7.18 Base MIPSzy + sub

A designer can extend the base MIPSzy to support the sub (subtract) instruction. The designer can include the sub instruct behavioral description. As a reminder, the add and sub instructions have the same opcode bits ir31_26 of 000000 but have function bits ir5_0 of 100000 (add) and 100010 (sub).

Table 7.18.1: Reminder: MIPSzy machine instructions for add and sub.

Assembly	Machine
`add $t0, $t1, $t2`	`000000 01001 01010 01000 00000 100000`
`sub $t0, $t1, $t2`	`000000 01001 01010 01000 00000 100010`

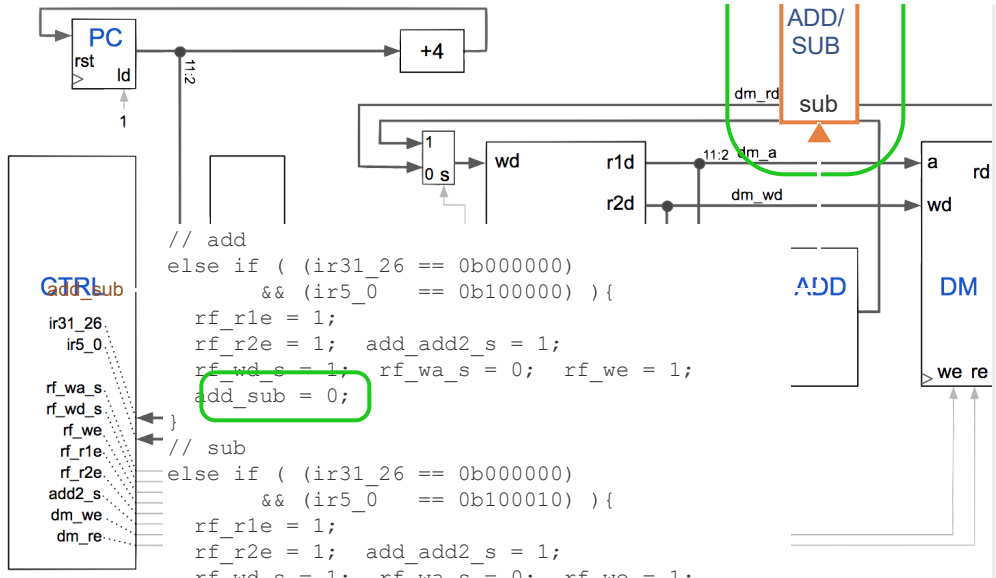
To support the behavioral description's actions in the processor's circuit, the designer can replace ADD with an ADD/SUB c performs addition when the component's control input sub is 0, and subtraction when sub is 1. Then CTRL sets the compo appropriately for each of the add and sub instructions.

PARTICIPATION ACTIVITY 7.18.1: Extending the base MIPSzy processor to support the subtraction 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)){
  add1 = RF[ir25_21];
  add2 = RF[ir20_16];
  RF[ir15_11] = add1 + add2;
}
// sub
else if ((ir31_26 == 0b000000)
      &&(ir5_0 == 0b100010)){
  add1 = RF[ir25_21];
  add2 = RF[ir20_16];
  RF[ir15_11] = add1 - add2;
}
```



PARTICIPATION ACTIVITY 7.18.2: Extending the base MIPSzy with the sub instruction.

1) In the base MIPSzy from a previous section, the add instruction's control logic actions set the add_sub control signal with 0.
   - ○ True
   - ○ False

2) In the base MIPSzy extended for sub, the add instruction's control logic

actions set the add_sub control signal with 0.

- ○ True
- ○ False

3) In the base MIPSzy extended for sub, the sub instruction's control logic actions set the add_sub control signal with 1.

- ○ True
- ○ False

CHALLENGE ACTIVITY 7.18.1: Subtraction extension in MIPSzy.

Start

1

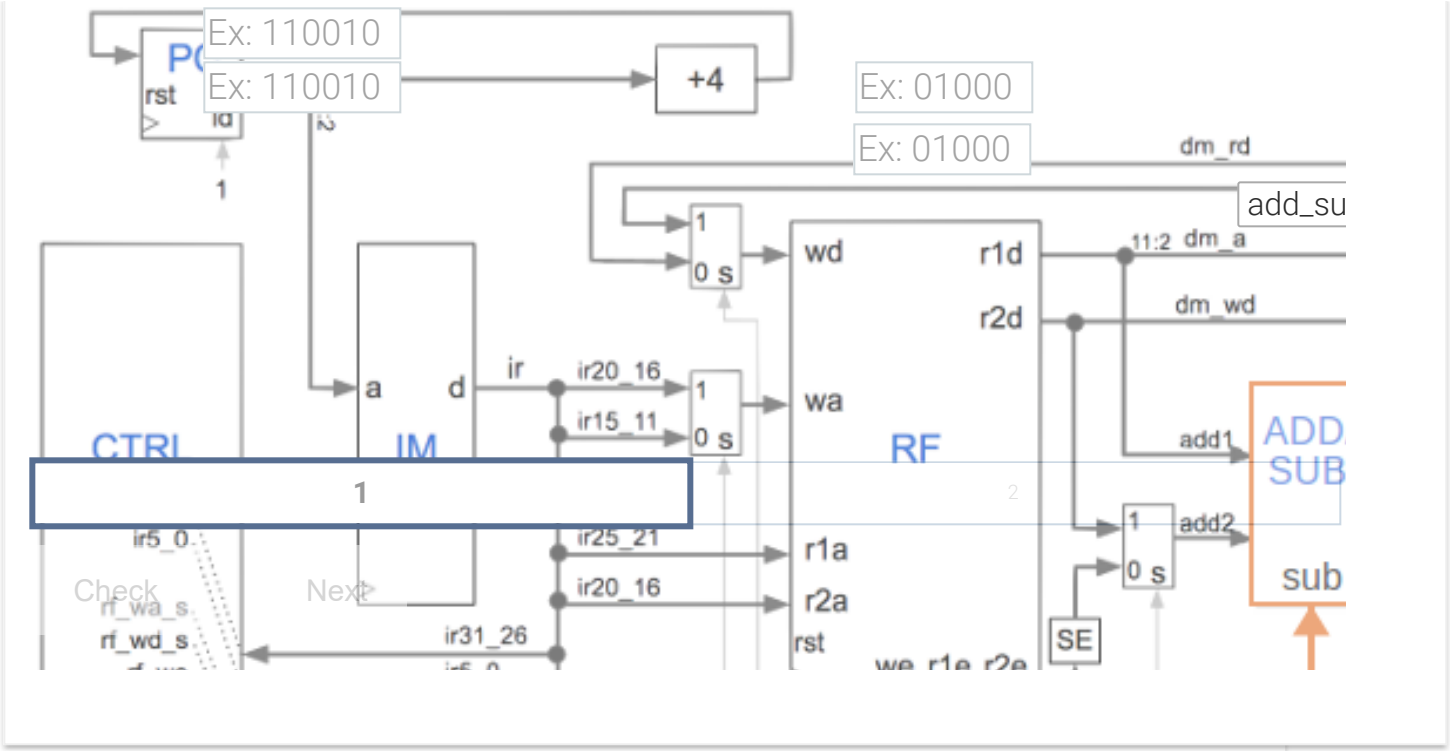
2

Enter the bits for each item.

asm: add $t7, $t1, $t5

ir: 000000 01001 01101 01111 0
$t1 $t5 $t7

Ex: 01000



Provide feedback on this section