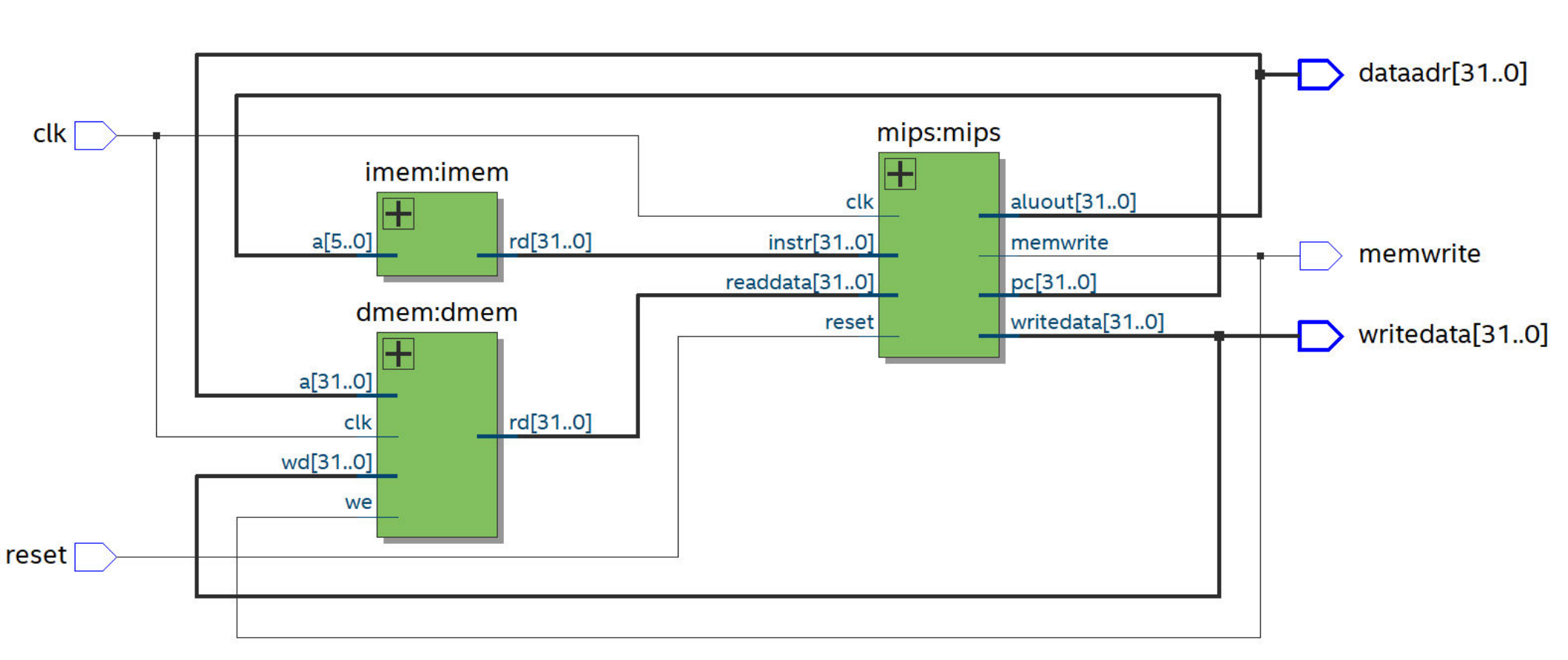
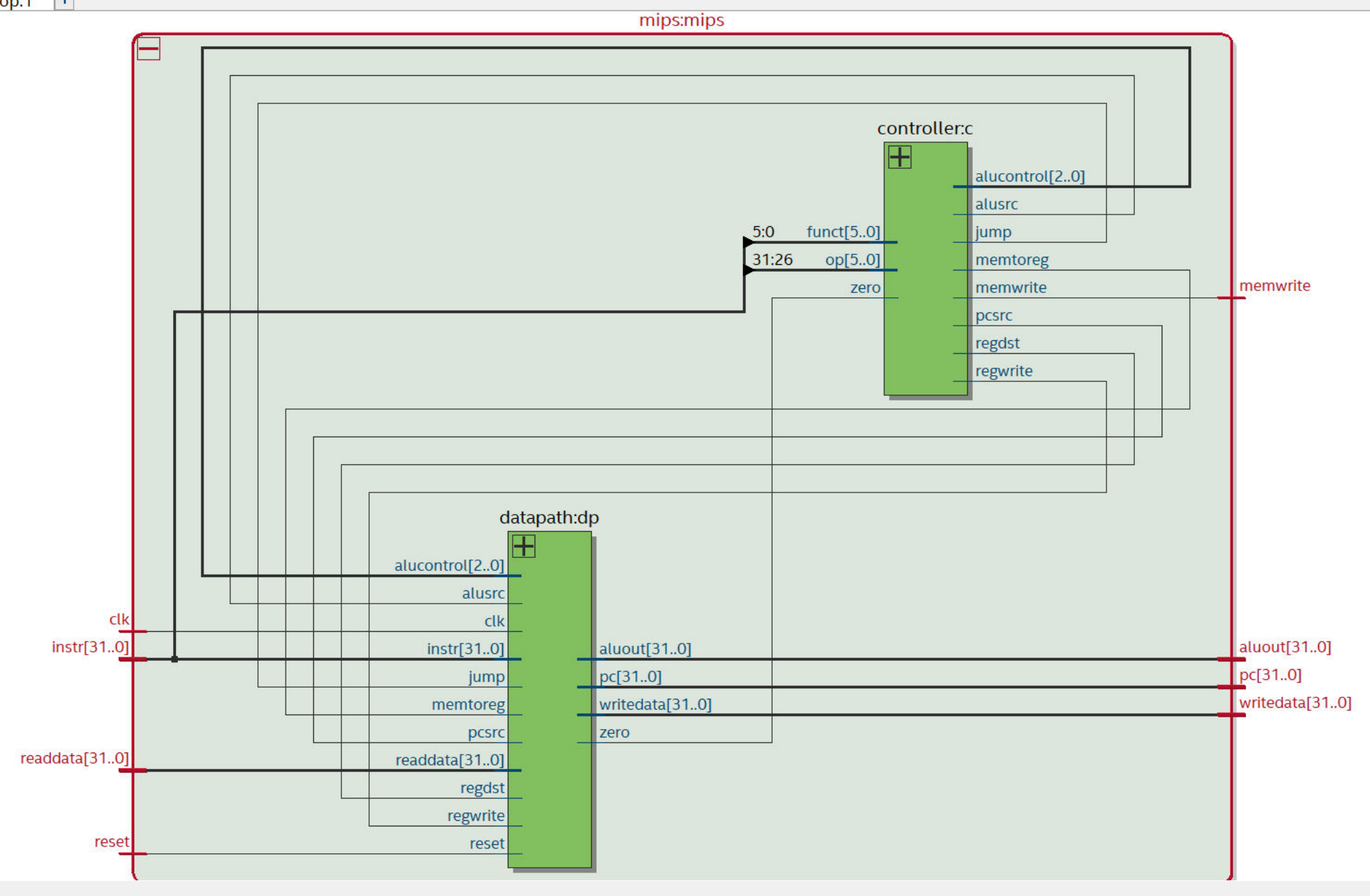
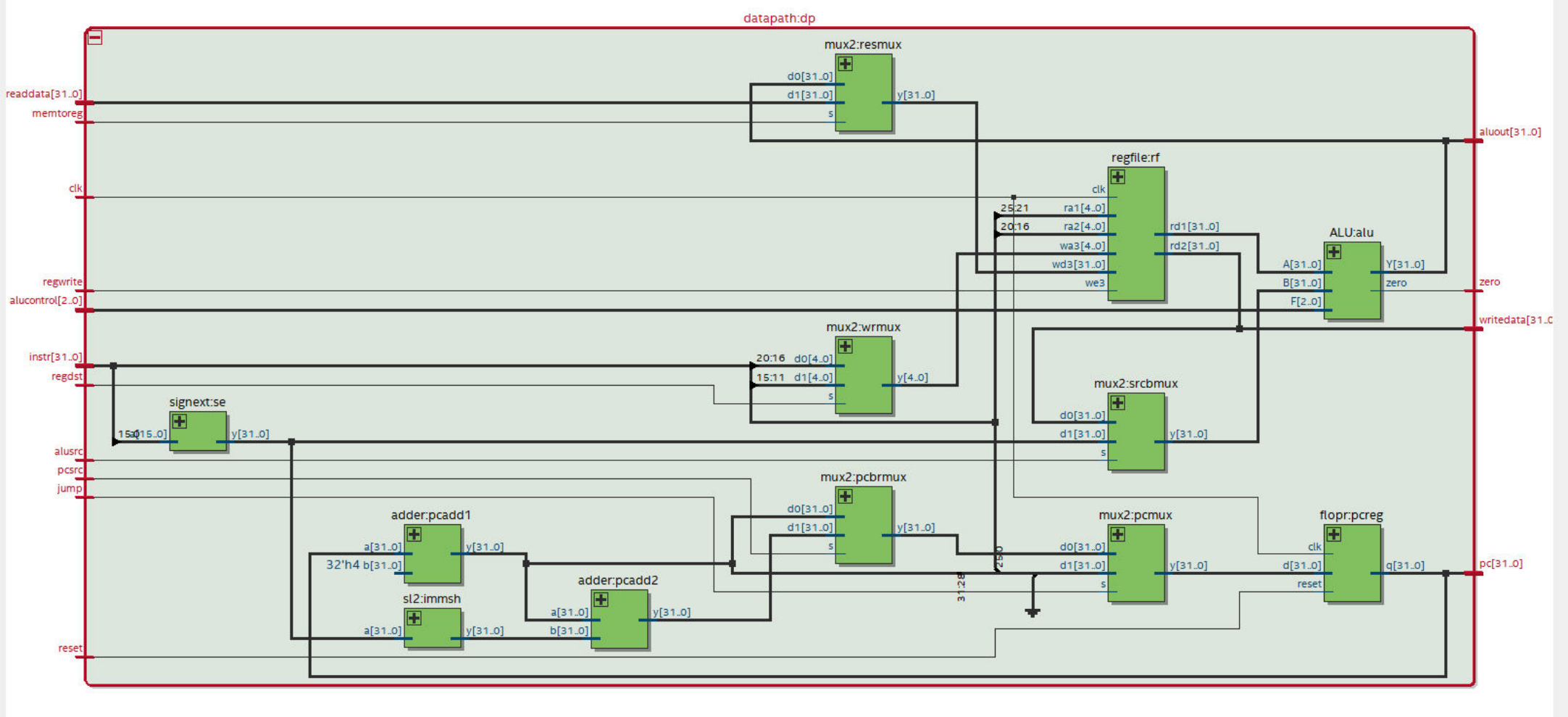
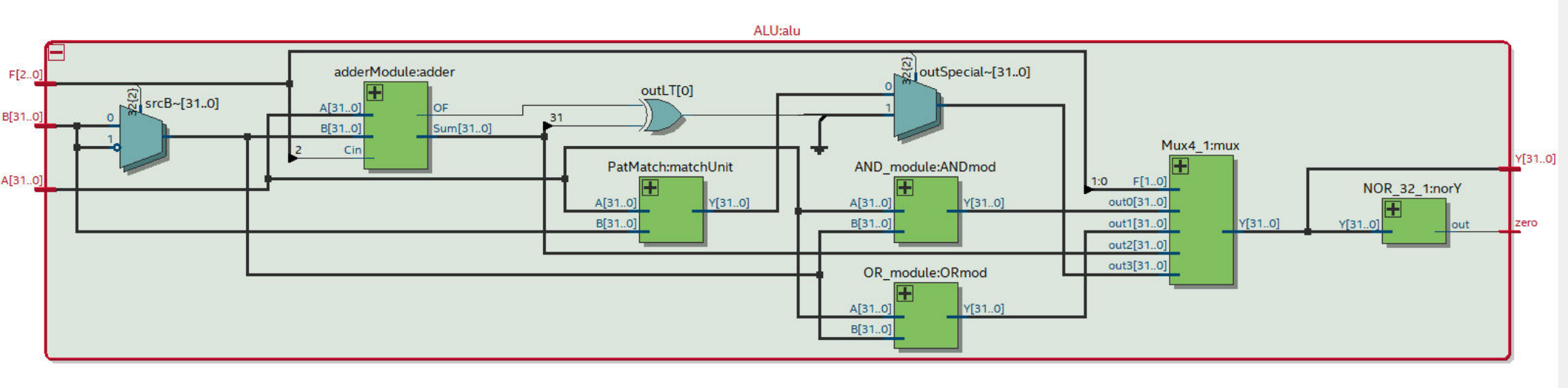
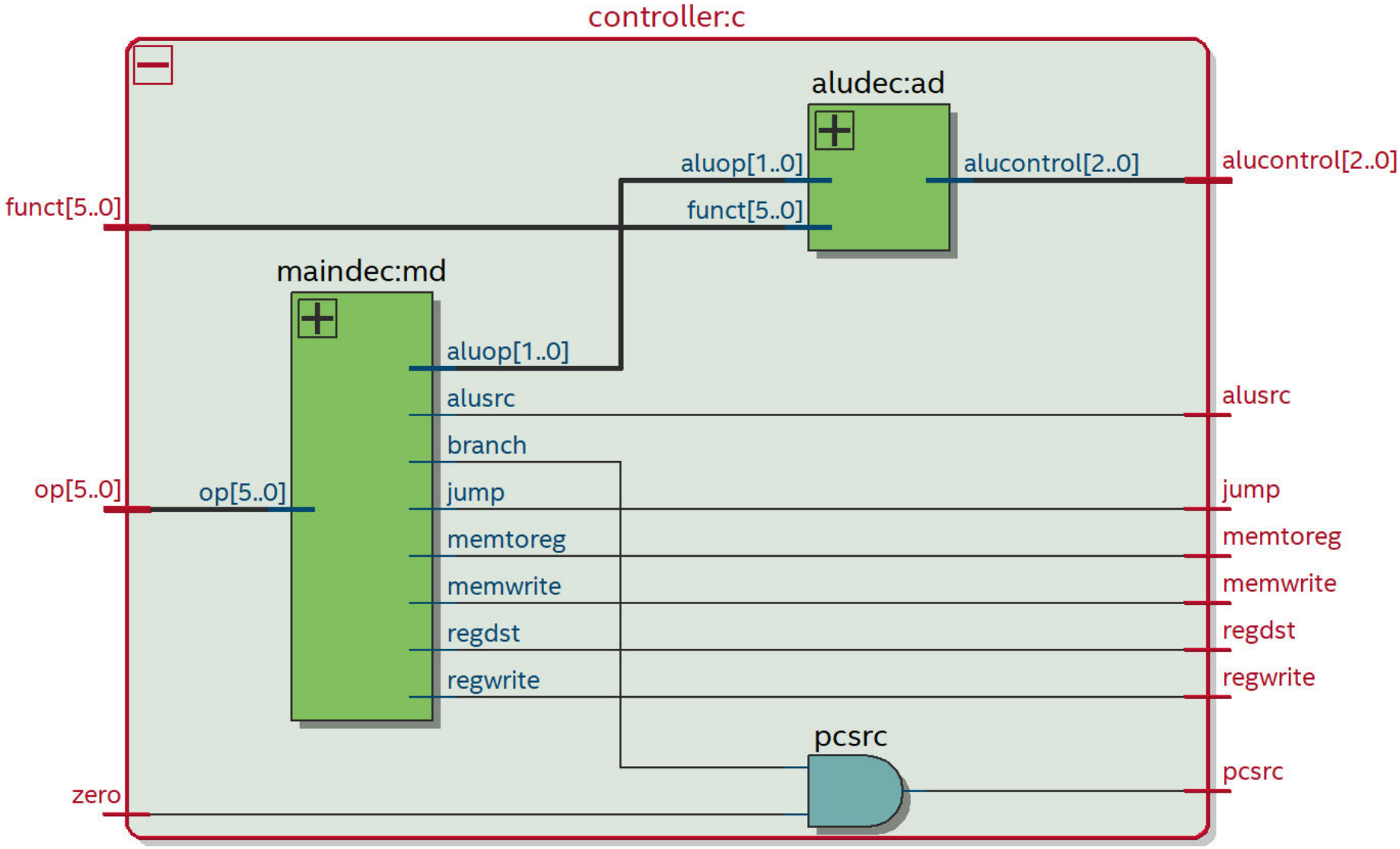
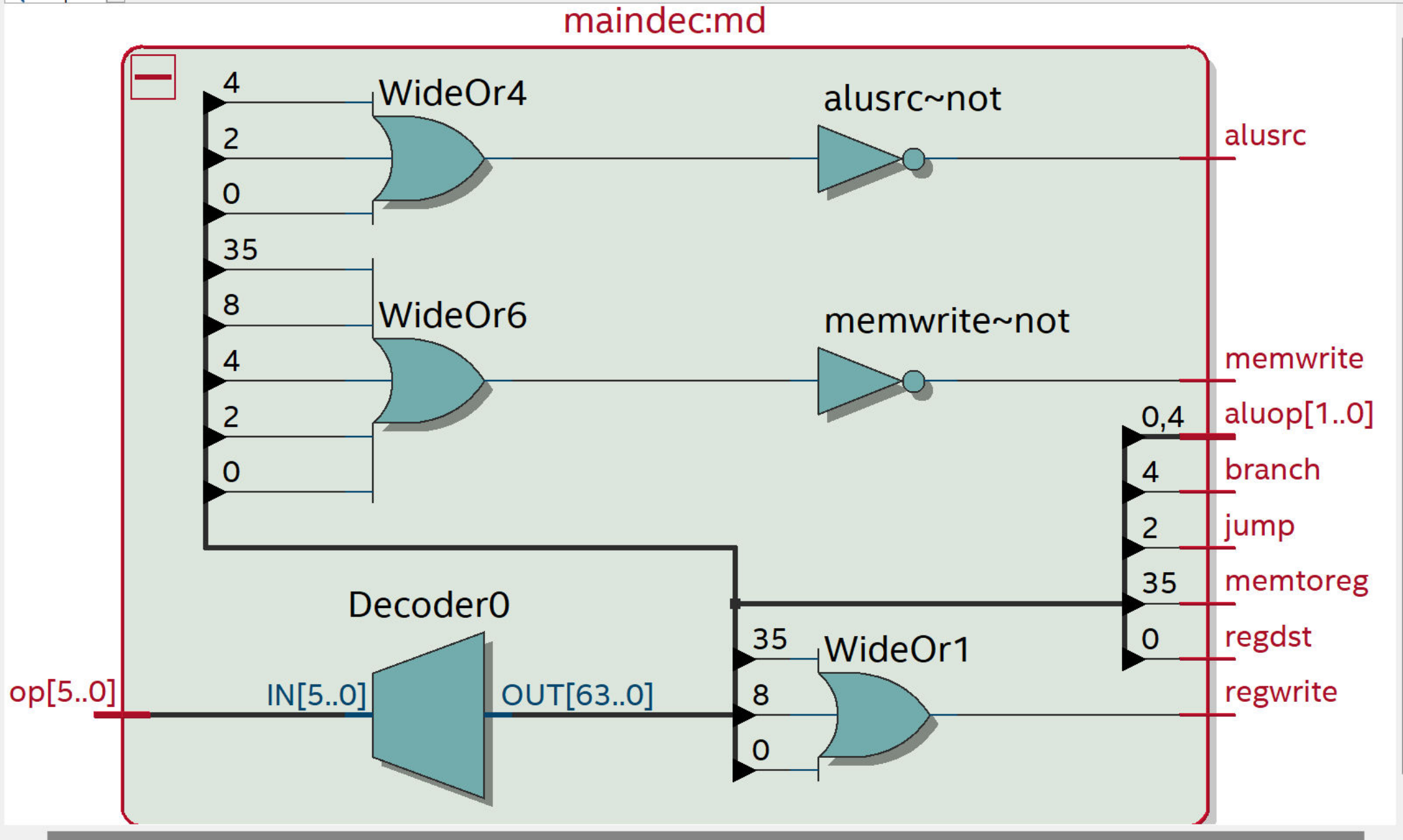

clk
reset
imem:imem
a[5..0]
rd[31..0]
dmem:dmem
a[31..0]
clk
wd[31..0]
we
rd[31..0]
mips:mips
clk
instr[31..0]
readdata[31..0]
reset
aluout[31..0]
memwrite
pc[31..0]
writedata[31..0]
dataadr[31..0]
memwrite
writedata[31..0]

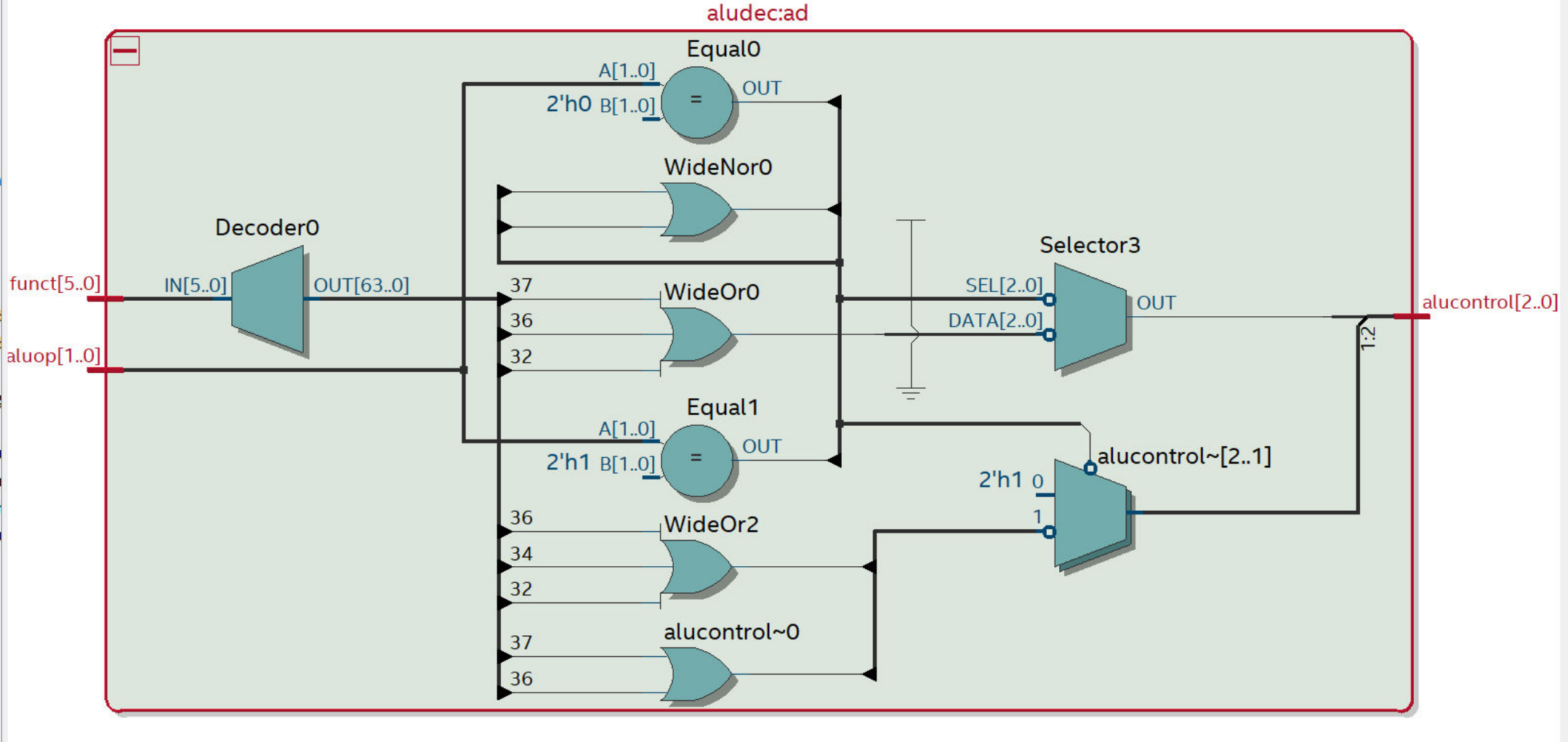

mips:mips
controller:c
alucontrol[2..0]
alusrc
jump
memtoreg
memwrite
pcsrc
regdst
regwrite
5:0
funct[5..0]
31:26
op[5..0]
zero
memwrite
datapath:dp
alucontrol[2..0]
alusrc
clk
instr[31..0]
jump
memtoreg
pcsrc
readdata[31..0]
regdst
regwrite
reset
aluout[31..0]
pc[31..0]
writedata[31..0]
zero
clk
instr[31..0]
readdata[31..0]
reset
aluout[31..0]
pc[31..0]
writedata[31..0]


datapath:dp
mux2:resmux
d0[31..0]
d1[31..0]
s
y[31..0]
readdata[31..0]
memtoreg
aluout[31..0]
clk
regfile:rf
clk
ra1[4..0]
ra2[4..0]
wa3[4..0]
wd3[31..0]
we3
rd1[31..0]
rd2[31..0]
25:21
20:16
ALU:alu
A[31..0]
B[31..0]
F[2..0]
Y[31..0]
zero
regwrite
alucontrol[2..0]
mux2:wrmux
20:16 d0[4..0]
15:11 d1[4..0]
s
y[4..0]
writedata[31..0]
instr[31..0]
regdst
mux2:srcbmux
d0[31..0]
d1[31..0]
s
y[31..0]
signext:se
15:0
y[31..0]
alusrc
pcsrc
jump
mux2:pcbrmux
d0[31..0]
d1[31..0]
s
y[31..0]
adder:pcadd1
a[31..0]
32'h4 b[31..0]
y[31..0]
mux2:pcmux
d0[31..0]
d1[31..0]
s
y[31..0]
31:28
25:0
flopr:pcreg
clk
d[31..0]
reset
q[31..0]
pc[31..0]
adder:pcadd2
a[31..0]
b[31..0]
y[31..0]
sl2:immsh
a[31..0]
y[31..0]
reset


ALU:alu
F[2..0]
B[31..0]
A[31..0]
srcB~[31..0]
adderModule:adder
A[31..0]
B[31..0]
Cin
OF
Sum[31..0]
outLT[0]
31
outSpecial~[31..0]
PatMatch:matchUnit
A[31..0]
B[31..0]
Y[31..0]
AND_module:ANDmod
OR_module:ORmod
Mux4_1:mux
1:0
F[1..0]
out0[31..0]
out1[31..0]
out2[31..0]
out3[31..0]
NOR_32_1:norY
out
Y[31..0]
zero


controller:c
aludec:ad
aluop[1..0]
funct[5..0]
alucontrol[2..0]
alucontrol[2..0]
funct[5..0]
maindec:md
aluop[1..0]
alusrc
branch
op[5..0]
op[5..0]
jump
memtoreg
memwrite
regdst
regwrite
alusrc
jump
memtoreg
memwrite
regdst
regwrite
pcsrc
zero
pcsrc


maindec:md
4
2
0
WideOr4
alusrc~not
alusrc
35
8
4
2
0
WideOr6
memwrite~not
memwrite
0,4
aluop[1..0]
4
branch
2
jump
35
memtoreg
0
regdst
regwrite
Decoder0
op[5..0]
IN[5..0]
OUT[63..0]
35
8
0
WideOr1


aludec:ad
Equal0
A[1..0]
2'h0
B[1..0]
=
OUT
WideNor0
Decoder0
funct[5..0]
aluop[1..0]
IN[5..0]
OUT[63..0]
37
36
32
WideOr0
Selector3
SEL[2..0]
DATA[2..0]
OUT
alucontrol[2..0]
1:2
Equal1
A[1..0]
2'h1
B[1..0]
=
OUT
alucontrol~[2..1]
2'h1
0
1
36
34
32
WideOr2
37
36
alucontrol~0