

# Codificação das Instruções do Processador

Gustavo Brunholi Chierici - 2126656  
Thiago de Mendonça Mildemberger - 1716980

27/05/2022

## 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 iiiiiiii  
 - MOV R7, *#imm16* → 11 1iii iiiiiiii 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) → 1000 iiiiiiii iiiiiiii rrrrrrr  
 - MOV [Rm + *#imm8*], Rn (escrita - w) → 1001 iiiiiiii iiiiiiii rrrrrrr

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* → 0 iiiiii 0sss iiiiiiii rrr  
 - CMP Rm, *#imm10* → 0 iiiiii 1001 iiiiiiii rrr  
 - MOV Rm, *#imm10* → 0 iiiiii 1010 iiiiiiii rrr

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):

-  $\{ULA\_OP\}$   $R_m, R_n \rightarrow$  101100 0sss 00 rrr rrr  
 - CMP  $R_m, R_n \rightarrow$  101100 1001 00 rrr rrr  
 - MOV  $R_m, R_n \rightarrow$  101100 1010 00 rrr rrr

6 bits	4 bits	5 bits	3 bits
101100	Opcode R	00 & $R_m$	$R_n$

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

- JMPA  $cc, caddr \rightarrow$  1010 0aaa aaaa 00 cccc  
 - CALLA  $cc, caddr \rightarrow$  1010 0aaa aaaa 01 cccc  
 - JMPR  $cc, rel \rightarrow$  1010 iiiii iiiii 10 cccc  
 - CALLR  $cc, rel \rightarrow$  1010 iiiii iiiii 11 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  $R_m \rightarrow$  101110 0000 10 000 rrr  
 - PUSH  $R_m \rightarrow$  101110 0011 00 000 rrr  
 - MOV  $R_m, SP \rightarrow$  101110 1000 10 000 rrr  
 - MOV  $SP, R_m \rightarrow$  101110 1001 01 000 rrr  
 - RET  $\rightarrow$  101111 0000 00 000 000

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, com a diferença de que há uma clausula extra para se assegurar que o NOP não altera as flags.