

# Codificação das Instruções do Processador

Gustavo Brunholi Chierici - a2126656

Thiago de Mendonça Mildemberger - 1716980

18/05/2022

## 1 Operações da ULA (ULA\_OP):

- **ADD:** 000
- **SUB:** 001
- **AND:** 010
- **OR:** 011
- **XOR:** 100
- **SHL:** 101
- **ASHR:** 110
- **ROL:** 111

## 2 Formatos das instruções:

- **Formato X** (para carregar constantes de 16 bits):
  - MOV R6, *#imm16* -> 11 0iii iiiiiiii
  - MOV R7, *#imm16* -> 11 1iii iiiiiiii

3 bits	15 bits
R6/R7	15 bits menos significantes da constante

- **Formato M** (para escrita e leitura na memória):

- MOV Rm, [Rn + #imm8] (leitura - r) -> 10 00ii iiiirrrrr
- MOV [Rm + #imm8], Rn (escrita - w) -> 10 01ii iiiirrrrr

4 bits	8 bits	3 bits	3 bits
Opcode M	constante	Rn(r) Rm(w)	Rm(r) Rn(w)

• **Formato I** (para operações com constantes de 10 bits):

- {ULA\_OP} Rm, #imm10 -> 0i iiiii0sss iiiirrrr
- CMP Rm, #imm10 -> 0i iiiii1001 iiiirrrr
- MOV Rm, #imm10 -> 0i iiiii1010 iiiirrrr

6 bits	4 bits	5 bits	3 bits
0 & 5 bits MSB da cons- tante M	Opcode I	5 bits LSB da cons- tante	Rm

• **Formato R** (para operações entre registradores):

- {ULA\_OP} Rm, Rn -> 10 1100 0sss 00rr rrrr
- CMP Rm, Rn -> 10 1100 1001 00rr rrrr
- MOV Rm, Rn -> 10 1100 1010 00rr rrrr

6 bits	4 bits	5 bits	3 bits
101100	Opcode R	00 & Rm	Rn

• **Formato J** (para Jumps e Calls, relativos e absolutos):

- JMPA cc, caddr -> 10 100a aaaa aa00 cccc
- CALLA cc, caddr -> 10 100a aaaa aa01 cccc
- JMPR cc, rel -> 10 10ii iiiii10 cccc
- CALLR cc, rel -> 10 10ii iiiii11 cccc

4 bits	8 bits	1 bit	1 bit	4 bits
1010	(0 & endereço absoluto) ou (constante)	A/R	JMP/ CALL	condição

- **Formato U** (para operações na stack):

- POP Rm -> 10 1110 0000 0000 0rrr
- PUSH Rm -> 10 1110 0001 0000 0rrr
- MOV Rm, SP -> 10 1110 1000 0000 0rrr
- MOV SP, Rm -> 10 1110 1001 0000 0rrr
- RET -> 10 1111 0000 0000 0000

15 bits	3 bits
Opcode U	Rm/000

- **Obs:** O registrador R0 é sempre 0 e a codificação da instrução NOP é a mesma codificação de ADD R0, #0