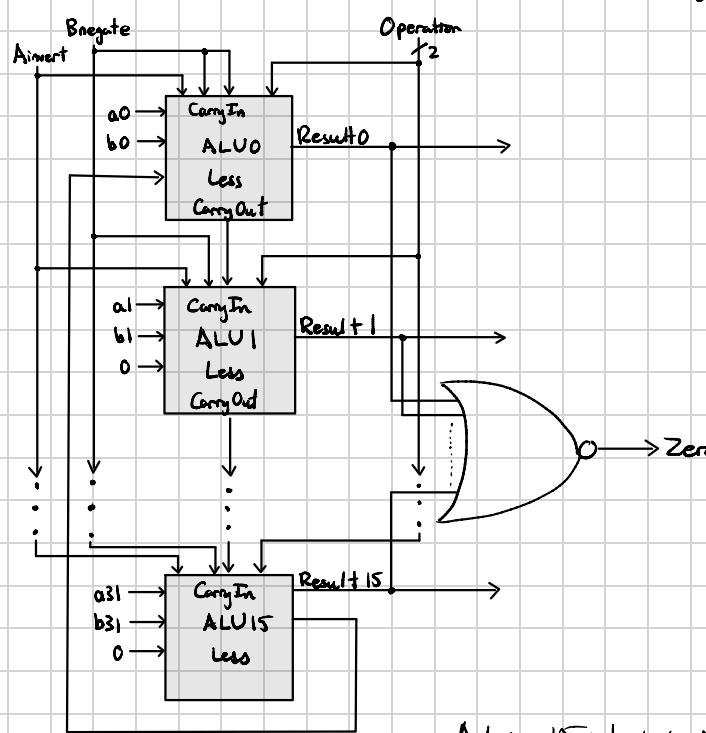
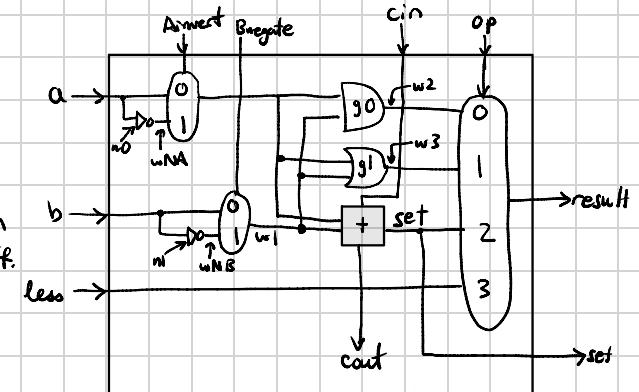
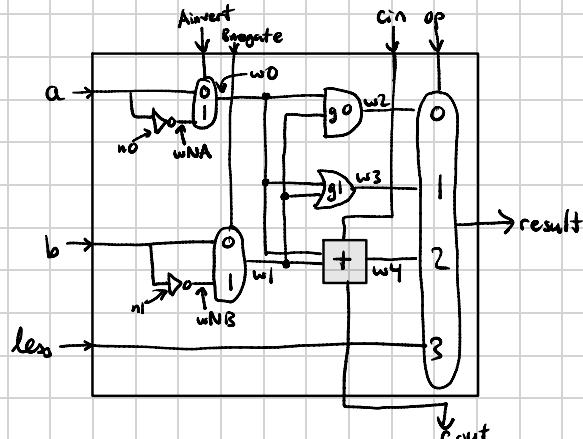
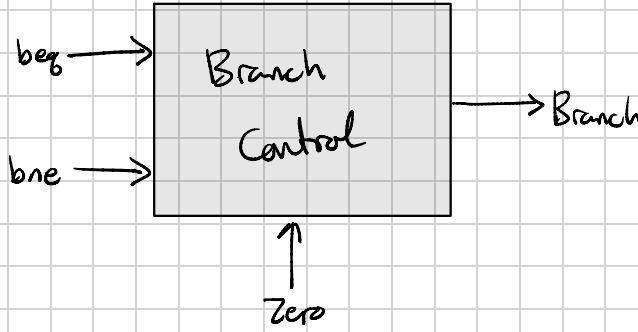
op	rs	rt	rd	unused
4	2	2	2	6

I-format: addi, lw, sw, beq, bne, j

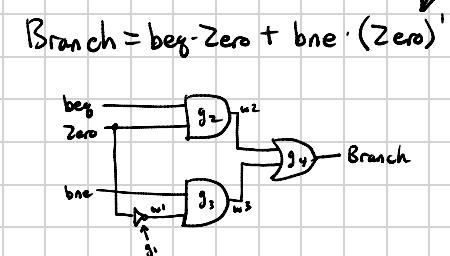
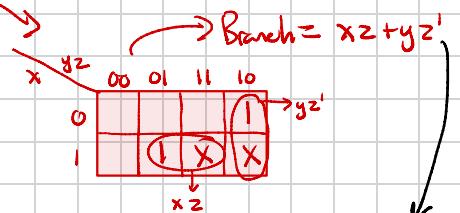
op	rs	rt	addr/value
4	2	2	8

ALU Control Lines:16-bit ALU: *No Overflow*

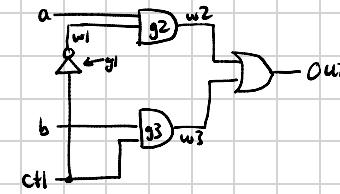
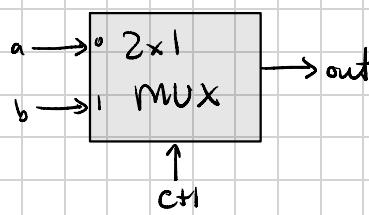
Anvert ALU<16>	Bnegate ALU<16>	Operation ALU<16>[1:0]	Inst.
0	0	10	add
0	0	10	sub
0	0	00	and
0	0	01	or
-1	0	00	nor
0	0	01	nand
0	0	11	slt
0	0	10	addi
0	0	10	lw
0	0	10	sw
0	0	10	beg
0	0	10	bne
0	-1	10	j

ALU0 to ALU14: 1-bit ALU, cascadedALU 15: 1-bit ALU, last of the cascaded chain of ALUs. Slightly diff.Branch Control Module: Aids in executing branch instructions, controlling the multiplexer.

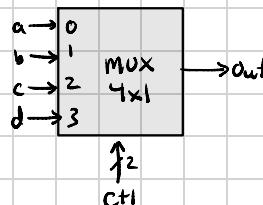
x	y	z	A	Branch
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	0	0
-1	0	0	0	0
-1	0	1	-1	X
-1	1	0	-1	X
-1	1	1	-1	X



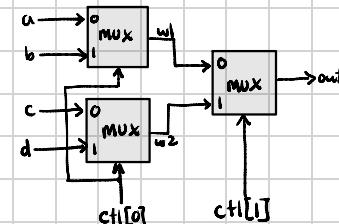
1-bit 2x1 Multiplexer:



1-bit 4x1 Multiplexer:

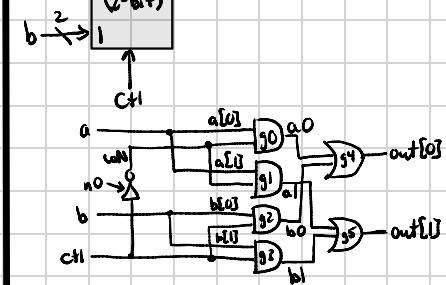
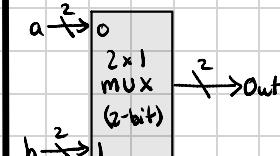


Implemented by cascading 2x1 multiplexers.



2-bit 2x1 Multiplexer:

Multiplexes 2 inputs of 2-bit data.



16-bit 2x1 Multiplexer: Multiplexes 2 inputs of 16-bit data.

→ Implemented by cascading 8 2-bit 2x1 multiplexers, as shown below.

