7.22 Base MIPSzy + j / jal

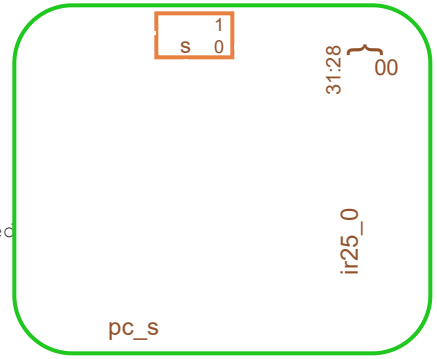
j (jump)

A designer can extend the base MIPSzy to support the j (jump) instruction by first modifying the behavioral description to d 000010, with the actions being to load PC with ir25_0 appropriately modified into a 32-bit address (described in an earlier s jump/branch immediates).

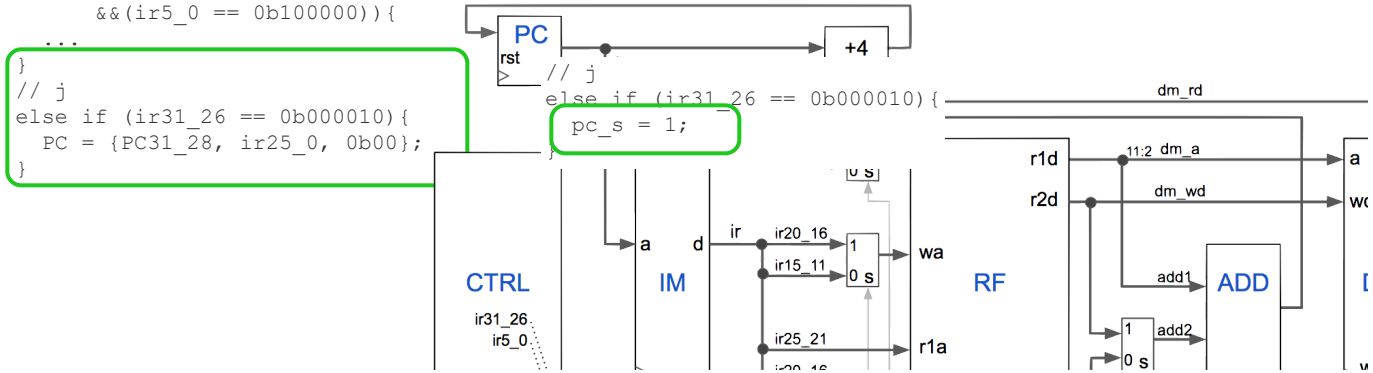
The designer can then modify the processor circuit to carry out those actions, by inserting a mux in front of the PC and app controlling that mux, with the new value coming from ir25_0 appropriately modified.

PARTICIPATION ACTIVITY 7.22.1: Extending the base MIPSzy processor to support the jump instruction.

Start 2x speed




```
ir = IM[PC/4];
PC = PC + 4;
// Assume ir31_26... extracted
// lw
if (ir31_26 == 0b100011)  {
  ...
}
// sw
else if (ir31_26 == 0b101011){
  ...
}
// addi
else if (ir31_26 == 0b001000){
  ...
}
// add
else if ((ir31_26 == 0b000000)
```



PARTICIPATION ACTIVITY 7.22.2: Extending the base MIPSzy processor to support the jump instruction.

1) In the base MIPSzy processor, CTRL sets pc_s with ___.
   - 0
   - 1
   - Not applicable

2) In the base MIPSzy processor extended for jump, for most instructions, CTRL sets pc_s with ___.
   - 0
   - 1
   - PC + 4

3) In the base MIPSzy processor extended for jump, for the jump instruction, CTRL sets pc_s with ___.
   - 0
   -

1

4) In the base MIPSzy processor extended for jump, how many bits is the jump target address that gets passed through the PC's mux?

- 26
- 28
- 32

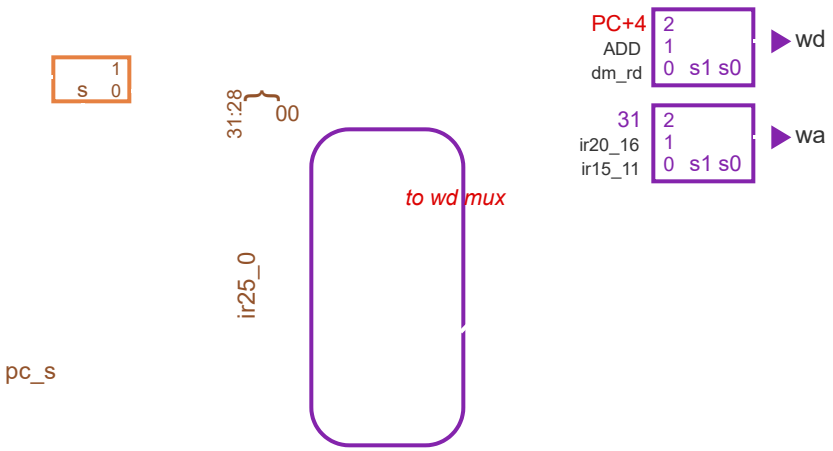
jal (jump and link)

The jal instruction also jumps to a target address like the j instruction, but also writes PC + 4 to the special $ra register, whi at register file location 31. The animation below shows the update to the behavioral description. The animation also shows muxes are needed in front of the RF's wd and wa inputs to pass PC + 4 and 31 respectively. The animation shows the contı to carry out jal on the revised processor circuit.

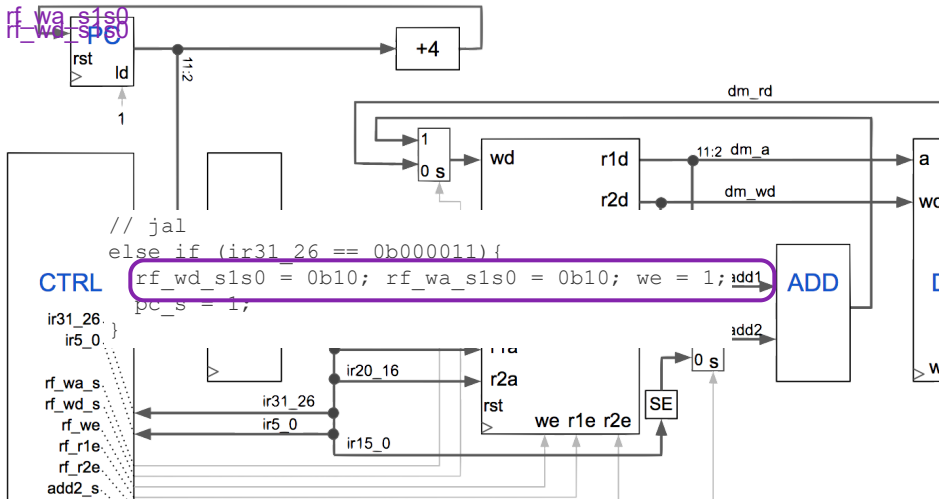
PARTICIPATION ACTIVITY 7.22.3: Extending MIPSzy to support the jal instruction as well as the j instruction.

Start 2x speed

```
ir = IM[PC/4];
PC = PC + 4;
// Assume ir31_26... extracted
// lw
if (ir31_26 == 0b100011)  {
  ...
}
// sw
else if (ir31_26 == 0b101011){
  ...
}
// addi
else if (ir31_26 == 0b001000){
  ...
}
```



```
// add
else if ((ir31_26 == 0b000000)
      &&(ir5_0 == 0b100000)){
  ...
}
// j
else if (ir31_26 == 0b000010){
  PC = {pc31_28, ir25_0, 0b00};
}
// jal
else if (ir31_26 == 0b000011){
  RF[31] = PC;
  PC = {PC31_28, ir25_0, 0b00);
}
```



PARTICIPATION ACTIVITY 7.22.4: Extending the base MIPSzy processor to support the jal instruction.

Consider the base MIPSzy extended to support jal.

1) An add instruction would set rf_wd_s1s0 with ___.
   - ○ 1
   - ○ 01

2) A jal instruction writes to the register at RF location ___.
   - ○ 0
   - ○ 31

3) The circuit for loading the PC is _____ for the j and jal instructions.
   - ○ the same
   - ○ different

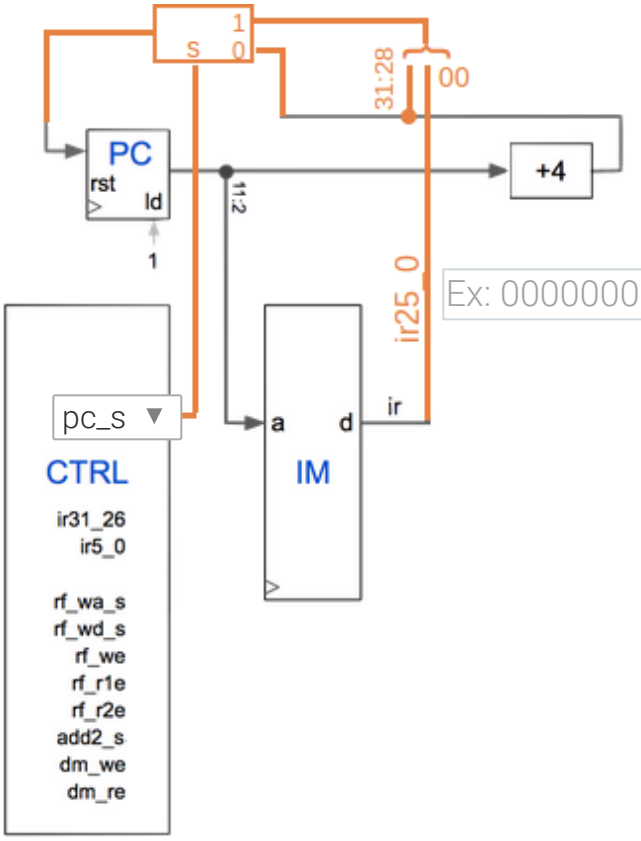
CHALLENGE ACTIVITY 7.22.1: j and jal extension in MIPSzy.

Start

1

2

Enter the bits for each item.
asm: j LABEL
ir: 000010 00000000000000000000100100



1 2

Check Next

Provide feedback on this section