



## ENE 111821 - Laboratório de Sistemas Digitais – 2019-1

### Projeto Final - Estacionamento automatizado usando MYCA-2

## Objetivos

- Implementar um Controlador com um Conjunto Fixo de Instruções e Capacidade para Sub-Rotinas. Myca-2
- Implementar memória ROM com rotina de programação.
- Implementar a máquina de refrigerantes usando o Myca-2.

## Projeto

1. Neste mês, a empresa precisa lançar um novo produto. Como a empresa não teve tempo suficiente para desenvolver um produto completamente novo, os diretores de marketing mandaram os engenheiros *requeentarem* um produto antigo, o microcontrolador programável Myca-2, para que eles possam lançá-lo nesse natal como **Myca-2**.

As funcionalidades são similares ao Myca-2. O novo Myca-2, a diferença do antecessor, terá uma instrução com a capacidade de se testar uma condição em um dado estado  $X$  e, baseado neste teste, desviar para uma outra sequência de estados, e finalmente retornar para o estado  $X + 1$ . Em software chamamos isso de *pular para uma sub-rotina*.

Os opcodes esperados no Myca-2S são:

Mnemonic	Opcode	Flag		Instruction Function
		FALSE	TRUE	
HIC	<b>000</b>	$NS = PS$	$NS = PS + 1$	<i>Hold then Increment on Condition</i>
HBC	<b>001</b>	$NS = PS$	$NS = BA$	<i>Hold then Branch on Condition</i>
IBC	<b>100</b>	$NS = PS + 1$	$NS = BA$	<i>Increment if Condition is false, Branch on Condition</i>
IUC	<b>010</b>	$NS = PS + 1$		<i>Increment Unconditionally</i>
BUC	<b>011</b>	$NS = BA$		<i>Branch Unconditionally</i>
BSR	<b>101</b>	$NS = PS + 1$	$NS = SRA$	<i>Branch to Sub-Routine</i>
RSR	<b>110</b>	$NS = BA$	$NS = TOS$	<i>Return from Sub-Routine</i>
LJA	<b>111</b>	$NS = PS + 1$	$NS = JA$	<i>Load Jam Address</i>

**NOTA:** NS = Próximo estado, PS = Estado atual, BA = Branch Address, JA = Jam Address, TOS = Top Of Stack, SRA = Sub-Routine Address.

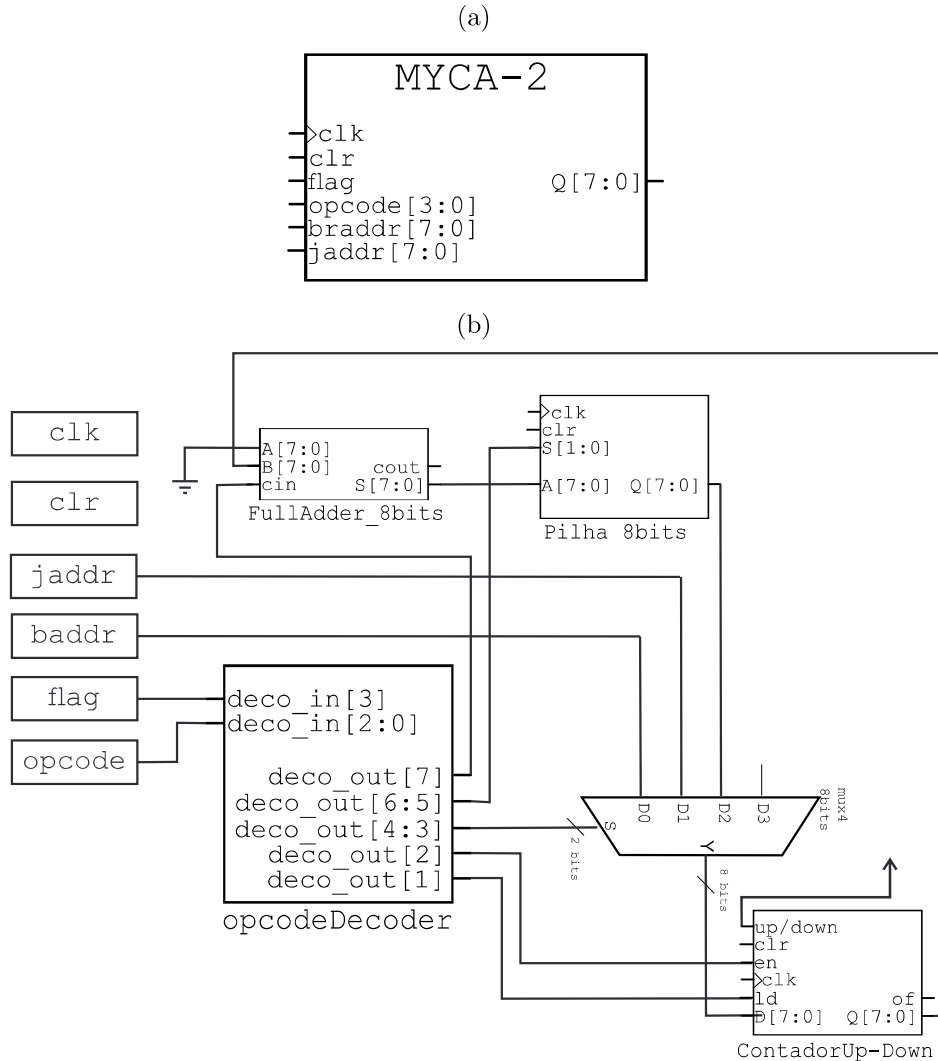


Figure 1: (a) Controlador Myca-2 e (b) Arquitetura interna (não esqueçam de ligar o clk e clr no contador e a pilha)

Note que cada opcode deve definir:

- Como o contador deve contar (**C**ount **E**nable e **L**oad).
- Qual endereço deve ser carregado no contador (Branch Address, Jam Address Input ou Top of Stack, selecionados por **S** do mux).
- Qual operação deve ser feita com a Stack (Push, Pop ou Hold, selecionados por **S** da pilha).
- Se o endereço deve ser incrementado para ser inserido na stack (**C**in).

Projete o microcontrolador Myca-2 utilizando o diagrama apresentado e considerando um PC (*program counter*, contador de programa) de 8 bits.

O conteúdo do *opcodeDecoder* está definido pela seguinte tabela:

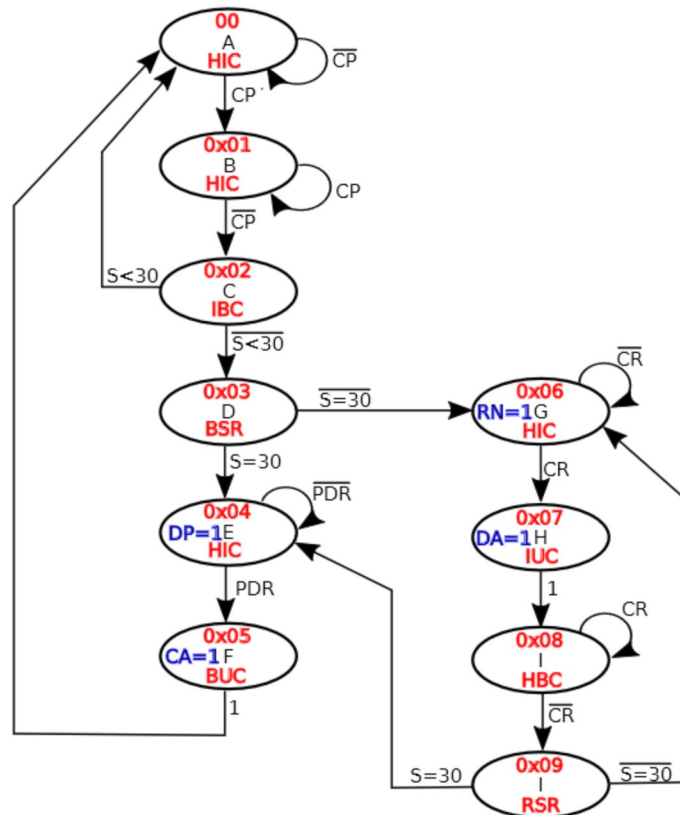


Instruction	deco_in				deco_out						
	OC <sub>2</sub>	OC <sub>1</sub>	OC <sub>0</sub>	FLAG	Counter		Address Mux		Shift Register		Adder Cin
					CE	LD	MS1	MS0	S1	S0	
HIC	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	1	1	0	0	0	0	0	0
HBC	0	0	1	0	0	0	0	0	0	0	0
	0	0	1	1	0	1	0	0	0	0	0
IUC	0	1	0	0	1	0	0	0	0	0	0
	0	1	0	1	1	0	0	0	0	0	0
BUC	0	1	1	0	0	1	0	0	0	0	0
	0	1	1	1	0	1	0	0	0	0	0
IBC	1	0	0	0	1	0	0	0	0	0	0
	1	0	0	1	0	1	0	0	0	0	0
BSR	1	0	1	0	1	0	0	0	0	0	0
	1	0	1	1	0	1	0	0	0	1	1
RSR	1	1	0	0	0	1	0	0	0	0	0
	1	1	0	1	0	1	1	0	1	0	0
LJA	1	1	1	0	1	0	0	0	0	0	0
	1	1	1	1	0	1	0	1	0	1	1

Para esta questão, deverá ser apresentado no relatório:

(a) [4 pts] Código VHDL e Esquemático RTL do Myca-2.

2. Use o Myca-2 para implementar a mesma máquina de refrigerantes do experimento 6. Para isto, considere o seguinte diagrama de estados:





É obrigatório que seja possível ver a linha da ROM atual em um display 7 segmentos. Da mesma forma também deve ser possível visualizar o opcode atual no display restante. Deve ficar desta forma:

```
input_1 <= preco_refri;  
input_2 <= contador_moeda;  
input_3 <= ROM_addr(3 downto 0);  
input_4 <= '0' & opcode;
```

Para esta questão, deverá ser apresentado no relatório:

- (a) [2 pts] Código VHDL e Esquemático RTL do sistema da maquina de refri com Myca-2.
- (b) [2 pts] Diagrama de estados dos sistema.
- (c) [2 pts] Rotina usando os mnemônicos e o conteúdo da **ROM4MB**.

#### NOTAS:

- O relatório deste projeto devera ser entregue junto com o visto.
- De uma olhada no exemplo do Myca-1 feito em aula pelo professor.
- Basicamente o que muda com respeito ao experimento 6 é que o *ControllerFSM* agora é trocado pelo Myca-2.
- Lembre que o Myca-2 apenas pode avaliar um sinal por estado. Você deve adiciona um mux para mexer com as entradas.
- De uma olhada nas aulas 23 e 24 da teoria de sistemas digitais para saber mais sobre o Myca.

## Vistos

1. [10 pts] Amostre a implementação do sistema da questão 2.