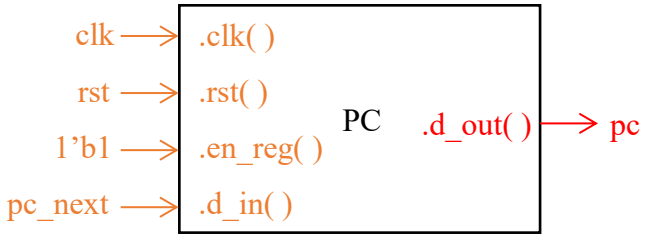
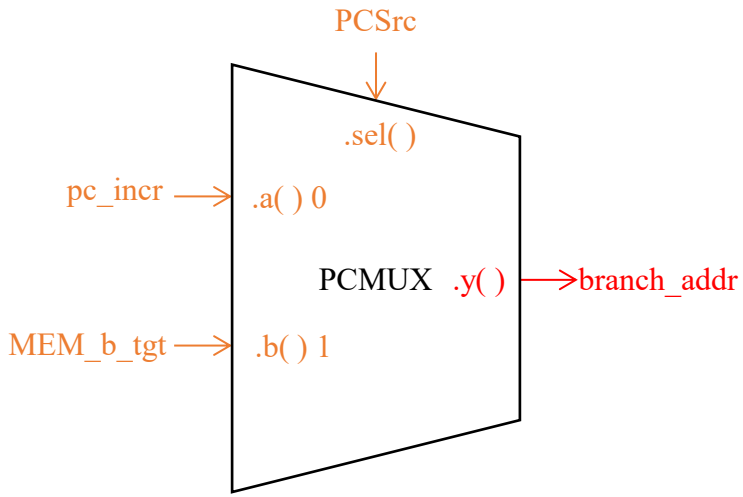
index: Final Project 線路架構圖

PC: module reg32

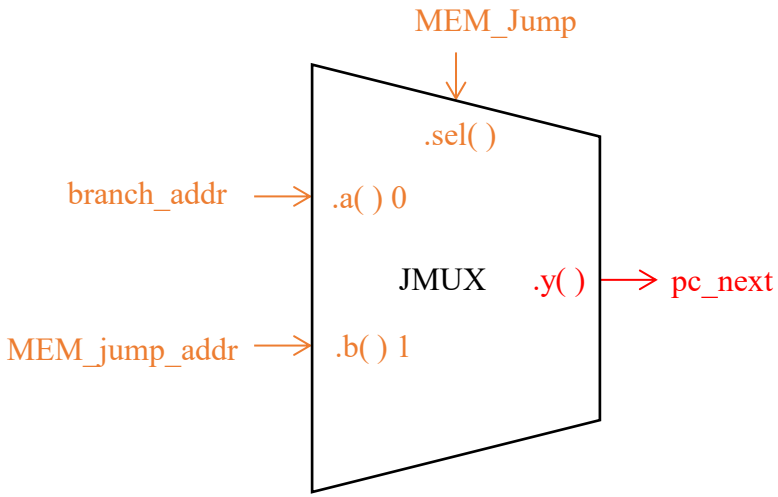


en_reg: 除了reset，其他都enable

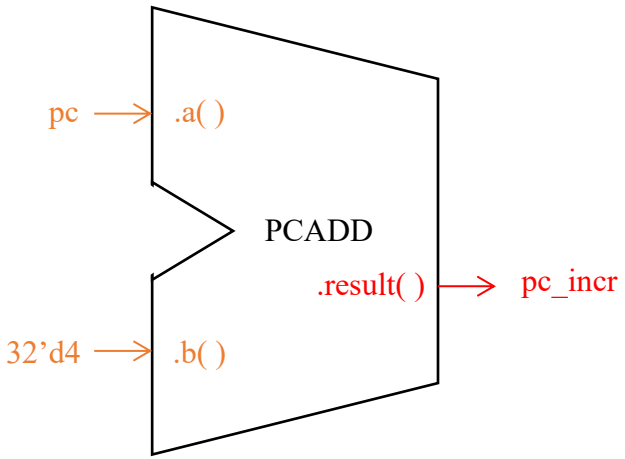
PCMUX: module mux2



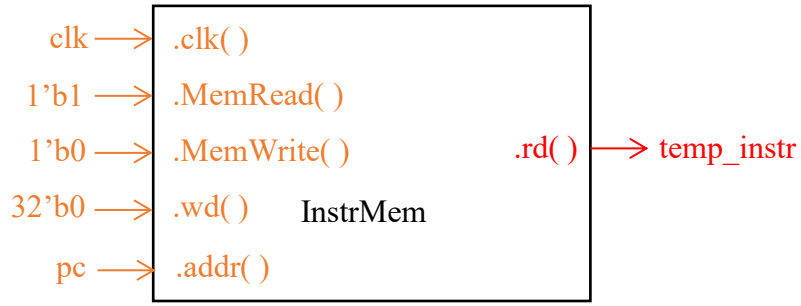
JMUX: module mux2



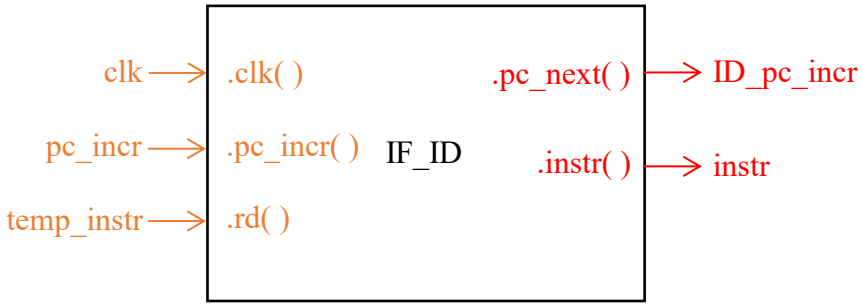
PCADD: module add32



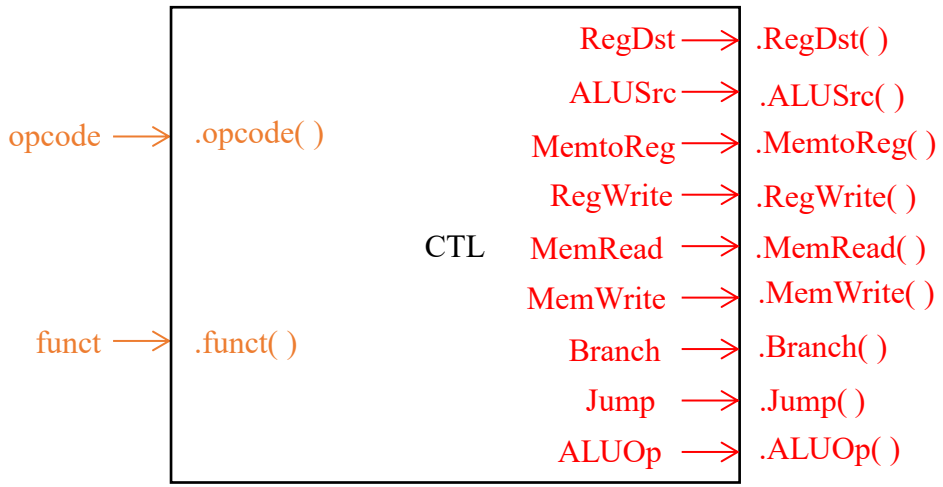
InstrMem: module memory



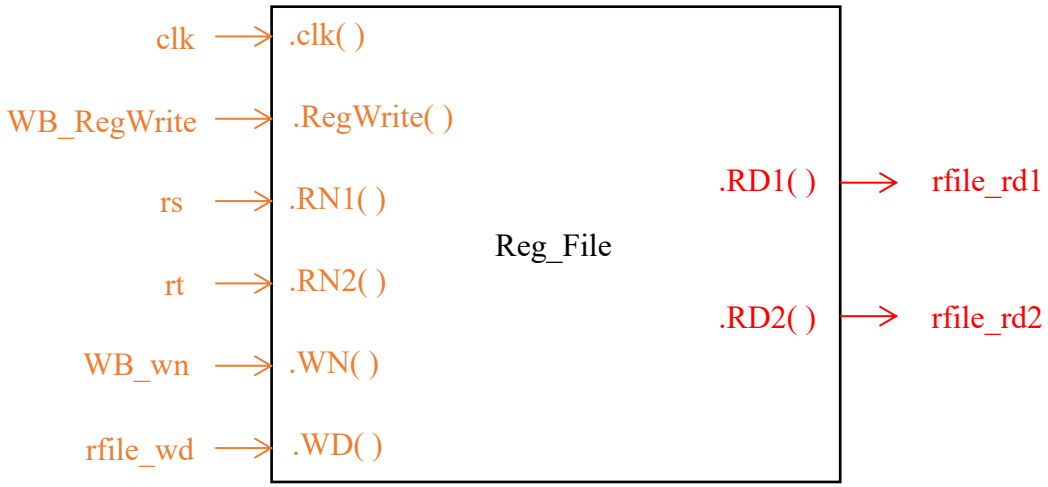
IF_ID: module IF_ID



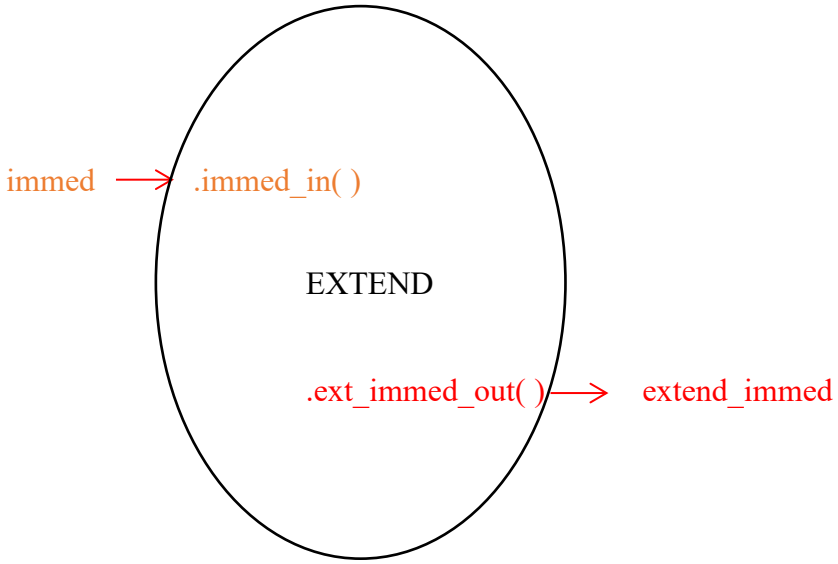
CTL: module control_pipeline



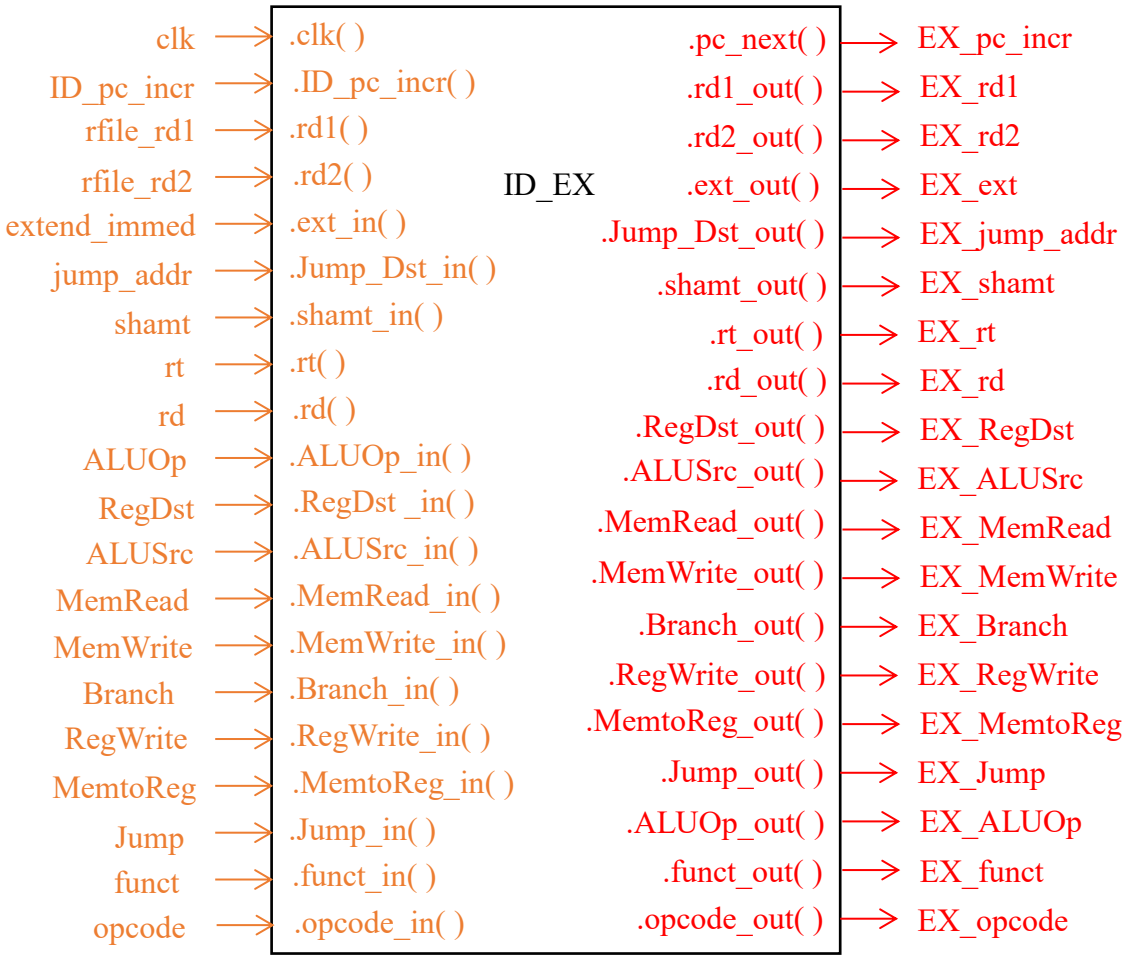
Reg_File: module reg_file



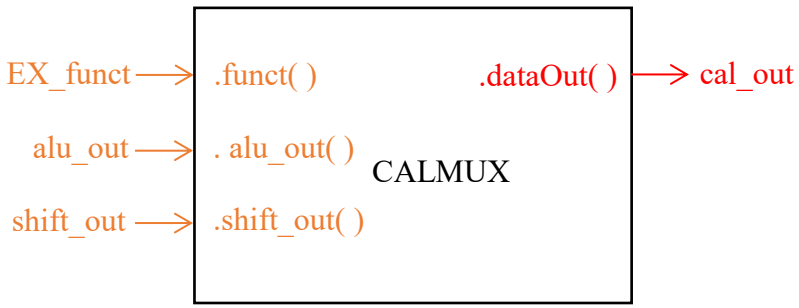
SignExt: module sign_extend



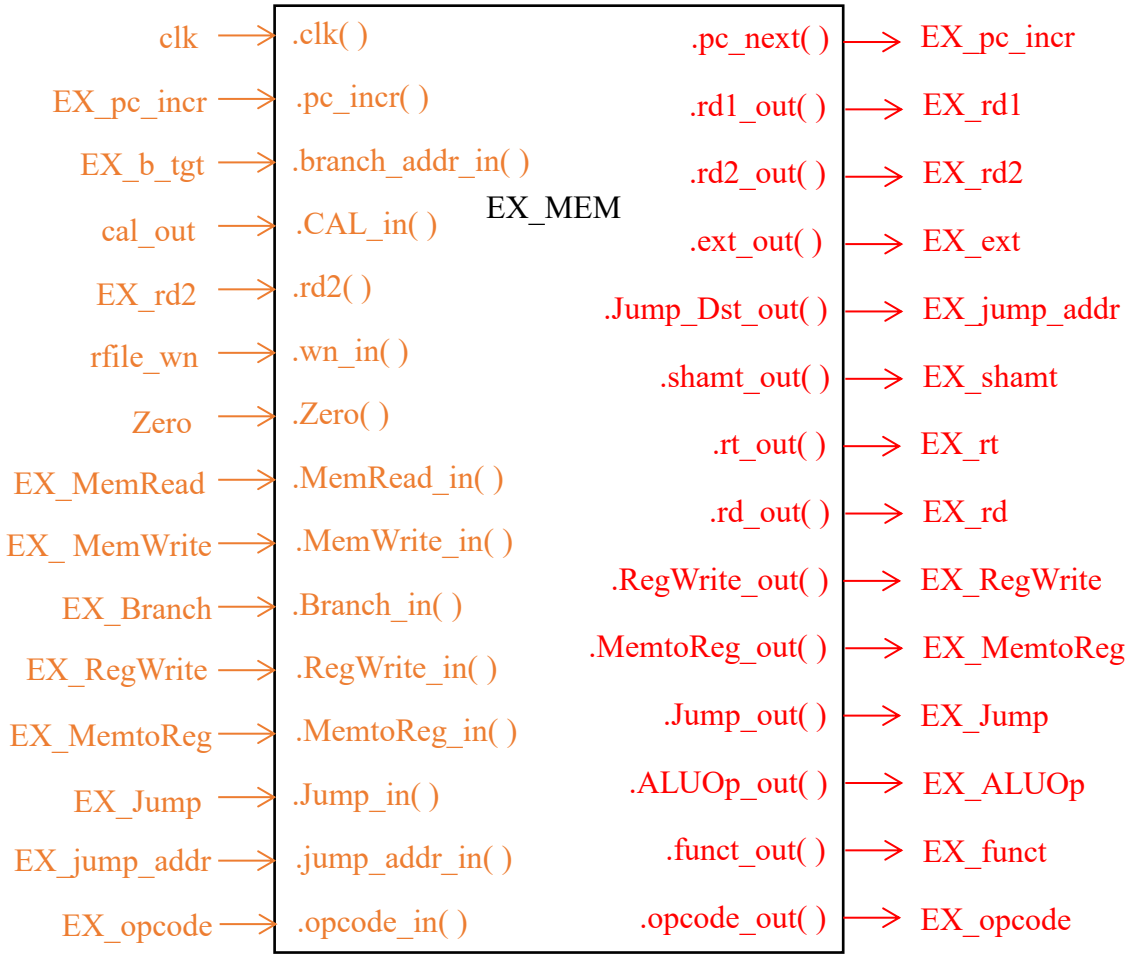
ID_EX: module ID_EX



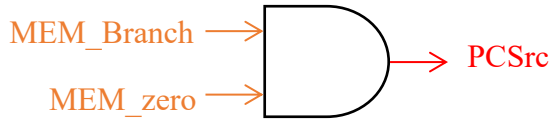
CALMUX: module mux2_cal



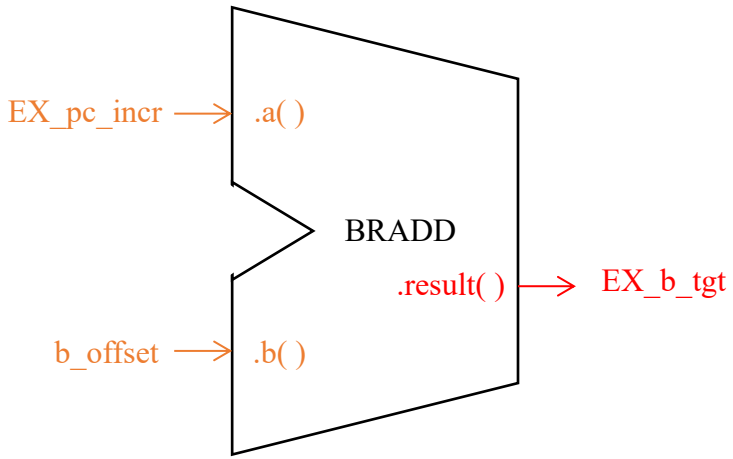
EX_MEM: module EX_MEM



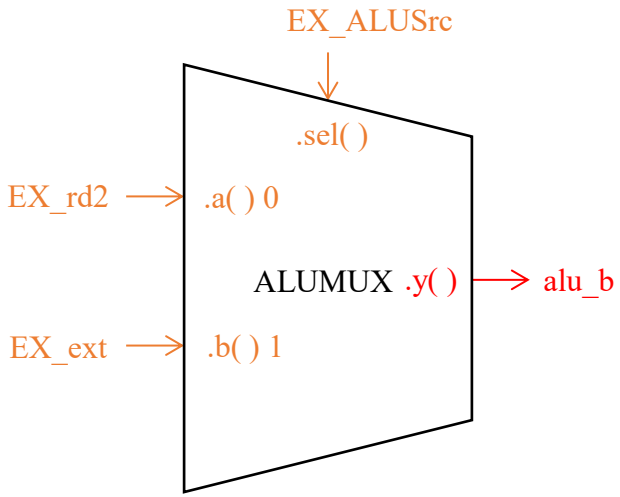
EX_MEM: module EX_MEM



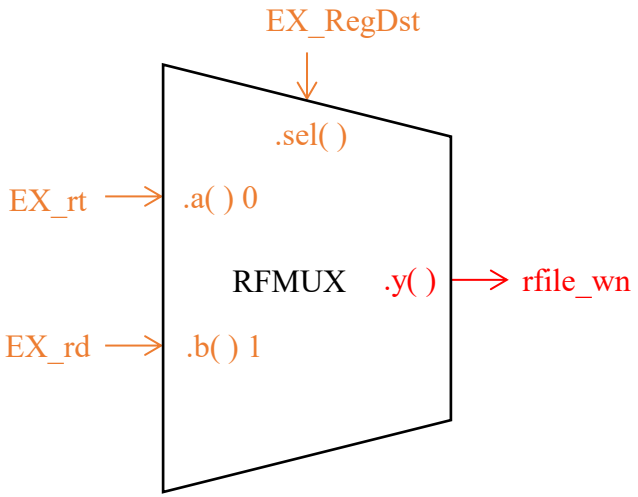
BRADD: module add32



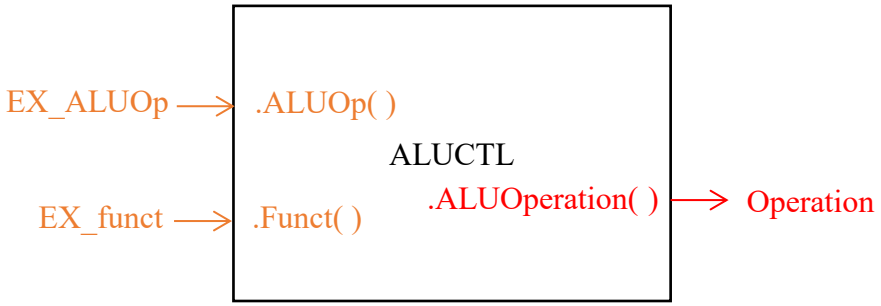
ALUMUX: module mux2



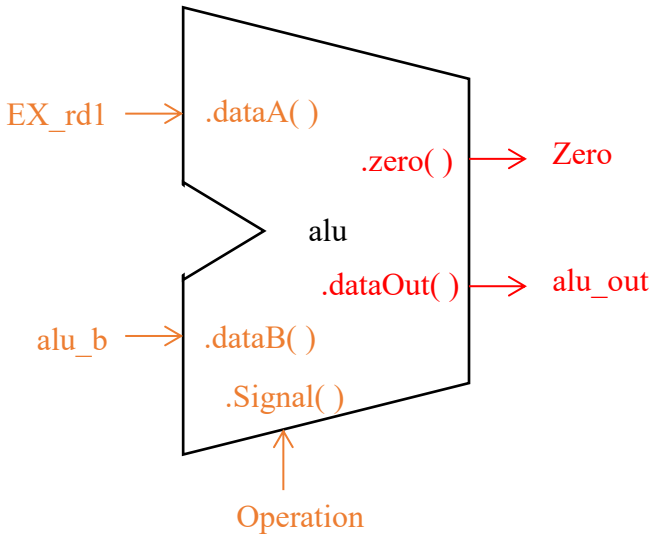
RFMUX: module mux_5bit



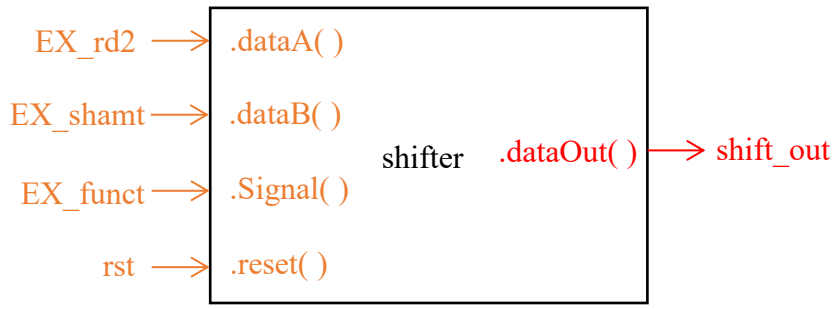
ALUCTL: module alu_ctl



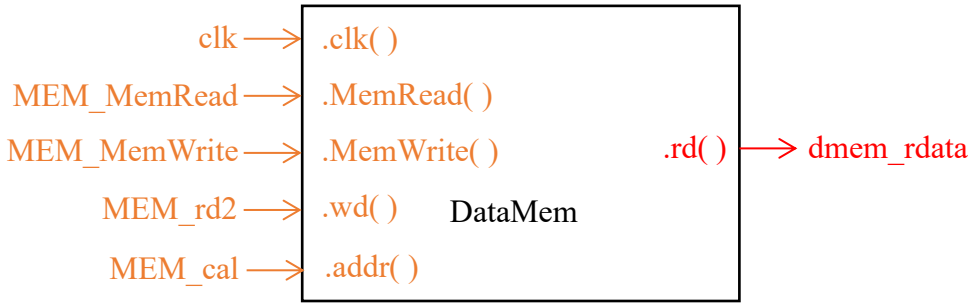
alu: module ALU



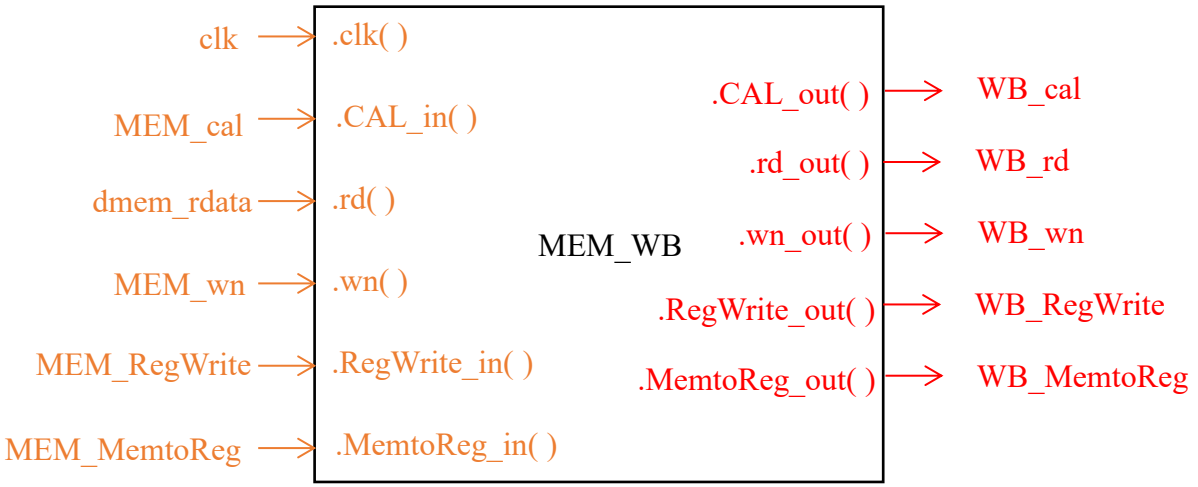
Shifter: module Shifter



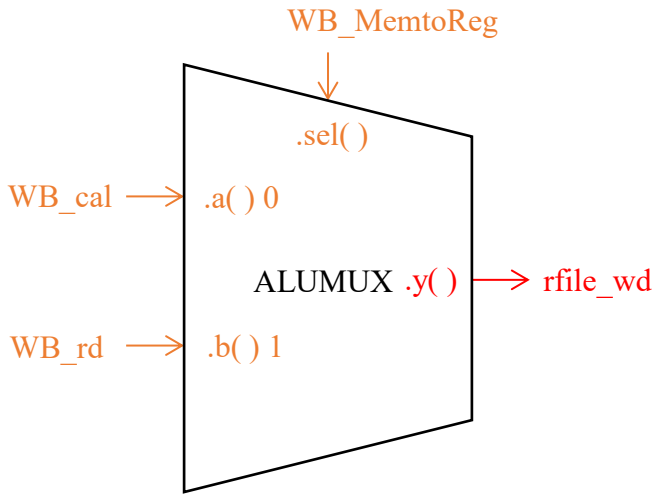
DataMem: module memory



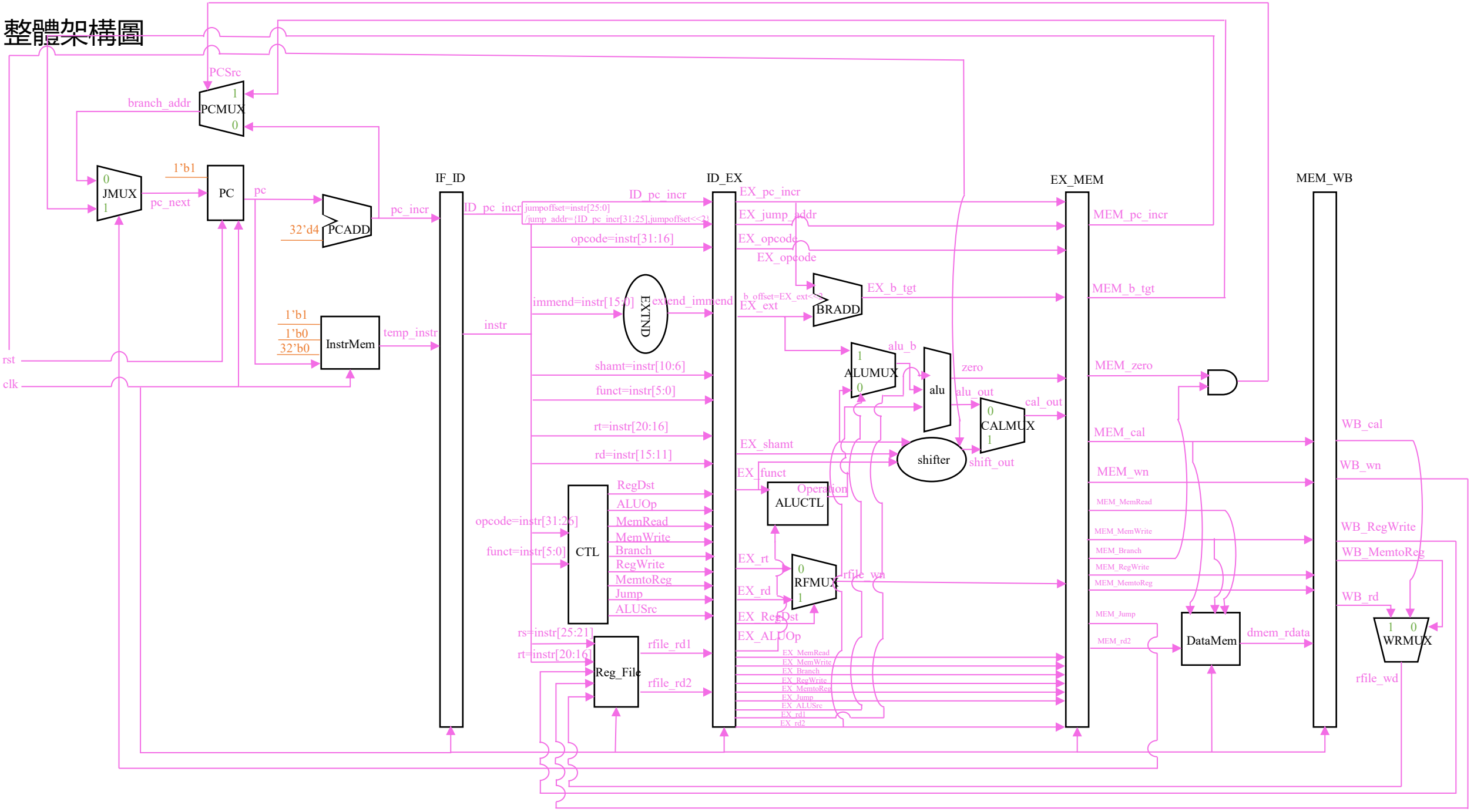
MEM_WB: module MEM_WB



WRMUX: module mux2

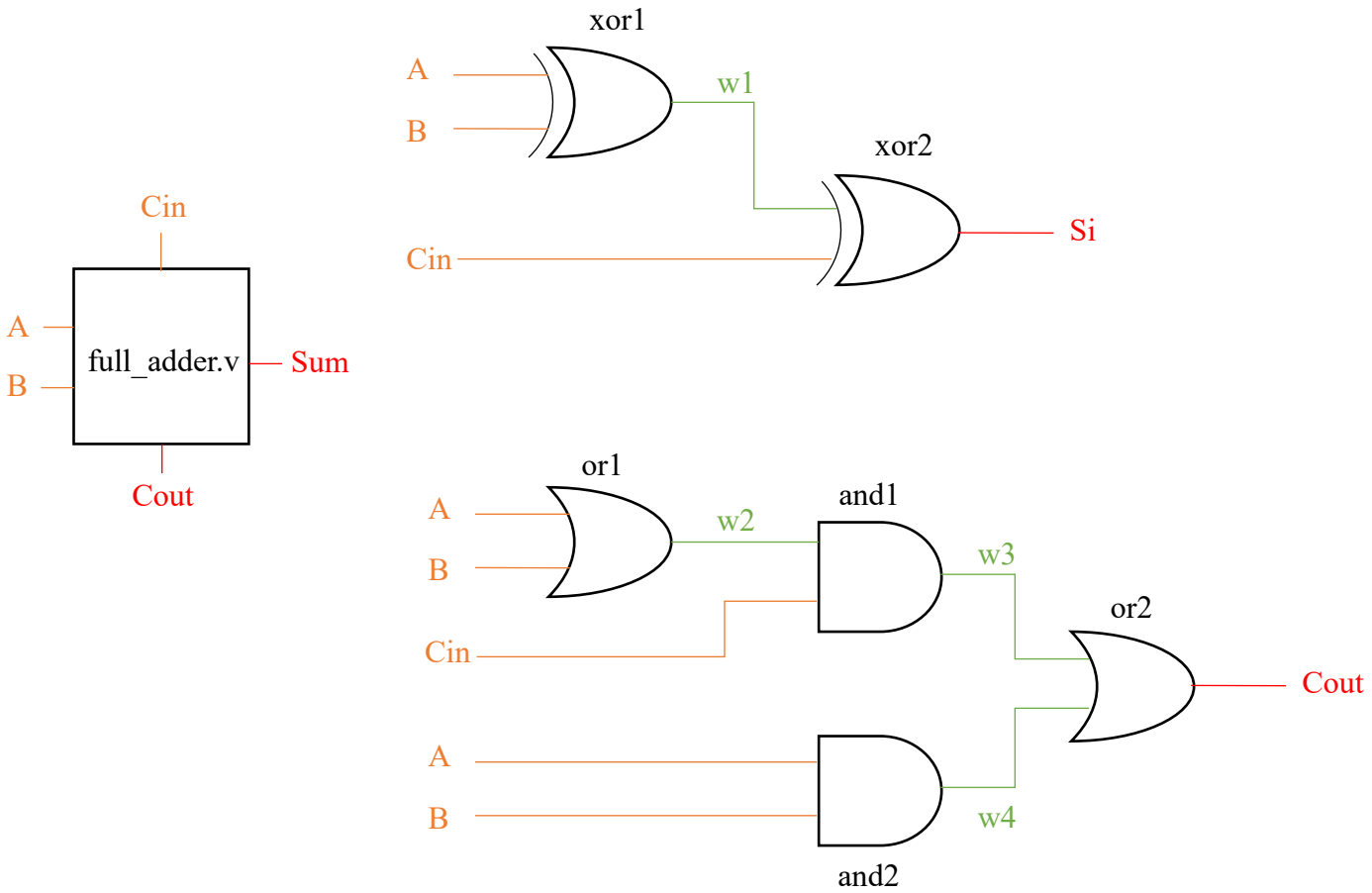


整體架構圖

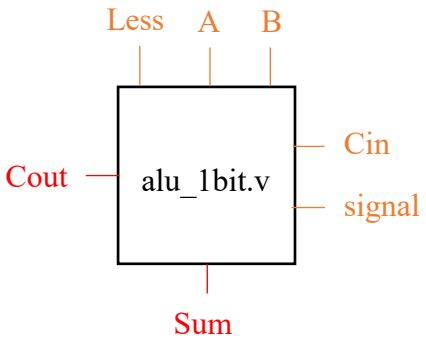

PCSrc
branch_addr
PCMUX
JMUX
pc_next
PC
pc
1'b1
32'd4
PCADD
pc_incr
IF_ID
ID_pc_incr
jumpoffset=instr[25:0]
/jump_addr={ID_pc_incr[31:25],jumpoffset<<2}
opcode=instr[31:16]
immend=instr[15:0]
EXTND
extend_immend
1'b1
1'b0
32'b0
InstrMem
temp_instr
instr
shamt=instr[10:6]
funct=instr[5:0]
rt=instr[20:16]
rd=instr[15:11]
rst
clk
RegDst
ALUOp
MemRead
MemWrite
Branch
RegWrite
MemtoReg
Jump
ALUSrc
opcode=instr[31:26]
funct=instr[5:0]
CTL
rs=instr[25:21]
rt=instr[20:16]
Reg_File
rfile_rd1
rfile_rd2
ID_EX
EX_pc_incr
EX_jump_addr
EX_opcode
EX_opcode
EX_b_tgt
b_offset=EX_ext<<2
EX_ext
BRADD
alu_b
ALUMUX
alu
zero
alu_out
cal_out
CALMUX
EX_shamt
shifter
shift_out
EX_funct
Operation
ALUCTL
EX_rt
EX_rd
EX_RegDst
EX_ALUOp
RFMUX
rfile_wn
EX_MemRead
EX_MemWrite
EX_Branch
EX_RegWrite
EX_MemtoReg
EX_Jump
EX_ALUSrc
EX_rd1
EX_rd2
EX_MEM
MEM_pc_incr
MEM_b_tgt
MEM_zero
MEM_cal
MEM_wn
MEM_MemRead
MEM_MemWrite
MEM_Branch
MEM_RegWrite
MEM_MemtoReg
MEM_Jump
MEM_rd2
DataMem
dmem_rdata
MEM_WB
WB_cal
WB_wn
WB_RegWrite
WB_MemtoReg
WB_rd
WRMUX
rfile_wd

附錄：midterm project內容

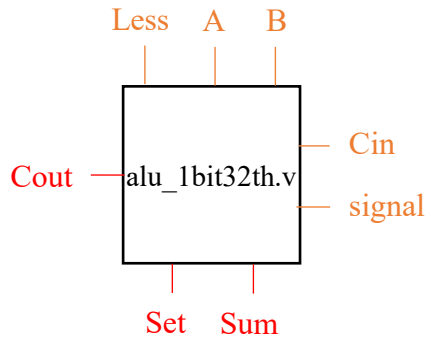
full_adder.v


xor1
A
B
w1
xor2
Cin
Si
Cin
A
full_adder.v
Sum
B
Cout
or1
A
B
w2
and1
w3
or2
Cout
Cin
A
B
w4
and2

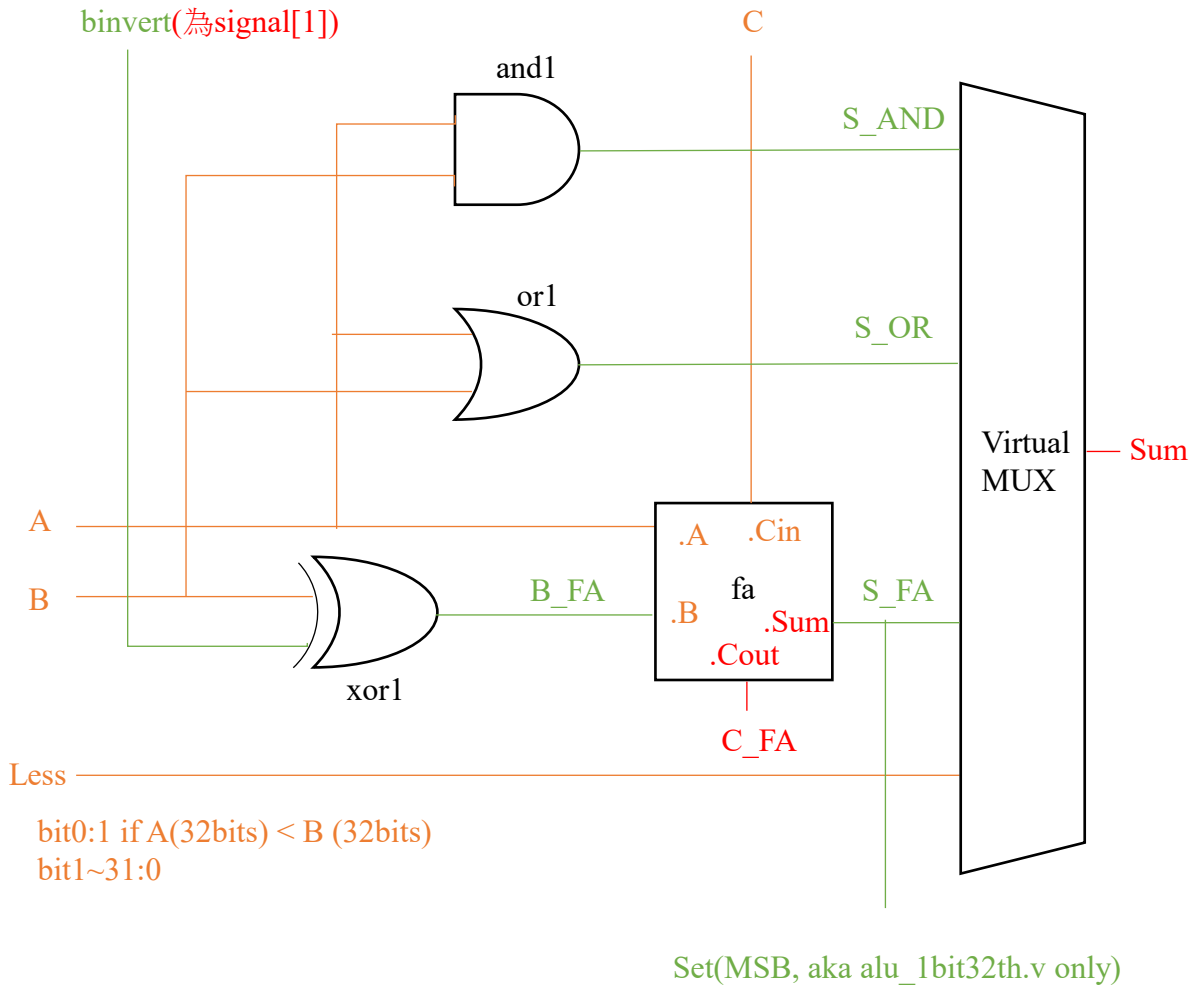
alu_1bit.v



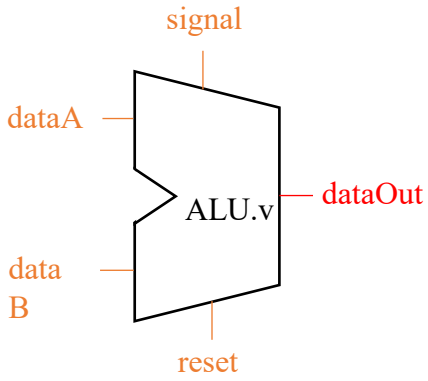
alu_1bit32th.v

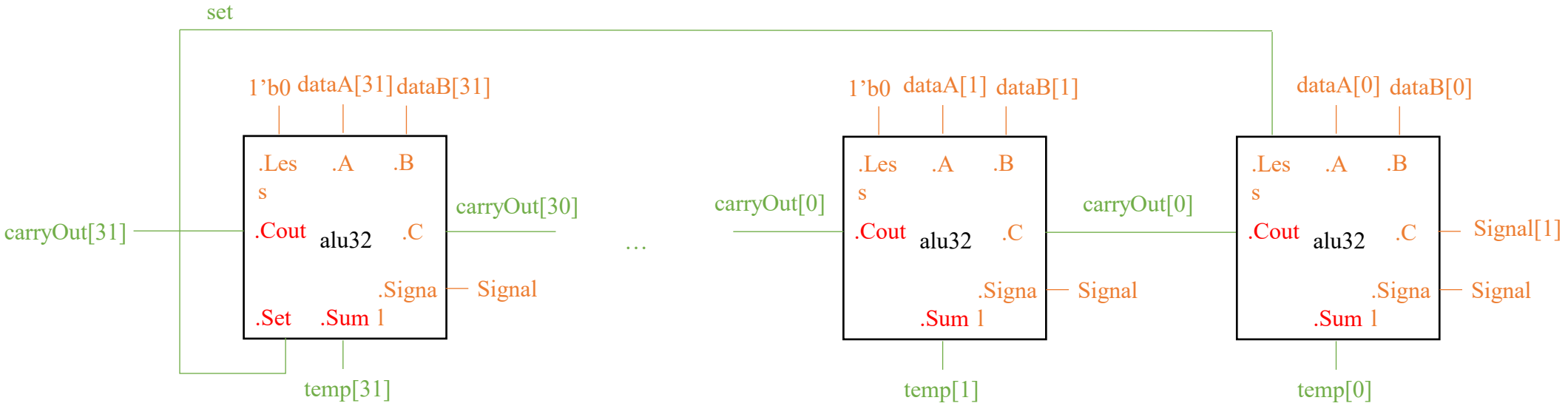


alu_1bit.v / alu_1bit32th.v (32bits比1bit多一個set)

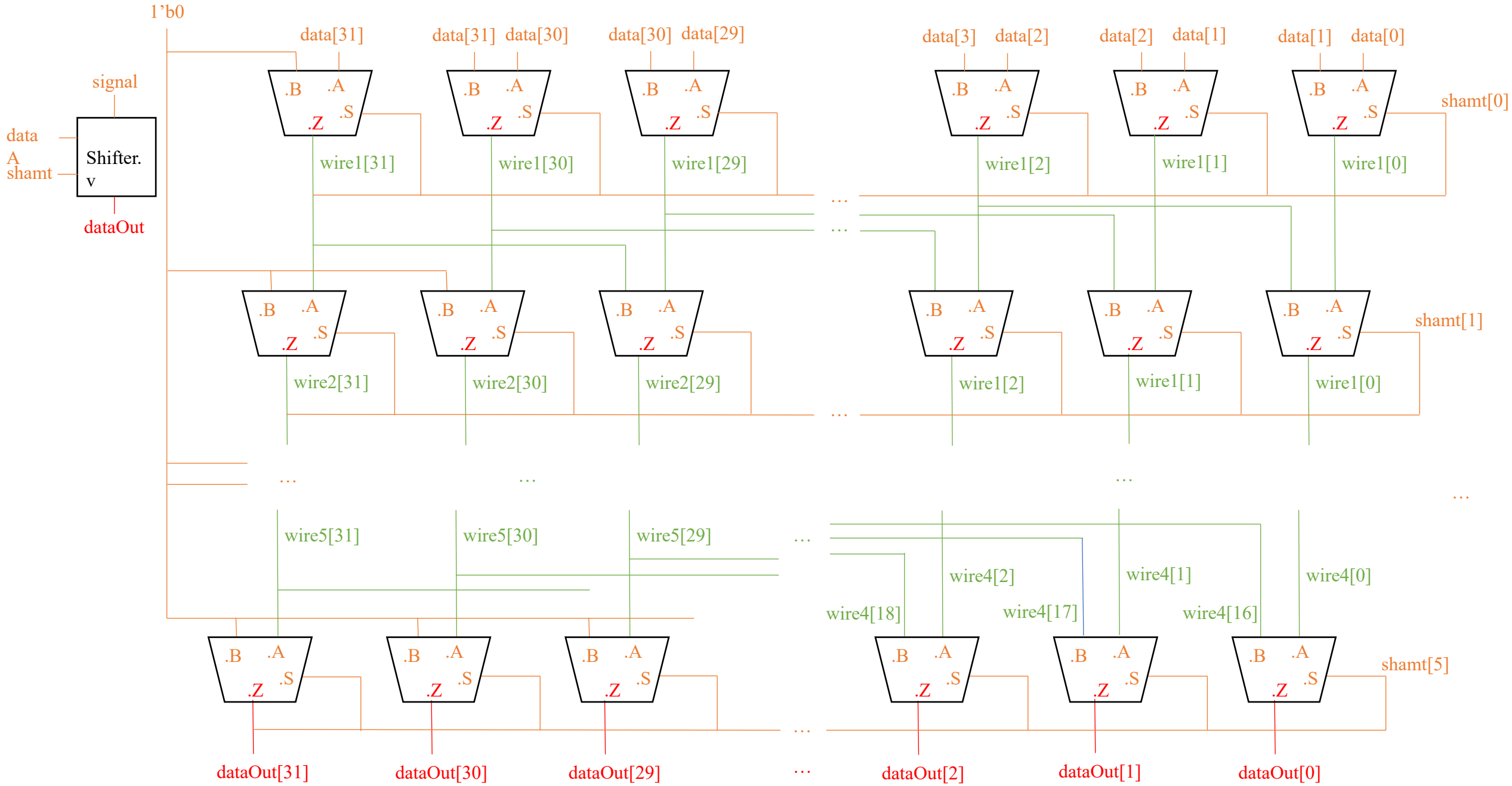


ALU.v


signal
dataA
ALU.v
dataOut
data
B
reset


set
1'b0
dataA[31]
dataB[31]
.Les
s
.A
.B
carryOut[31]
.Cout
alu32
.C
carryOut[30]
.Signa
Signal
.Set
.Sum
l
temp[31]
…
1'b0
dataA[1]
dataB[1]
carryOut[0]
temp[1]
dataA[0]
dataB[0]
Signal[1]
temp[0]

Shifter.v


1'b0
signal
data
A
shamt
Shifter.v
dataOut
data[31]
data[30]
data[29]
data[3]
data[2]
data[1]
data[0]
shamt[0]
shamt[1]
shamt[5]
wire1[31]
wire1[30]
wire1[29]
wire1[2]
wire1[1]
wire1[0]
wire2[31]
wire2[30]
wire2[29]
wire5[31]
wire5[30]
wire5[29]
wire4[18]
wire4[17]
wire4[16]
wire4[2]
wire4[1]
wire4[0]
dataOut[31]
dataOut[30]
dataOut[29]
dataOut[2]
dataOut[1]
dataOut[0]
.B
.A
.S
.Z