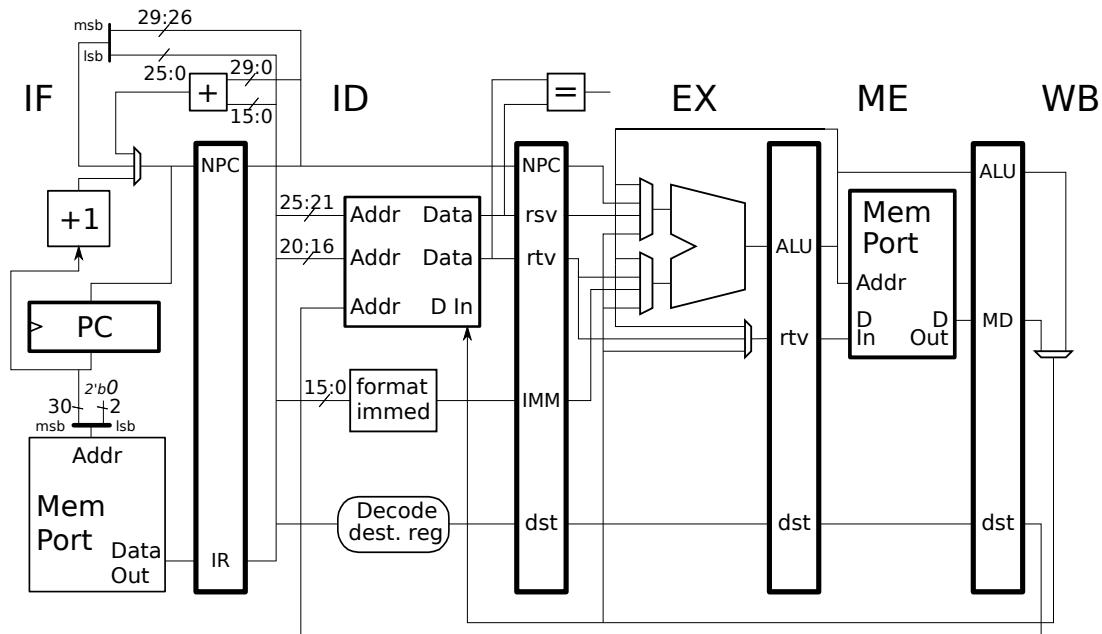


**Problem 1:** The code fragment below is to execute on the illustrated MIPS implementation. Unfamiliar instructions can be looked up on the MIPS ISA manual linked to the course references page. Show the execution of the code fragment below on the illustrated MIPS implementation. All branches are taken.

- Pay close attention to dependencies, including those for the branch.
- Note that unnecessary stalls are just as incorrect as not stalling when a stall is necessary.



```

add r4, r2, r3

lw r6, 8(r4)

sub r1, r6, r5

bltz r1 TARG

and r8, r7, r10

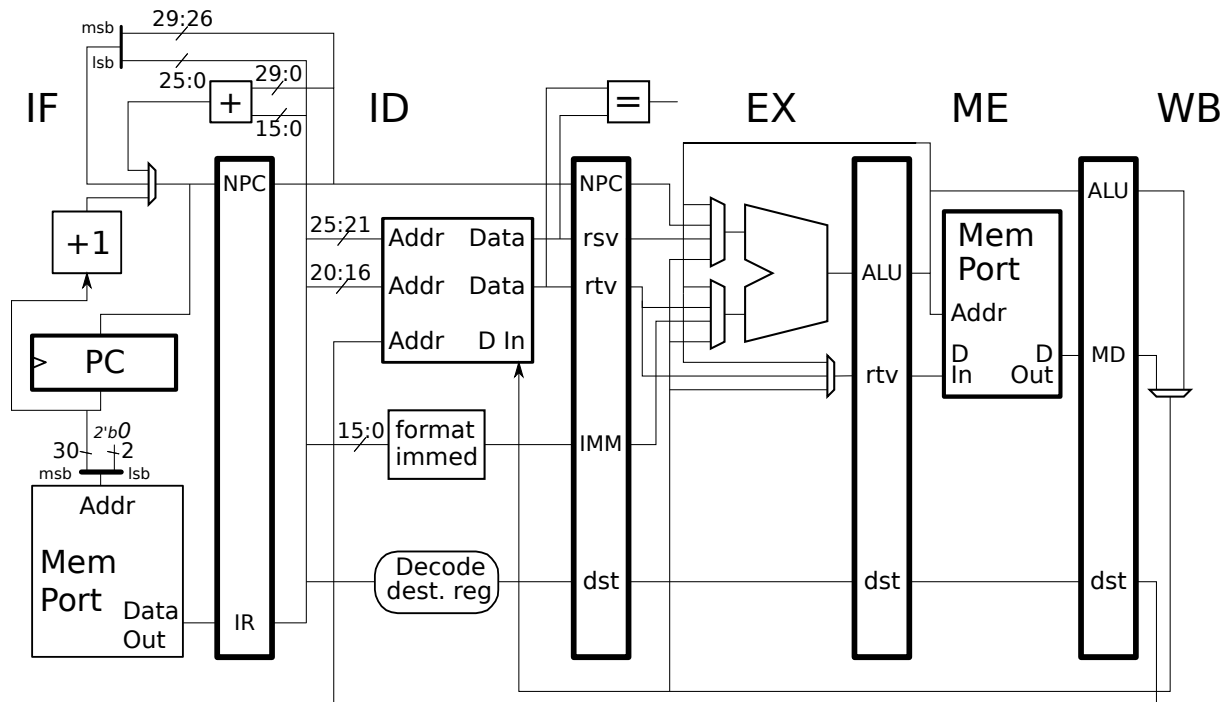
or r11, r12, r13

xor r14, r11, r8

TARG:
sw r1, 0(r2)
    
```

Spoiler Alert: Hint about solution on next page.

**Problem 2:** The implementation below (which is the same as the implementation for the previous problem) lacks hardware needed for the `bltz` instruction. In this problem design such hardware as described in the parts below. *Note: An Inkscape SVG version of the implementation can be found at <http://www.ece.lsu.edu/ee4720/2016/mpipei3.svg>.*



- (a) Add the hardware needed to detect when a `bltz` is taken. The hardware should have an output labeled **TAKEN**, which should be set to logic 1 if there is a taken `bltz` in ID. Include control logic, including the logic for detecting `bltz`.
- (b) The solution to the previous problem (not the previous part to this problem) should have included a stall due to the branch instruction. Add a bypass path to the hardware designed above so that the branch from the previous problem can execute without stalling.
- (c) Design control logic for the bypass path.

**Problem 3:** The code below is similar to the code from the first problem, the only difference is in the branch instruction. In this problem explain some bad news and good news about that branch.

```

add r4, r2, r3
lw r6, 8(r4)
sub r1, r6, r5
beq r0, r1 TARG
and r8, r7, r10
or r11, r12, r13
xor r14, r11, r8

```

TARG:

```

sw r1, 0(r2)

```

- (a) The bad news is that adding bypass paths for a `beq` would not be a good idea, even though adding bypass paths for the `bltz` was a good idea. Explain why.
- (b) The good news is that the program above can easily avoid the stalls by just changing the branch instruction. Explain how. (Of course, it should go without saying that the changed program must do the same thing as the original one.)

