

2.5. Descripción de máquinas de estado (FSM) en VHDL

Objetivos

- Qué es una FSM
- Tipos de FSM:
 - Mealy
 - Moore
- Descripción de máquinas de estado en VHDL:
 - Un proceso
 - Dos procesos

2.4.1. Qué son las máquinas de estado

Máquinas de estado finita (FSM)

- Es un circuito donde su salida depende del **valor de la entradas** ahora y del valor de las entradas antes (**estado actual**). → Cualquier circuito secuencial
- **Ejemplo: ascensor, contador, etc.**
- Basan su funcionamiento en los AUTÓMATAS FINITOS (matemáticas)
- La FSM avanza hacia cada estado según una secuencia predeterminada (tabla o diagrama de transición de estados).

Máquinas de estado (FSM)

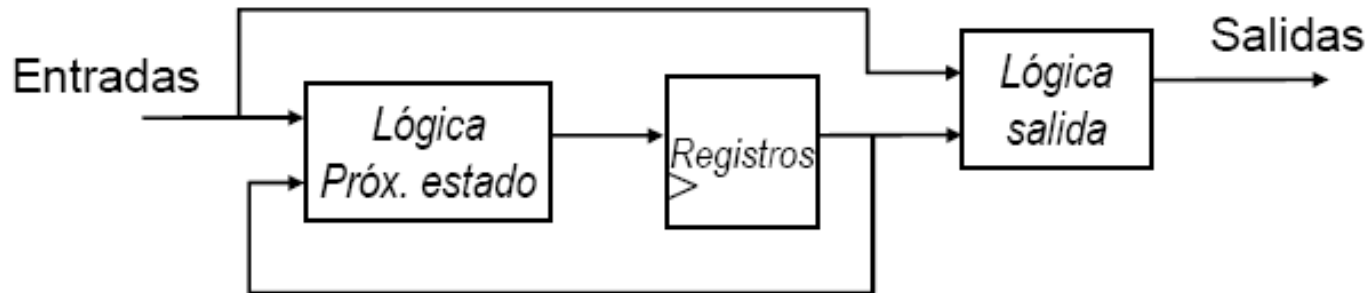
Proceso de diseño:

1. Análisis del problema para determinar entradas y estados posibles.
2. Decidir qué tipo de máquina se diseñará (Mealy o **Moore**)
3. Diseño del **diagrama** de transición **de estados** (grafo):
 - Transición entre estados
 - Salidas
4. Codificación VHDL

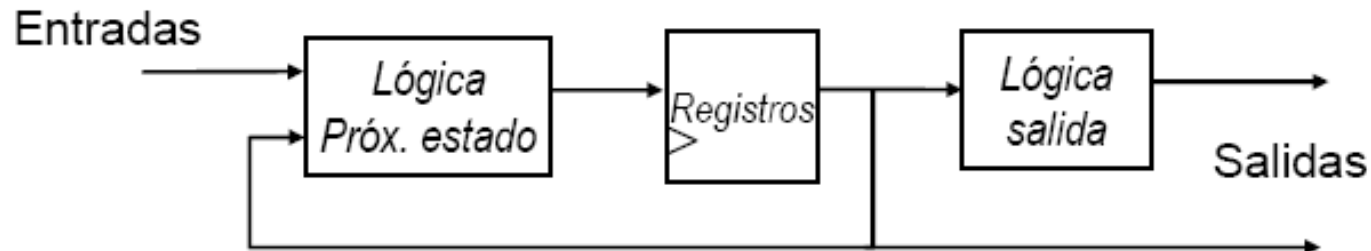
2.4.2. Tipos de máquinas de estado

Máquinas de estado (FSM)

- ✓ **Mealy**: las salidas son función del estado y entradas actuales



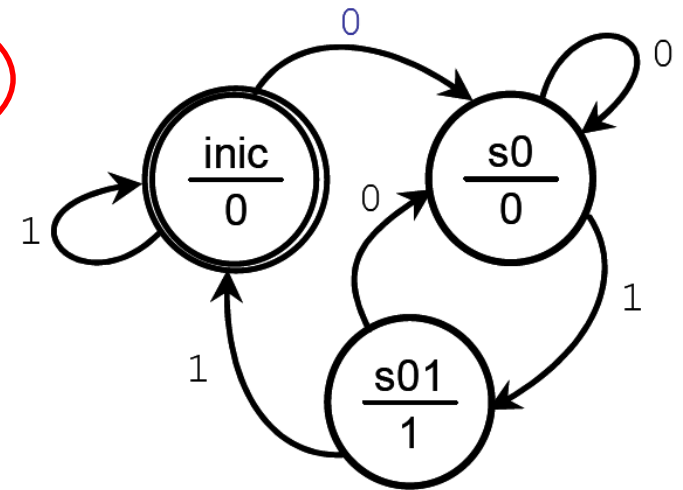
- ✓ **Moore**: las salidas son función sólo del estado actual



Máquinas de estado (FSM): ejemplo

Detector de flancos (Secuencia "01")

PULSADOR (IN)	ESTADO ACTUAL	ESTADO SIGUIENTE	Salida
0	inic	S0	0
1	inic	inic	0
0	S0	S0	0
1	S0	S01	1
0	S01	S0	0
1	S01	inic	0



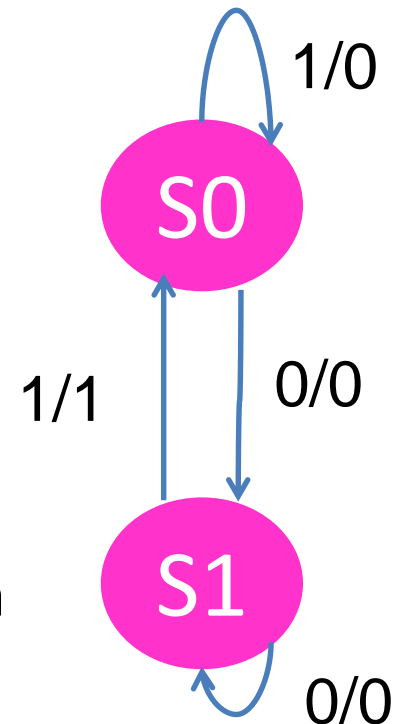
Solución A: Moore

Inic → Esperando '0'
S0 → Llegó un '0'
S01 → Llegó "01"

Máquinas de estado (FSM): ejemplo

Detector de flancos (Secuencia "01")

PULSADOR (IN)	ESTADO ACTUAL	ESTADO SIGUIENTE	Salida
0	S0	S1	0
1	S0	S0	0
0	S1	S1	0
1	S1	S0	1



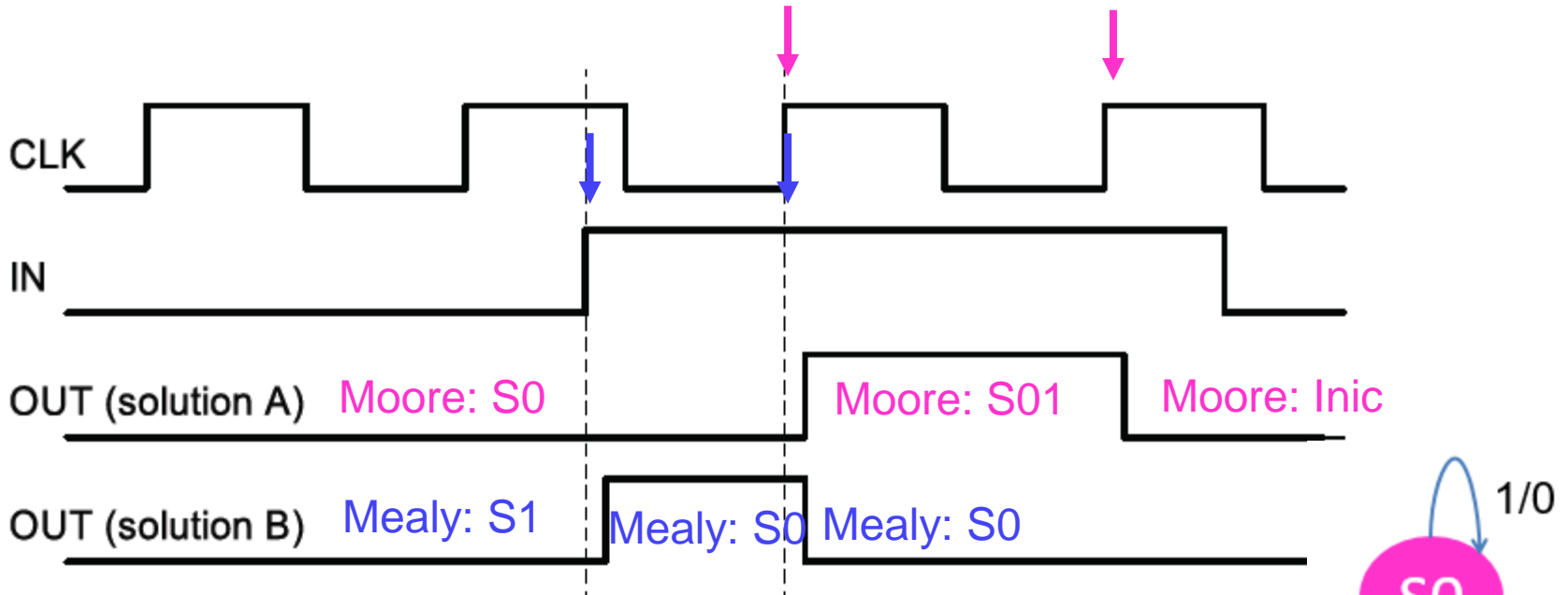
S0 → Esperando que empiece la secuencia

S1 → Llegó un "0"

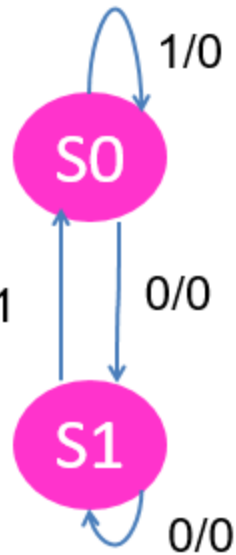
Solución B: Mealy

Máquinas de estado (FSM): ejemplo

Edge detector timing diagrams



- Solution A: output follows the clock
- Solution B: output changes with input rising edge and is^{1/1} asynchronous wrt the clock.



Máquinas de estado (FSM)

Moore	Mealy
Tiene más estados	Tiene menos estados
Tarda un ciclo más en dar la salida	Tarda un ciclo menos en dar la salida
Salida estable durante un ciclo completo de reloj	Salida NO estable durante un ciclo completo de reloj
Salida síncrona	Salida asíncrona

2.4.3. Descripción VHDL de FSM

Descripción VHDL de FSM: ejemplo secuencia “01”

Descripción de FSM MOORE

Máquina de estados en la que las salidas cambian sólo cuando cambia el estado, la salida no depende del valor de las entradas.

Estilos de descripción VHDL:

1. Dos procesos (o un proceso y sentencias concurrentes):
 - Combinacional → Salidas según estado
 - Secuencial → Estado siguiente
2. Un proceso:
 - Estado y salida

Tipo de dato enumerado definido por el usuario

- El usuario establece el NOMBRE del tipo y los elementos que lo forman
- Sintaxis:

Type <nombre tipo de dato> **is** (<elementos del tipo de dato que se define>);

- Luego podrá utilizarse este tipo de dato con cualquier objeto

Signal <nombre del dato> : < nombre tipo de dato >;

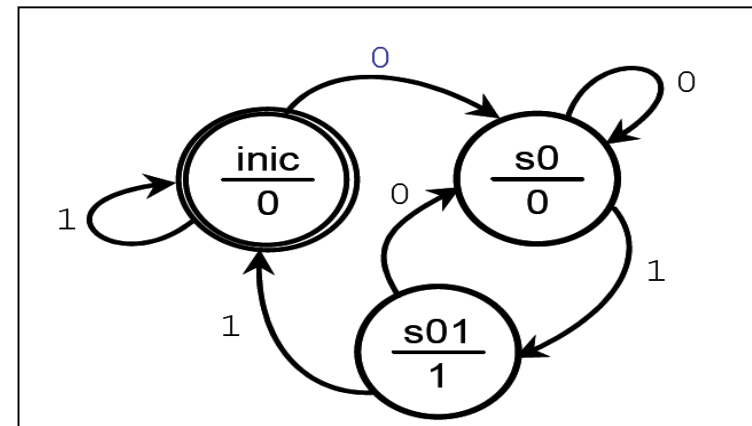
Descripción VHDL de FSM: ejemplo secuencia "01"

P28_FSM_EDGE_DETECTOR

```
entity FSM_EDGE_DETECTOR is
    Port ( RST_i,CLK_i,PUSH_i: in STD_LOGIC;
           PULSE_o           : out STD_LOGIC);
end FSM_EDGE_DETECTOR;
architecture Behavioral of FSM_EDGE_DETECTOR is
-----
-- 1. USER ENUMERATED TYPE FOR FSM
-----

type EdgeDetector_States is (inic,S0,S01);
signal State: EdgeDetector_States ;

begin
...
```



Descripción VHDL de FSM: ejemplo secuencia "01"

Sincronizando las entradas

P28_FSM_EDGE_DETECTOR

```
signal PUSH: std_logic;

begin
-----
-- 2. INPUTs synchronisation
-----

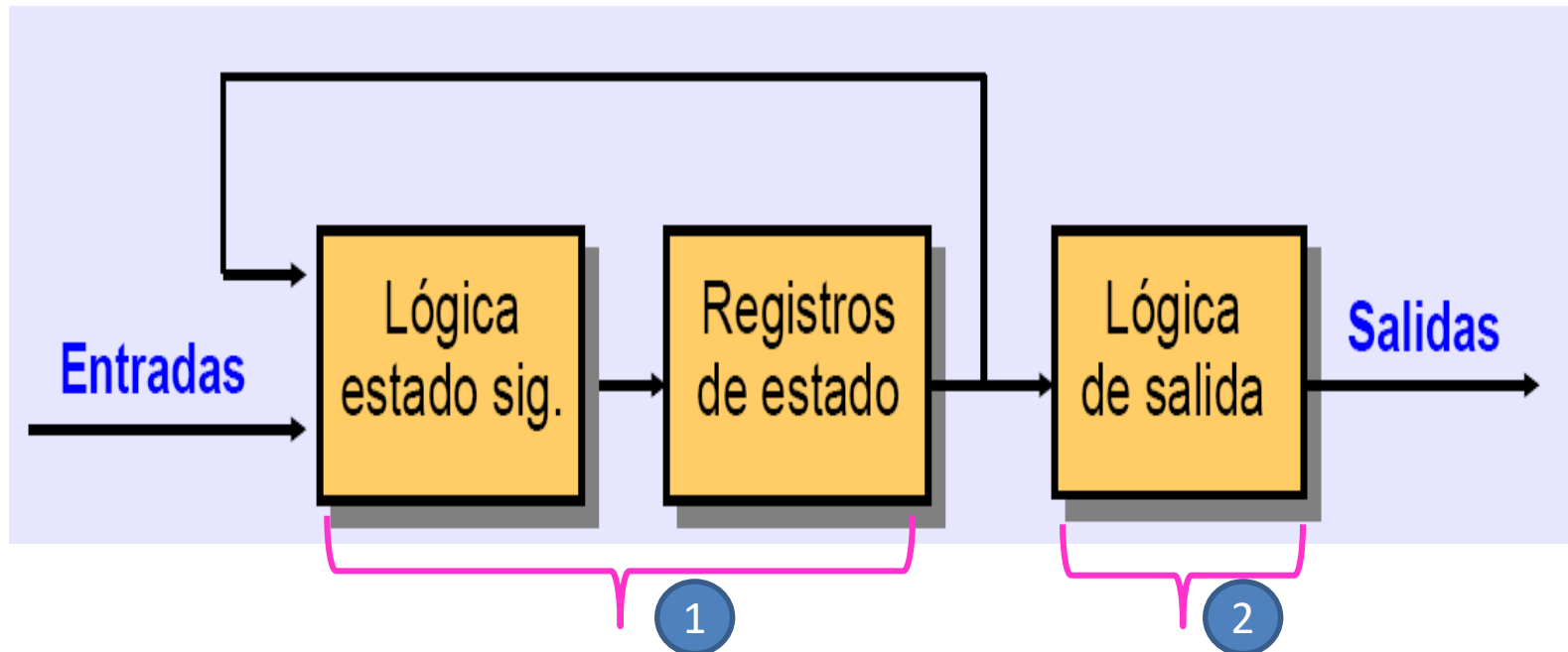
SYNC_INPUT: Process (RST_i, CLK_i)
begin
    if RST_i = '1' then PUSH <= '0';
    -- INICIO CON RESET
    elsif rising_edge(CLK_i) then
        PUSH <= PUSH_i;
    end if;
end process SYNC_INPUT;
```

Fuente: Dpto. Tec.Electrónica U.Vigo

2.4.3.1. FSM mediante dos procesos

Descripción VHDL de FSM: ejemplo secuencia "01"

Dos procesos



Diferenciamos dos bloques:

- 1- **Proceso secuencial** → Estado futuro/siguiente
- 2- **Proceso combinacional** o Sentencias concurrentes → Valor de las salidas según el estado.

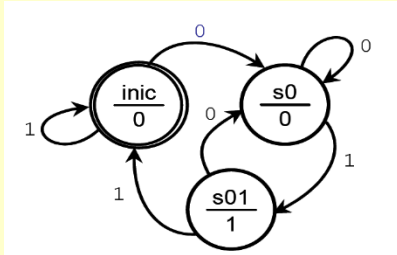
Descripción VHDL de FSM: ejemplo secuencia "01"

```
Process (RST_i, CLK_i)
begin
```

Proceso 1

```
    if RST_i = '1' then
        STATE <= inic; -- INICIO CON RESET
    elsif rising_edge(CLK_i) then
        case STATE is
            when inic => if PUSH = '0' then
                           STATE <= S0;      --llega "0-"
                        end if;
            when S0    => if PUSH = '1' then
                           STATE <= S01;     --llega "01"
                        end if;
            when S01   => if PUSH = '0' then
                           STATE <= S0;      --"0" para "0-"
                        else
                           STATE <= inic;    --llega un "1",
                        end if;
            when others => STATE <= inic;
        end case;
    end if;
end process;
```

Estado Actual
Entradas



Descripción VHDL de FSM: ejemplo secuencia "01"

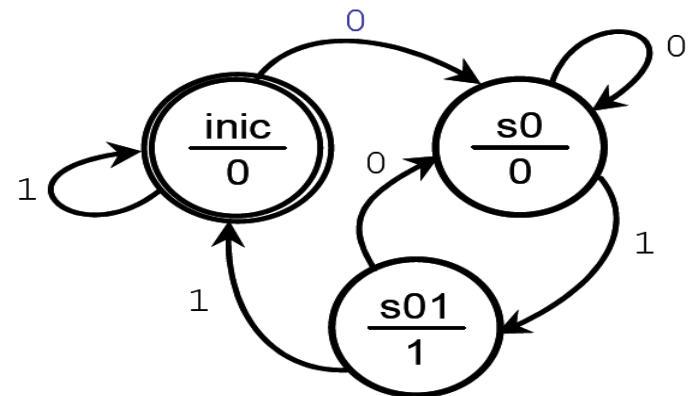
Dos procesos:

Primera opción → Sentencias concurrentes

Proceso 2

```
-----  
-- TWO PROCESS: CONCURRENTS ASSIGNMENTS TO DECODE OUTPUTS  
-----
```

```
with STATE select  
    PULSE_o <= '0' when inic,  
                '0' when s0,  
                '1' when s01,  
                '0' when others; -- catch all  
end Behavioral;
```



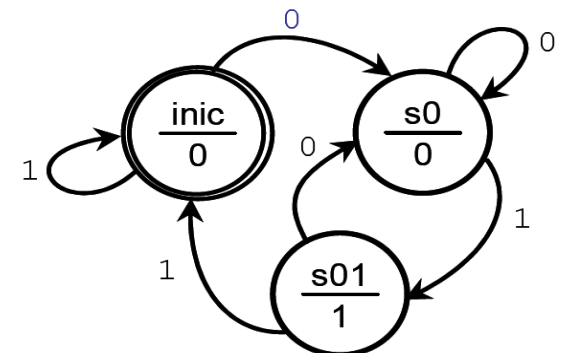
Descripción VHDL de FSM: ejemplo secuencia "01"

Dos procesos:

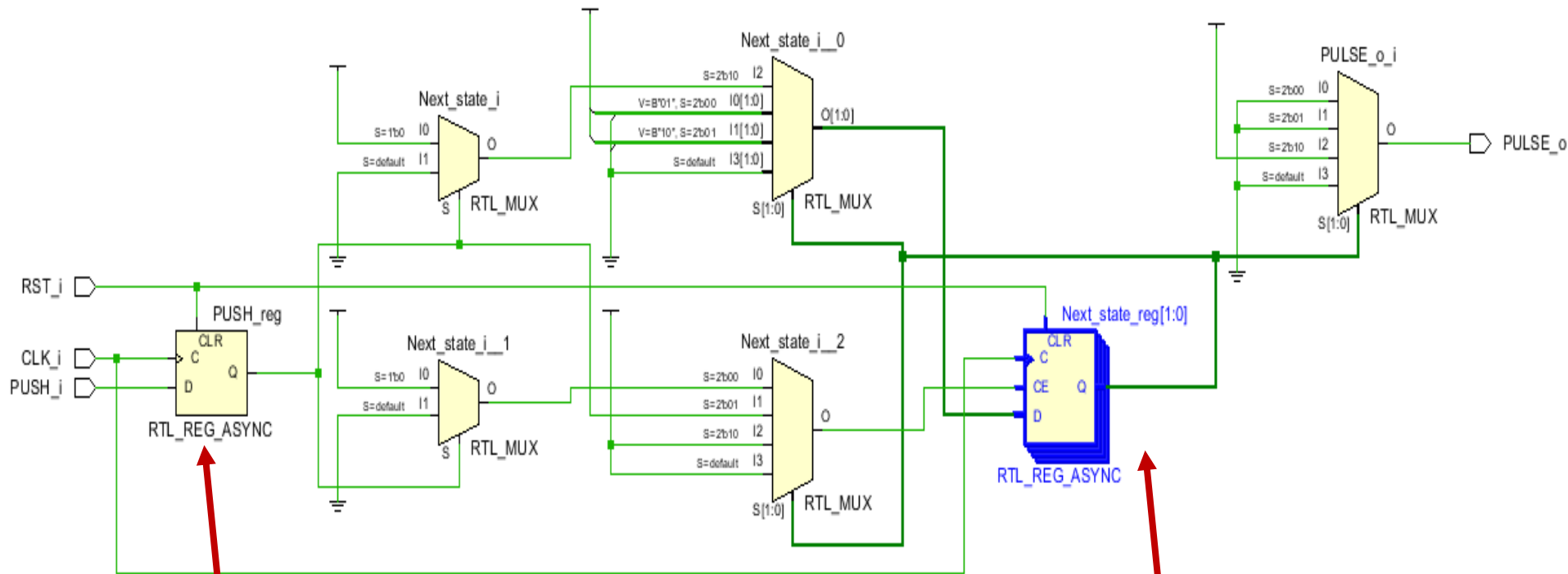
Segunda opción → Proceso combinacional

Proceso 2

```
-----  
-- TWO PROCESS: COMBINATIONAL PROCESS TO DECODE OUTPUTS  
-----  
  
process (STATE)  
begin  
    case STATE is  
        when inic      => PULSE_o <= '0';  
        when S0        => PULSE_o <= '0';  
        when S01       => PULSE_o <= '1';  
        when others    => PULSE_o <= '0';  
    end case;  
end process;
```



Descripción VHDL de FSM: ejemplo secuencia "01"



Sincroniza la entrada

Registra STATE

Descripción VHDL de FSM: ejemplo secuencia "01"

