



PROJETO1 - AV3 – SDA **LABORATÓRIO DE SISTEMAS DIGITAIS AVANÇADOS**

Prof. Imbiriba

State Machine – Semáforo com counter interno **Máquina de Estado**

O objetivo destes exercícios é revisar, através da síntese de circuitos implementados em VHDL, os conceitos e os circuitos digitais relacionados às máquinas de estados.

INTRODUÇÃO

Construir circuitos digitais que executam operações complexas pode ser trabalhoso e demorado. Felizmente há uma forma de sistematizar o processo. Sistemas que requerem como entrada o estado anterior para decidir o estado futuro enquadram-se no que convencionou-se chamar de circuitos sequenciais. Estes são circuitos nos quais as saídas não são simplesmente determinadas pelas entradas mas sim por uma combinação das entradas com o estado anterior do circuito. Há diversas formas de projetar tais sistemas. Por exemplo, podemos aproveitar a experiência adquirida com *latches e flip-flops* e simplesmente retroalimentar no sistema o estado anterior como entrada(s). Infelizmente esta não é uma abordagem prática por diversos motivos. Inicialmente, os sistemas podem ser complexos, e tratar de sinais realimentados pode tornar o projeto proibitivamente complexo. Outro motivo deriva de dificuldades em controlar no sistema quando as entradas/saídas estarão efetivamente estáveis e que pode levar a grandes complexidades no projeto e dificultar o processo de correção de erros. Felizmente há formas de sistematicamente especificar tais sistemas. Nesta prática estudaremos uma metodologia de modelagem e especificação de sistemas sequenciais chamada de máquinas de estados finitos – MEF ou *finite state machine* – FSM.

Modelo de Moore e Mealy

O modelo de máquinas de Moore especifica que a saída dependa apenas do estado anterior armazenado na memória da MEF. Usualmente associa-se uma célula de memória para cada estado. Consequentemente a MEF sempre estará em algum estado. Por fim o modelo de Moore especifica que as saídas sejam síncronas. Uma consequência interessante do Modelo de Moore é o fato de que a cada estado sempre haverá associado um conjunto de saídas. Tal fato impõe um limite ao número de transições que podem partir de cada estado. Este número será limitado num máximo de $2^{|\Sigma|}$ onde Σ é o conjunto dos sinais de entrada. O mesmo não pode ser dito acerca do número máximo de estados. Uma conclusão compreensível porém errônea seria assumir que o número máximo de estados fosse 2^z , onde z representa o número de bits na saída. No entanto, este não é o caso, pois pode surgir a necessidade de que dois estados sejam definidos

representando exatamente o mesmo conjunto de saídas mas alcançáveis a partir de diferentes transições e originando diferentes transições.

No modelo de máquinas de Mealy o estado atual é definido por uma função lógica combinacional entre o estado anterior e um conjunto de entradas, conforme ilustra a Figura 1. Adicionalmente, a saída do sistema dependerá do estado atual assim como das entradas do sistema. Consequentemente as saídas podem mudar de modo assíncrono.

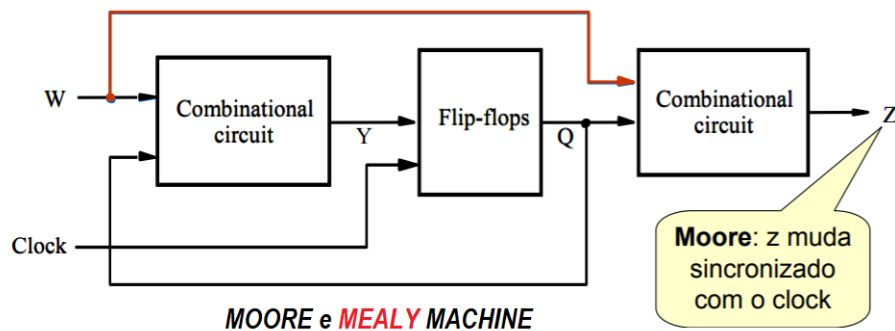
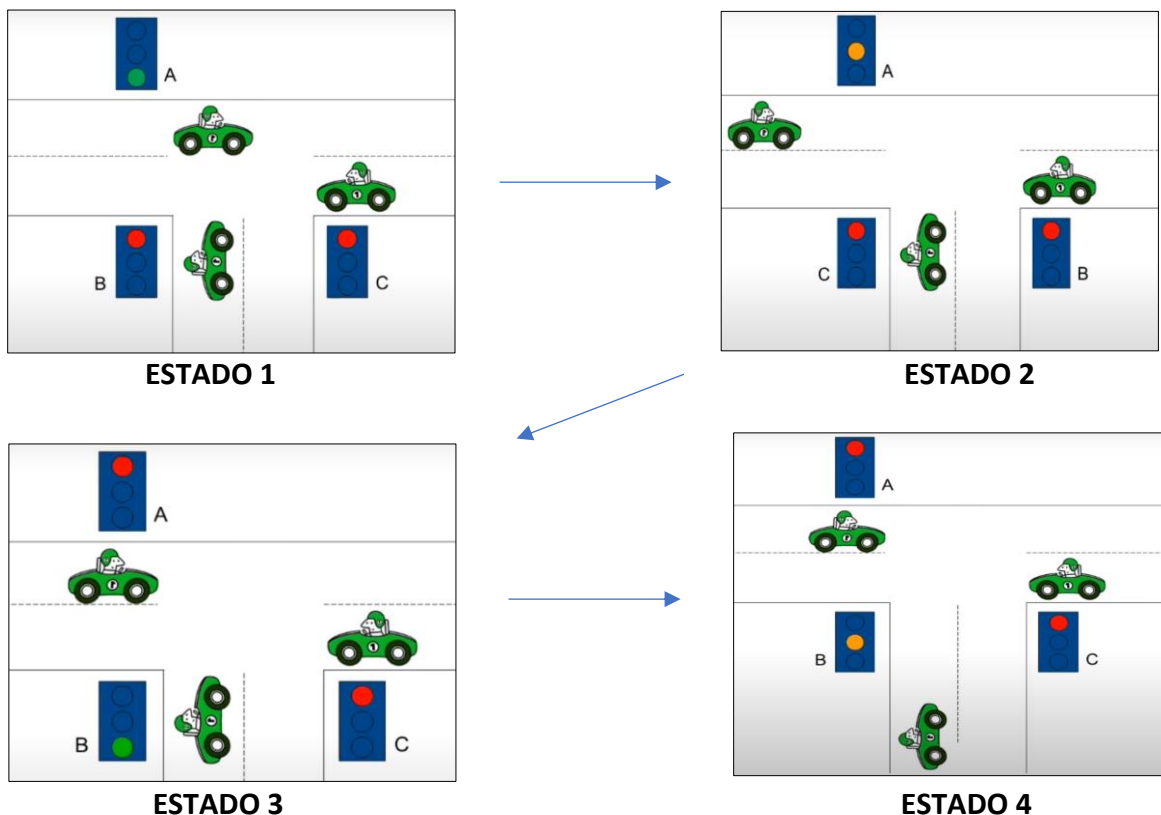
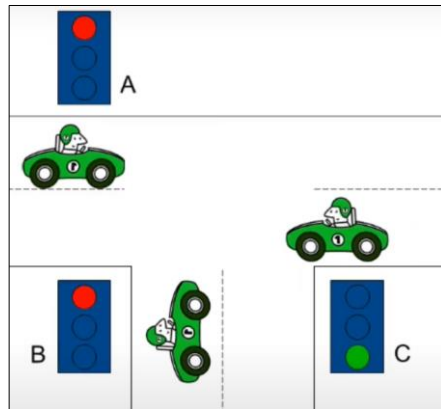


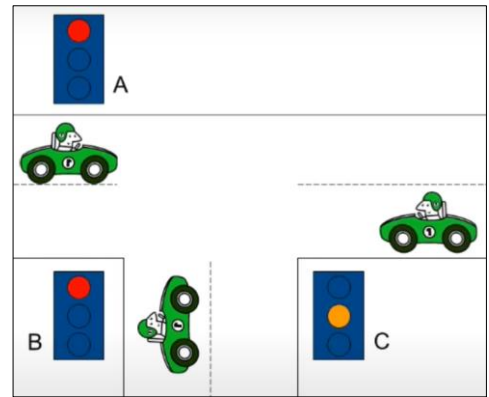
Figura 1 – Modelo de máquina de estado Moore e Mealy

EXERCÍCIO – IMPLEMENTAÇÃO DE UM SEMÁFORO USANDO DEFINIÇÃO DE MÁQUINA DE ESTADO.





ESTADO 5



ESTADO 6

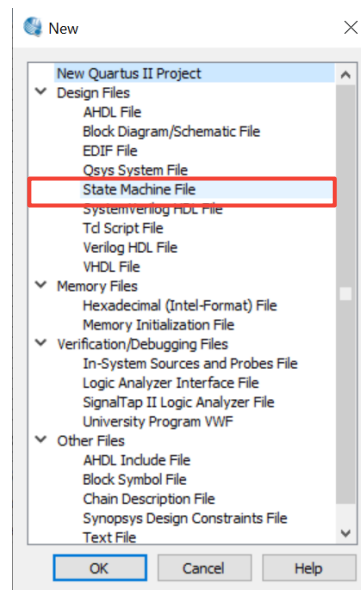
TABELA VERDADE

State	A			B			C		
	Red	Yellow	Green	Red	Yellow	Green	Red	Yellow	Green
1	0	0	1	1	0	0	1	0	0
2	0	1	0	1	0	0	1	0	0
3	1	0	0	0	0	1	1	0	0
4	1	0	0	0	1	0	1	0	0
5	1	0	0	1	0	0	0	0	1
6	1	0	0	1	0	0	0	1	0

TEMPO DE VERDE: 5 SEG - TEMPO DE AMARELO: 2 SEG

PARTE 1:

1 - Abra o Quartus e crie um novo STATE MACHINE FILE:



2 – Crie os 6 estados definidos para o sistema (semáforo);

3 – Adicione a transição de cada estado;



- 4 – Utilizando a ferramenta STATE MACHINE WIZARD configure os parâmetros da máquina de estados conforme descrito abaixo:
 - Reset síncrono ativo em alto;
 - Transição para o estado fonte se alguma transição não for satisfeita.
- 5 – Em entradas (inputs) adicione um contador de 5 bits (counter) para a contagem do tempo (transição entre os estados da máquina);
- 6- Adicione as saídas do sistema (GREEN, YELLOW, RED) como vetores de 3 bits (3 semáforos);
- 7- Configure as condições de mudança de um estado para outro, conforme figura 1 abaixo. Nesse caso, as mudanças se darão em função do tempo especificado para cada estado somente.
- 8- Defina as condições de saídas do sistema (luzes do semáforo) para cada um dos estados.
- 9- Compile e gere o código VHDL. Faça a simulação do circuito usando o University Programm VWF.
- 10- Adicione a condição de reset do temporizador (counter) quando findar o ciclo e retorno ao estado 1 conforme apresentado na Figura 2.

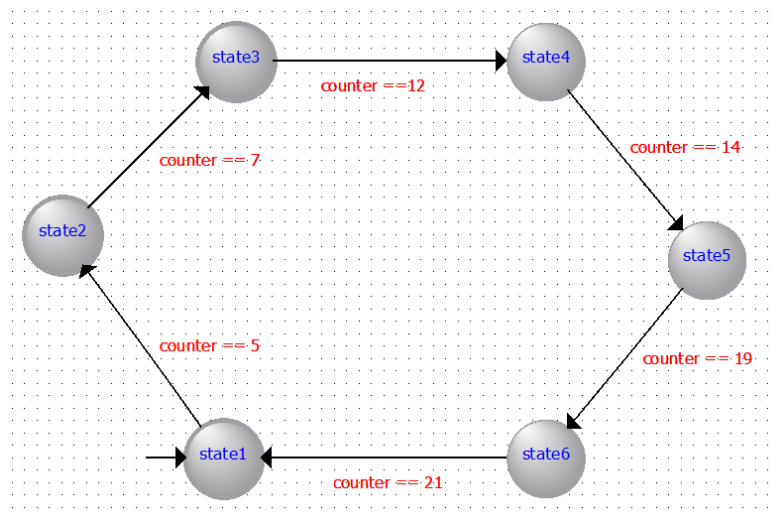


Figura 1 – Condições de transição dos estados

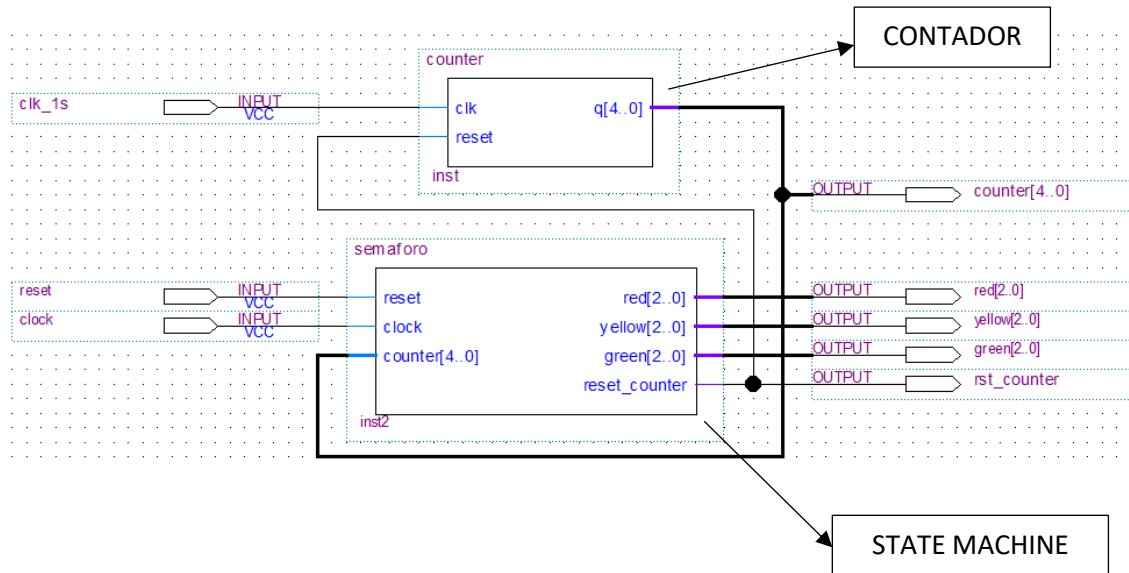
State Properties		
General	Incoming Transitions	Outgoing Transitions
Output Port	Output Value	Additional Conditions
red[2:0]	3	
yellow[2:0]	0	
green[2:0]	4	
reset_counter	1	counter == 21
< New >		

Figura 2 – Definição de saídas do estado 1



PARTE2:

1 - Implementar um controlador com contador para satisfazer a temporização do semáforo. Um exemplo de implementação pode ser visto abaixo:



2 – As formas resultado da simulação deve ser tal como apresentado abaixo:

