instruction	icode	ifun	address	destination		Mem. read (1: read)	Mem. write (1: write)	Mem. addr valE / valA	Mem. data valA / valP
irmovl imm, rA	0000	0000	rA	F	imm	0	0	X	X
rmovl rA, D(rB)	0001	0000	rA	rB	D	0	0	X	X
mrmovl ld rA, <addr>	0010	0000	<addr>	rA		1	0	valE	X
rmmovl st rA, <addr>	0011	0000	rA	<addr>		0	1	valE	valA
OPl rA, rB	0100	fn	rA	rB		0	0	X	X
jmp <addr>	0101	0000	<addr>			0	0	X	X
jz <addr>	0110	fn				0	0	X	X
halt	0111	0000				0	0	X	X

valM ← RAM[addr]

RAM[addr] ← valA

N/A

N/A

valE if icode = 0010 or 0011

valA if icode = 0011

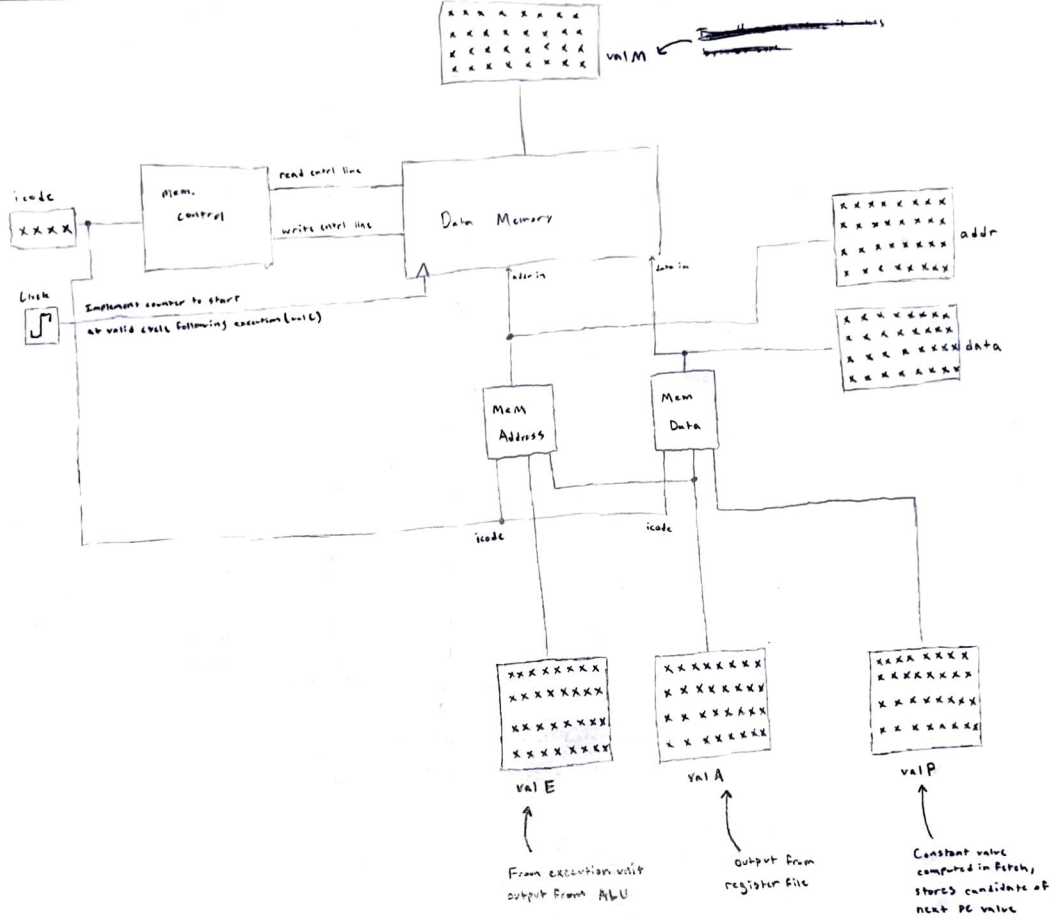
Read control signal
1 if icode = 0010

Write control signal
1 if icode = 0011

3

valC on cycle 3
valC → valE on cycle 4

4 → 5


valM
Mem. control
read control line
write control line
Data Memory
icode
Clock
Implement counter to start
at valid event following execution (cycle 6)
addr in
data in
addr
data
Mem Address
Mem Data
icode
icode
valE
valA
valP
From execution unit output from ALU
output from register file
Constant value computed in Fetch; stores candidate of next PC value

instruction	icode	ifun	output ALU A 1: valA	output ALU B 1: valB	output ALU FUN. 1: Fn
irmovl imm,rA	0000	0000	valC	0 (32-bit)	0000
rmmovl rA,D(rB)	0001	0000	valA	0	0000
mrmovl D(rB),rA	0010	0000	valC	0	0000
OPl rA, rB	0100	Fn	valA	valB	ifun=Fn
Jmp <addr>	0101	0000	valC	0	ifun=Fn
Jxx <addr>	0110	Fn	valC	0	ifun=Fn
halt	0111	0000	valC	0	0000

valC if icode = 0000 + 0010 + 0011 + 0101 + 0110 + 0111

valA if icode = 0001 + 0100

valB if icode = 0100

0 otherwise

ifun/fn if icode = 0100 + 0101 + 0110

0 otherwise

instruction	icode	ifun	dstE output	dstM output	srcA output	srcB output
			1: rA, 0: 1111/F	1: rA, 0: 1111/F	1: rA, 0: 1111/F	1: rA, 0: 1111/F
irmovl imm, rA	0000	0000	0000	1111	1111	1111
rrmovl rA, D(rB)	0001	0000	0000	1111	0000	1111
ld rA, <addr>	0010	0000	0000	0000	1111	1111
st rA, <addr>	0011	0000	1111	1111	0000	1111
OPl rA, rB	0100	fn	1111	1111	0000	0000
Jmp <addr>	0101	0000	1111	1111	1111	1111
Jz <addr>	0110	fn	1111	1111	1111	1111
halt	0111	0000	1111	1111	1111	1111

0000 if icode = 0100, 0xF/1111 otherwise

0000 if icode = 0001 + 0011 + 0100, 0xF/1111 otherwise

0000 if icode = 0010, 0xF/1111 otherwise

0000 if icode = 0000 + 0001 + 0010

0xF/1111 otherwise

instruction	icode	ifun	cnd	Output New PC
irmovl imm, rA	0000	0000	x	valP
rmovl rA, D(rB)	0001	0000	x	valP
ld rA, <addr>	0010	0000	x	valP
st rA, <addr>	0011	0000	x	valP
OPl rA, rB	0100	fn	x	valP
Jmp <addr>	0101	0000	x	valP
Jz <addr>	0110	fn	1	valC
			0	valP
halt	0111	0000	x	valP

valC if icode = 0110
and cond = 1
otherwise valP