

Projeto de Lógica Sequencial

S. W. Song

MAC 412 - Organização de Computadores

Lógica Sequencial

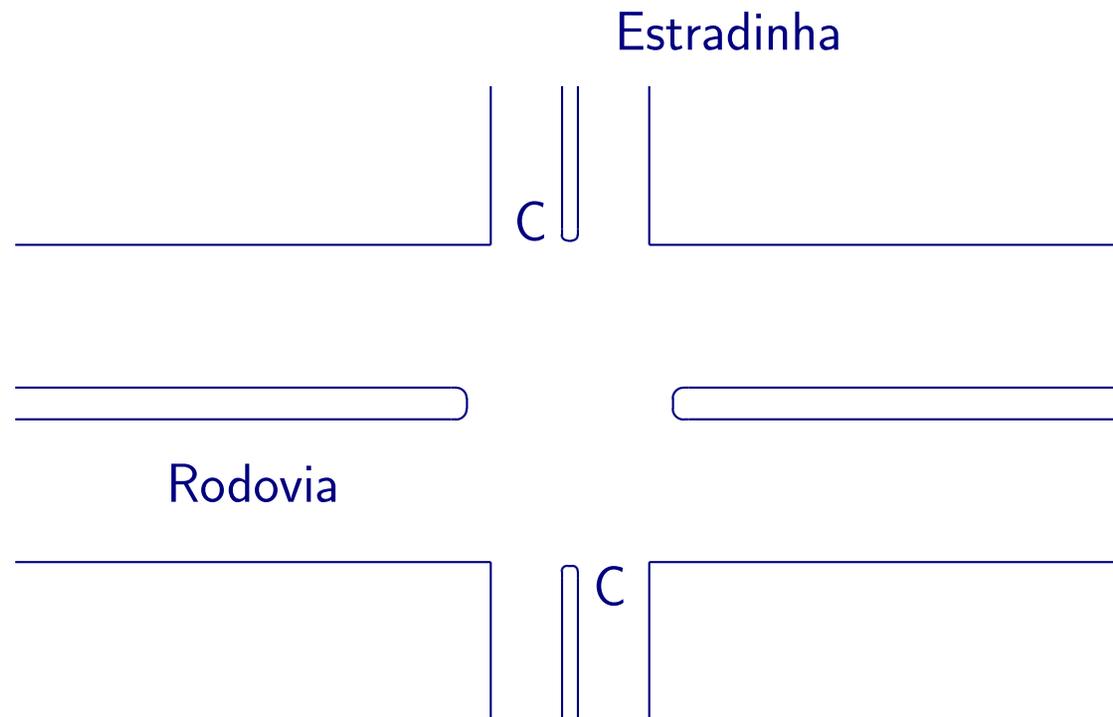
Esse assunto já foi visto em Álgebra Booleana.

Aqui vamos apenas tratar de alguns tópicos específicos.

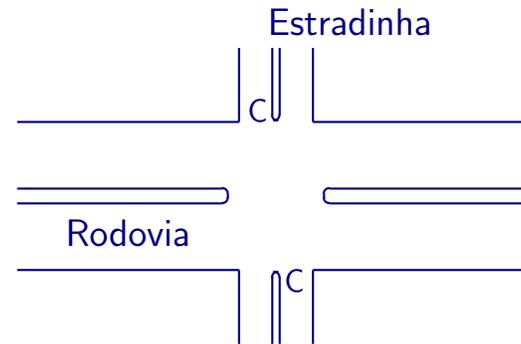
- Projeto de lógica sequencial usando diagrama de estados finitos e a sua implementação usando uma PLA.
- Veremos um exemplo: projeto de um circuito de lógica sequencial para controlar os semáforos de um cruzamento.

Projeto de semáforos

Uma **rodovia** expressa é intersectada por uma **estradinha** pouco movimentada. Um semáforo SemRod controla o tráfego na rodovia e um semáforo SemEst controla o da estradinha.



Funcionamento



- Se $C = 0$ (não detecta carros), o semáforo SemRod deve permanecer **verde**.
- Se $C = 1$, SemRod deve passar para **amarelo**, permanece amarelo por um tempo de duração T_{curto} , e então passa para **vermelho**. (É claro que aí SemEst deve mudar para **verde**.)
- SemEst permanece **verde** apenas se C continua 1, mas nunca por um período superior a T_{longo} .
- Passado este período T_{longo} , ou C ficar 0, SemEst deve mudar para **amarelo**, por uma duração T_{curto} , depois para **vermelho**, quando então SemRod passa novamente para **verde**.
- SemRod só pode mudar de estado decorrido tempo igual a T_{longo} .

Entradas

As entradas são:

- C (1 bit): sensor de carro na estradinha. Vale 1 quando há carros e 0 caso contrário.
- TC e TL (1 bit cada):
Usamos um temporizador (*timer*) que, uma vez disparado com o sinal DISPARA, emite um sinal $TC = 1$ quando ultrapassar um período igual a T_{curto} e emite um sinal $TL = 1$ quando ultrapassar T_{longo} .
- O estado presente (veremos a seguir).

Saídas

As saídas são:

- DISPARA (1 bit): quando vale 1 dispara o temporizador.
- SemRod e SemEst (2 bits cada): as cores dos semáforos.
- O estado próximo (veremos a seguir).

Quatro Estados

- RodVerde
- RodAmarelo
- EstVerde
- EstAmarelo

Estando num estado e conforme as entradas, muda-se para o próximo estado (que pode ser o mesmo estado atual). Em cada transição são definidas as saídas (inclusive o estado próximo).

Diagrama de estados

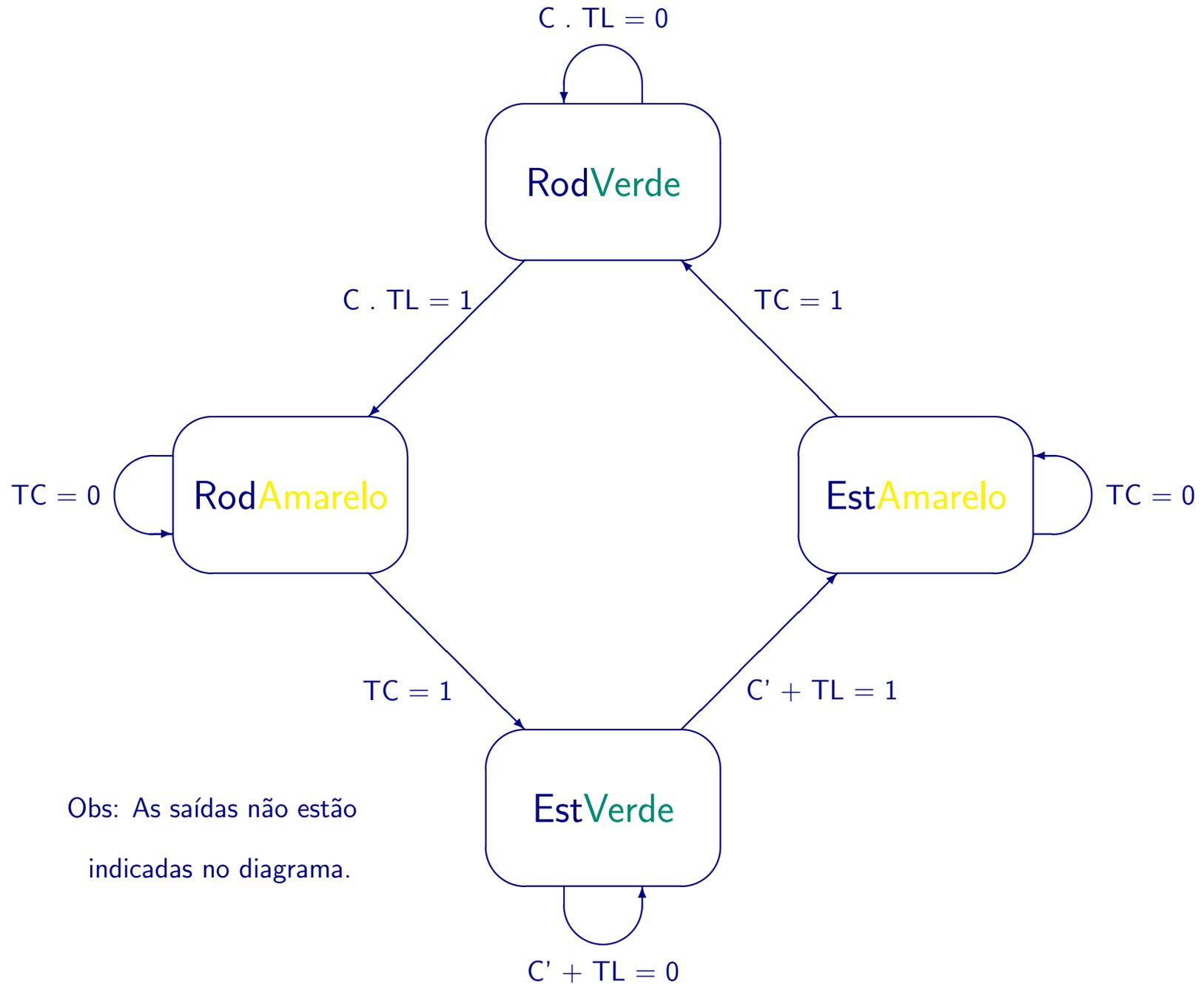


Tabela de Transição

Por razões didáticas, fazemos duas tabelas, uma mais alto nível (para entender melhor) e outra mais baixo nível (mais adequada para a implementação).

Tabela de transição (alto nível):

Entradas:		Saídas:			
presente estado	outras entradas	SemRod	SemEst	DISPARA	próximo estado
RodVerde	C.TL=0	Verde	Vermelho	Não	RodVerde
RodVerde	C.TL=1	Verde	Vermelho	Sim	RodAmarelo
RodAmarelo	TC=0	Amarelo	Vermelho	Não	RodAmarelo
RodAmarelo	TC=1	Amarelo	Vermelho	Sim	EstVerde
EstVerde	C'+TL=0	Vermelho	Verde	Não	EstVerde
EstVerde	C'+TL=1	Vermelho	Verde	Sim	EstAmarelo
EstAmarelo	TC=0	Vermelho	Amarelo	Não	EstAmarelo
EstAmarelo	TC=1	Vermelho	Amarelo	Sim	RodVerde

Codificações

Estados

RodVerde	00
RodAmarelo	01
EstVerde	11
EstAmarelo	10

Cores

Verde	00
Amarelo	01
Vermelho	10

Semáforos

R0 R1 indicam a cor de SemRod
E0 E1 indicam a cor de SemEst

Estados

X0 X1 indicam o estado presente
Y0 Y1 indicam o estado próximo

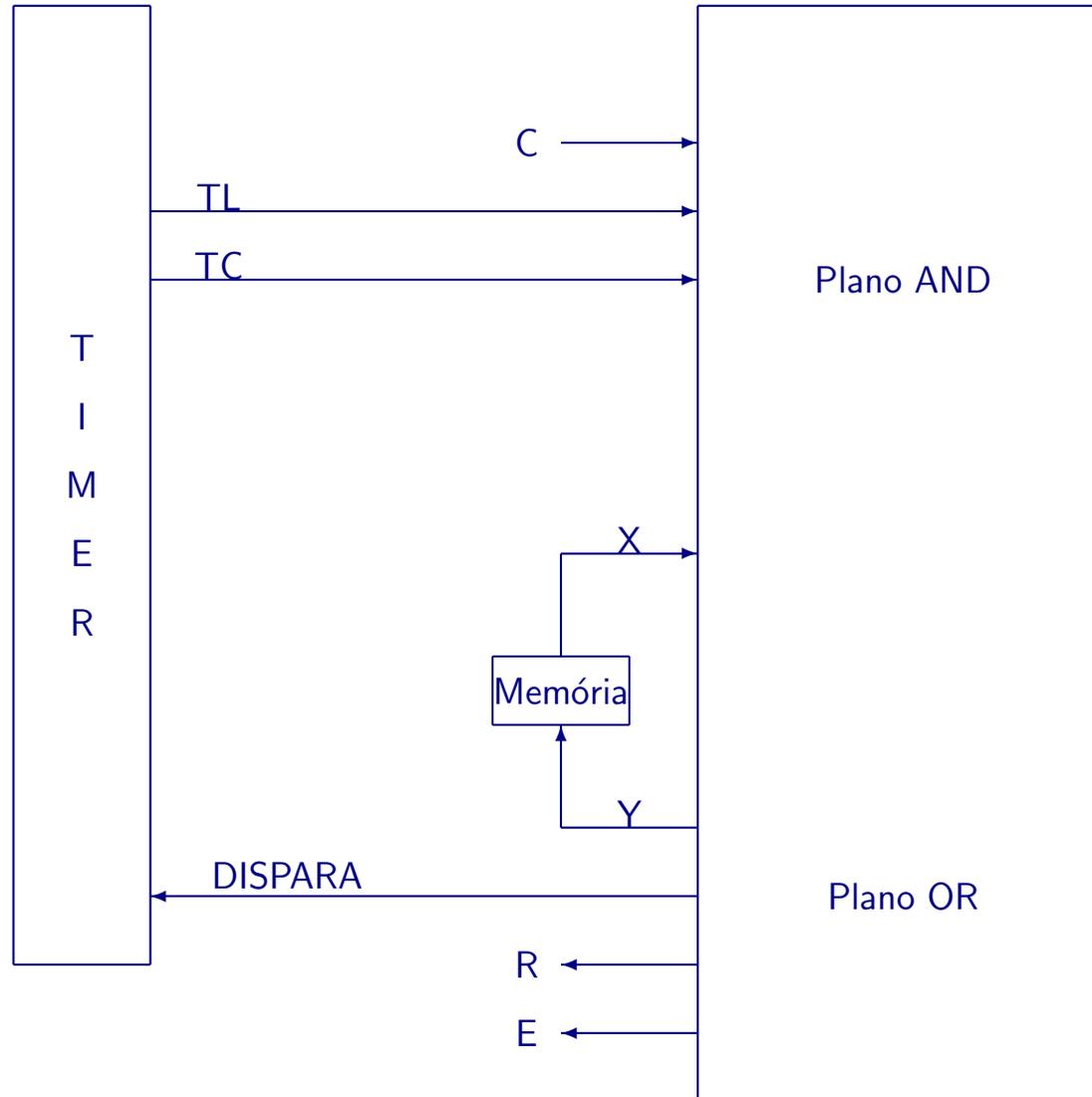
Tabela de Transição codificada

Essa tabela é a própria tabela de verdade.

Entradas:					Saídas:						
X0	X1	C	TL	TC	DISPARA	R0	R1	E0	E1	Y0	Y1
0	0	0	X	X	0	0	0	1	0	0	0
0	0	X	0	X	0	0	0	1	0	0	0
0	0	1	1	X	1	0	0	1	0	0	1
0	1	X	X	0	0	0	1	1	0	0	1
0	1	X	X	1	1	0	1	1	0	1	1
1	1	1	0	X	0	1	0	0	0	1	1
1	1	0	X	X	1	1	0	0	0	1	0
1	1	X	1	X	1	1	0	0	0	1	0
1	0	X	X	0	0	1	0	0	1	1	0
1	0	X	X	1	1	1	0	0	1	0	0

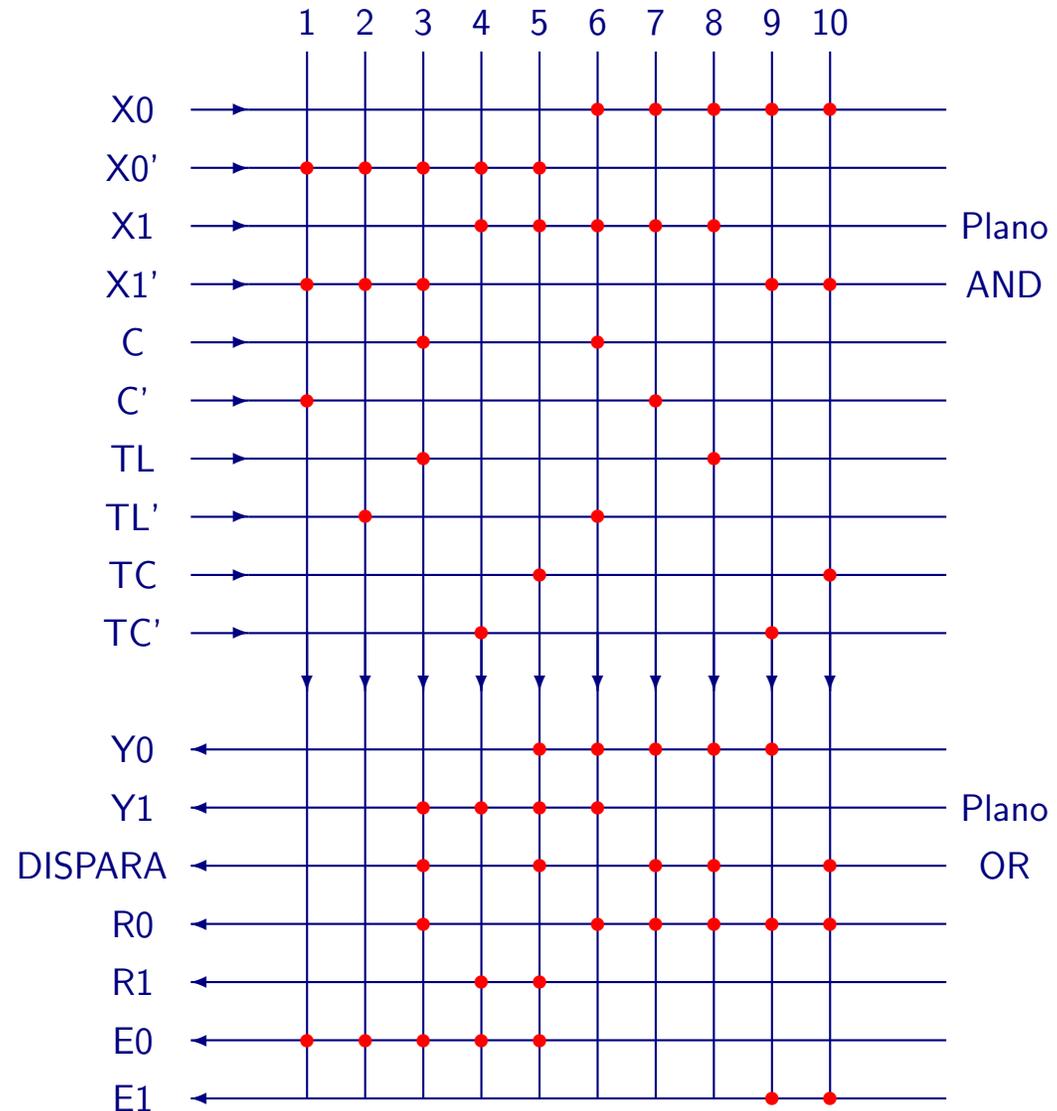
Lógica Seqüencial

Obs: Os sinais X, Y, R, E têm 2 bits cada.



Lógica Seqüencial implementada por PLA

A partir da tabela de transição, podemos chegar a seguinte implementação.



Implementação em PLA

Source: Mead and Conway - *Introduction to VLSI Systems*

