

Projeto #1

FSC / L.EIC
Adaptado de Vasco Mela

1. FSM State Diagram

- Dedução do algoritmo

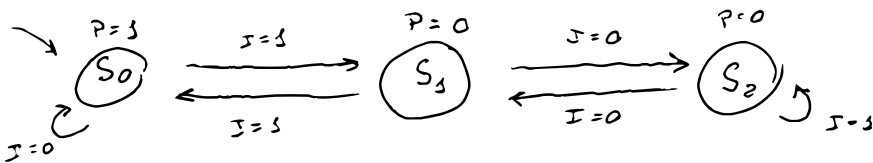
Introduzir:	Valor inicial	Resultado
1	x	$2x + 1$
0		$2x$
1	$1 \bmod 3$	$0 \bmod 3$
0		$2 \bmod 3$
1	$2 \bmod 3$	$2 \bmod 3$
0		$1 \bmod 3$
1	$0 \bmod 3$	$1 \bmod 3$
0		$0 \bmod 3$

Ex

	Valor	mod 3
(1)	1	1
(2)	1 0	2
(4)	1 0 0	1
(17)	1 0 0 1	2
(35)	1 0 0 1 1	2
...		
(1)	1	1
(3)	1 1	0
(6)	1 1 0	0
(12)	1 1 0 0	0
(25)	1 1 0 0 1	1

$0 \bmod 3 = 3 \bmod 3 = 0$

- Máquina de estados



- Legenda:

S_0 | Estado "0 mod 3"
I | "Input"

S_1 | Estado "1 mod 3"
P | "Output"

S_2 | Estado "2 mod 3"

2. State encoding

- No mínimo de FF

É impossível simplificar o n° de estados do diagrama, portanto o necessário, no mínimo, 2 flip-flops para representar os 3 estados.

⊛ $2^1 = 2 < 3$, mas $2^2 = 4 \geq 3$

- Codificação de estados

Por exemplo:

$$S_0 = \begin{matrix} Q_1 & Q_0 \\ 0 & 0 \end{matrix}$$

(0 mod 3)

$$S_1 = \begin{matrix} Q_1 & Q_0 \\ 0 & 1 \end{matrix}$$

(1 mod 3)

$$S_2 = \begin{matrix} Q_1 & Q_0 \\ 1 & 0 \end{matrix}$$

(2 mod 3)

⊛ O estado 3 não será tido em conta; assume-se como impossível.

3. State transition functions

Input	Estado atual		Próx. estado		Output
I	Q ₁	Q ₀	D ₁	D ₀	P
1	1	1	X	X	X
1	1	0	1	0	0
1	0	1	0	0	0
1	0	0	0	1	1
0	1	1	X	X	X
0	1	0	0	1	0
0	0	1	1	0	0
0	0	0	0	0	1

⊛ Implementação Moore

⊛ D₁, D₀ representam o próximo estado

⊛ X = "Don't care"

4. Simplified algebraic representation

• D_1 (2º bit do próximo estado)

Q_1, Q_0

	00	01	11	10
0	0	1	X	0
1	0	0	X	1

$$D_1 = \overline{I} \cdot Q_0 + I \cdot Q_1$$

• D_0 (1º bit do próximo estado)

Q_1, Q_0

	00	01	11	10
0	0	0	X	1
1	1	0	X	0

$$D_0 = I \cdot \overline{Q_0} \cdot \overline{Q_1} + \overline{I} \cdot Q_1$$

5. Next part function

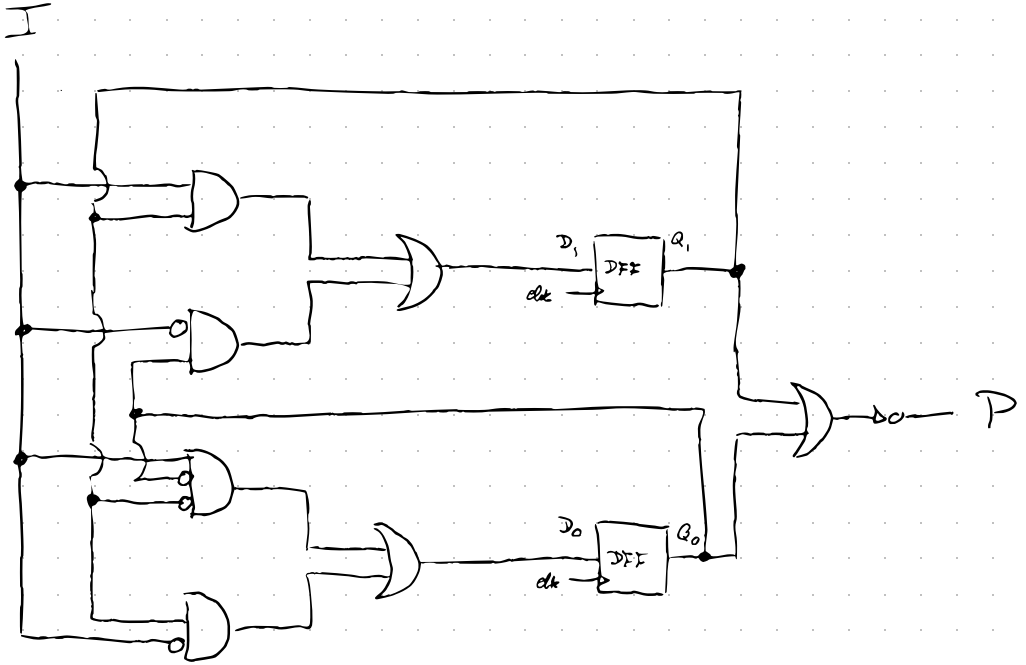
Q_1, Q_0

	00	01	11	10
0	1	0	X	0
1	1	0	X	0

$$P = \overline{Q_1} \cdot \overline{Q_0}$$

$$\text{aux} = \overline{(Q_1 + Q_0)}$$

6. Implementation schematic



Projeto #2

FSC 12. ETC

1. Análise e simulação de um programa

- a) $0 \times 5005005e$ ($50050000 + 5e$)
- b) $0 \times 5005000e$ (4ª posição)
- c) 28 bytes (7×4 bytes)
- d) 7 vezes
- e) 6, -50, 42, 4, -20, 500 e 58
- f) 6, -50, 42, 2, -50, 50 e 9
- g) Multiplica por 8 todas as elementos armazenadas em memória

2. Extração das componentes de um número

⊕ ao 1 Número IEEE

• sinal :

srli a0, a0, 31
ret

• expoente real :

srli a0, a0, 23
andi a0, a0, 0x000000FF
addi a0, a0, -127
ret

• mantissa :


li t0, 0x007FFFFFFF
li t3, 0x00800000
and a0, a0, t0
or a0, a0, t3
ret

3. Adição e subtração de dois números

- soma:
- subtração:

4. Multiplicação entre dois números

- multiplica:


$$\begin{array}{r} 5.00505 \\ + 5.09505 \\ \hline 10.10010 \end{array}$$