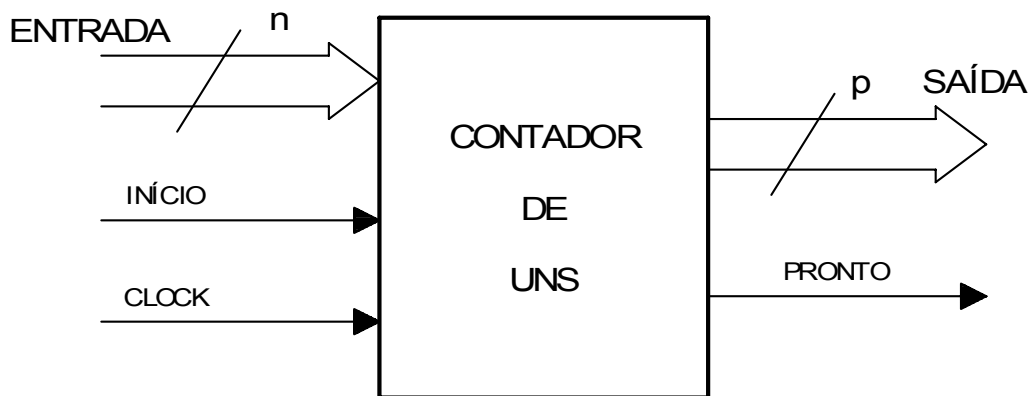


## Exemplo 2 de Projeto de Circuito Síncrono

### Contador de Uns

(Resumo elaborado por Edith Ranzini, a partir do exemplo extraído do livro GAJSKI, DANIEL D. Principles of Digital Design - 1997 com algumas alterações)

#### 1. Especificação



Após o acionamento do botão Início, o circuito deve mostrar, na SAÍDA, o código binário do número de bits iguais a UM presentes na ENTRADA. ( ENTRADA com  $n$  bits e SAÍDA com  $p$  bits, sendo  $2^p \geq n$ )

O sinal PRONTO deve ser acionado após a operação ser completada e deve durar apenas 1 ciclo de CLOCK.

O resultado deve ficar presente na SAÍDA até uma nova operação ser iniciada.

#### 2. Encaminhamento da Solução Adotada

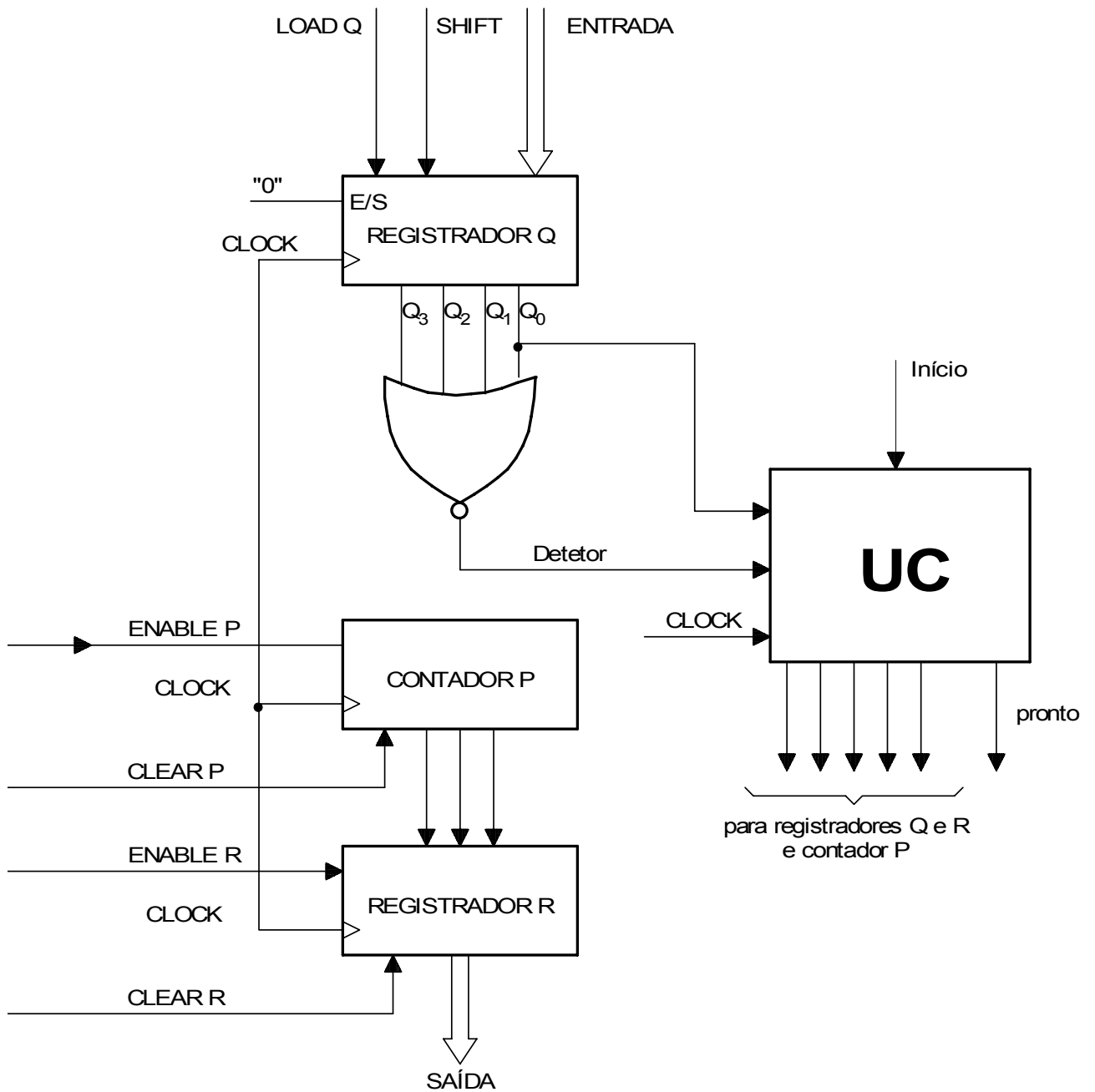
Ao receber o comando Início, carregar ENTRADA num registrador de deslocamento à direita (Q), com entrada serial = 0. A cada deslocamento, examinar o bit menos significativo ( $Q_0$ ), incrementando um Contador P, sempre que  $Q_0 = 1$ .

A operação estará completada sempre que, na saída paralela do registrador Q, todos os bits foram iguais a zero. (Dessa forma, elimina-se presença de um 2º contador, que controlaria a realização do número necessário de deslocamentos).

O resultado (saída do contador P) deve ser armazenado num registrador R.

### 3. Projeto, para n = 4

#### 3.1- Diagrama em Blocos



Hipótese: Todos os sinais de comando (Load, Clear, Shift, Enable) são síncronos e ativos no nível 1.

Figura 1 – Contador de UNS

### 3.2- Projeto da UC

a) Diagrama ASM – Solução MOORE

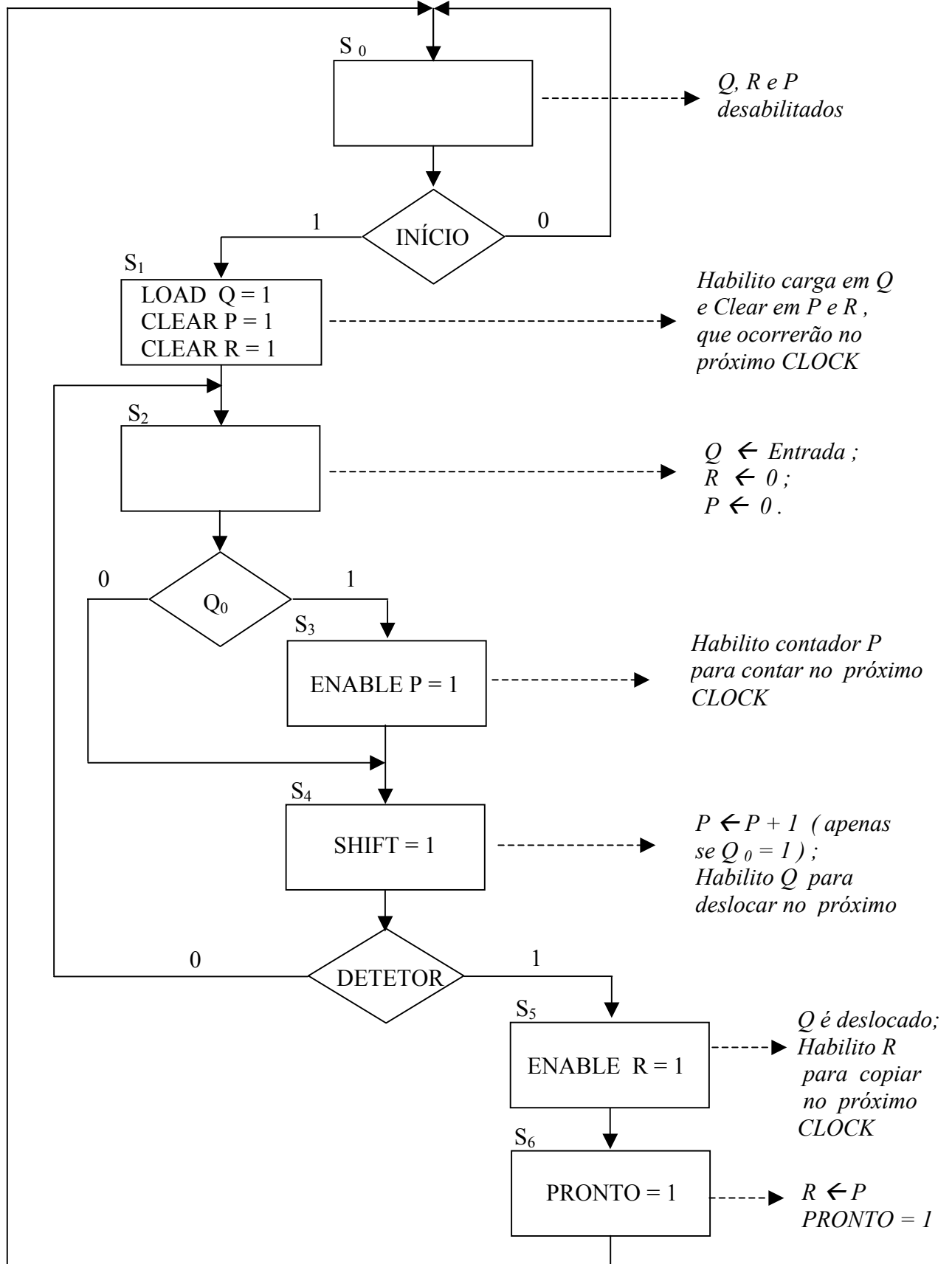


Figura 2 – Contador de UNS (Moore)

b) Diagrama ASM – Solução Mealy

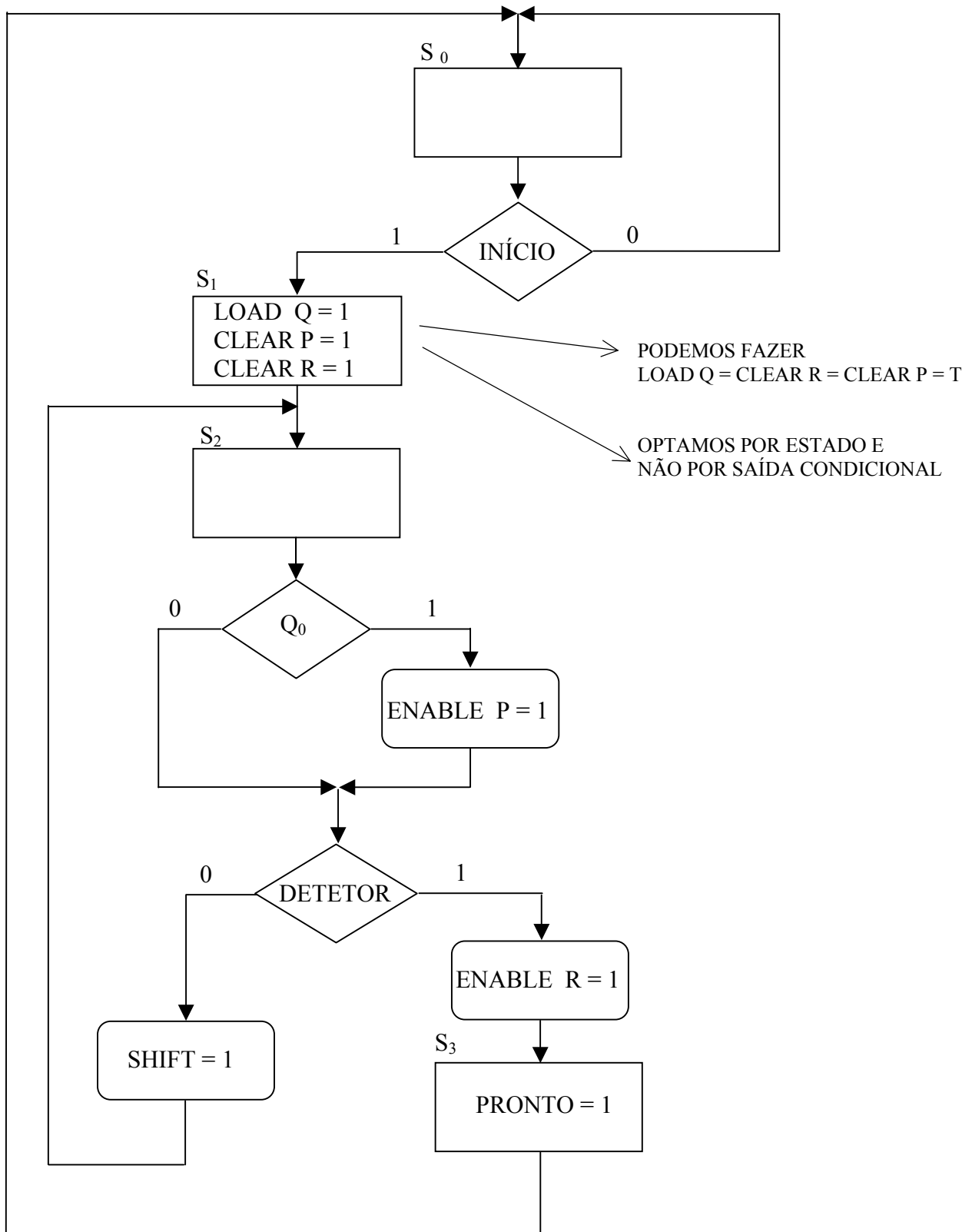
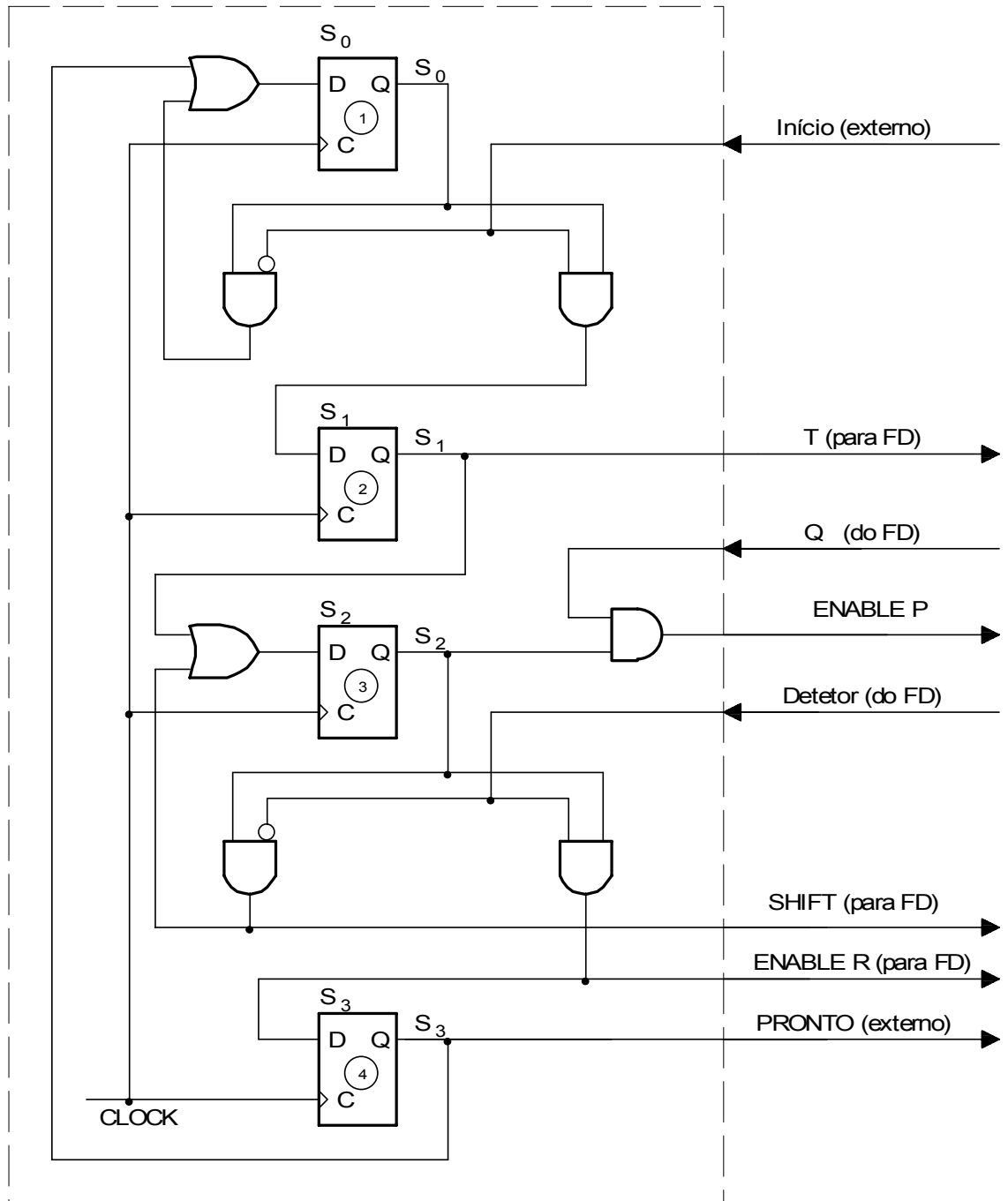


Figura 3 – Contador de UNS (Mealy)

### 4. Implementação da UC (apenas para a solução MEALY)

a) Com UM Flip-Flop por estado



Condição inicial :    FF ① - 1  
 (ao ligar)            FF ② - 0  
                           FF ③ - 0  
                           FF ④ - 0

Figura 4 – Circuito obtido por mapeamento direto do ASM da fig.3

b) Método Clássico (4 estados → 2 Flip-flops)

Tabela

Estado atual		Próximo Estado		Saídas	
		Condições	Prox. Estado	Condições	Saídas
Nome	y <sub>1</sub> y <sub>0</sub>	Detetor Início	y <sub>1</sub> y <sub>0</sub>	Q <sub>0</sub> Detetor	
S <sub>0</sub>	0 0	X 0	0 0	X X	-----
	0 0	X 1	0 1	X X	-----
S <sub>1</sub>	0 1	X X	1 0	X X	T=1
S <sub>2</sub>	1 0	0 X	1 0	1 X	ENABLE P = 1
	1 0	1 X	1 1	X 0 X 1	SHIFT = 1 ENABLE R=1
S <sub>3</sub>	1 1	X X	0 0	X X	PRONTO=1

Equações

$$\begin{cases} D_0 = y_1' y_0' (\text{Início}) + y_1 y_0' (\text{Detetor}) \\ D_1 = y_1' y_0 + y_1 y_0' \end{cases}$$

$$\begin{cases} T = y_1 y_0' (\text{LOADQ=CLEARP=CLEARR}) \\ \text{ENABLE P} = y_1 y_0' Q_0 \\ \text{SHIFT} = y_1 y_0' (\text{Detetor})' \\ \text{ENABLE R} = y_1 y_0' (\text{Detetor}) \\ \text{PRONTO} = y_1 y_0 \end{cases}$$

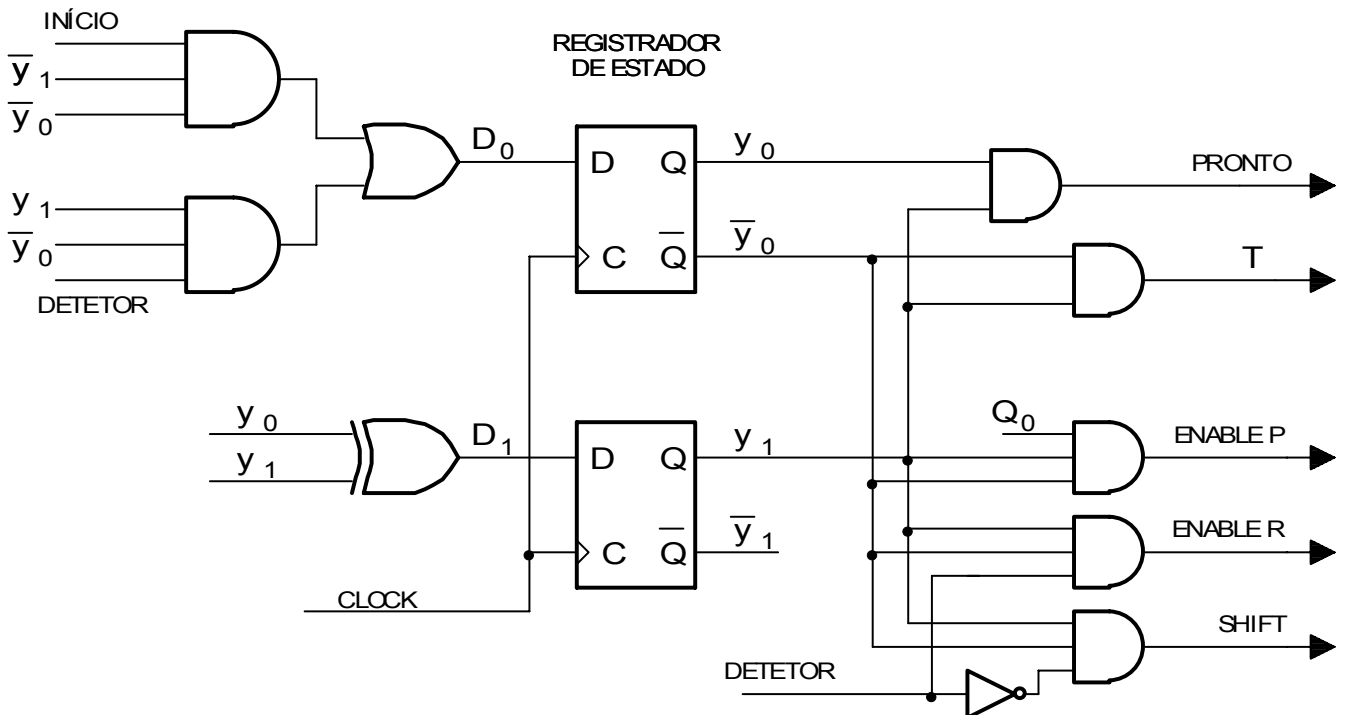


Figura 5 - Método clássico - Circuito

c) Modelo Registrador de estados → decodificador

Tabela de transição

Estado atual		Próximo Estado		Decodificador			
		Condições	Prox. Estado				
Nome	y <sub>1</sub> y <sub>0</sub>	Detetor Início	y <sub>1</sub> y <sub>0</sub>	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>
S <sub>0</sub>	0 0	X 0	0 0	1	0	0	0
	0 0	X 1	0 1	1	0	0	0
S <sub>1</sub>	0 1	X X	1 0	0	1	0	0
S <sub>2</sub>	1 0	0 X	1 0	0	0	1	0
	1 0	1 X	1 1	0	0	1	0
S <sub>3</sub>	1 1	X X	0 0	0	0	0	1

Equações

$$\begin{cases} D_0(t) = y_0(t+1) = S_0 \text{ Início} + S_2 \text{ Detetor} \\ D_1(t) = y_1(t+1) = S_1 + S_2 \end{cases}$$

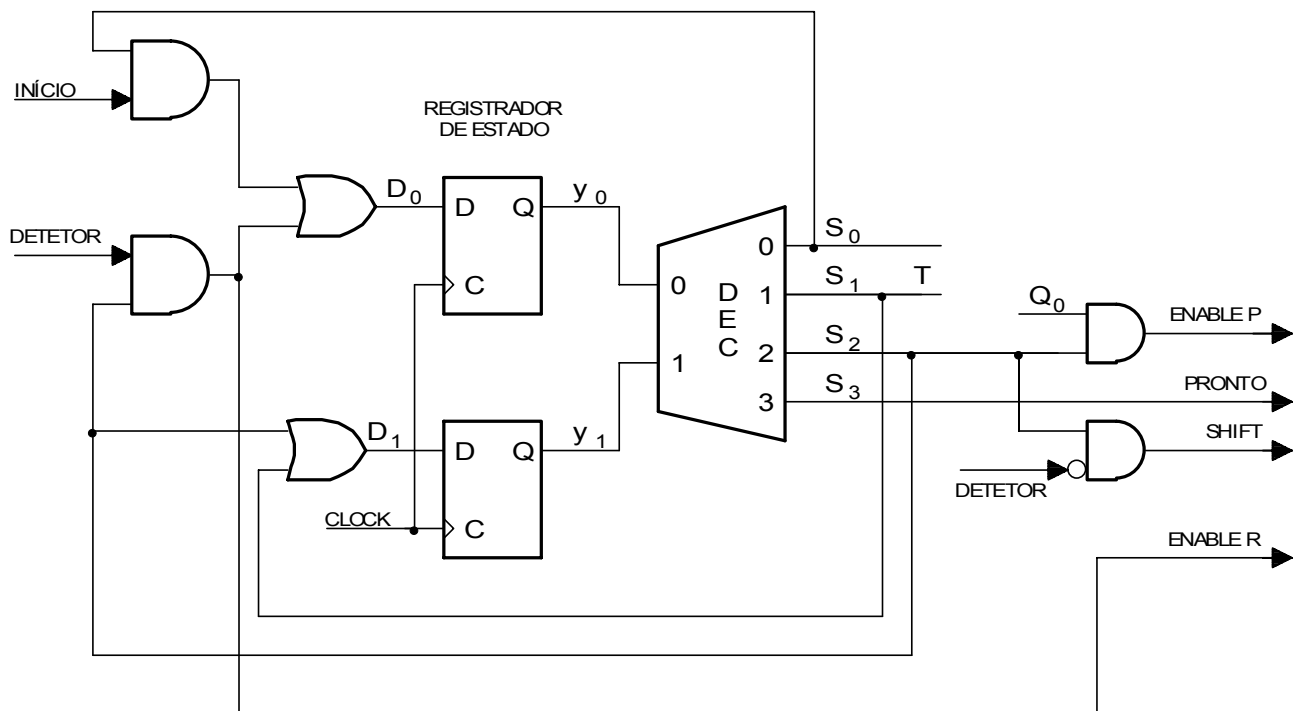


Figura 6 - Circuito com registrador de estados - decodificador

## d) Modelos alternativos

As partes combinatórias dos circuitos apresentados nas figuras 5 e 6 podem também ser implementados com Multiplexadores, PALs OU EPROMs.

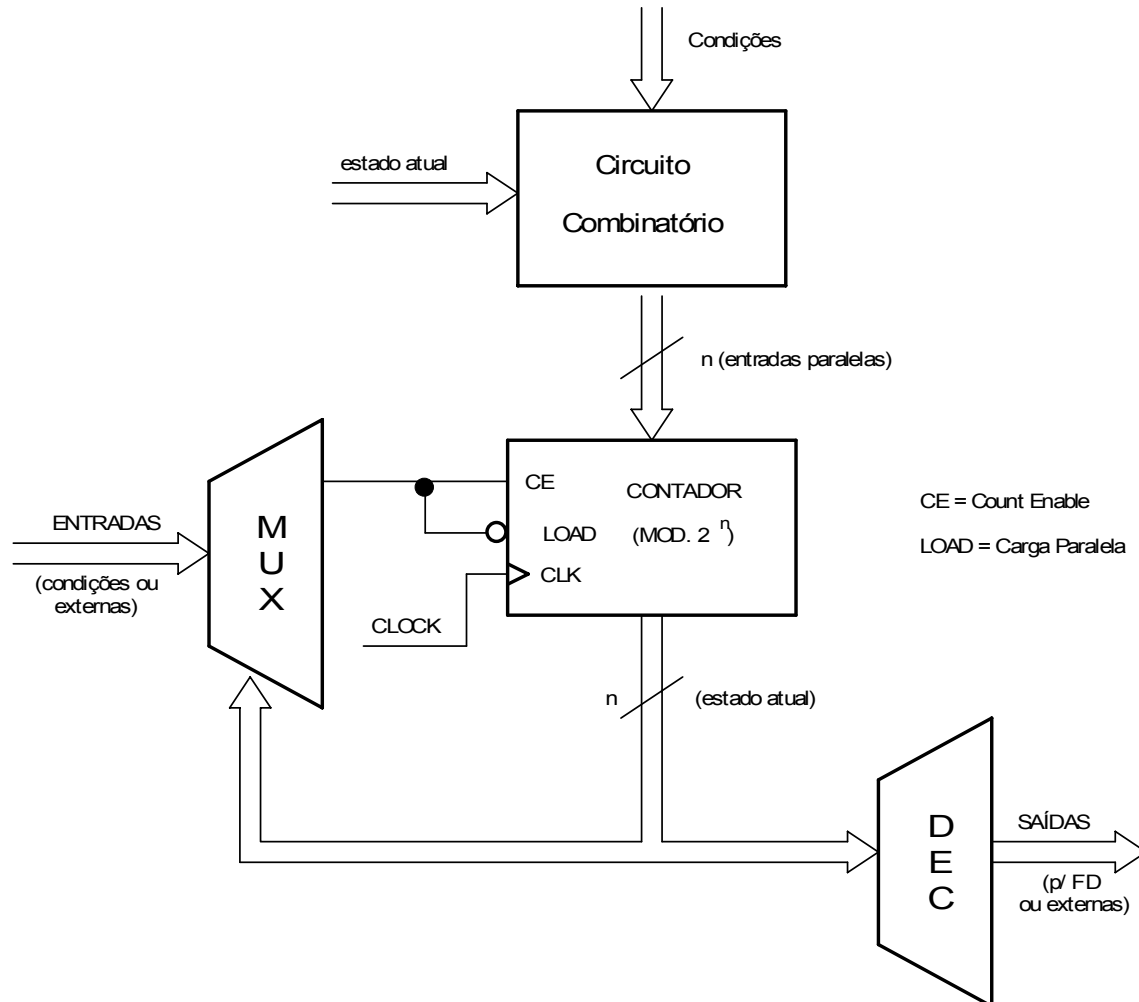
Além disso, os circuitos das figuras 4 e 5 podem ser constituídos com uma PAL com registrador (do tipo PAL y Rx).

A medida que o número de estados cresce, pode ser conveniente usar um contador ou um registrador de deslocamento como "registrador de estado". Nesse caso, as condições que determinam o próximo estado são usadas para gerar os sinais que selecionam a função a ser executada pelo contador, (por exemplo, contar, carregar, zerar, inibir) ou pelo registrador de deslocamento, (por exemplo, deslocar, carregar, inibir) quando ocorrer o próximo CLOCK. Apenas para exemplificar, se fosse usado um contador (não flip-flops isolados), **a tabela de transição do item c) seria modificada para:**

Estado atual			Próximo Estado				Decodificador				Contador
			Condições		Prox. Estado						
Nome	y <sub>1</sub>	y <sub>0</sub>	Detetor Início		y <sub>1</sub>	y <sub>0</sub>	S <sub>0</sub>	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	Função
S <sub>0</sub>	0	0	X	0	0	0	1	0	0	0	INIBE
	0	0	X	1	0	1	1	0	0	0	CONTA
S <sub>1</sub>	0	1	X	X	1	0	0	1	0	0	CONTA
S <sub>2</sub>	1	0	0	X	1	0	0	0	1	0	INIBE
	1	0	1	X	1	1	0	0	1	0	CONTA
S <sub>3</sub>	1	1	X	X	0	0	0	0	0	1	CONTA

Como nesse exemplo só temos 4 estados, não é possível mostrar nenhuma vantagem em substituir 2 flip-flops por um contador de 2 bits.

Em circuitos maiores, pode ser vantajosa a utilização de contadores que, associados a decodificadores e multiplexadores permitem a obtenção de circuitos como o esquematizado na figura 7.



**Figura 7 - Esquema de UC tipo MOORE, com CONTADOR-DEC e MUX**

Obviamente, a utilização de contadores é vantajosa quando a seqüência de estados a ser percorrida explora ao máximo a função de "contar" do contador e, só em algumas transições entre estados recorreu-se à função de carga paralela. Se não for assim, o circuito combinatório para gerar as entradas de carga paralela será muito complexo.

Soluções usando registradores de deslocamento, no lugar de contadores, são análogas. Também vale a mesma observação sobre explorar ao máximo a função de "deslocar".

e) Outras soluções

Quanto a complexidade da UC (e/ou da UC+FD) é muito grande, há outras alternativas:

- Microprogramação (no caso da UC de unidades centrais de processamento)
- Utilização de microcontroladores
- Utilização de linguagens de descrição de hardware e simuladores e implementação com dispositivos lógicos programáveis complexos (CPLD).

Nas disciplinas PCS2308 e PCS2355, esse circuito do Contador de Uns (ou um muito parecido) será projetado e montado utilizando uma ferramenta computacional e dispositivos lógicos programáveis.