

Organização de Computadores

Versão Hands-on com Logisim

Prof. Juan G. Colonna
juancolonna@icomp.ufam.edu.br
Instituto de Computação (IComp)
Universidade Federal do Amazonas (UFAM)
Semestre 2024/01

I/O

Entrada de dados

Se quisermos ter mais flexibilidade para carregar dados nos registradores, o que poderíamos usar no lugar da instrução DATA?

Em outras palavras, Quais fontes de dados do usuário podemos usar?

Endereço	Programa	Codes	Hexa decimal
	DATA R2,0x20 # endereço base DATA R3,0x01 # incremento LD R2,R0 # load primeiro dado ADD R3,R2 # incremento LD R2,R1 # load segundo dado ADD R0,R1 # realizar operação ADD R3,R2 ST R2,R1		

Entrada de dados

Endereço	Programa	Codes	Hexa decimal
	<p>DATA R2,0x?? # endereço do periférico em R2</p> <p>OUT addr,R2 # endereço do periférico em R2</p> <p>IN data,R0 # lê o dado do teclado para R0</p> <p>LD R2,R0 # load primeiro dado</p> <p>DATA R3,0x01 # incremento</p> <p>ADD R3,R2 # incremento</p> <p>LD R2,R1 # load segundo dado</p> <p>ADD R0,R1 # realizar operação</p> <p>ADD R3,R2</p> <p>ST R2,R1</p>		

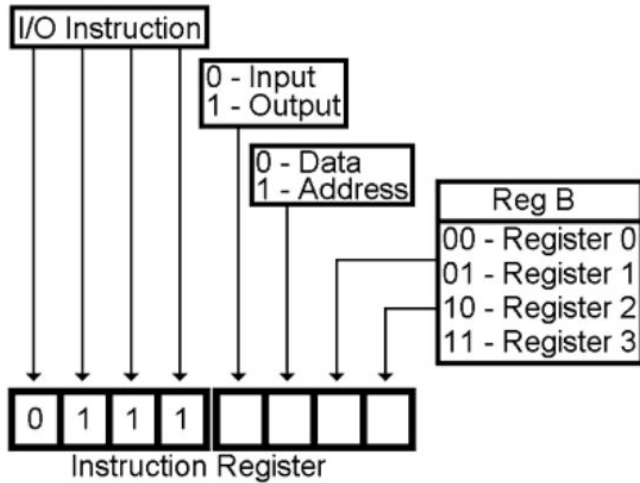
Instruções

<u>Instruction Code</u>	<u>Language</u>	<u>Meaning</u>
1000 rarb	ADD RA,RB	Add
1001 rarb	SHR RA,RB	Shift Right
1010 rarb	SHL RA,RB	Shift Left
1011 rarb	NOT RA,RB	Not
1100 rarb	AND RA,RB	And
1101 rarb	OR RA,RB	Or
1110 rarb	XOR RA,RB	Exclusive OR
1111 rarb	CMP RA,RB	Compare
0000 rarb	LD RA,RB	Load RB from RAM addr in RA
0001 rarb	ST RA,RB	Store RB to RAM addr in RA
0010 00rb	DATA RB,Addr	Load these 8 bits into RB
0011 00rb	JMPR RB	Jump to the address in RB
0100 0000	JMP Addr	Jump to the addr in the next byte
0101 caez	JCAEZ Addr	Jump if any tested Flag is on
0110 0000	CLF	Clear all Flags

<u>Language</u>	<u>Meaning</u>
IN Data,RB	Input I/O Data to RB
IN Addr,RB	Input I/O Address to RB
OUT Data,RB	Output RB to I/O as Data
OUT Addr,RB	Output RB to I/O as Address

Completamos o conjunto de instruções com as instruções que faltavam.

Instruções



Language	Meaning
IN Data,RB	Input I/O Data to RB
IN Addr,RB	Input I/O Address to RB
OUT Data,RB	Output RB to I/O as Data
OUT Addr,RB	Output RB to I/O as Address

Completamos o conjunto de instruções com as instruções que faltavam.

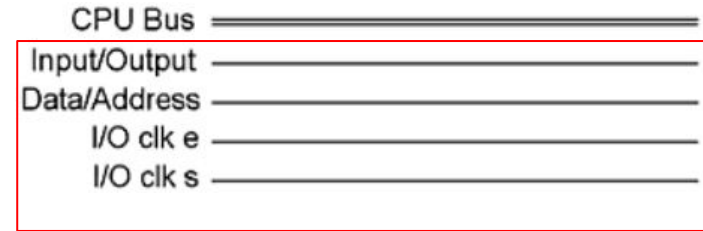
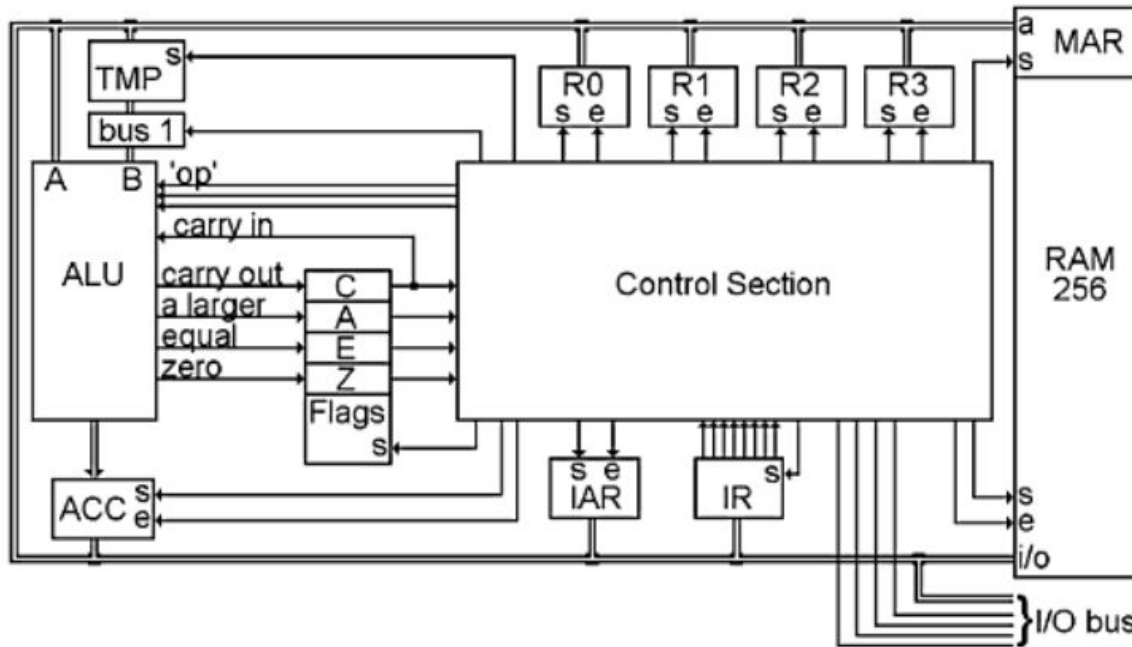
Programa	Codes	Hexa decimal
OUT addr,R2 # endereço do periférico em R2 IN data,R0 # lê o dado do teclado para R0		



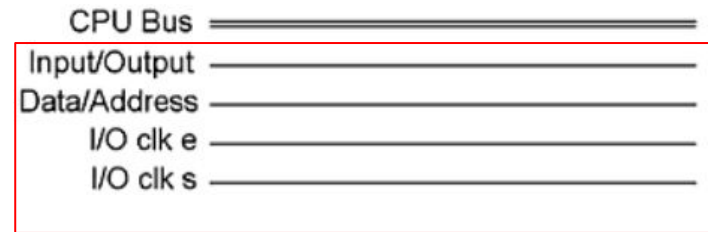
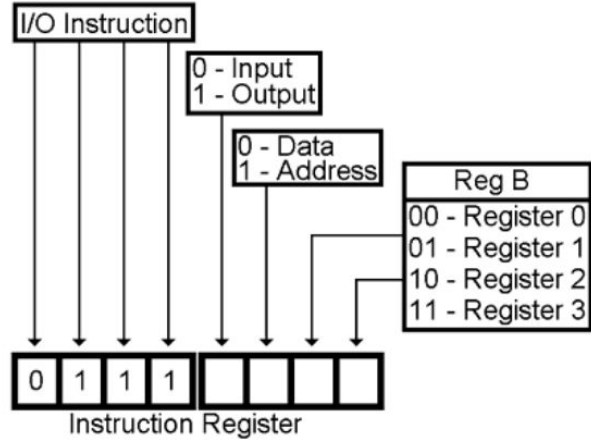
Como são conectados os periféricos?

I/O Devices

Cada device tem suas próprias características, funções e endereço. Vamos ter que adicionar quatro saídas ao projeto.



Steps.



I/O Devices

Exemplos:

A instrução **OUT addr,RB** faz....?

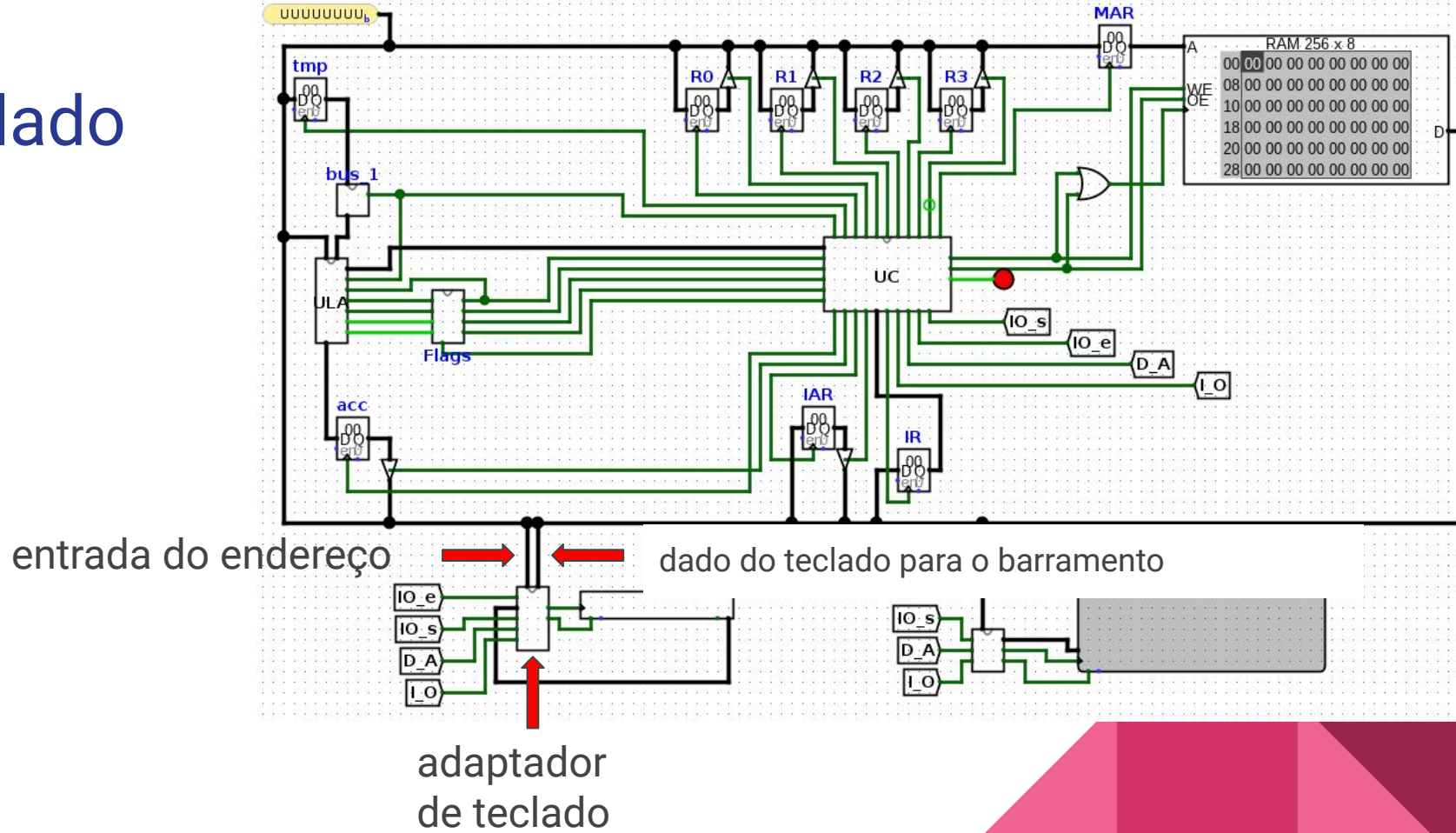
A instrução **OUT data,RB** faz....?

A instrução **IN data,RB** faz....?

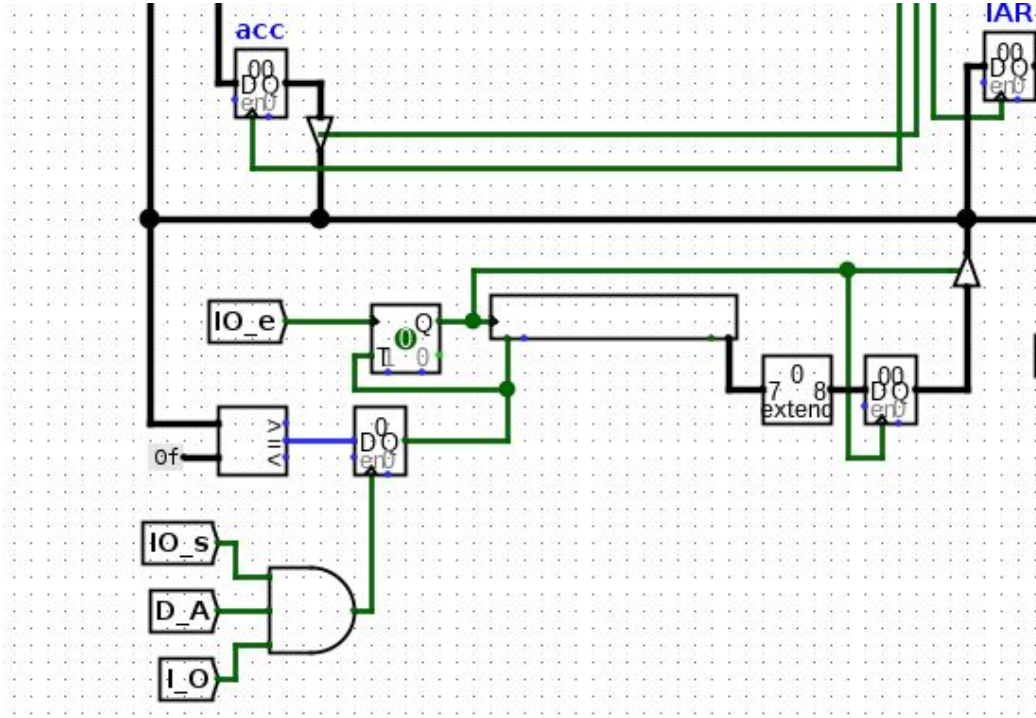
A instrução **IN addr,RB** faz....?



Teclado

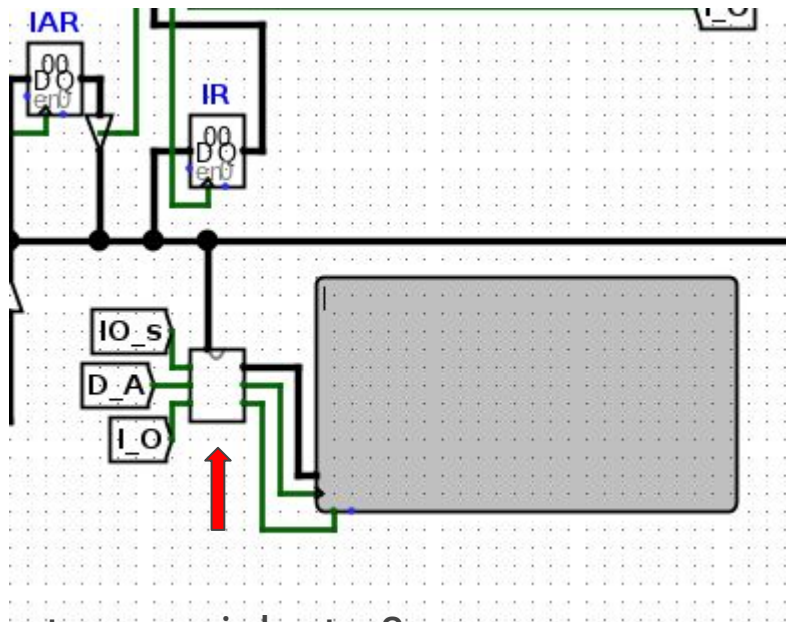


Teclado



Para usar o teclado sempre teremos que utilizar uma combinação de instruções, primeiro OUT para selecionar o teclado e depois IN para ler o dado

Monitor



Para usar o monitor sempre teremos que utilizar uma combinação de instruções, primeiro OUT para selecionar o monitor e depois OUT para escrever no monitor

O que tem aqui dentro?

Porque tem uma única conexão com o barramento?

Atividade

Adicionar a interface do teclado e do monitor

