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

Programas em assembly de 8bits

Programa assembly

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		

https://docs.google.com/spreadsheets/d/1oG0TEvD9Er8TiZOSF350yBLQeIkHU4iUIgLgawqji54/edit?usp=sharing

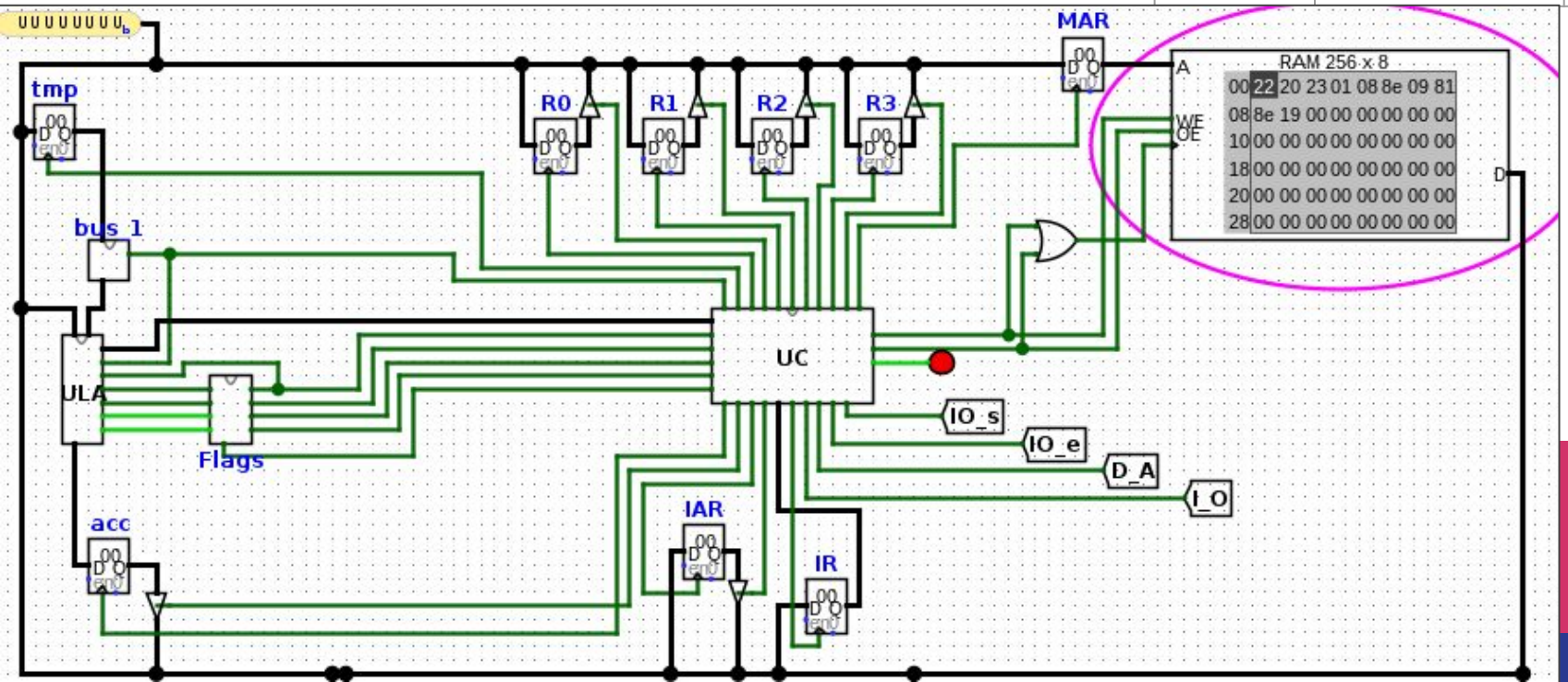
Instruções

Instruction Code	Language		Meaning
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

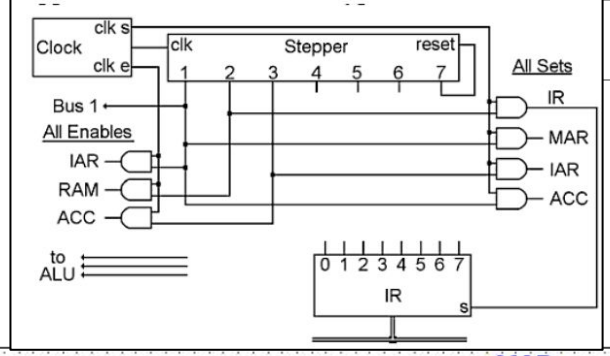
Como são executados os programas?

Step-by-step

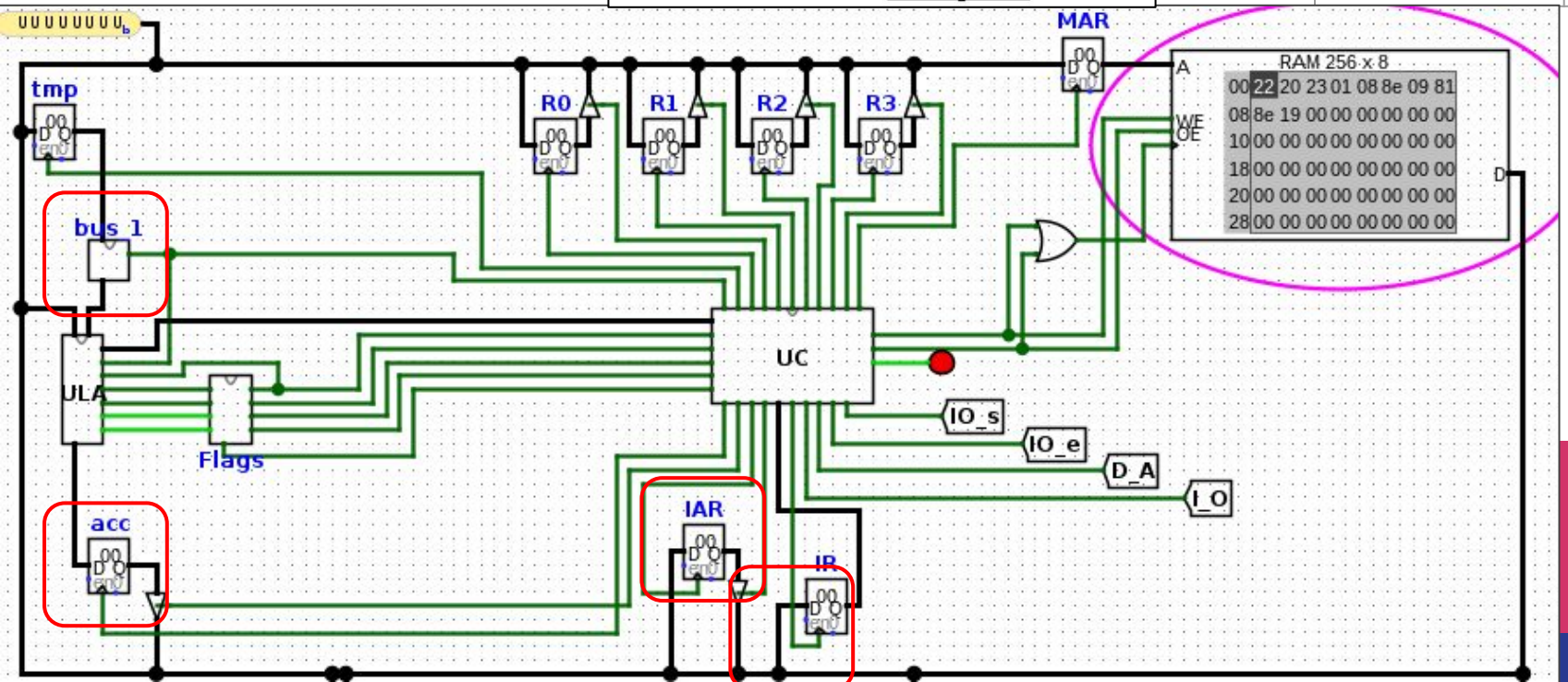
Endereço	Programa	Hexa
0x00 0x01	DATA R2,0x20	22 20
0x02 0x03	DATA R3,0x01	23 01
0x04	LD R2,R0	08
0x05	ADD R3,R2	8E
0x06	LD R2,R1	09
0x07	ADD R0,R1	81
0x08	ADD R3,R2	8E
0x09	ST R2,R1	19



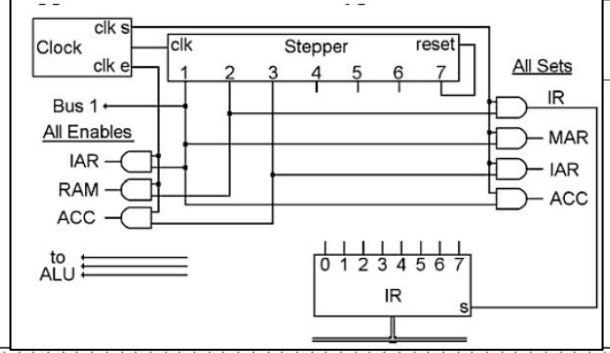
Elementos



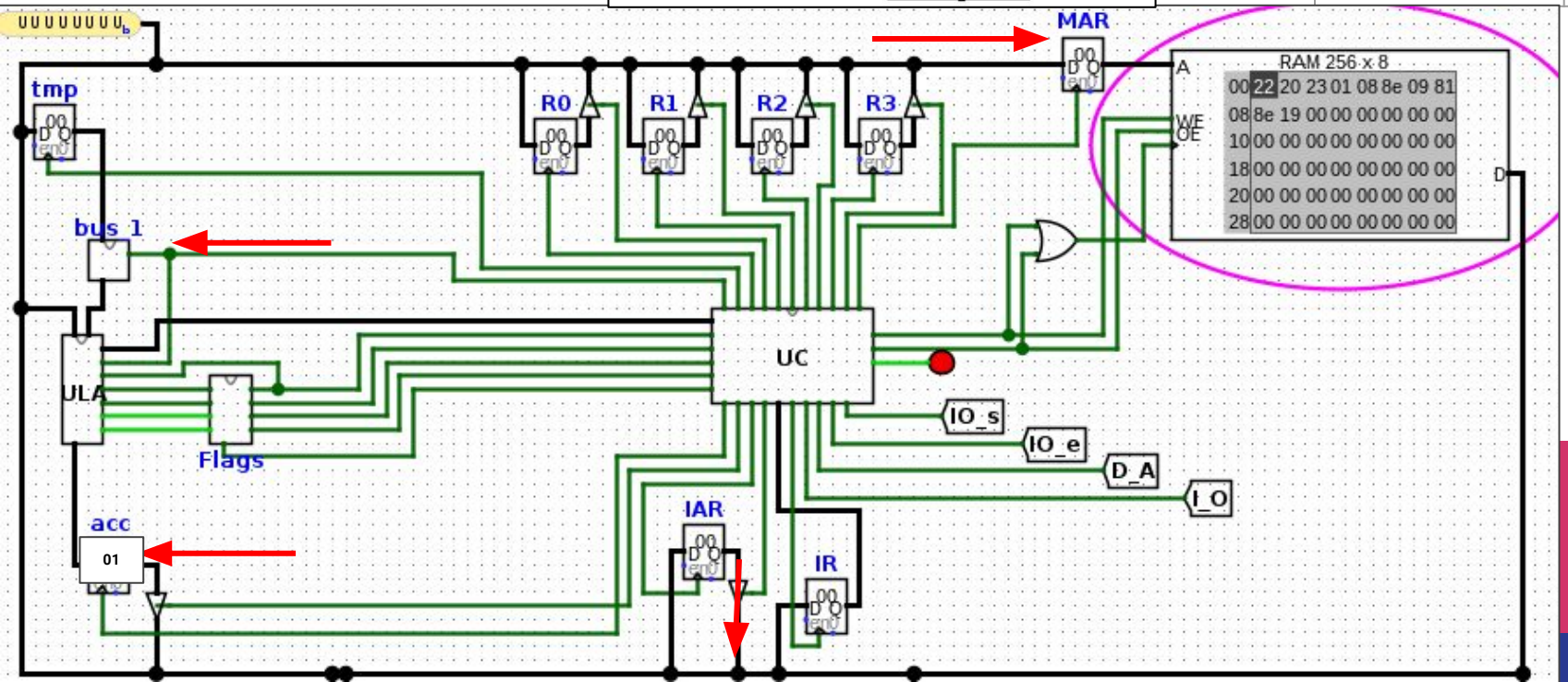
Endereço	Programa	Hexa
0x00 0x01	DATA R2,0x20	22 20
0x02 0x03	DATA R3,0x01	23 01
0x04	LD R2,R0	08
0x05	ADD R3,R2	8E
0x06	LD R2,R1	09
0x07	ADD R0,R1	81
0x08	ADD R3,R2	8E
0x09	ST R2,R1	19



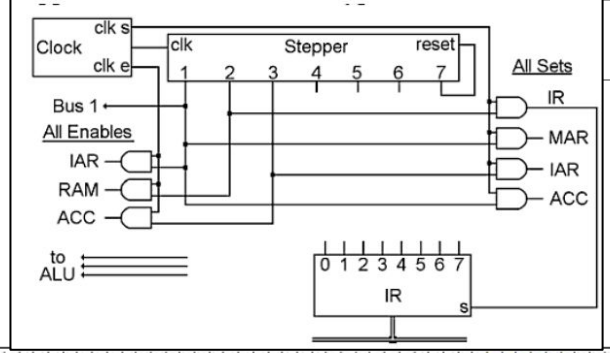
Step 1



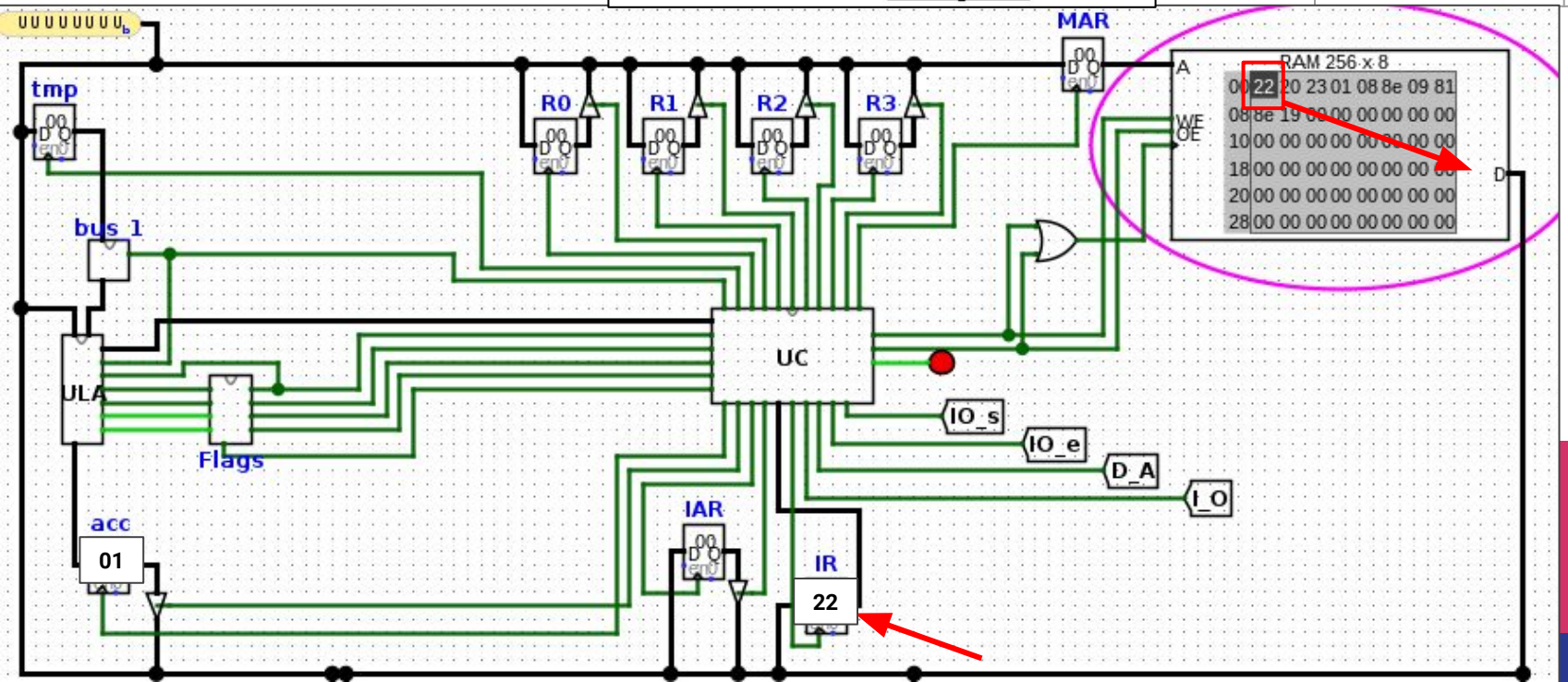
Endereço	Programa	Hexa
0x00 0x01	DATA R2,0x20	22 20
0x02 0x03	DATA R3,0x01	23 01
0x04	LD R2,R0	08
0x05	ADD R3,R2	8E
0x06	LD R2,R1	09
0x07	ADD R0,R1	81
0x08	ADD R3,R2	8E
0x09	ST R2,R1	19



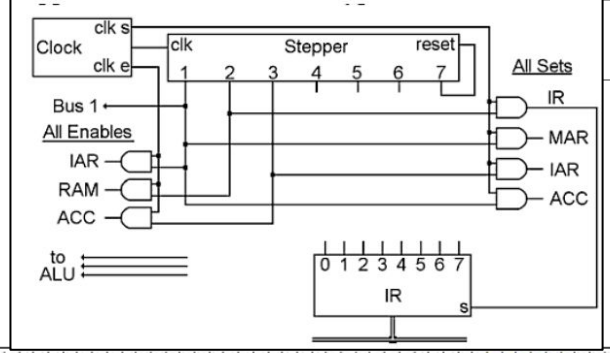
Step 2



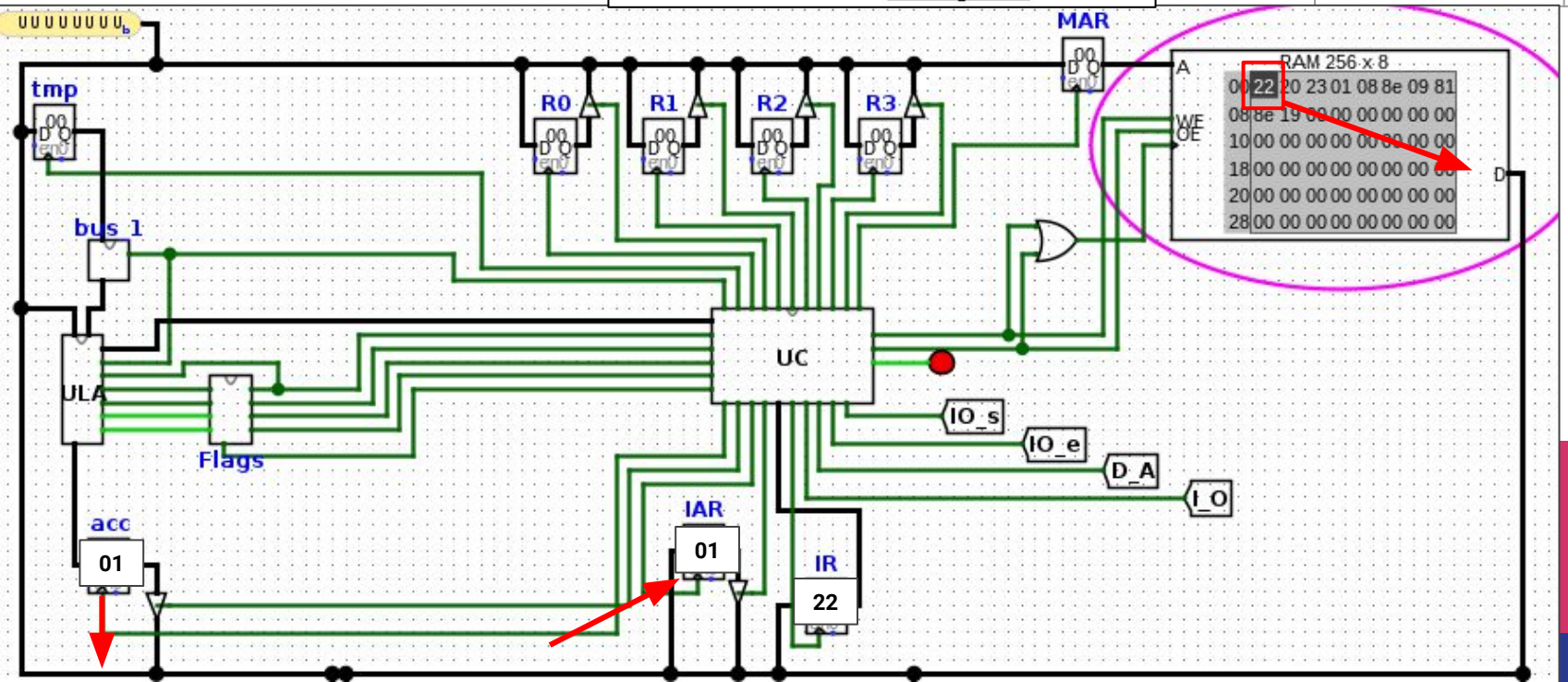
Endereço	Programa	Hexa
0x00 0x01	DATA R2,0x20	22 20
0x02 0x03	DATA R3,0x01	23 01
0x04	LD R2,R0	08
0x05	ADD R3,R2	8E
0x06	LD R2,R1	09
0x07	ADD R0,R1	81
0x08	ADD R3,R2	8E
0x09	ST R2,R1	19



Step 3

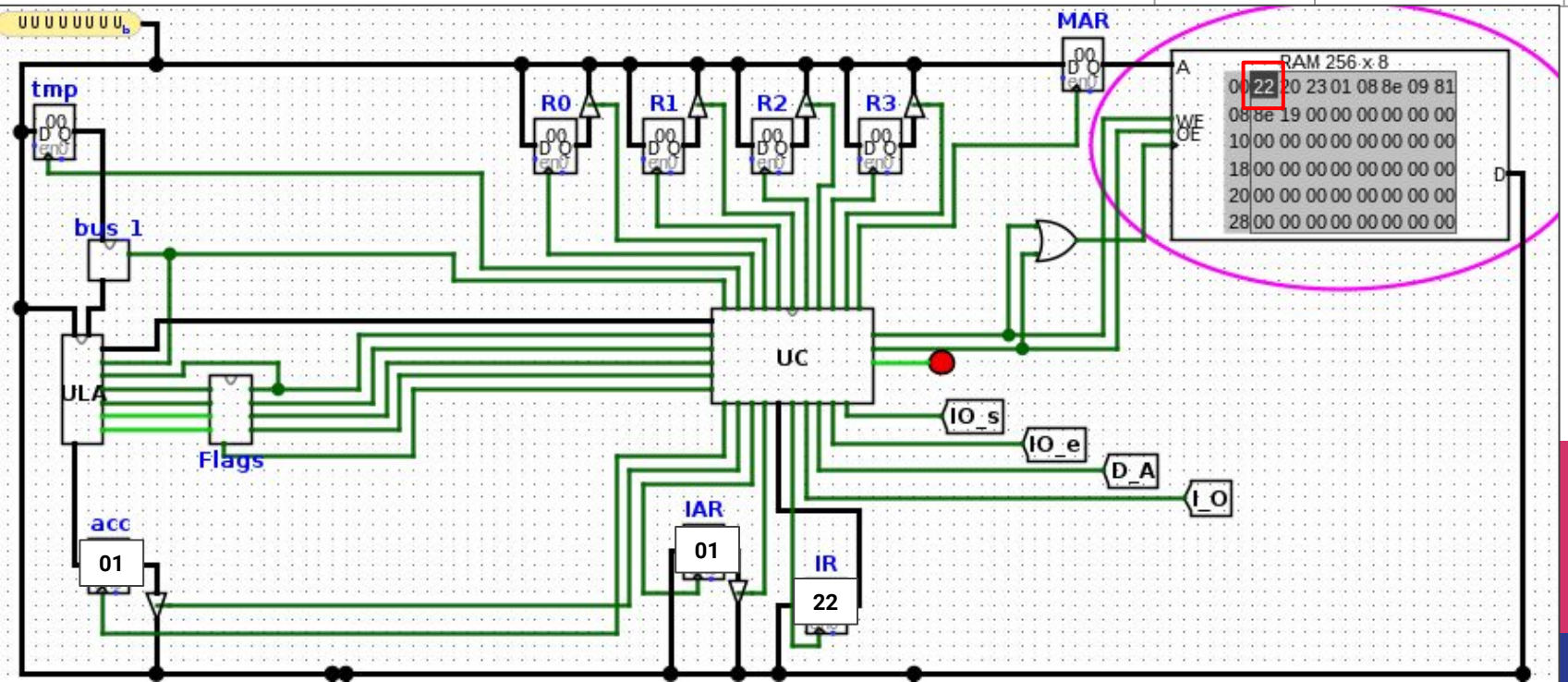


Endereço	Programa	Hexa
0x00 0x01	DATA R2,0x20	22 20
0x02 0x03	DATA R3,0x01	23 01
0x04	LD R2,R0	08
0x05	ADD R3,R2	8E
0x06	LD R2,R1	09
0x07	ADD R0,R1	81
0x08	ADD R3,R2	8E
0x09	ST R2,R1	19



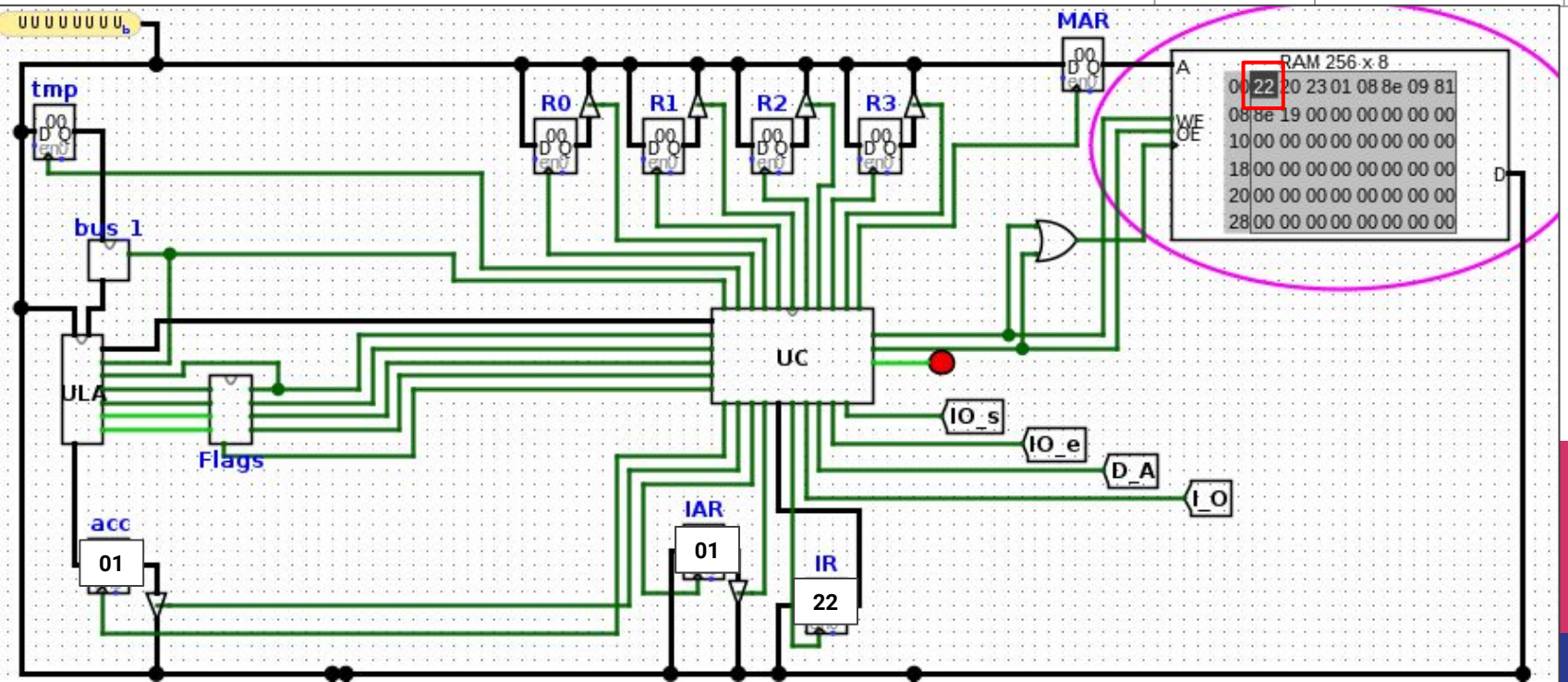
Os step 4, 5 e 6 dependem da instrução que é executa

Endereço	Programa	Hexa
0x00 0x01	DATA R2,0x20	22 20
0x02 0x03	DATA R3,0x01	23 01
0x04	LD R2,R0	08
0x05	ADD R3,R2	8E
0x06	LD R2,R1	09
0x07	ADD R0,R1	81
0x08	ADD R3,R2	8E
0x09	ST R2,R1	19

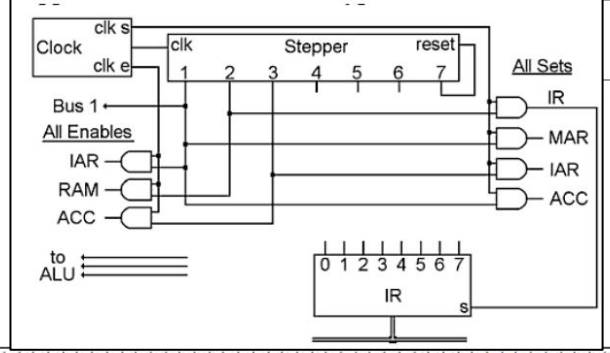


Executar os passos da segunda instrução agora

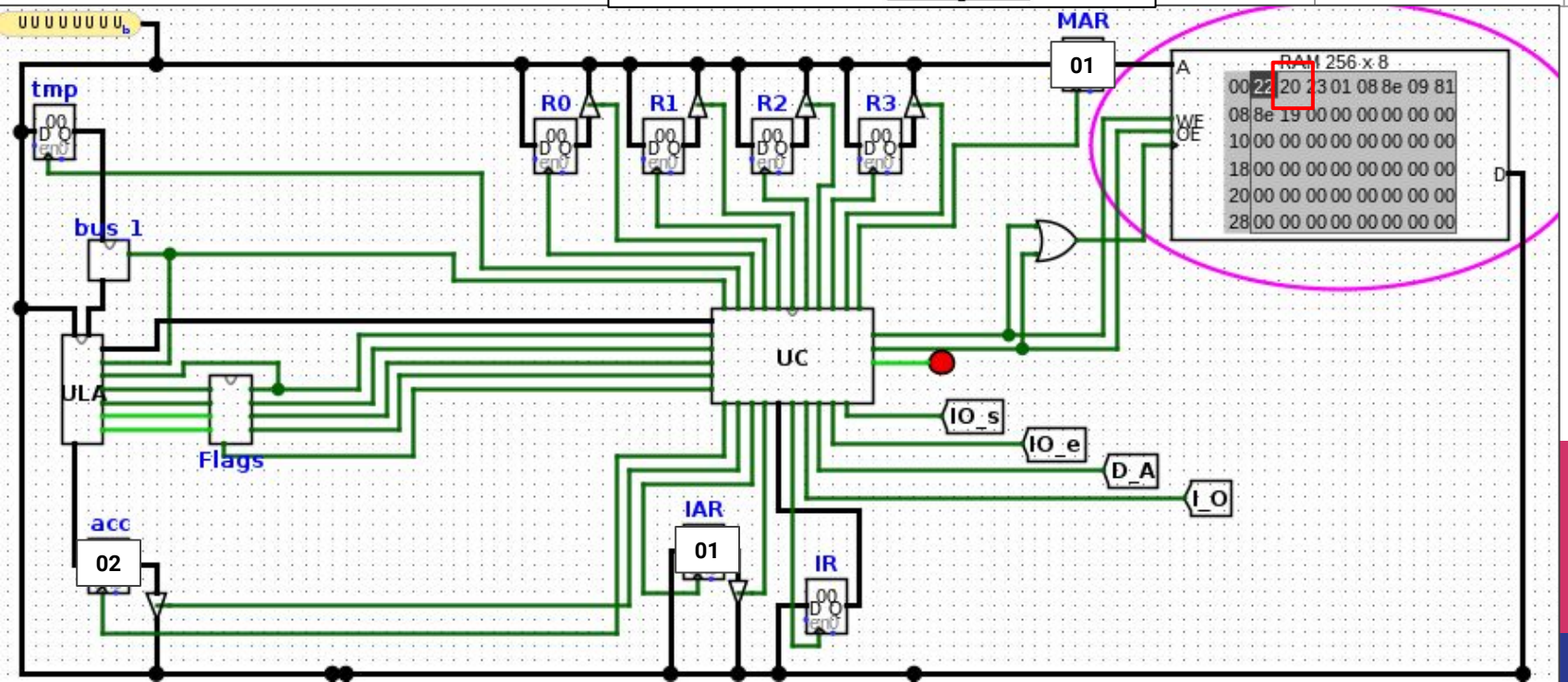
Endereço	Programa	Hexa
0x00 0x01	DATA R2,0x20	22 20
0x02 0x03	DATA R3,0x01	23 01
0x04	LD R2,R0	08
0x05	ADD R3,R2	8E
0x06	LD R2,R1	09
0x07	ADD R0,R1	81
0x08	ADD R3,R2	8E
0x09	ST R2,R1	19



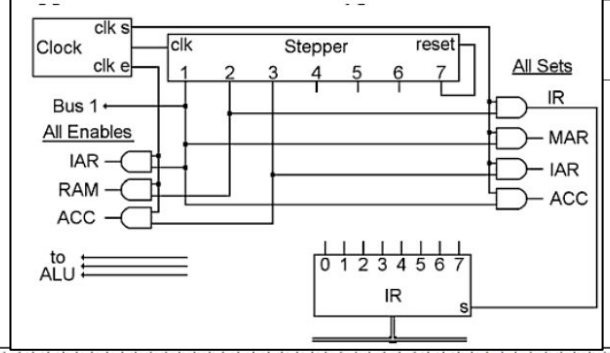
step 1



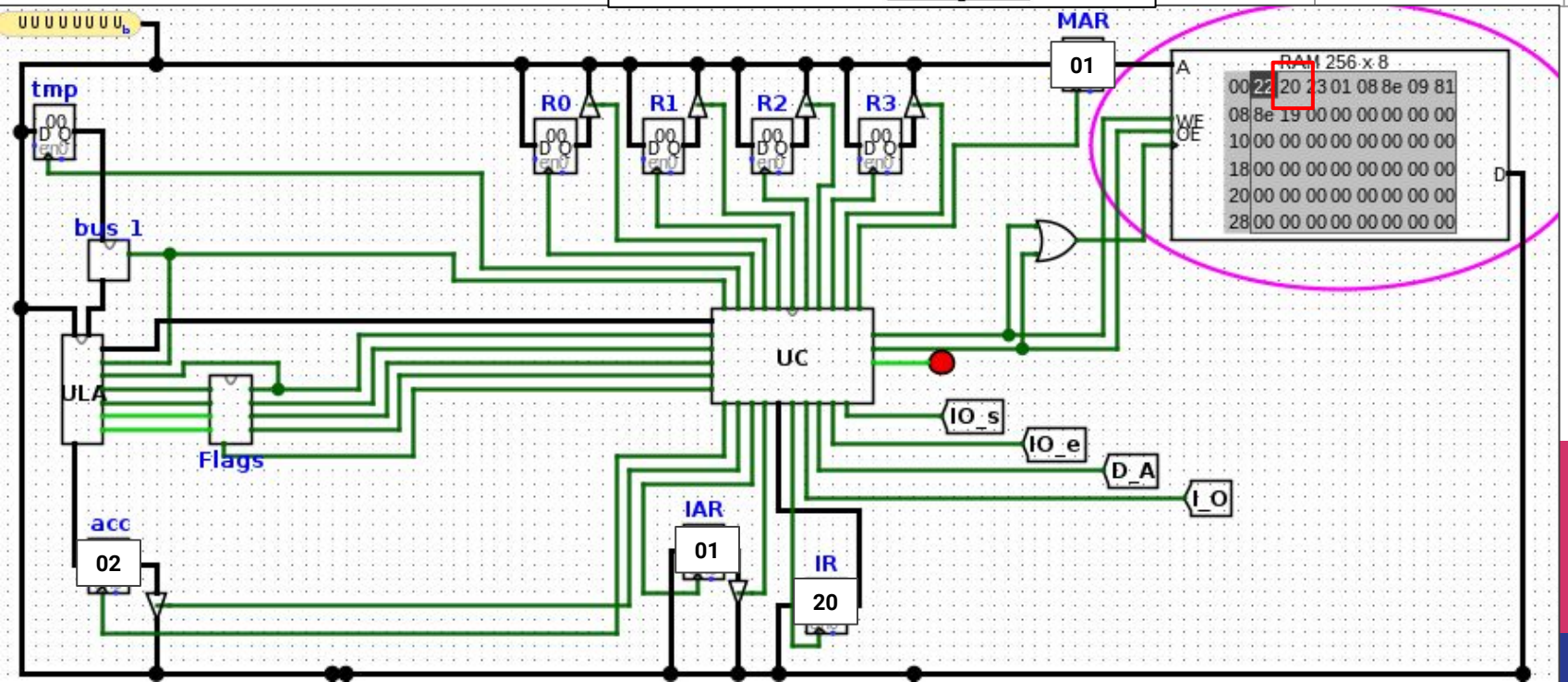
Endereço	Programa	Hexa
0x00 0x01	DATA R2,0x20	22 20
0x02 0x03	DATA R3,0x01	23 01
0x04	LD R2,R0	08
0x05	ADD R3,R2	8E
0x06	LD R2,R1	09
0x07	ADD R0,R1	81
0x08	ADD R3,R2	8E
0x09	ST R2,R1	19



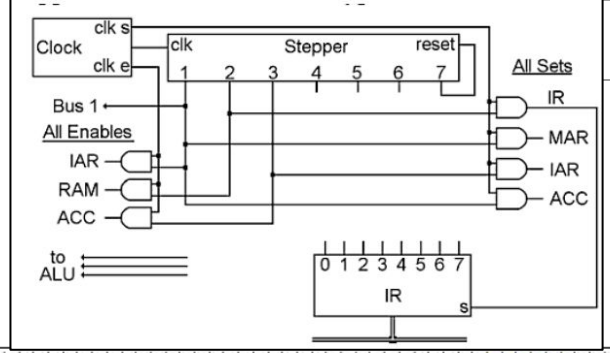
step 2



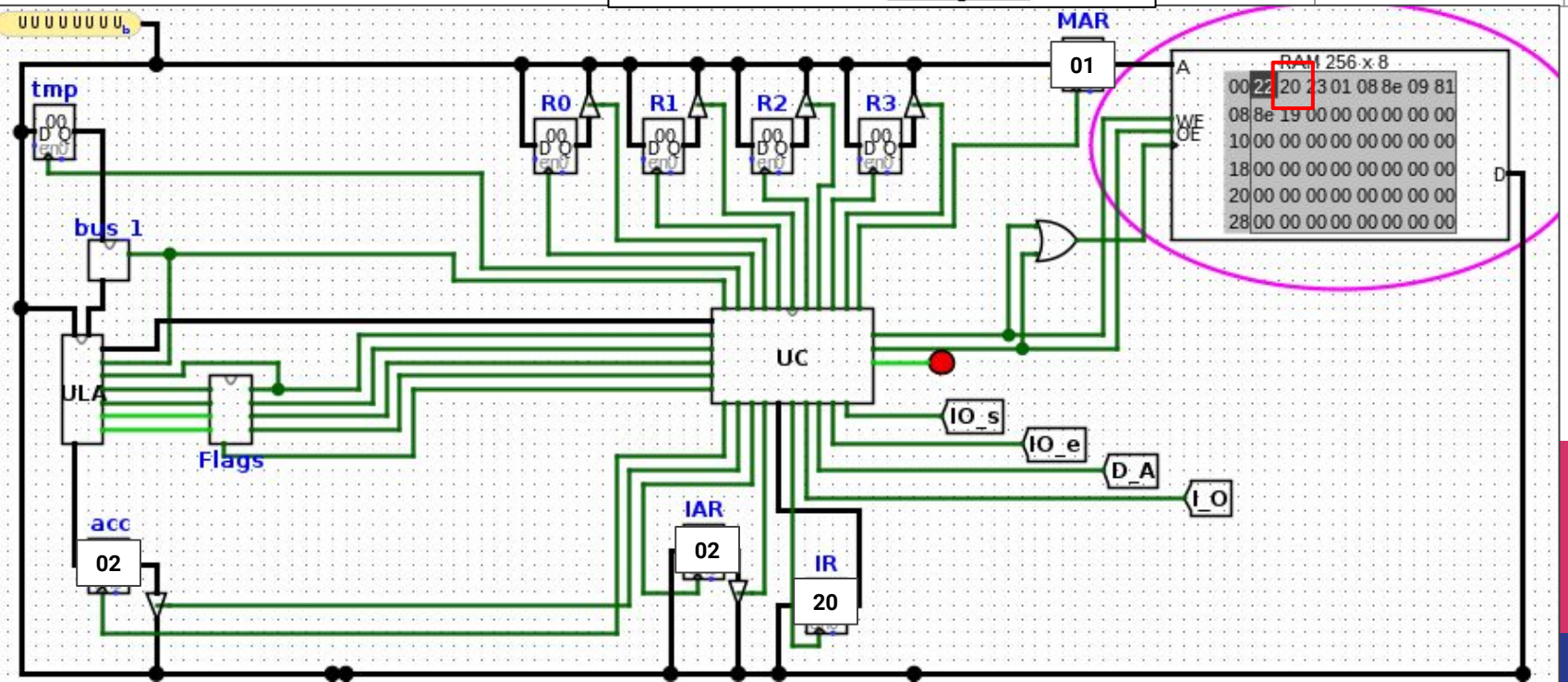
Endereço	Programa	Hexa
0x00 0x01	DATA R2,0x20	22 20
0x02 0x03	DATA R3,0x01	23 01
0x04	LD R2,R0	08
0x05	ADD R3,R2	8E
0x06	LD R2,R1	09
0x07	ADD R0,R1	81
0x08	ADD R3,R2	8E
0x09	ST R2,R1	19



step 3



Endereço	Programa	Hexa
0x00 0x01	DATA R2,0x20	22 20
0x02 0x03	DATA R3,0x01	23 01
0x04	LD R2,R0	08
0x05	ADD R3,R2	8E
0x06	LD R2,R1	09
0x07	ADD R0,R1	81
0x08	ADD R3,R2	8E
0x09	ST R2,R1	19



Atividade

Incorporar o componentes IAR, IR e bus 1 no circuito, enviar a atividade (os únicos elementos que podem faltar são o UC e flag)