

Putting together all the ideas about single cycle MIPS machine design.

① What needs to be loaded into the PC?

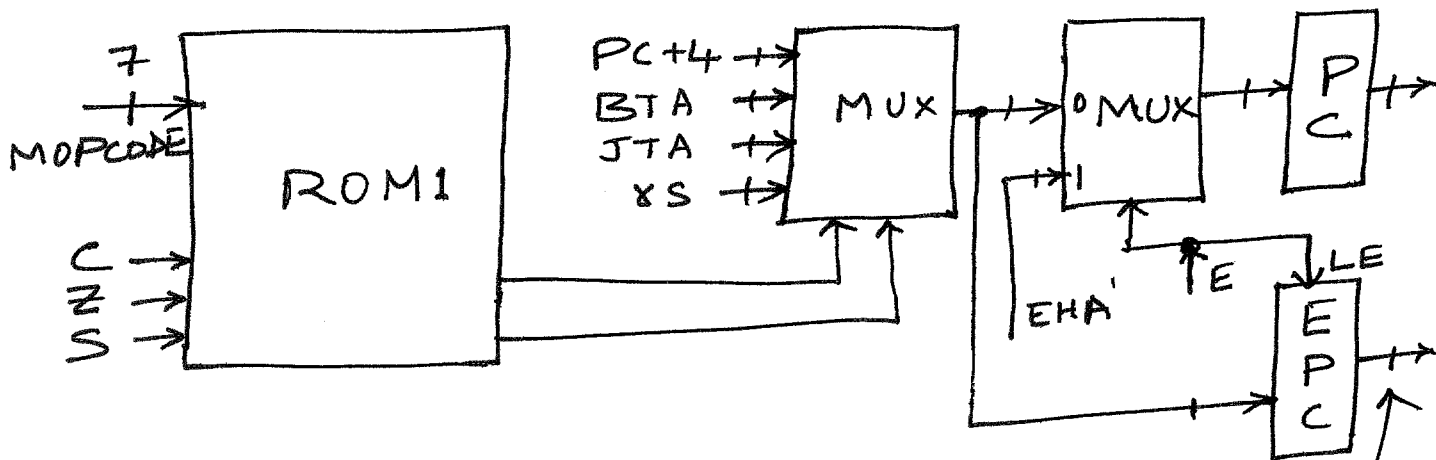
- PC+4
- BTA
- JTA
- [rs]
- EHA (Exception handler address).

② What should be loaded into EPC, if EHA is loaded into PC?

One of the above 4, depending on "conditions."

Conclusion: We need to multiplex PC+4, BTA, JTA, [rs] before we decide to load EHA

Loading into PC & EPC



E: Exception flag

To be multiplexed into WD3 in register file

③ What do we write into a register in the register file?

EPC ← done using an instruction

PC+4

ALUout

RAMout

ROM2 is another control circuit with input MOPCODE

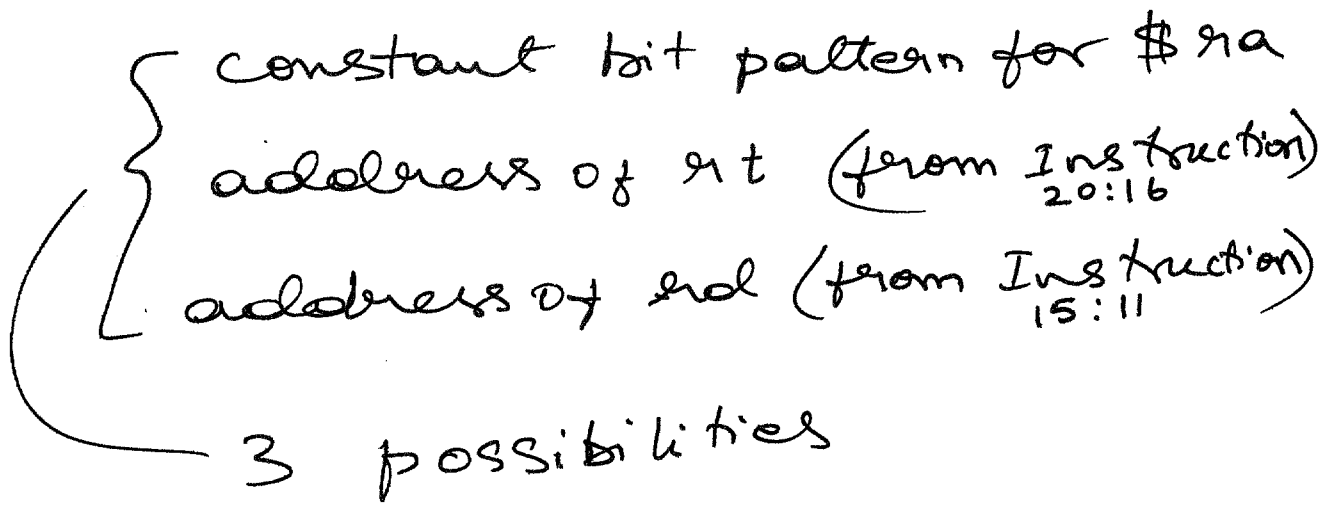
What is loaded depends

only on the MOPCODE. We need

two "select source for WD3"

control outputs from ROM2
with MOPCODE WD3 for WRITE enable.

④ Address to select which register to write into?
 That is the 5 bit A3; it should come from



We need two "select register address" control outputs from ROM 2

⑤ Address of RD1 (A1) comes directly from instruction 25:21
 Address of RD2 (A2) comes directly from instruction 20:16

6 Inputs to ALU: The first operand always comes from rs, that is at the RDI output of register file.

The second operand for ALU comes from rt or Extended Immediate operand.
 → at RD2

We need one control output from ROM2 to select between rt and Extended Imm operand.

We also need another control to specify signed/unsigned Imm. operand.

7 Inputs to RAM:

Address is always the output of ALU

Data for WRITE to RAM comes from rt, always

"Write enable RAM" is one control bit output from ROM2

8

That leaves us with ALU functions. Looking through the instructions, we need

ADD

SUB

Output 1 if ~~one~~ ^{first} operand is less than the other.

0 if ~~one~~ ^{first} operand is ~~not~~ less than the other.

" " same operation with unsigned comparison

AND

OR

XOR

SLL

SRL

SRA

SLL

SRL

SRA

} Shift operations based on shamt

} Shift operations based on [rs]_{4:0}

NOR, a few more ~~operations~~ involving unsigned operations.

There are ~~less than~~ 16 operations

We need ~~4~~⁵ ALU funct lines out of

ROM 2

