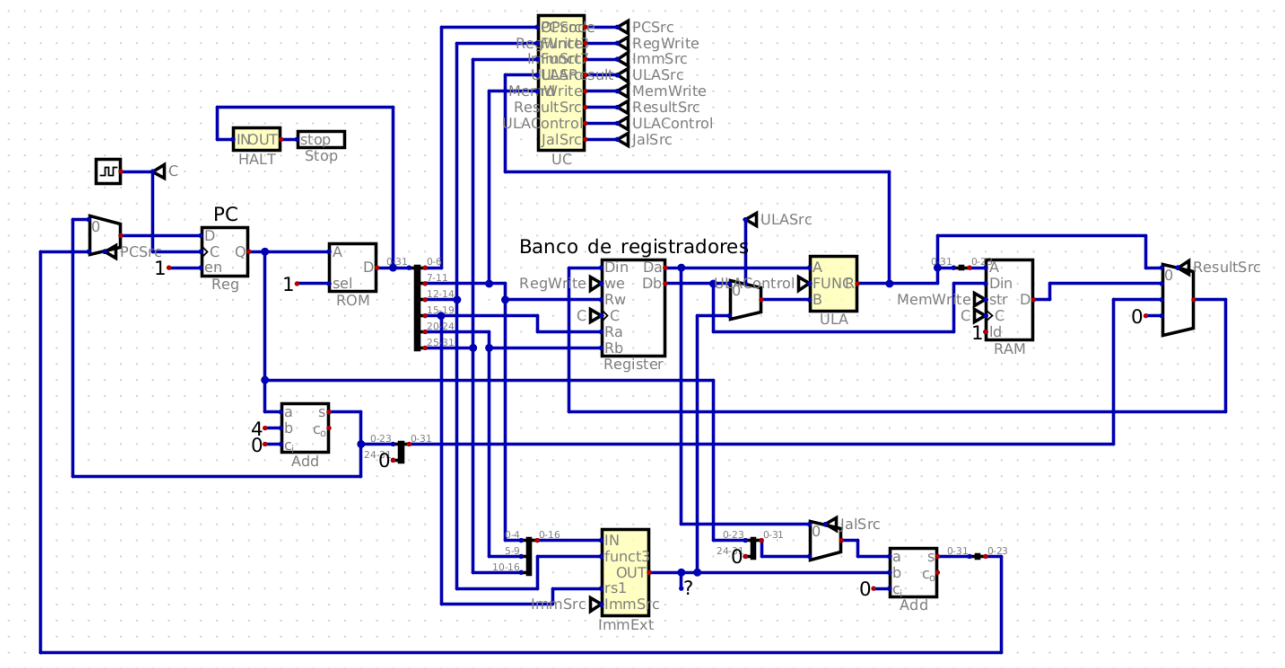


Relatório microprocessador RISC-V



Nesse microprocessador o código é guardado na ROM e o registrador PC aponta para instrução que deve ser executada.

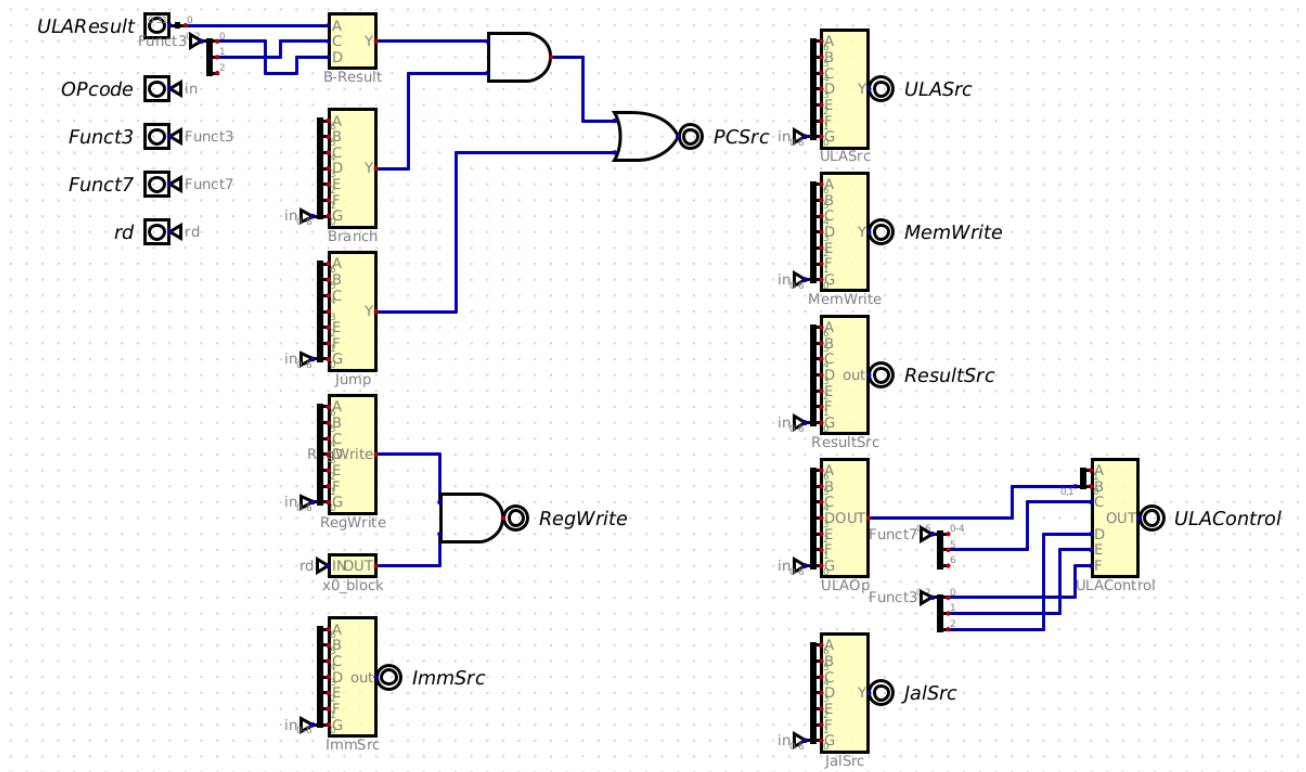
Após isso, os bits da instrução são divididos e vão para a unidade de controle e para o banco de registradores. A unidade de controle manda sinais para o circuito todo, dependendo da instrução, já o bando de registradores tem 31 registradores que podem ser alterados e 1 que não, o x0.

Então, a operação definida pela instrução deve ser feita na ula e ir para a RAM, se necessário. Após isso o valor de PC muda dependendo da instrução.

Quando a intrução for 0, o HALT é ativado e para o circuito.

Inspirado nas aulas de David Harris

Unidade de controle



A unidade de controle controla:

- Qual operação deve ser feita para o PC, com o PCSrc
- Se algum registrador deve ser escrito, com o RegWrite
- Como o imediato deve ser reescrito, com o ImmSrc
- Se a ULA deve receber um imediato ou um valor de registrador, com a ULASrc
- Se deve ser escrito algo na memória, com o MemWrite
- Qual dos valores calculados deve ser considerado, com o ResultSrc
- Qual a operação da ULA deve ser feita com o ULAControl
- Se é um jal ou jalr, com JalSrc

As tabelas verdades dos circuitos usados na unidade de controle ficou assim:

OPCODE	RegWrite	ImmSrc	ULASrc	MemWrite	ResultSrc
lw	1	00	1	0	01
sw	0	01	1	1	XX
R-type	1	XX	0	0	00
B-type	0	10	0	0	XX
I-type	1	00	1	0	00
jal	1	11	X	0	10
jalr	1	00	1	0	10

OPCODE	Branch	U LAOp	Jump	JalSrc
lw	0	01	0	X
sw	0	01	0	X
R-type	0	00	0	X
B-type	1	10	0	X
I-type	0	11	0	X
jal	0	XX	1	1
jalr	0	01	1	0

Operação	ULAControl
Soma	0
Subtração	1
XOR	2
OR	3
AND	4
Shift	5
Igual	6
Menor	7

Para o HALT, caso todos os bits da instrução forem 0, ativa 1.

Para o x0_block, caso todos os bits de rd forem 0, ativa 0.