

Prática 7: Máquina de Estados para identificar código Morse

Nesta prática iremos desenvolver uma máquina de estados para identificar letras do código morse. **Baixe no moodle os arquivos de apoio para a prática**

① Máquina de Estados Finitos:

A figura 1 ilustra o diagrama de blocos de uma máquina de estados. Temos três estágios:

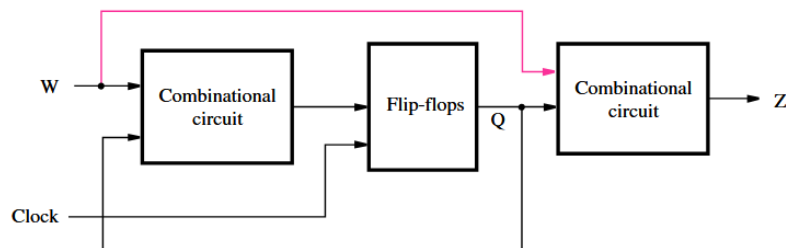


Figura 1: Estrutura de uma máquina de estado

- 1.) Circuito combinacional de entrada: responsável por definir qual o **próximo estado** (Y_n)
- 2.) Circuito sequencial: armazena o **estado atual** (y_n) da FSM
- 3.) Circuito combinacional de saída: controla qual o valor da saída, de acordo com a entrada e o estado atual

Vamos analisar cada uma destas etapas, e como implementá-las em Verilog para identificar caracteres em código Morse:

② Circuito Combinacional de Entrada:

A entrada da FSM é simbolizada por pontos e traços, onde, de acordo com a sequência transmitida, uma letra é identificada como ilustra a figura 2. Desta forma, teremos **26 estados**, um estado simbolizando cada uma das letras e simbolizaremos o **ponto como 1 e o traço como 0**

Sendo assim, de acordo com o bit da entrada e o estado atual da FSM, podemos identificar qual a próxima letra.

A ● -	J ● - - -	S ● ● ●
B - ● ● ●	K - ● -	T -
C - ● - ●	L ● - ● ●	U ● ● -
D - ● ●	M - -	V ● ● ● -
E ●	N - ●	W ● - -
F ● ● - ●	O - - -	X - ● ● -
G - - ●	P ● - - ●	Y - ● - -
H ● ● ● ●	Q - - ● -	Z - - ● ●
I ● ●	R ● - ●	

Figura 2: Código Morse

- 3) **Circuito Sequencial:** De acordo com o estado atual, o circuito sequencial, na **borda do clock**, realiza a transição entre o estado atual e o próximo estado, seguindo o esquema do diagrama de transição a seguir:

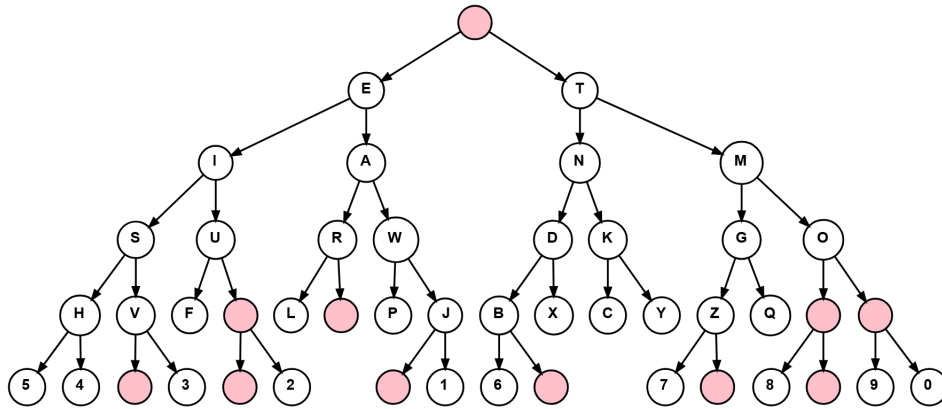


Figura 3: Diagrama de Transição para os estados da FSM

4) **Circuito Combinacional de Saída:**

Por fim, de acordo com o estado atual da FSM, utilizarem um circuito combinacional para converter o estado em um caractere ASCII de acordo com a letra a ser identificada.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Figura 4: Tabela ASCII

5) **Tarefas:**

- Desenvolva em Verilog um módulo que identifique letras em código morse e faça a conversão para ASCII. Mostre o resultado no GTKWave e anexe no Moodle os códigos desenvolvidos.