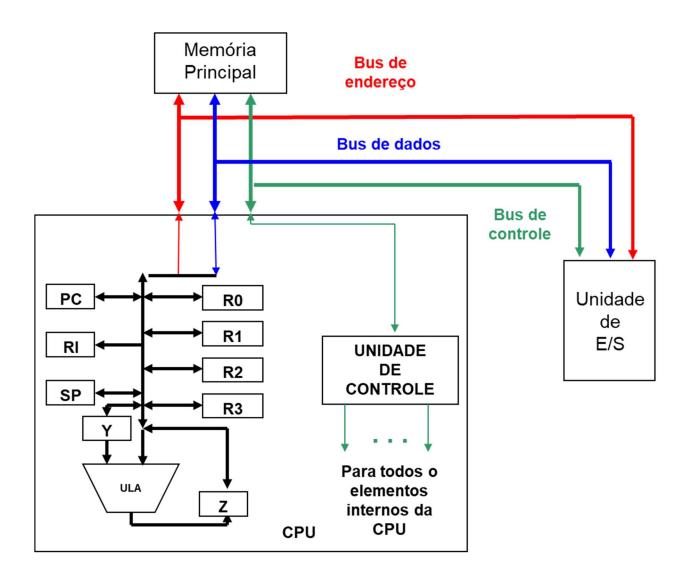


## CPU HIPOTÉTICA 2





## Laboratório de Organização de Computadores **Atividade 4**

Instruções de Movimentação de Dados       MOV     Reg1,Reg2     0000     Reg1 ← Reg2       MOV     Reg,imed     1000     Reg ← imed       MOV     Reg,[end]     1001     Reg ← MEM[end]       MOV     [end],Reg     1010     MEM[end] ← Reg       Instruções Aritméticas e Lógicas				INSTRUÇÕES	CONJUNTO DE	
Instruções de Movimentação de Dados       MOV     Reg1,Reg2     0000     Reg1 ← Reg2       MOV     Reg,imed     1000     Reg ← imed       MOV     Reg,[end]     1001     Reg ← MEM[end]       MOV     [end],Reg     1010     MEM[end] ← Reg       Instruções Aritméticas e Lógicas						
MOV         Reg1,Reg2         0000         Reg1 ←Reg2           MOV         Reg,imed         1000         Reg ← imed           MOV         Reg,[end]         1001         Reg ← MEM[end]           MOV         [end],Reg         1010         MEM[end] ← Reg           Instruções Aritméticas e Lógicas		Significado	Opcode	Operandos	Mnemônico	
MOVReg,imed1000Reg ← imedMOVReg,[end]1001Reg ← MEM[end]MOV[end],Reg1010MEM[end] ← RegInstruções Aritméticas e Lógicas	Instruções de Movimentação de Dados					
MOV Reg,[end] 1001 Reg ← MEM[end] MOV [end],Reg 1010 MEM[end] ← Reg Instruções Aritméticas e Lógicas		Reg1 ←Reg2	0000	Reg1,Reg2	MOV	
MOV [end],Reg 1010 MEM[end] ← Reg Instruções Aritméticas e Lógicas		Reg ← imed	1000	Reg,imed	MOV	
Instruções Aritméticas e Lógicas		$Reg \leftarrow MEM[end]$	1001	Reg,[end]	MOV	
<u> </u>		MEM[end] ← Reg	1010	[end],Reg	MOV	
	Instruções Aritméticas e Lógicas					
<b>\DD</b>	2	Reg1 ← Reg1 + Reg2	0001	Reg1,Reg2	ADD	
ADD Reg,imed 1011 Reg ← Reg + imed		Reg ← Reg + imed	1011	Reg,imed	ADD	
SUB Reg1,Reg2 0010 Reg1 ← Reg1 - Reg2		Reg1 ← Reg1 - Reg2	0010	Reg1,Reg2	SUB	
SUB Reg,imed 1100 Reg ← Reg – imed		Reg ← Reg – imed	1100	Reg,imed	SUB	
AND Reg1,Reg2 0011 Reg1 ← Reg1 <u>e</u> Reg2	<u> </u>	Reg1 ← Reg1 <u>e</u> Reg2	0011	Reg1,Reg2	AND	
AND Reg,imed 1101 Reg ← Reg <u>e</u> imed		Reg ← Reg <u>e</u> imed	1101	Reg,imed	AND	
DR Reg1,Reg2 0100 Reg1 $\leftarrow$ Reg1 $ou$ Reg2	յ2	Reg1 ← Reg1 <u>ou</u> Reg2	0100	Reg1,Reg2	OR	
Instruções de Manipulação de Pilha						
PUSH   Reg   0101   SP, MEM[SP] $\leftarrow$ Reg	∍g	$SP, MEM[SP] \leftarrow Reg$	J 0101	Reg	PUSH	
POP Reg 0110 Reg←MEM[SP], SP++	+	Reg←MEM[SP], SP++	0110	Reg	POP	
Instruções de Controle de Fluxo de Execução						
JMP   end   1110   PC ← end		PC ← end	1110	end	JMP	
CALL end 1111 SP , MEM[SP]←PC , PC←e	←end	SP, MEM[SP] $\leftarrow$ PC, PC $\leftarrow$ e	1111	end	CALL	
RET 0111 PC← MEM[SP], SP++	<b>++</b>	$PC \leftarrow MEM[SP]$ , $SP++$	0111		RET	

Mostrar o ciclo de execução de instruções para todas as instruções do ISA da CPU Hipotética 2 – ENTREGAR PELO CANVAS

1) MOV R1,R0

- 2) MOV R2,16
- 3) MOV R3,[4]
- 4) MOV [4],R2
- 5) **ADD R1,R2**
- 6) **AND R0,10**
- 7) **JMP 6**
- 8) **PUSH R2**
- 9) **POP R3**
- 10) **CALL 3**
- 11) **RET**