
G
O
O
D
O
O
O
D

9.4 Main, ALU, and PC ctrl

ctrl	input	output
Main	OPcode	ctrl signals

ALUctrl opcode+funct ALUop

Pcctrl opcode, zero? PCOP

Op	RegDst	RegWr	ExtOp	ALUSrc	MemRd	MemWr	WBdata
R-type	1 = Rd	1	X	0 = BusB	0	0	0 = ALU
ADDI	0 = Rt	1	1 = sign	1 = Imm	0	0	0 = ALU
SLTI	0 = Rt	1	1 = sign	1 = Imm	0	0	0 = ALU
ANDI	0 = Rt	1	0 = zero	1 = Imm	0	0	0 = ALU
ORI	0 = Rt	1	0 = zero	1 = Imm	0	0	0 = ALU
XORI	0 = Rt	1	0 = zero	1 = Imm	0	0	0 = ALU
LW	0 = Rt	1	1 = sign	1 = Imm	1	0	1 = Mem
SW	X	0	1 = sign	1 = Imm	0	1	X
BEQ	X	0	1 = sign	0 = BusB	0	0	X
BNE	X	0	1 = sign	0 = BusB	0	0	X
J	X	0	X	X	0	0	X

How do we build this circ (not K-map)

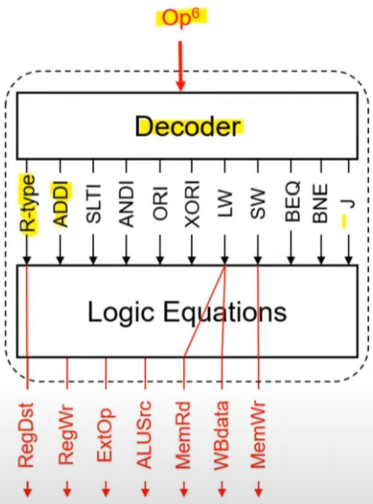
Decoder + OR gates
to preconfigure signal vals

Think sheet cheat:

- More 0s than 1s: connect Minterms to an OR gate
- More 1s than 0s: connect Maxterms to a NAND gate

Signals for opcoder

RegDst	=	R-type
RegWrite	=	$\overline{(SW + BEQ + BNE + J)}$
ExtOp	=	$\overline{(ANDI + ORI + XORI)}$
ALUSrc	=	$\overline{(R\text{-}type + BEQ + BNE)}$
MemRd	=	LW
MemWr	=	SW
WBdata	=	LW



While some differing functions use the same opcode, we must utilize the Funct field to adjust the data path accordingly.

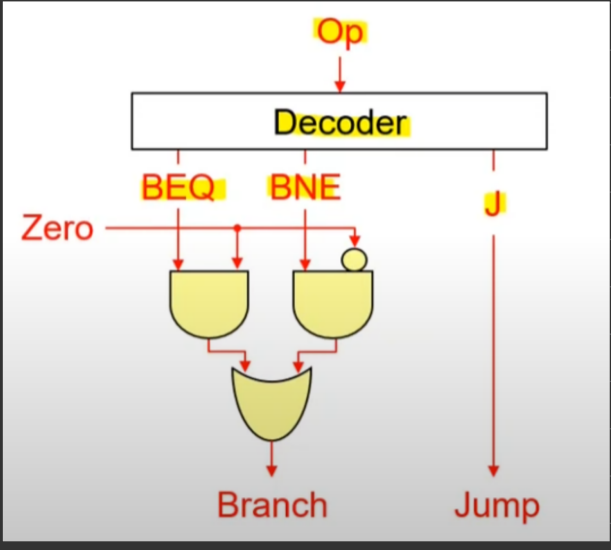
ALU ctrl TT

Op	funct	ALUOp	4-bit Coding
R-type	AND	AND	0001
R-type	OR	OR	0010
R-type	XOR	XOR	0011
R-type	ADD	ADD	0100
R-type	SUB	SUB	0101
R-type	SLT	SLT	0110
ADDI	X	ADD	0100
SLTI	X	SLT	0110
ANDI	X	AND	0001
ORI	X	OR	0010
XORI	X	XOR	0011
LW	X	ADD	0100
SW	X	ADD	0100
BEQ	X	SUB	0101
BNE	X	SUB	0101
J	X	X	X

Immediate -> Funct == 0
because we don't have
Funct Field For imm & j instructions

PC Ctrl

Op	Zero flag	PCSrc
R-type	X	0 = Increment PC
J	X	1 = Jump Target Address
BEQ	0	0 = Increment PC
BEQ	1	2 = Branch Target Address
BNE	0	2 = Branch Target Address
BNE	1	0 = Increment PC
Other than Jump or Branch	X	0 = Increment PC



Jal has two stages
link: $r7 = Pc + 1
jump: Pc = $rs
new signal jal

- So at WB, we need a 3rd input Pc
- We need to hard wire Busw at 111 to write in $r7, so RegDst is 0 Rt, but Rt is hard wired to 111

1- at instruction to RegFile wire Mux is Jal

2- At WB Mux, WB = 2 when opcode = jal

4 of 4 rows shown

regDist[1..0]	PortB	PortW	
0 0	0	1	R-Type AND NOT REV
0 1	1	1	REV
1 0	-	0	Immediate / Lw / BEq
1 1	-	-	

What are our main Ctrl signals do we have?

1. isJal
2. RegDist [1:0]
3. ALUSrc
4. ExtoP
5. RegWrite
6. WBData [1:0]
7. MemRd
8. MemWr

■ Dont care

OPCode	RtR7Rt	RegDst	ALUSrc	ExtoP	RegW	WB	MemRd	MemW
Arithmetic R	00	00	00	0	1	00	0	0
Logical R	00	00	00	0	1	00	0	0
Shift R	00	00	00	0	1	00	0	0
Rev R	00	01	00	0	1	00	0	0
ADDI	00	10	01	0	1	00	0	0
ANDI	00	10	01	0	1	00	0	0
ORI	00	10	01	0	1	00	0	0
XORI	00	10	01	0	1	00	0	0
LW	00	10	01	1	1	01	1	0
SW	00	00	01	1	0	00	0	1
Beq	00	00	00	1	0	00	0	0
BNE	00	00	00	1	0	00	0	0
J	00	00	00	0	0	00	0	0
Jal	01	10	00	0	1	10	0	0
lui	10	10	10	0	1	00	0	0

ALU ctrl

■ Dont care

opcode	Funct	ALUop in		ALUop out
Arithmetic R	Add	0	0	0 0 0
Arithmetic R	Sub	0	1	0 0 0
Arithmetic R	SLt	0	0	0 0 1
Arithmetic R	Sltu	0	1	0 0 1
Logical R	AND	0	0	0 1 0
Logical R	OR	0	1	0 1 0
Logical R	XOR	1	0	0 1 0
Logical R	NOR	1	1	0 1 0
ShiFt R	Sll	0	0	0 1 1
ShiFt R	Srl	0	1	0 1 1
ShiFt R	SrA	1	0	0 1 1
ShiFt R	RoR	1	1	0 1 1
RevR	RevL	0	0	1 0 0
RevR	RevH	0	1	1 0 0
RevR	RevA	1	0	1 0 0
RevR	JR	0	2	1 0 0
ADDI	X	0	2	0 0 0
ANDI	X	0	0	0 1 0
ORI	X	0	1	0 1 0
XORI	X	1	0	0 1 0
LW	X	0	0	0 0 0
SW	X	0	0	0 0 0
BEQ	X	0	1	0 0 0
BNE	X	0	1	0 0 0
J	X	0	0	0 0 0
Jal	X	0	0	0 0 0
LUI	X	0	0	0 0 0

Jal jump PC jV

00000001
00010010

PC Ctl V

	OPcode	Funct	Zero	PCSrc
3 = 00011	RetV R	11	X	11
17 = 10001	jal	X X	X	10
16 = 10000	j	X X	X	10
14 = 01110	BEQ	X X	1	01
.	BEQ	X X	0	00
15 = 01111	BNE	X X	1	00
	BNE	X X	0	01
	NOR	X X	X	00