Sistemas Reconfiguráveis
Eng. de Computação

Profs. Francisco Garcia e Antônio Hamilton Magalhães

Aula 12 – Introdução às máquina de estados finitos (FSM)

Resposta ao exercício da aula 11

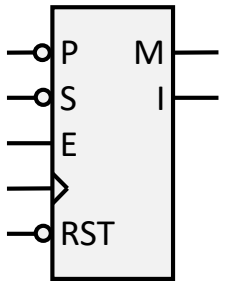
rev. C - 31/10/23

Máquina de estados finitos
Exercício fsm_2

- **Exercício fsm_2:**
- **Desenhar o diagrama de transição de estados** de um máquina de estados finitos que controla um sistema de reprodução de áudio descrita a seguir
- **Escrever o código em VHDL** para essa máquina de estados, usando dois PROCESS, um sequencial para a memória de estado e outro combinacional para as lógicas do próximo estado e de saída.
- **Testar o projeto** no módulo DE2

Máquina de estados finitos
Exercício fsm_2

- **Exercício fsm_2 – especificações:**
- Esse sistema tem dois botões: PLAY e STOP. Esses botões, quando pressionados, geram um nível lógico '**0**' e, quando liberados, voltam para a posição original por efeito de um mola, gerando o nível lógico '**1**'. Existe também um sinal END, ativo em nível lógico alto, que indica que o áudio chegou ao final.
- O sistema é controlado através de duas saídas, uma para ativar a reprodução e outra para voltar a gravação para o início. Esse último não pode ser ativado durante a reprodução nem durante a pausa.



Entradas:

nrst	reset assíncrono ativo baixo
clk	clock borda de subida
p	PLAY ativo baixo
s	STOP ativo baixo
e	END ativo alto

Saídas:

m	ativa reprodução ativo alto
i	volta para início ativo alto

Máquina de estados finitos
Exercício fsm_2

- **Exercício fsm_2 – especificações (continuação):**

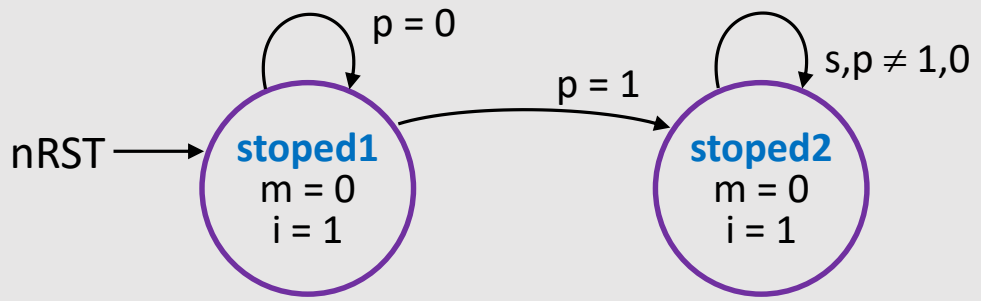
- O estado inicial do sistema, após a energização, é parado. Estando o botão de STOP liberado, a reprodução é iniciada quando o botão PLAY é pressionado. Manter o botão PLAY pressionado depois do início da reprodução não faz efeito algum. Se o botão PLAY for liberado, a reprodução continua normalmente.
- Estando reproduzindo, pressionar o botão PLAY novamente faz a reprodução pausar. Manter o botão PLAY pressionado não faz efeito algum. Se o botão PLAY for liberado, o sistema continua pausado.
- Estando em pausa, pressionar o botão PLAY novamente faz a reprodução retornar a partir do ponto onde foi pausada. Manter o botão PLAY pressionado não faz efeito algum. Se o botão PLAY for liberado, a reprodução continua normalmente.
- Se, durante a reprodução o sinal END for ativado ou, em qualquer situação, o botão STOP for pressionado, o sistema volta para o estado inicial. Se isso acontecer com o botão PLAY pressionado, ele deve ser liberado antes de um novo comando de PLAY.

Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2:

Após a inicialização, a FSM vai para o estado **stoped1**. Se o botão PLAY estiver pressionado, permanece nesse estado. Caso contrário, vai para **stoped2.**

Nos dois estados, o sistema de reprodução está parado (m='0') e comandando para voltar para o início da reprodução (i='1').

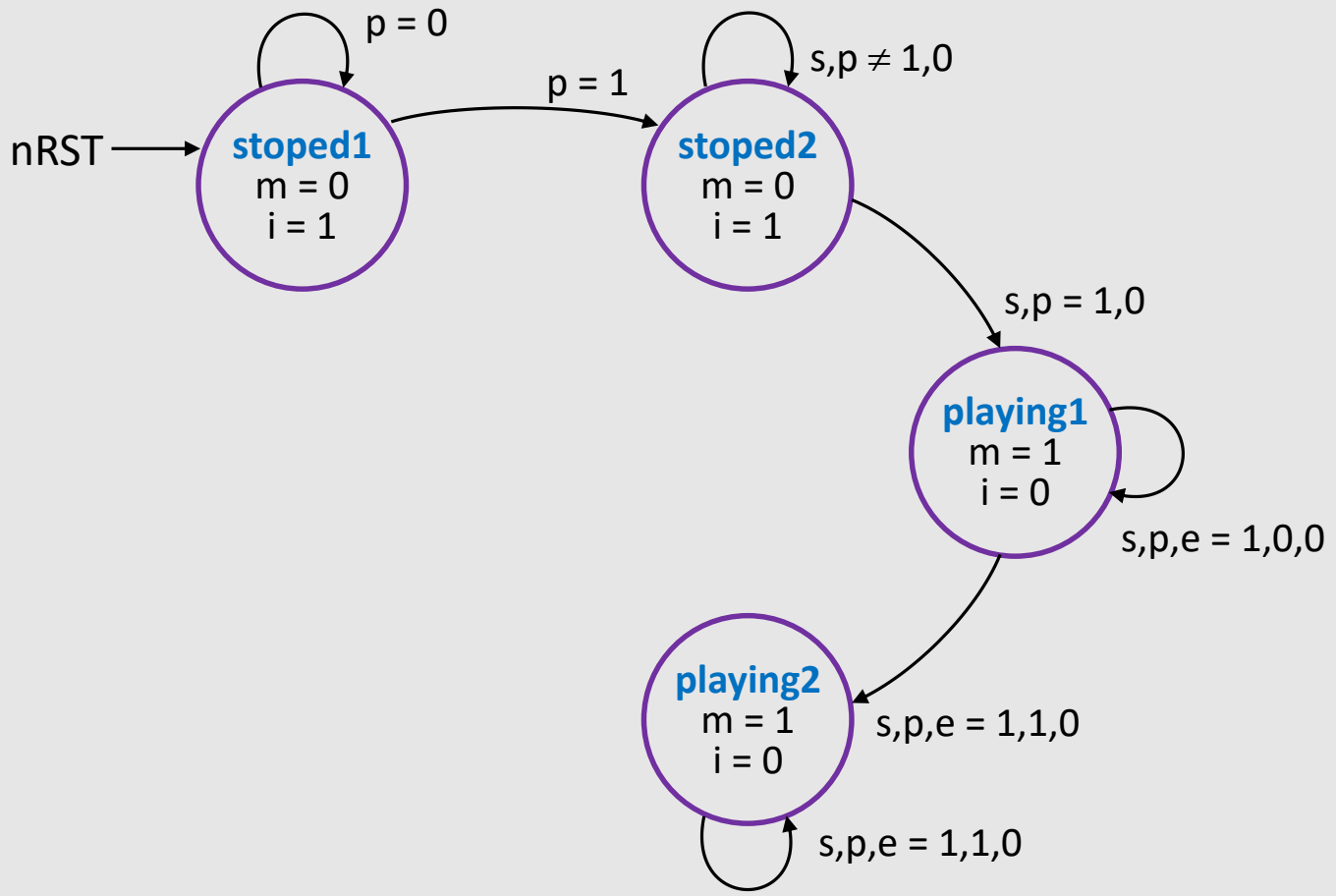


Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2:

Estando o botão STOP liberado, quando PLAY for pressionado, a FSM vai para **playing1**, onde fica esperando o botão ser liberado, quando passa para **playing2.**

Nos dois estados, o sistema de reprodução está reproduzindo, (m='1‘ e i='0').



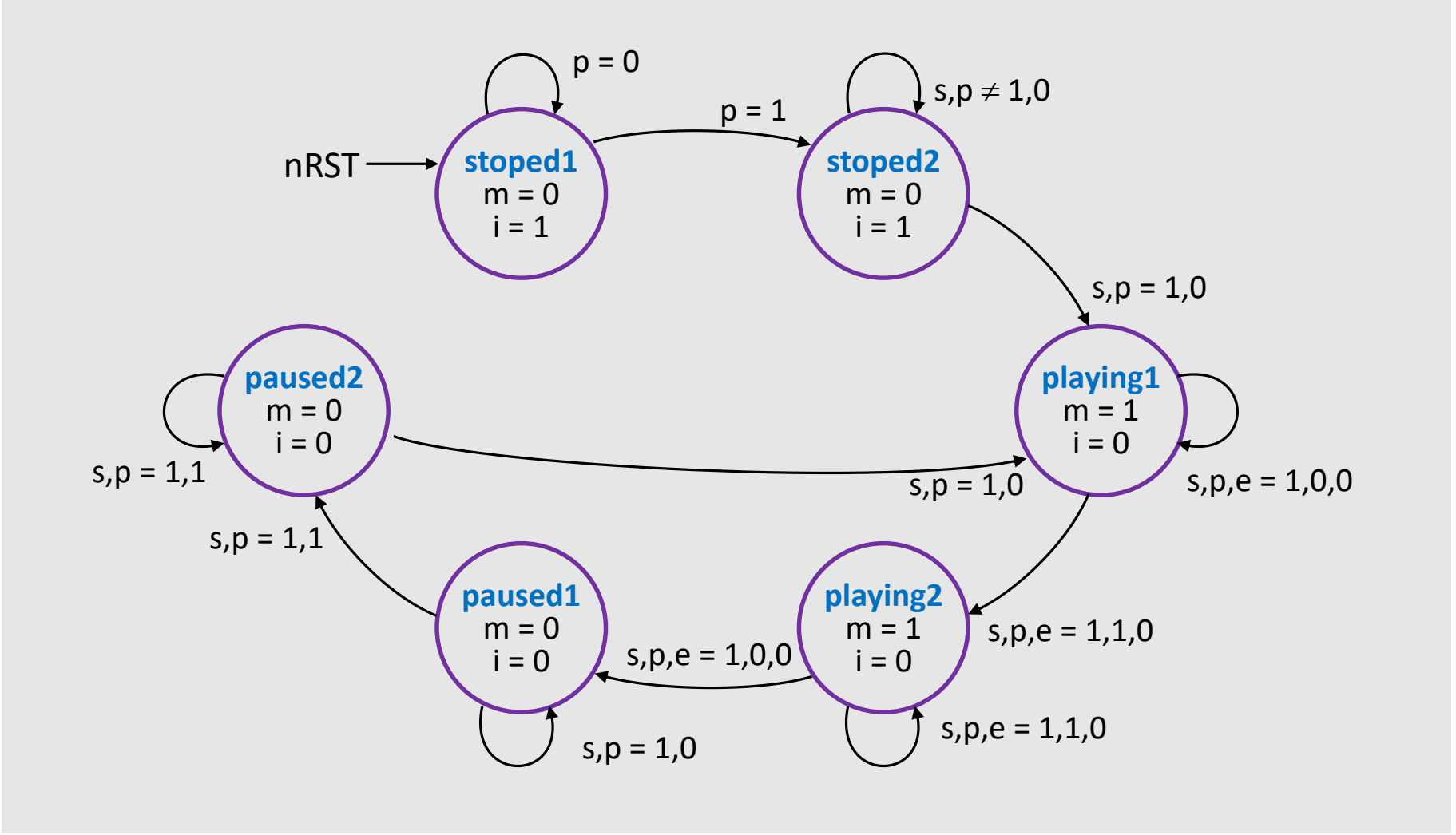
Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2:

Pressionando PLAY novamente, a FSM vai para **paused1**, onde fica esperando o botão ser liberado, quando passa para **paused2.**

Nos dois estados, o sistema de reprodução está pausado, (m='0' e i='0').

Pressionando PLAY mais uma vez, a FSM volta para **playing1**.



Máquina de estados finitos
Exercício fsm_2

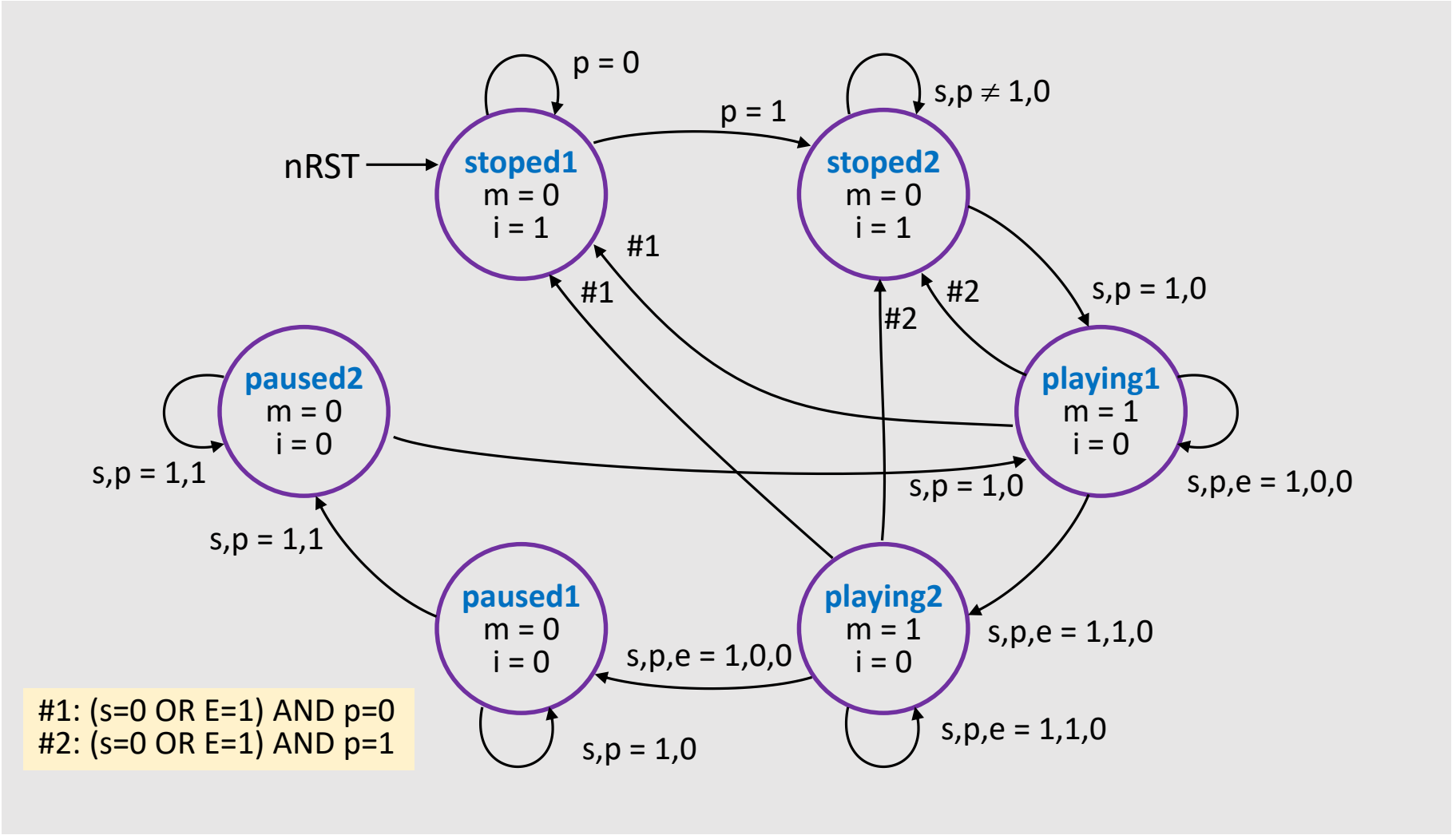
Exercício fsm_2:

Se, durante a reprodução, o sinal END for ativado ou se o botão STOP for pressionado, a FSM vai para:

Com o botão PLAY pressionado: **stoped1**;

Com o botão PLAY liberado: **stoped2**.

Foi considerada aqui a possibilidade da mudança simultânea mais de uma entrada.



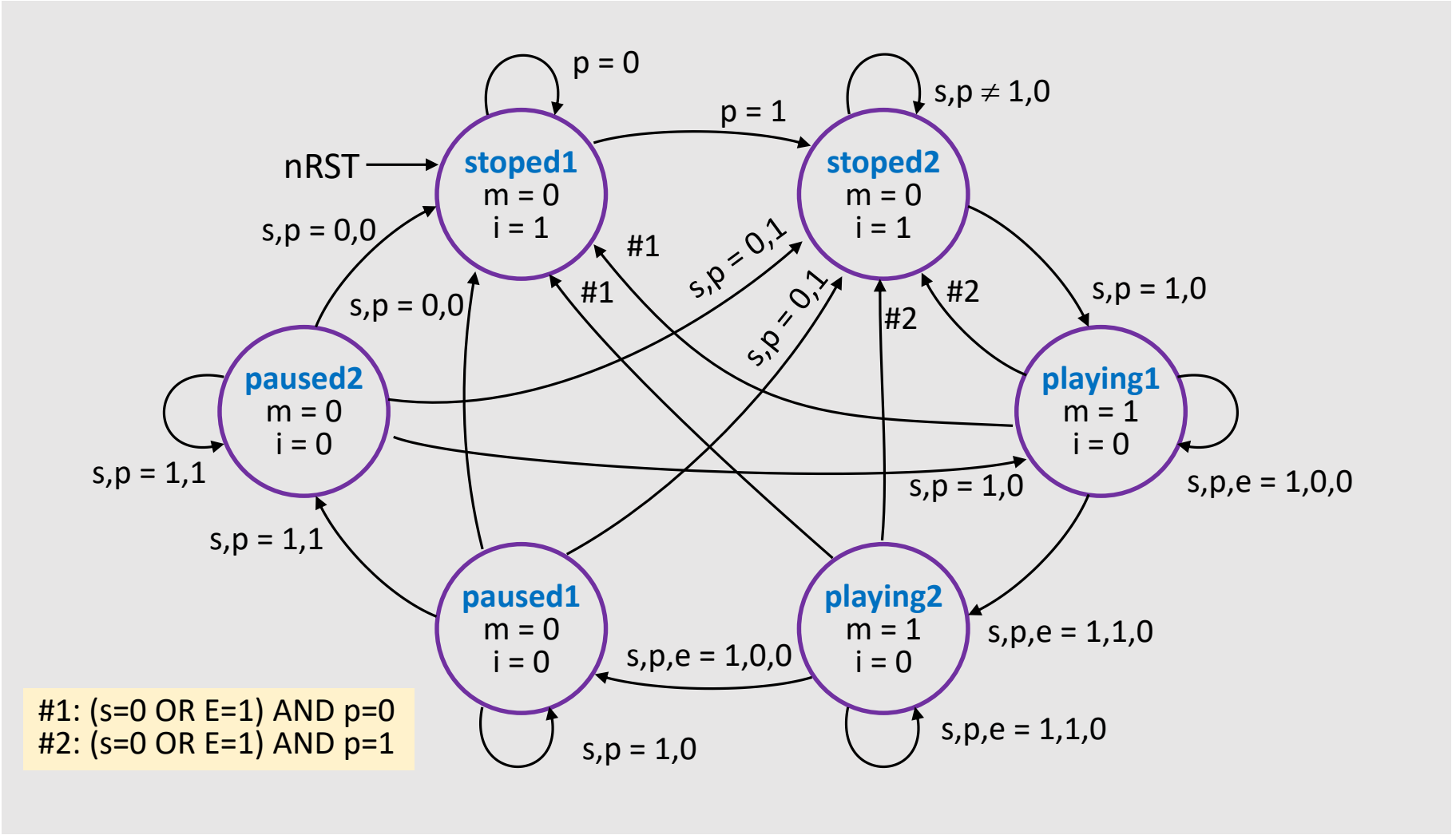
Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2:

Se, durante a pausa, o botão STOP for pressionado, a FSM vai para:

Com o botão PLAY pressionado: **stoped1**;

Com o botão PLAY liberado: **stoped2**.

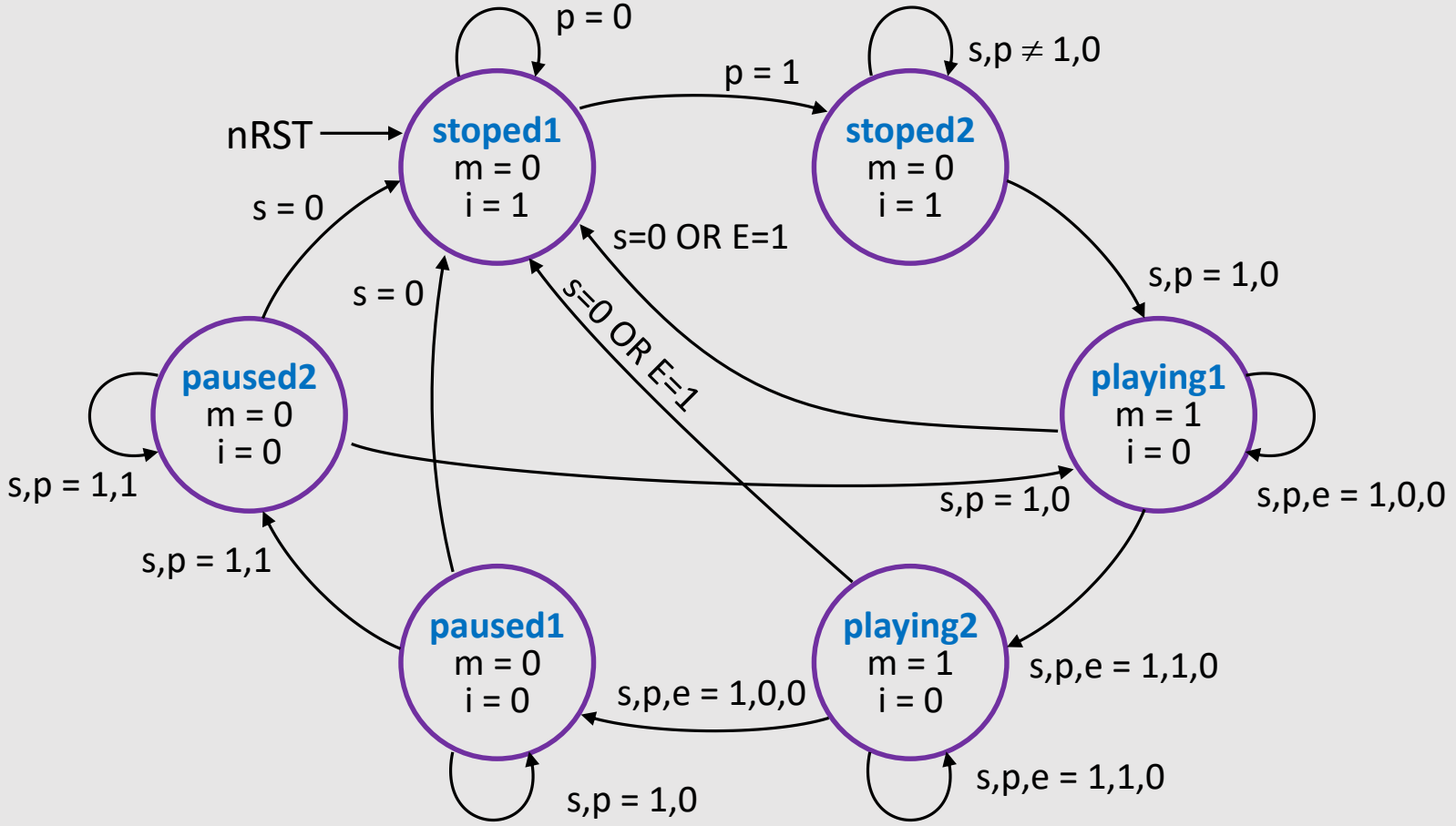


Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2:

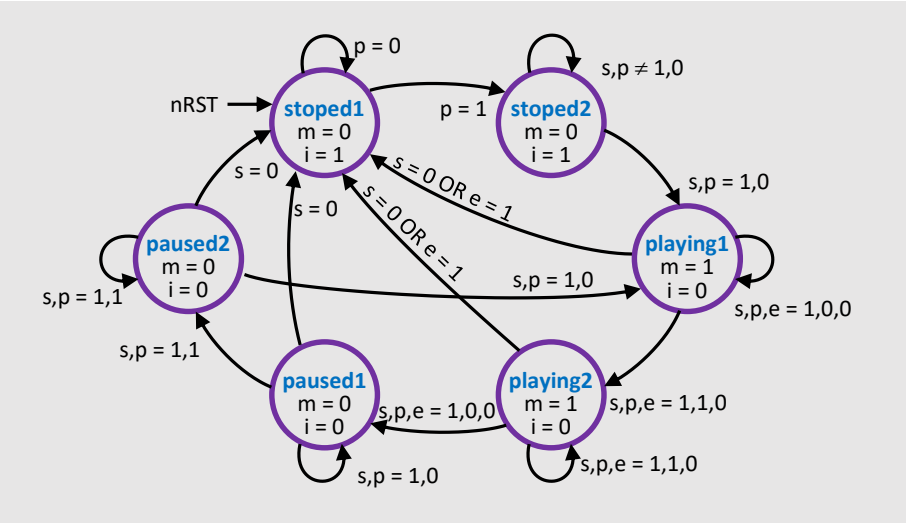
Uma alternativa mais simples é, nos dois casos anteriores, sempre ir para **stoped1**, independentemente do botão PLAY.

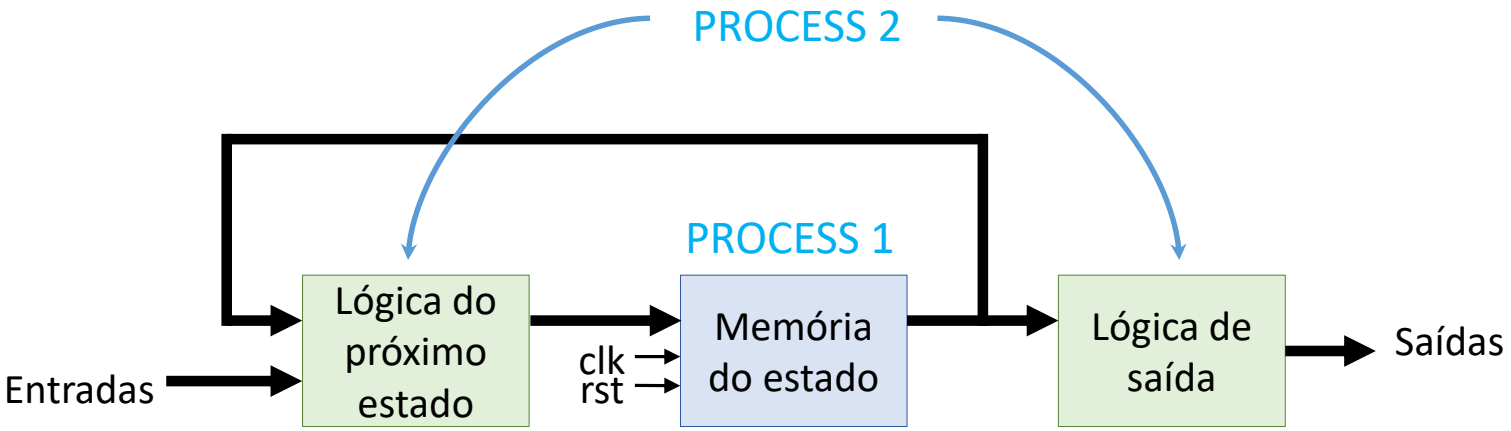
A diferença é que, dessa forma, ao se pressionar STOP ou ocorrer um END com o botão PLAY liberado, serão necessárias duas transições para a FSM chegar em **stoped2.**



Máquina de estados finitos
Exercício fsm_2

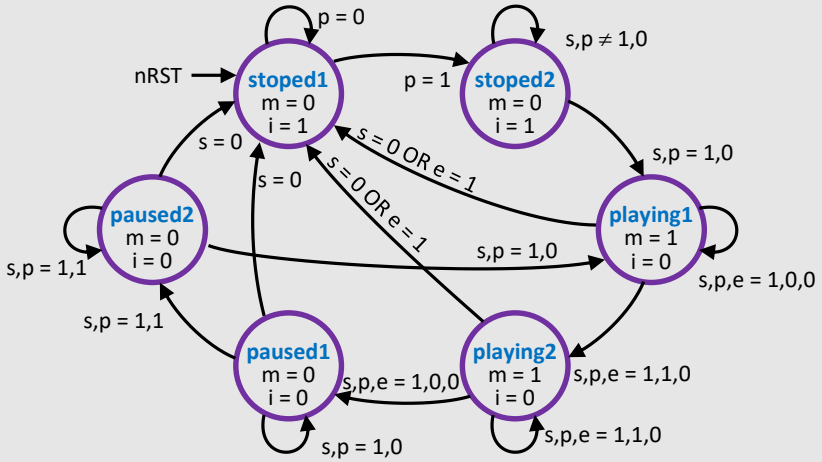
Exercício fsm_2: Descrever em VHDL a FSM cujo diagrama de transição de estados é mostrado abaixo, usando dois PROCESS.





Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2: Descrever em VHDL a FSM cujo diagrama de transição de estados é mostrado abaixo:

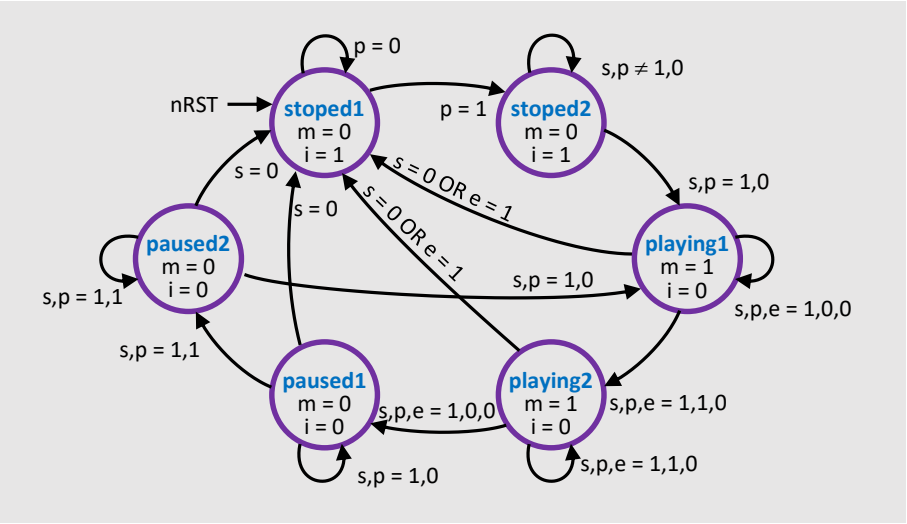



```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
-----------------------------------------------
ENTITY fsm_2 IS
   PORT (
      clk, nrst, p, s, e : IN STD_LOGIC;
      m, i : OUT STD_LOGIC
   );
END ENTITY;
-----------------------------------------------
```

Continua na próxima página

Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2: Descrever em VHDL a FSM cujo diagrama de transição de estados é mostrado abaixo:

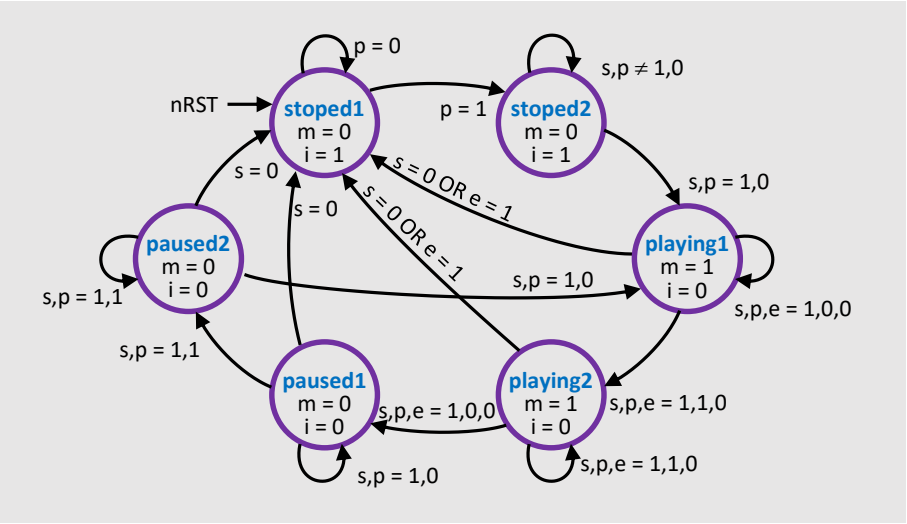



```
ARCHITECTURE arch1 OF fsm_2 IS
   TYPE state_type IS (stopped1, stopped2, playing1,
            playing2, paused1, paused2);
   SIGNAL next_state, pres_state : state_type;
BEGIN
   -- Parte sequencial da máquina de estados:
   PROCESS(nrst, clk)
   BEGIN
      IF nrst = '0' THEN
         pres_state <= stopped1;
      ELSIF RISING_EDGE(clk) THEN
         pres_state <= next_state;
      END IF;
   END PROCESS;
```

Continua na próxima página

Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2: Descrever em VHDL a FSM cujo diagrama de transição de estados é mostrado abaixo:

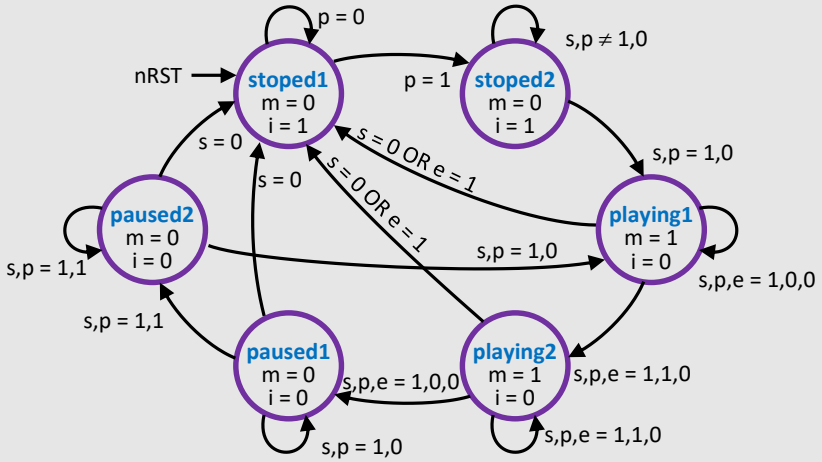



```
-- Parte combinacional da máquina de estados:
PROCESS(pres_state, p, s, e)
BEGIN
   CASE pres_state IS
      ----------------------------------------
      -- STOPPED 1
      -- Aguarda liberação do botão PLAY
      ----------------------------------------
      WHEN stopped1 =>
         ---------------------- Saídas:
         m <= '0';
         i <= '1';
         -------------- Próximo estado:
         IF p = '1' THEN
            next_state <= stopped2;
         ELSE
            next_state <= stopped1;
         END IF;
```

Continua na próxima página

Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2: Descrever em VHDL a FSM cujo diagrama de transição de estados é mostrado abaixo:

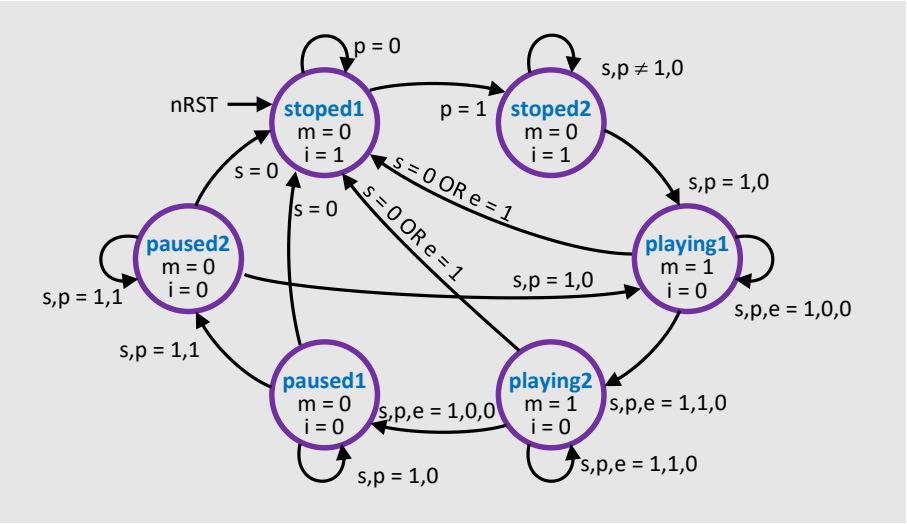



```
-----------------------------------------
-- STOPPED 2
-- Aguarda ativação do botão PLAY
-----------------------------------------
WHEN stopped2 =>
   ---------------------- Saídas:
   m <= '0';
   i <= '1';
   ---------------------- Próximo estado:
   IF s = '1'AND p = '0' THEN
      next_state <= playing1;
   ELSE
      next_state <= stopped2;
   END IF;
```

Continua na próxima página

Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2: Descrever em VHDL a FSM cujo diagrama de transição de estados é mostrado abaixo:

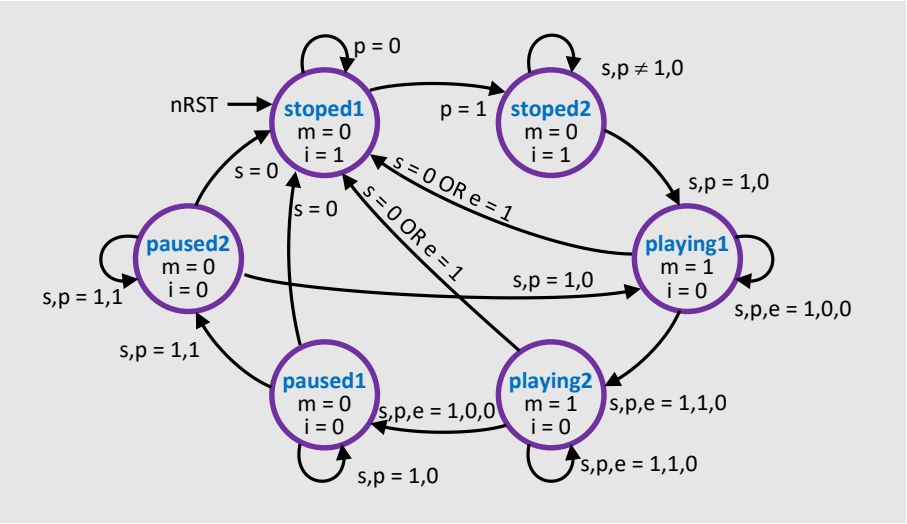



```
-----------------------------------------
-- PLAYING 1
-- Aguarda liberação do botão PLAY ou
-- ativação de STOP ou END
-----------------------------------------
WHEN playing1 =>
   ---------------------- Saídas:
   m <= '1';
   i <= '0';
   ---------------------- Próximo estado:
   IF s = '0' OR e = '1' THEN
      next_state <= stopped1;
   ELSIF p = '1' THEN
      next_state <= playing2;
   ELSE
      next_state <= playing1;
   END IF;
```

Continua na próxima página

Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2: Descrever em VHDL a FSM cujo diagrama de transição de estados é mostrado abaixo:

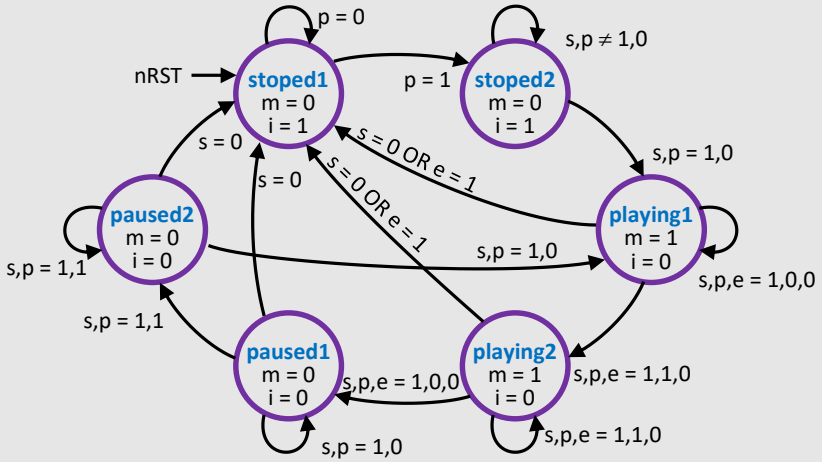



```
        ----------------------------------------
        -- PLAYING 2
        -- Aguarda ativação do botão PLAY (pausa)
        -- ou ativação de STOP ou END
        ----------------------------------------
        WHEN playing2 =>
           ---------------------- Saídas:
           m <= '1';
           i <= '0';
           ----------------------- Próximo estado:
           IF s = '0' OR e = '1' THEN
              next_state <= stopped1;
           ELSIF p = '0' THEN
              next_state <= paused1;
           ELSE
              next_state <= playing2;
           END IF;
```

Continua na próxima página

Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2: Descrever em VHDL a FSM cujo diagrama de transição de estados é mostrado abaixo:

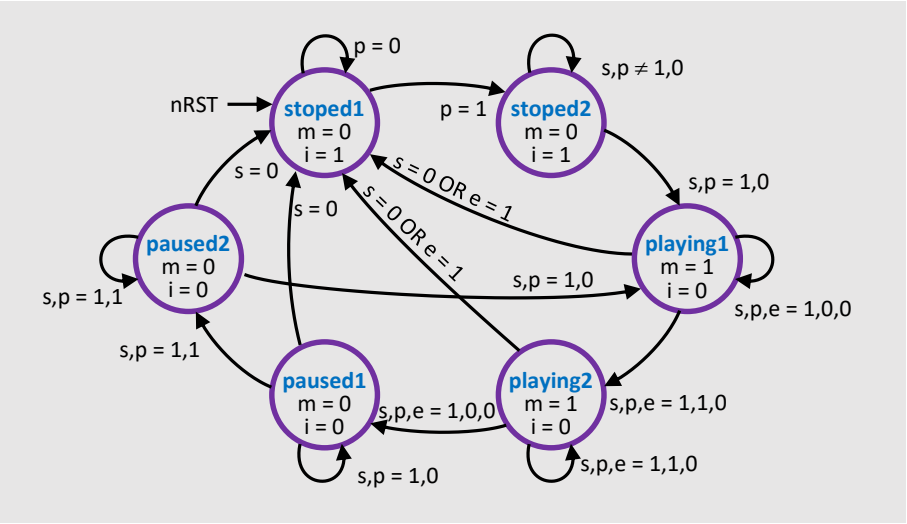



```
-----------------------------------------
-- PAUSED 1
-- Aguarda liberação do botão PLAY ou
-- ativação de STOP
-----------------------------------------
WHEN paused1 =>
   ---------------------- Saídas:
   m <= '0';
   i <= '0';
   ---------------------- Próximo estado:
   IF s = '0' THEN
         next_state <= stopped1;
   ELSIF p = '1' THEN
         next_state <= paused2;
   ELSE
         next_state <= paused1;
   END IF;
```

Continua na próxima página

Máquina de estados finitos
Exercício fsm_2

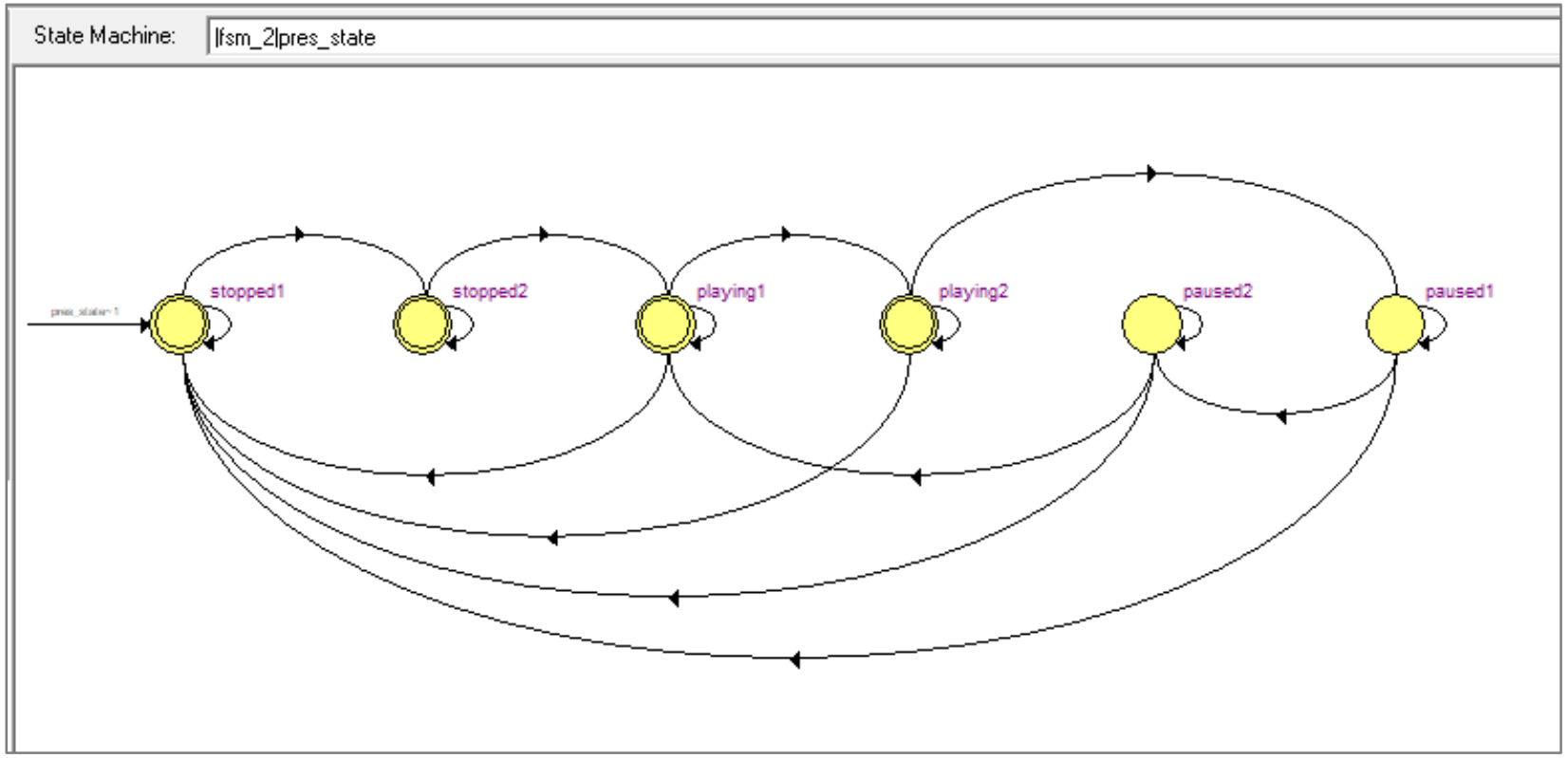
Exercício fsm_2: Descrever em VHDL a FSM cujo diagrama de transição de estados é mostrado abaixo:




```
        ----------------------------------------
        -- PAUSED 2
        -- Aguarda ativação do botão PLAY (fim de
        -- pausa) ou ativação de STOP
        ----------------------------------------
        WHEN paused2 =>
           ---------------------- Saídas:
           m <= '0';
           i <= '0';
           ---------------------- Próximo estado:
           IF s = '0' THEN
              next_state <= stopped1;
           ELSIF p = '0' THEN
              next_state <= playing1;
           ELSE
              next_state <= paused2;
           END IF;
     END CASE;
  END PROCESS;
END arch1;
```

Máquina de estados finitos
Exercício fsm_2

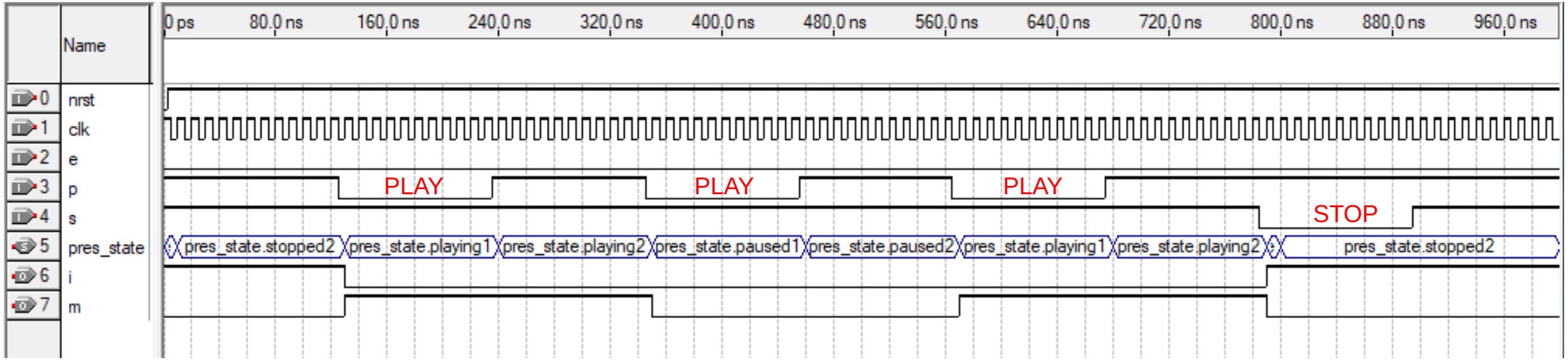
Exercício fsm_2 – Visualização dos estados no Quartus II:



	Source State	Destination State	Condition
1	paused1	paused1	(!p).(s)
2	paused1	paused2	(p).(s)
3	paused1	stopped1	(!s)
4	paused2	paused2	(!p).(s)
5	paused2	playing1	(p).(s)
6	paused2	stopped1	(!s)
7	playing1	playing1	(p).(s).(!e)
8	playing1	playing2	(!p).(s).(!e)
9	playing1	stopped1	(!s) + (s).(e)
10	playing2	paused1	(!p).(s).(!e)
11	playing2	playing2	(p).(s).(!e)
12	playing2	stopped1	(!s) + (s).(e)
13	stopped1	stopped1	(!p)
14	stopped1	stopped2	(p)
15	stopped2	playing1	(!p).(s)
16	stopped2	stopped2	(!p).(!s) + (p)

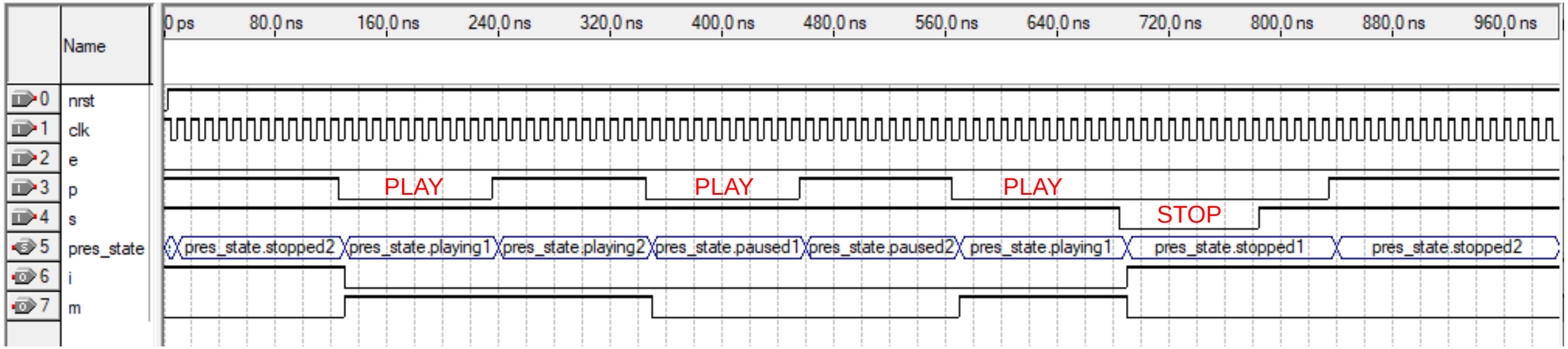
Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2 – Simulação funcional – sequência PLAY, PLAY, PLAY, STOP:



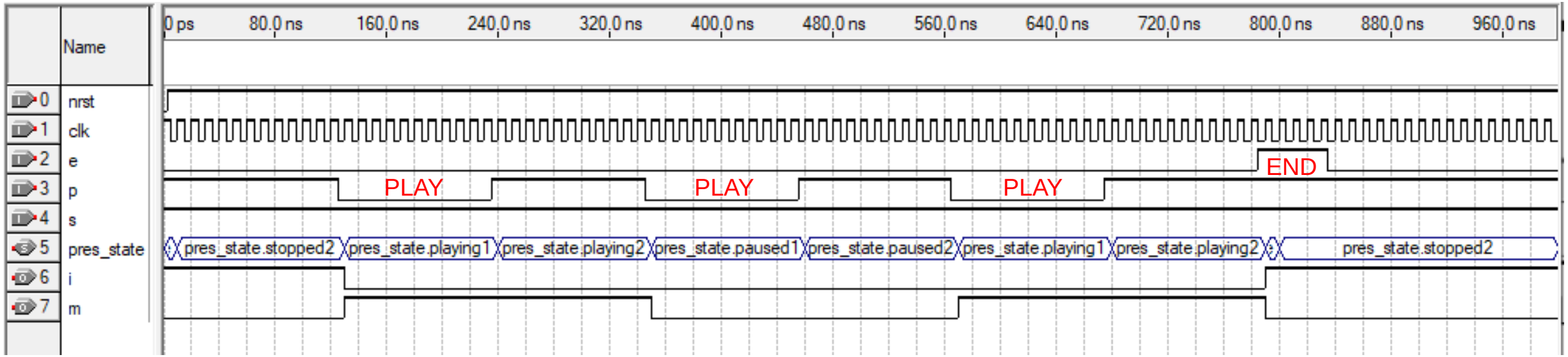
Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2 – Simulação funcional – sequência PLAY, PLAY, PLAY, STOP pressionado junto com PLAY:



Máquina de estados finitos
Exercício fsm_2

Exercício fsm_2 – Simulação funcional – sequência PLAY, PLAY, PLAY, END:

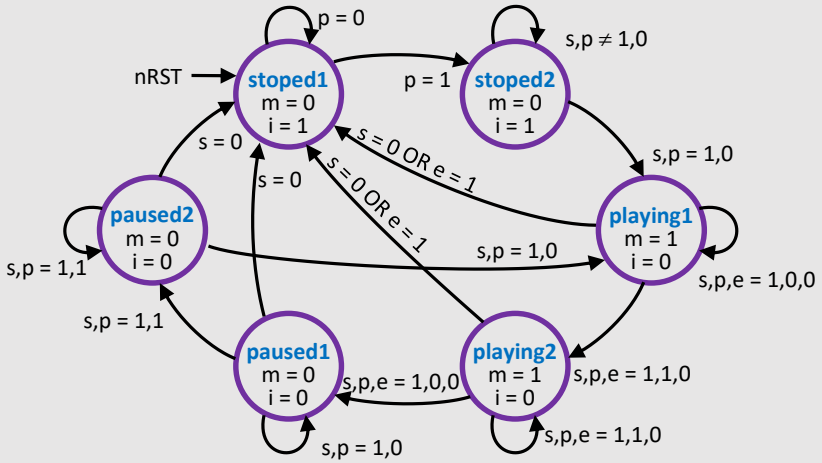


Máquina de estados finitos
Exercício fsm_2 – opção 2

Exercício fsm_2 – opção 2

Uma outra possibilidade para descrever a FSM em VHDL é usar um valor *default* para o próximo estado.

Pode ser visto que, em todos os estados, sempre existe uma condição que permanece no estado.



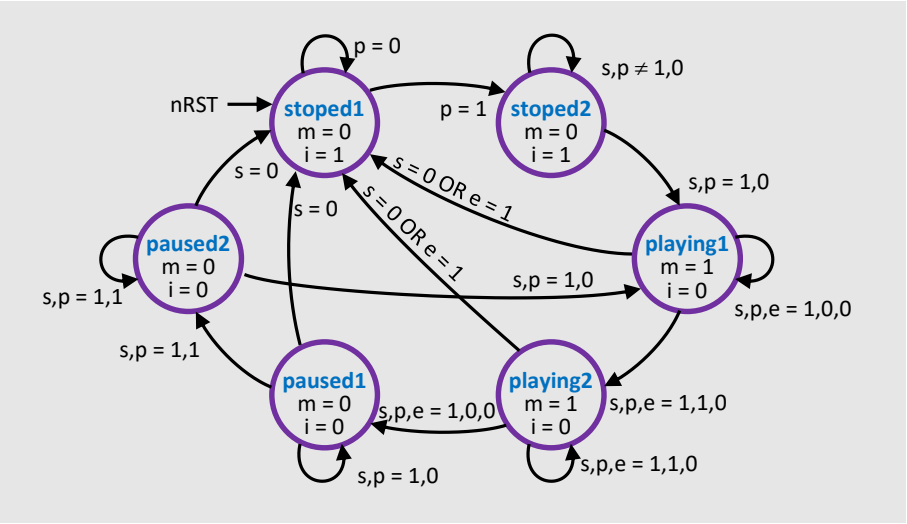

```
ARCHITECTURE arch2 OF fsm_2 IS
   TYPE state_type IS (stopped1, stopped2, playing1,
            playing2, paused1, paused2);
   SIGNAL next_state, pres_state : state_type;
BEGIN
   -- Parte sequencial da máquina de estados:
   PROCESS(nrst, clk)
   BEGIN
      IF nrst = '0' THEN
         pres_state <= stopped1;
      ELSIF RISING_EDGE(clk) THEN
         pres_state <= next_state;
      END IF;
   END PROCESS;
```

Continua na próxima página

Máquina de estados finitos
Exercício fsm_2 – opção 2

Exercício fsm_2 – opção 2

Nesse exemplo, vamos fazer a condição *default* para o próximo estado seja o estado atual.




```
-- Parte combinacional da máquina de estados:
PROCESS(pres_state, p, s, e)
BEGIN
   next_state <= pres_state;

   CASE pres_state IS
      ---------------------------------------
      -- STOPPED 1
      -- Aguarda liberação do botão PLAY
      ---------------------------------------
      WHEN stopped1 =>
         ---------------------- Saídas:
         m <= '0';
         i <= '1';
         -------------- Próximo estado:
         IF p = '1' THEN
            next_state <= stopped2;
         END IF;
```

Condição *default*

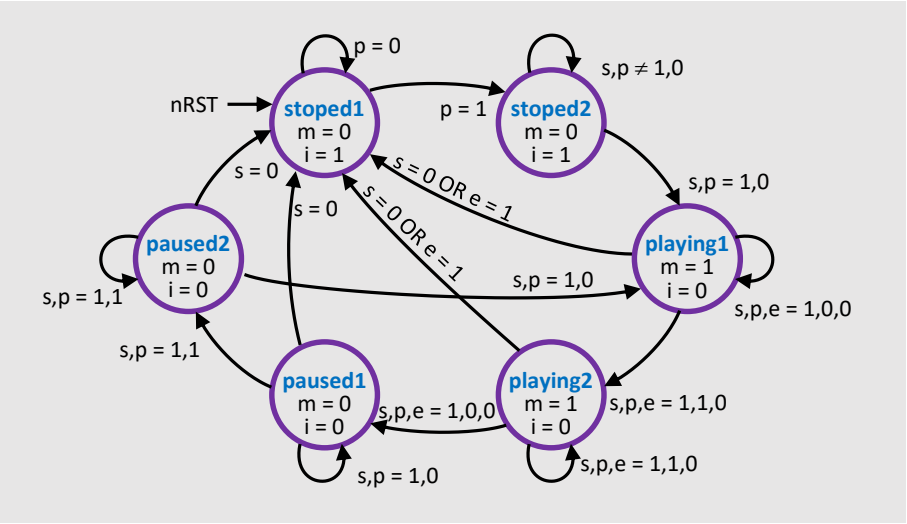
Não precisa do ELSE

Continua na próxima página

Máquina de estados finitos
Exercício fsm_2 – opção 2

Exercício fsm_2 – opção 2

Nesse exemplo, vamos fazer a condição *default* para o próximo estado seja o estado atual.



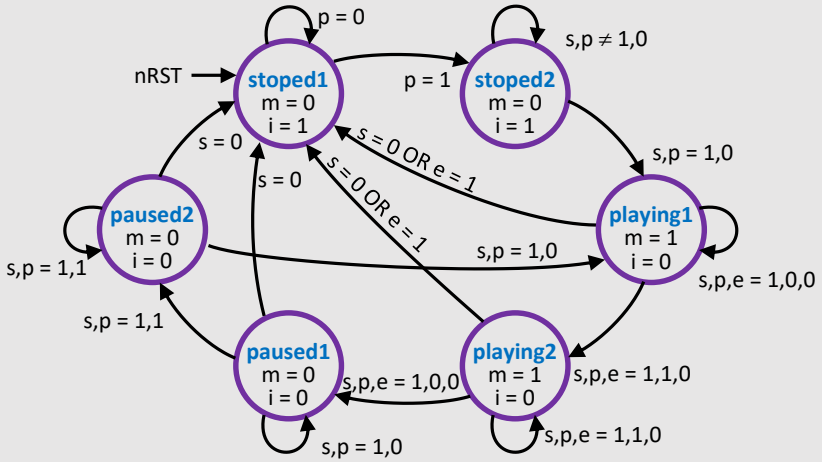

```
        ----------------------------------------
        -- STOPPED 2
        -- Aguarda ativação do botão PLAY
        ----------------------------------------
        WHEN stopped2 =>
           ---------------------- Saídas:
           m <= '0';
           i <= '1';
           ---------------------- Próximo estado:
           IF s = '1'AND p = '0' THEN
              next_state <= playing1;
           END IF;
```

Continua na próxima página

Máquina de estados finitos
Exercício fsm_2 – opção 2

Exercício fsm_2 – opção 2

Nesse exemplo, vamos fazer a condição *default* para o próximo estado seja o estado atual.



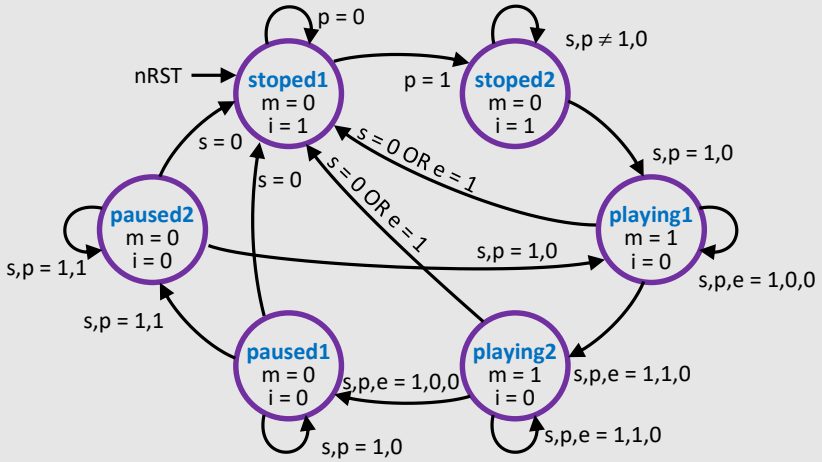

```
        ----------------------------------------
        -- PLAYING 1
        -- Aguarda liberação do botão PLAY ou
        -- ativação de STOP ou END
        ----------------------------------------
        WHEN playing1 =>
           ---------------------- Saídas:
           m <= '1';
           i <= '0';
           ---------------------- Próximo estado:
           IF s = '0' OR e = '1' THEN
              next_state <= stopped1;
           ELSIF p = '1' THEN
              next_state <= playing2;
           END IF;
```

Continua na próxima página

Máquina de estados finitos
Exercício fsm_2 – opção 2

Exercício fsm_2 – opção 2

Nesse exemplo, vamos fazer a condição *default* para o próximo estado seja o estado atual.



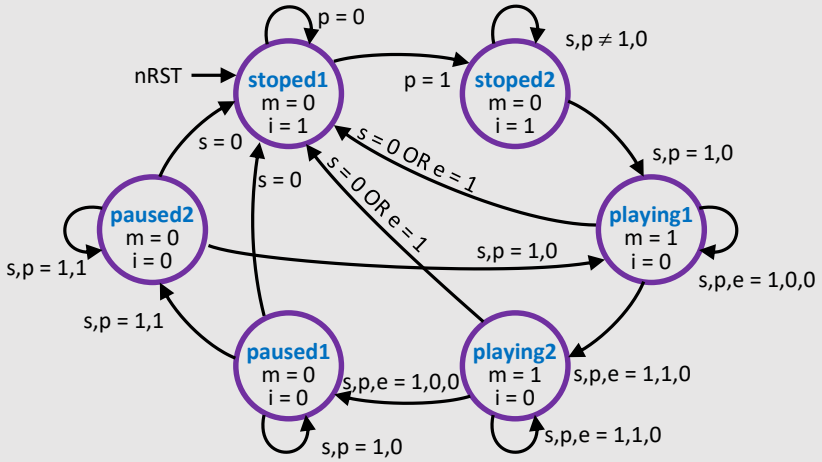

```
        ----------------------------------------
        -- PLAYING 2
        -- Aguarda ativação do botão PLAY (pausa)
        -- ou ativação de STOP ou END
        ----------------------------------------
        WHEN playing2 =>
           ---------------------- Saídas:
           m <= '1';
           i <= '0';
           ----------------------- Próximo estado:
           IF s = '0' OR e = '1' THEN
              next_state <= stopped1;
           ELSIF p = '0' THEN
              next_state <= paused1;
           END IF;
```

Continua na próxima página

Máquina de estados finitos
Exercício fsm_2 – opção 2

Exercício fsm_2 – opção 2

Nesse exemplo, vamos fazer a condição *default* para o próximo estado seja o estado atual.



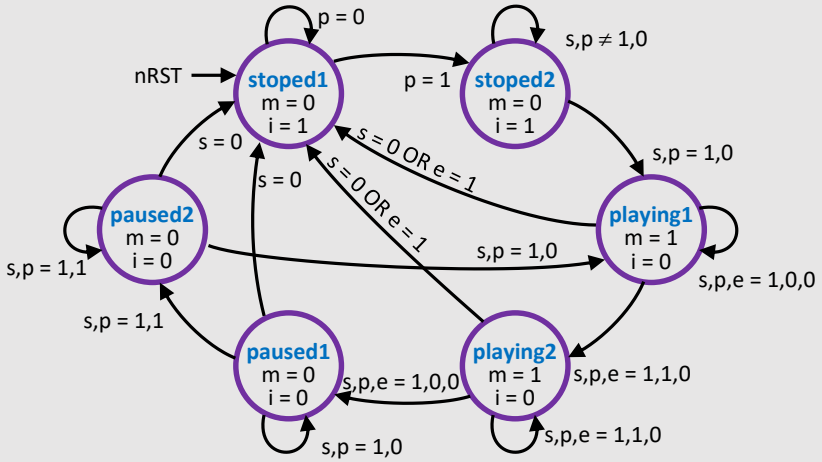

```
        ----------------------------------------
        -- PAUSED 1
        -- Aguarda liberação do botão PLAY ou
        -- ativação de STOP
        ----------------------------------------
        WHEN paused1 =>
           ---------------------- Saídas:
           m <= '0';
           i <= '0';
           ---------------------- Próximo estado:
           IF s = '0' THEN
              next_state <= stopped1;
           ELSIF p = '1' THEN
              next_state <= paused2;
           END IF;
```

Continua na próxima página

Máquina de estados finitos
Exercício fsm_2 – opção 2

Exercício fsm_2 – opção 2

Nesse exemplo, vamos fazer a condição *default* para o próximo estado seja o estado atual.




```
        ----------------------------------------
        -- PAUSED 2
        -- Aguarda ativação do botão PLAY (fim de
        -- pausa) ou ativação de STOP
        ----------------------------------------
        WHEN paused2 =>
           ---------------------- Saídas:
           m <= '0';
           i <= '0';
           ---------------------- Próximo estado:
           IF s = '0' THEN
              next_state <= stopped1;
           ELSIF p = '0' THEN
              next_state <= playing1;
           END IF;
     END CASE;
  END PROCESS;
END arch2;
```

Fim