

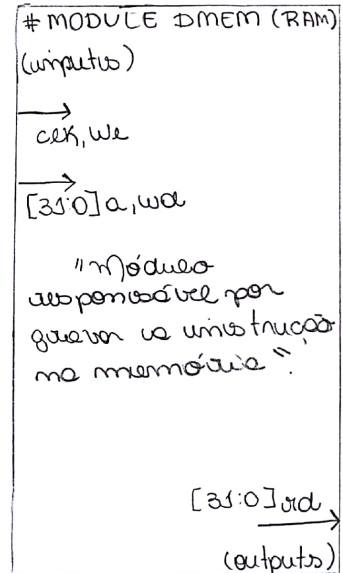
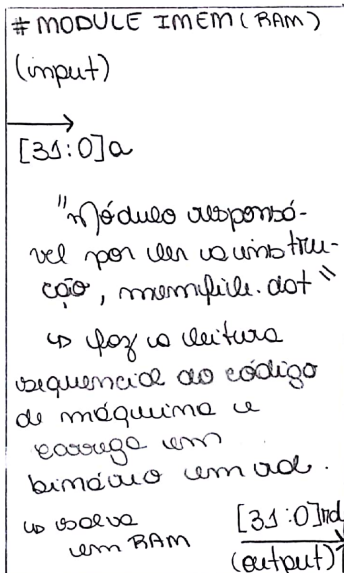
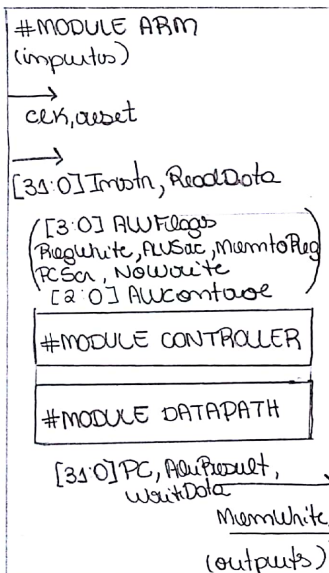
```
#MODULE TESTBENCH ( clk; reset; writeData; DataAddr; MemWrite)
```

(inputs)

clk

reset

```
#MODULE TOP ( PC[31:0]; InstrIn[31:00]; ReadData[31:0])
```



(output)

writeData

DataAddr

MemWrite

#MODULE ABM

#MODULE CONTROLLER

#MODULE DECODER

(inputs)

Op, Rd

Funct

"Módulo responsável por fazer a leitura da operação assim como condicione ou atualização ou mais das flags."

[2:0] AluControl
Nowrite
(outputs)

#MODULE CONDLOGIC

(input)

Nowrite

#CONDHECK

"Módulo responsável por fazer a condição das condições:

EQ, NE, CS... = Verifica se a instrução está executada.

#MODULE DATAPATH

#MODULE REGFILES

#MODULE EXTEND

#MODULE ALU

[31:0] AluResult

[3:0]
AluFlags

ALGUMAS DEFINIÇÕES

Instala [31:0]:

[31:28] = cond → condição de realização
ou modo de instalação
(EA, NE...)

[27:26] = usp

[25:20] = funct

[25]: 1 for immediate
0 for register

[24:21]: ADD/SUB/AND/ORR

[20]: S (1 = para setar flags)

[19:16] = um

[15:12] = ud

[8:11] = 0000

[7:0] = imm8 for immediate
0000 for register

ADD 0100

SUB 0010

AND 0000

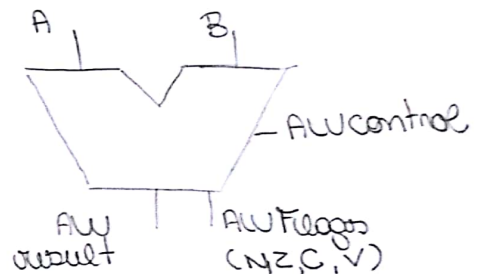
ORR 1100

[TST 1000 → S = 1 → utiliza a operação AND
CMP 1010 → S = 1 → utiliza a operação SUB
→ ambos retornam os flags

Podemos ter ARM monoprocesso:
→ faz uma instalação / clock.

Review ALU:

ALUcontrol	Function
? 00	ADD
? 01	SUB
? 10	AND
? 11	OR



→ Como foi feito
para implementar
a instalação LST?
↳ modificamos a
entrada da ALU
para 3 bits.