Laboratório de Sistemas Digitais

Aula Teórica 7

Ano Letivo 2014/15

Modelação em VHDL e Síntese de Máquinas de Estados Finitos

Aspetos Gerais

Modelo de *Moore*

Arnaldo Oliveira, Augusto Silva, Iouliia Skliarova, Manuel Bernardo Cunha

Conteúdo

- O processo de síntese de Máquinas de Estados Finitos (MEFs)
- *Workflow* da síntese
- Construção do diagrama de estados a partir da especificação
- Abordagens de modelação baseadas em VHDL
  - Modelo de *Moore*
- Simulação de MEFs

Síntese de Unidades de Controlo e Circuitos Sequenciais em Geral

- Passagem da especificação à implementação
  - Conceção de uma solução para um problema concreto (muitas vezes) inicialmente descrito em linguagem natural ou na forma de “use-cases”
  - Formalização/abstração segundo o modelo de Máquina de Estados Finitos (MEF)
  - Processo composto por diferentes passos de modelação, otimização e geração de hardware
  - Solução frequentemente
    - Não única
    - Sub-ótima

Instrumentos de Modelação e Síntese de MEFs

- Diagramas de Estado
- Cartas ASM: *Algorithmic State Machines**
- Tabelas de Estado/Saídas
- Tabelas de Transição/Saídas
- Tabelas de Excitação
- Diagramas temporais
- Linguagens de Descrição de Hardware
  - **VHDL**
- …

*Matéria não abordada nesta UC

Workflow da Síntese de MEFs

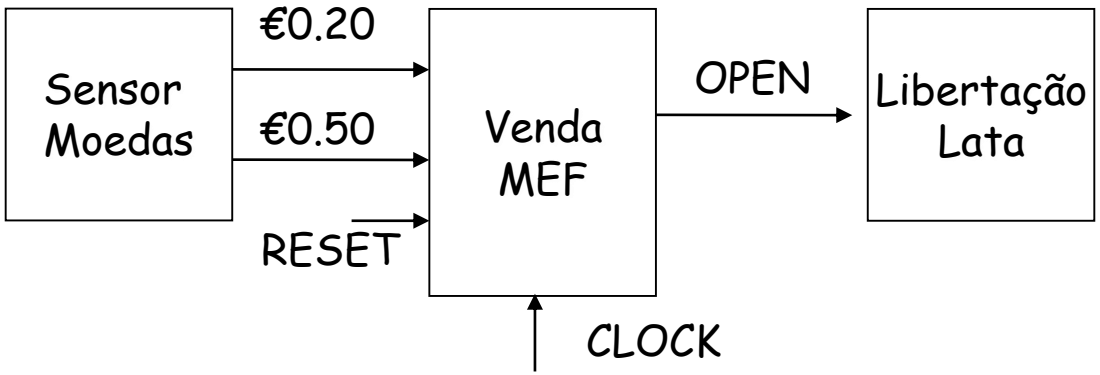
- Entender a especificação inicial
- Obter um representação abstrata da MEF
  - Diagrama de Estados/Saídas, Tabelas de Estado/Saídas, …
- Modelação em VHDL
- Simulação funcional da MEF
  - *Testbenches*
- Síntese lógica e implementação em hardware (em FPGA no contexto de LSDig)
- Verificação do comportamento da MEF em hardware

MEFs: Especificação → Formalização

- Problema principal (aspeto chave / mais importante)
  - Como passar da especificação ao diagrama de estados?
    - Qual do significado de cada estado?
- Especificação de ser
  - tão precisa quanto possível …
  - identificar as entradas e as saídas (E/S)
  - Identificar o comportamento E/S
  - incluir “use cases”
- Vamos analisar alguns exemplos...

Exemplo 1 - Especificação

- Máquina de venda de bebidas
  - Requisitos gerais:
    - entrega lata de cerveja (sem álcool ☺) após depósito de € 0.60
    - uma única entrada para moedas de € 0.20 e € 0.50
    - a máquina não dá troco
  - Passo 1: perceber o problema (fazer um desenho / diagrama de blocos!...)

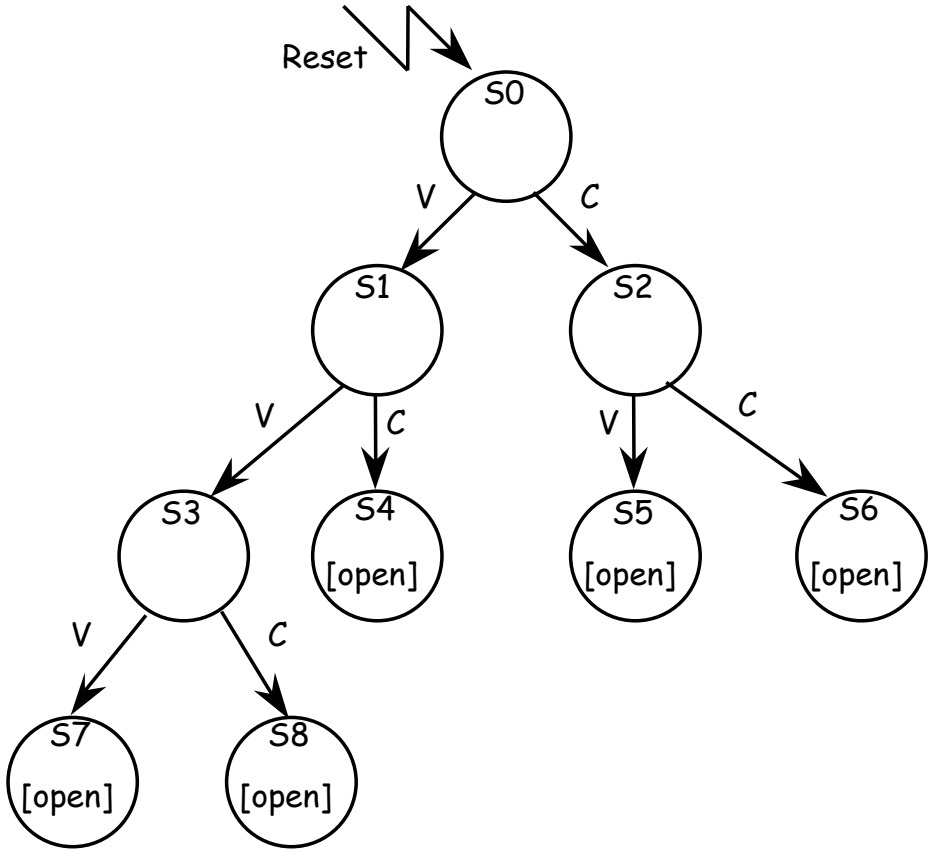


Exemplo 1 – Análise de Requisitos

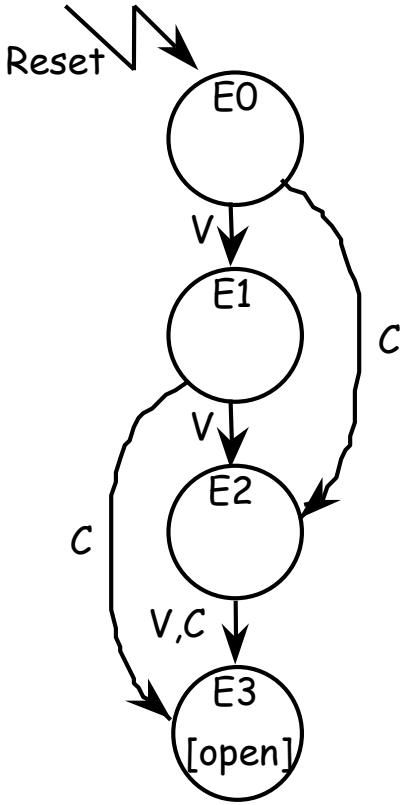
- Começar por identificar as sequências de entradas que levam diretamente à abertura
  - 3 moedas de €0.20
  - 1 moeda de €0.20 + 1 moeda de €0.50
  - 1 moeda de €0.50 + 1 moeda de €0.20
  - 2 moedas de €0.50
  - 2 moedas de €0.20 + 1 moeda de €0.50
- Identificar entradas de saídas:
  - Entradas:
    - V (sensor ativo para €0.20)
    - C (sensor ativo para €0.50)
  - Saída
    - OPEN

Exemplo 1 – Diagrama de Estados

- Diagrama de estados primário (inicial)

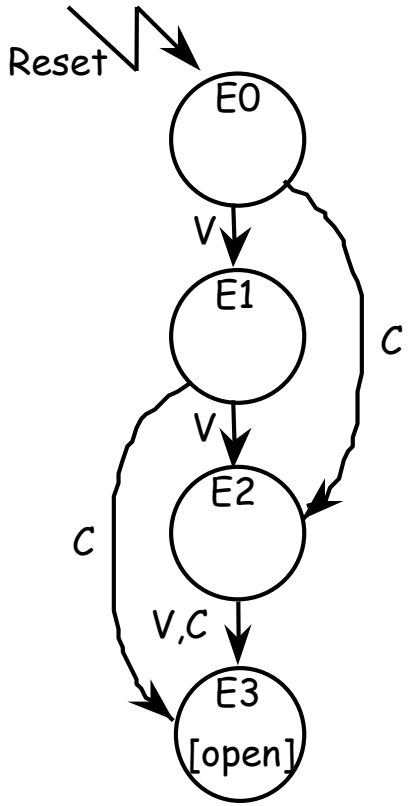


- Diagrama de estados correto com reutilização de estados



Exemplo 1 – Tabela de Estados

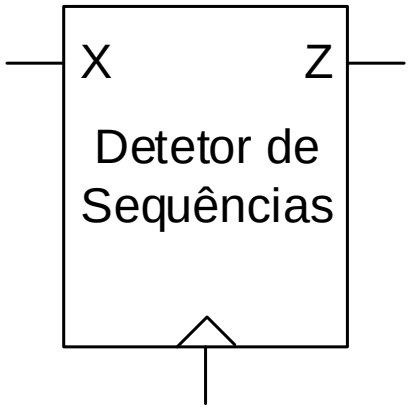
- Tabela de Estados/Saídas decorre diretamente do DE



PState	Inputs		NState	Output
	V	C		open
E0	0	0	E0	0
	0	1	E2	0
	1	0	E1	0
	1	1	X	X
E1	0	0	E1	0
	0	1	E3	0
	1	0	E2	0
	1	1	X	X
E2	0	0	E2	0
	0	1	E3	0
	1	0	E3	0
	1	1	X	X
E3	X	X	E3	1

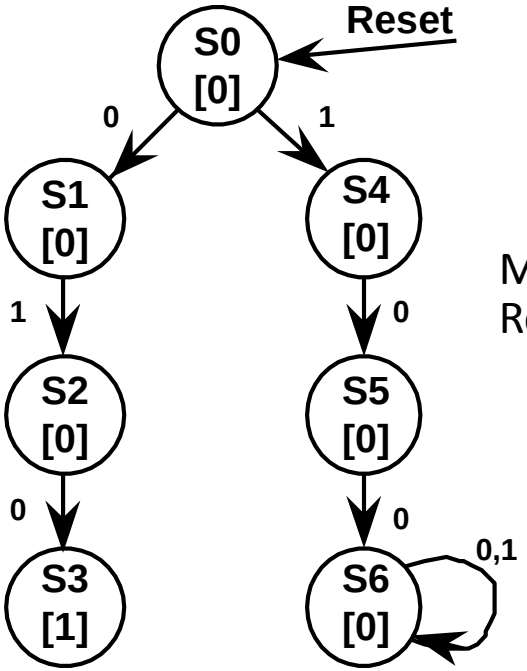
Exemplo 2: Detector de Sequências

- Reconhecimento de padrões em frases de comprimento finito. Exemplo:
  - Um reconhecedor de frases finitas tem uma entrada (X) e uma saída (Z). A saída é activa sempre que a sequência de entrada …010…é observada, desde que a sequência 100 nunca tenha surgido.
  - Exemplo do comportamento entrada/saída:
    - X: 011**010**00~~010~~…
    - Z: 00000**1**00000…
    - X: 0**0101010**~~010~~…
    - Z: 000**1**0**1**0**1**000…



Exemplo 2: Diagrama de Estados

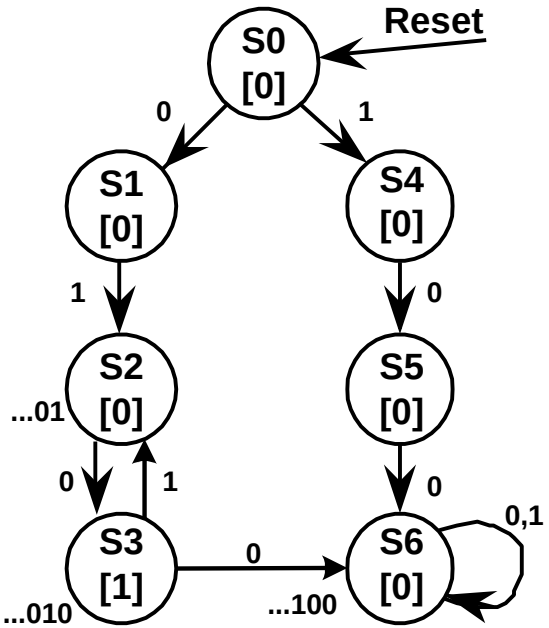
– Desenhar o diagrama de estados para os padrões que devem ser reconhecidos i.e., 010 e 100.



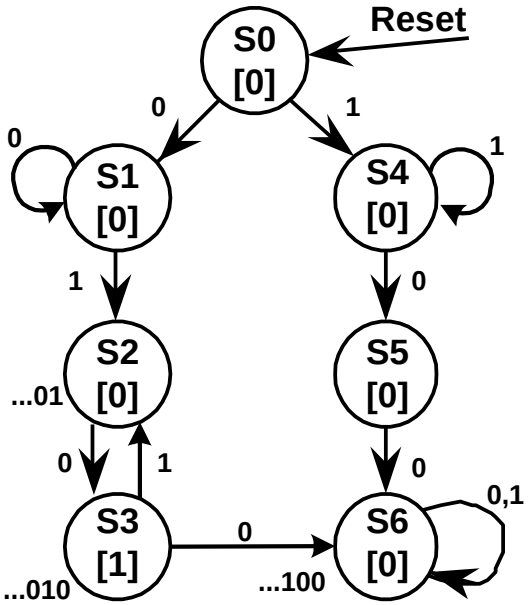
Modelo de Moore
Reset “leva” a máquina para o estado S0

Exemplo 2: Diagrama de Estados

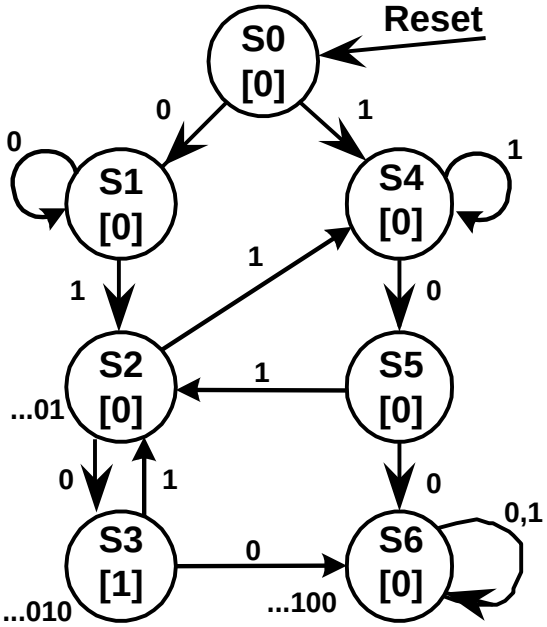
– Completar o diagrama analisando as condições de transição de cada estado



Transições em S3



Transições em S1 e S4



Transições em S2 e S5

Exemplo 2: Revisão do Procedimento

- Escrever sequências de teste com as entradas/saídas para perceber a especificação
- Criar uma sequência de estados e transições para as sequências que se pretende ver reconhecidas
- Acrescentar transições em falta; reusar estados existentes o mais possível
- Verificar o comportamento E/S do diagrama de estados para assegurar que funciona como pretendido

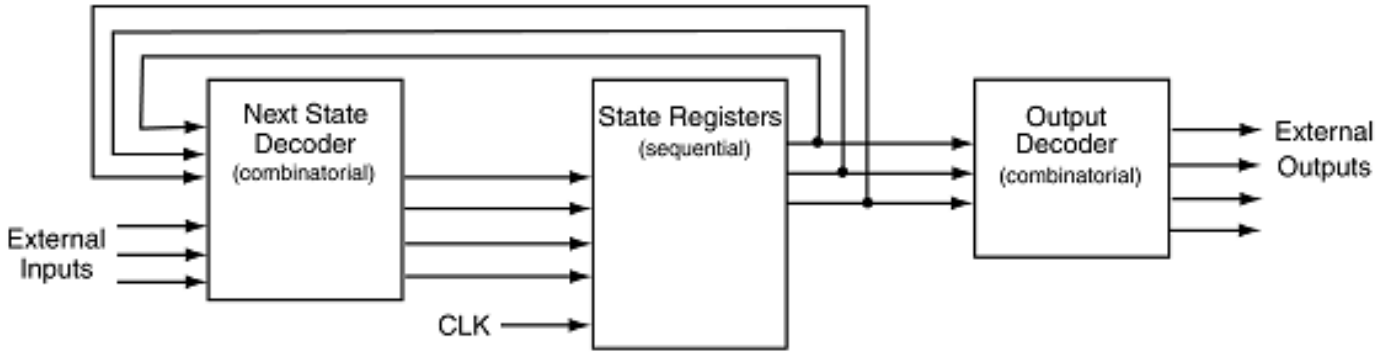
Agora que vimos alguns exemplos de passagem da especificação para o diagrama de estados, vamos ver como descrever eficientemente a MEF em VHDL…

Modelos Comportamentais Textuais de MEFs

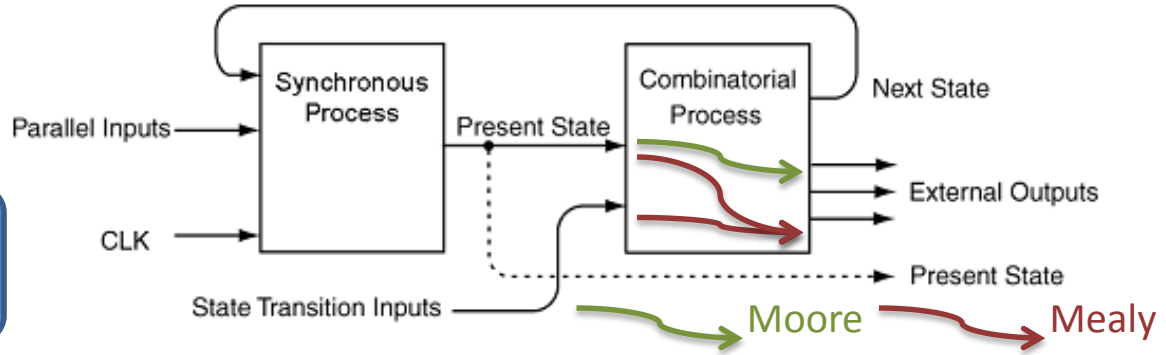
- A linguagem VHDL proporciona descrições de alto nível (comportamentais) de MEFs
  - Muito próximas dos diagrama de estados
  - Não são necessárias equações para explicitar saídas e o "próximo estado"
  - Por omissão é o sintetizador que determina a codificação dos estados com base em estados simbólicos
  - Vários estilos de escrita possíveis em VHDL (mais frequentemente com 2 ou 3 processos)
  - Tradução direta do digrama de estados para VHDL no IDE de desenvolvimento

MEFs em VHDL - O Estilo de Codificação Baseado em "2" Processos

- Modelo de *Moore* revisitado



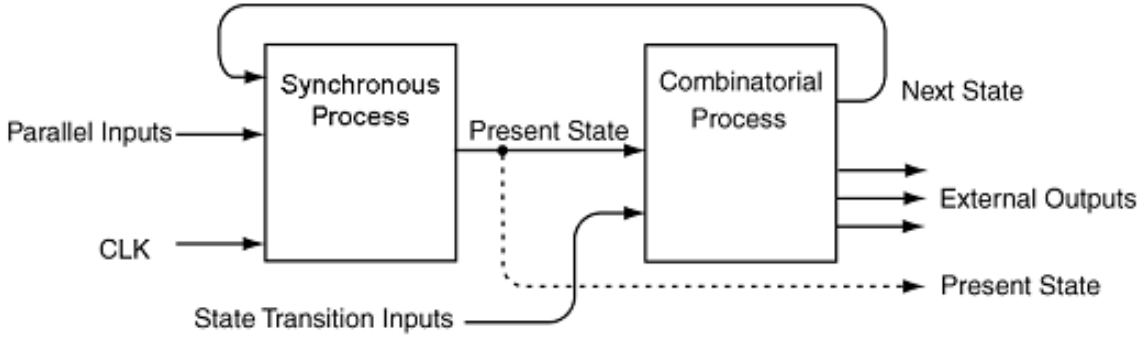
- Podemos reorganizar sinais e portos segundo dois processos (compatível com *Moore* e *Mealy*)



Entradas que condicionam em paralelo os flip-flops: RESET, SET, CLEAR, etc…

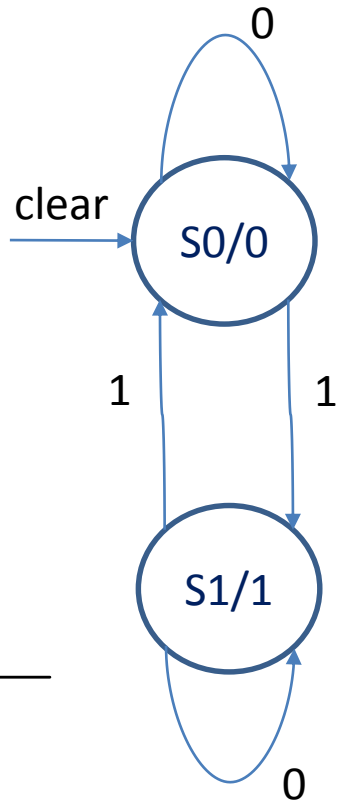
MEFs em VHDL - O Estilo de Codificação Baseado em "2" Processos

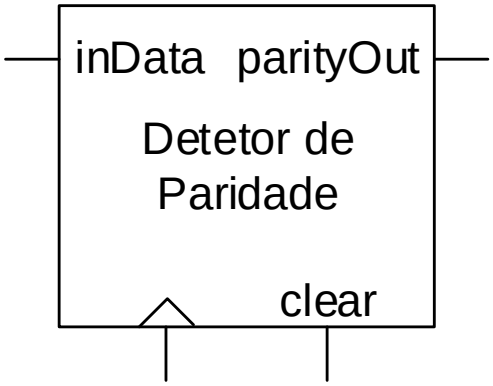
- Processo "*synchronous*" (apenas uma designação)
  - Atribuições dependentes dum evento de relógio e/ou de atribuição/inicialização assíncrona dos elementos de memória
- Processo "*combinatorial*" (apenas uma designação)
  - Atribuições relacionadas com a determinação de
    - Saídas
    - Estado seguinte
- Os 2 processos são interdependentes



Exemplo (segundo *Moore*) – Detetor de Paridade

- Detetor de paridade
  - Deteta se o número de ‘1’s ocorridos numa *stream* (série) de bits é par ou ímpar
  - Entradas
    - **`clk`**
    - **`clear / reset (assíncrono)`**
    - **`inData`**
  - Saída
    - **`parityOut`**

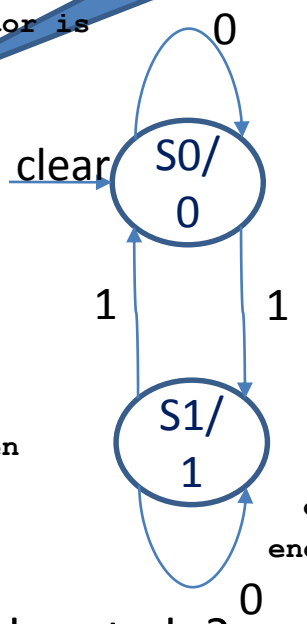




Exemplo (segundo *Moore*) – Detetor de Paridade

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity ParityDetector is
   port(clear     : in std_logic;
        clk       : in std_logic;
        inData    : in std_logic;
        parityOut : out std_logic);
end ParityDetector;

architecture Behav of ParityDetector is

   type TState is (S0,S1);
   signal pState, nState: TState;

begin
   sync_proc: process(clk, clear)
   begin
      if (clear = '1') then
         pState <= S0;
      elsif (rising_edge(clk)) then
         pState <= nState;
      end if;
   end process;
```

Criação dum novo tipo de dados "enumerado"

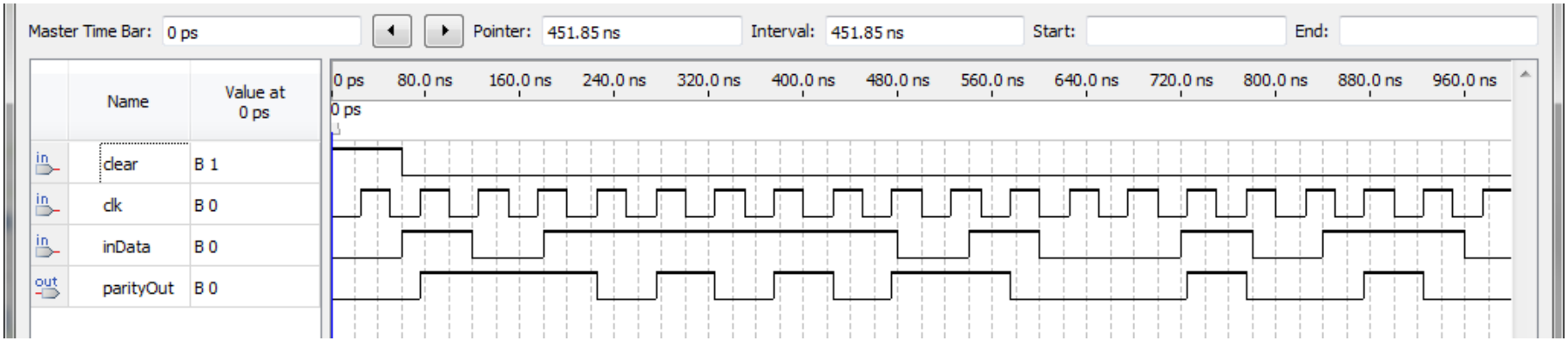



```
comb_proc: process(pState, inData)
begin
   case pState is
      when S0 =>
         parityOut <= '0'; -- Moore output
         if (inData = '1') then
            nState <= S1;
         else
            nState <= S0;
         end if;
      when S1 =>
         parityOut <= '1'; -- Moore output
         if (inData = '1') then
            nState <= S0;
         else
            nState <= S1;
         end if;
      when others => -- Catch all condition
         nState    <= S0;
         parityOut <= '0';
   end case;
end process;
end Behav;
```

Tradução direta do Diagrama de Estados

- Onde estão as variáveis de estado?

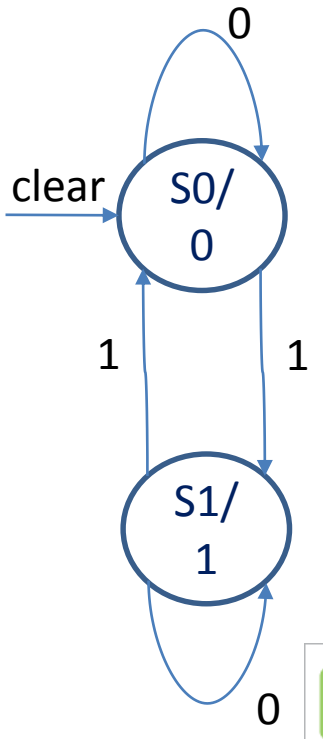
Simulação – Detetor de Paridade



Simulação com *testbench*:

- Gerada a partir do ficheiro VWF
- Gerada diretamente em VHDL de acordo com o *template* para componentes sequenciais
  - 1 processo para geração do sinal de relógio
  - 1 processo para inicialização e geração da entrada

- **<u>Muito importante:</u>**
  - Não comutar as entradas na vizinhança das transições ativas do sinal de relógio (refletir na simulação o cumprimento dos tempos de *setup* e de *hold*)



Comentários Finais

- No final desta aula e do trabalho prático 7 de LSDig, deverá ser capaz de:
  - Construir diagramas de estados com base na especificação de uma MEF
  - Conhecer os passos necessários à síntese de máquinas de estados finitos
  - Recorrer a descrições comportamentais VHDL próximas do diagrama de estados saídas
    - Modelo de *Moore*
  - Conceber *testbenches* para a simulação funcional das MEF
- … bom trabalho prático 7, disponível no site da UC
  - elearning.ua.pt