Asignación de señales en VHDL

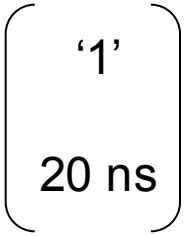
Microarquitecturas y softcores

Laboratorio de Sistemas Embebidos

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

VHDL: Asignación de señales

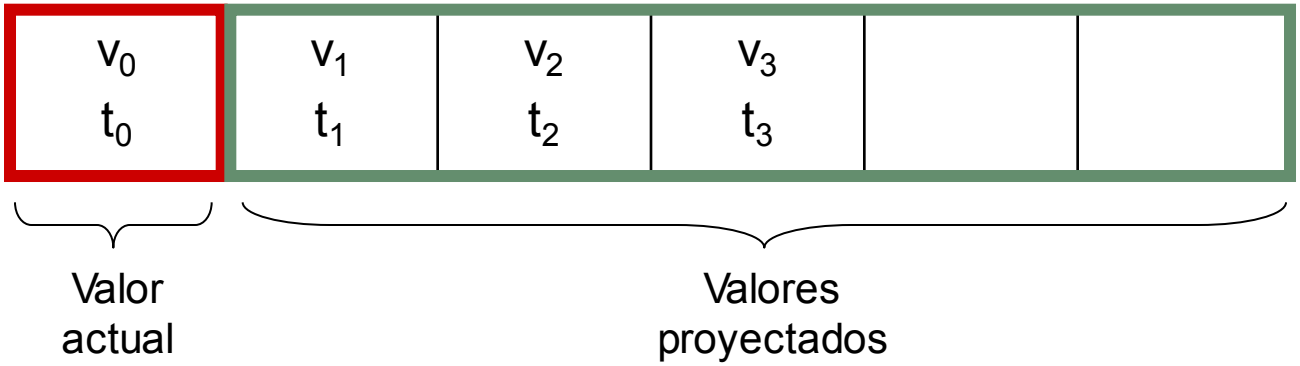
- La asignación a una señal establece una o más transacciones sobre la misma
- Cada señal tiene asociada una forma de onda proyectada, que es una lista de transacciones que indica los valores futuros de la señal.
- Por ejemplo: s <= ‘1’ after 20 ns;
  causará que la señal tome el valor ‘1’, 20 ns después de ejecutada la asignación. Esto se representa de la siguiente manera:



- Cuando la señal cambia de valor se dice que ha ocurrido un **evento** sobre ella.

VHDL: Asignación de señales

- El valor actual y los proyectados constituyen el driver de la señal



Ejemplo: a <= ‘1’ after 10 ns, ‘0’ after 15 ns, ‘1’ after 30 ns;

‘1’ 10 ns	‘0’ 15 ns	‘1’ 30 ns			

VHDL: Asignación de señales

La asignación puede realizarse de acuerdo a **dos tipos de retardo**: **inercial y transporte**.

Retardo de transporte

- Caso 1: la nueva transacción creada tiene un tiempo superior al de las demás transacciones existentes en el driver

  La nueva transacción se añade al final del driver

- Caso 2: la nueva transacción creada tiene un tiempo inferior que el de alguna de las demás transacciones existentes en el driver

  La nueva transacción se añade al driver y todas las transacciones que poseen tiempos superiores al de ella son eliminadas del driver

VHDL: Asignación de señales

La asignación puede realizarse de acuerdo a **dos tipos de retardo**: **inercial y transporte**.

Retardo de transporte

Ejemplo:

Se considera declarada la señal a (*signal a: std_logic := 'Z';*)

```
pp: process
begin
    a <= '1' after 10 ns, '0' after 25 ns, '1' after 40 ns, '0' after 50 ns;
    a <= transport '0' after 30 ns;
    wait;
end process;
```

'1' 10 ns	'0' 25 ns	'1' 40 ns	'0' 50 ns

Primera asignación

'1' 10 ns	'0' 25 ns	'0' 30 ns	

Segunda asignación

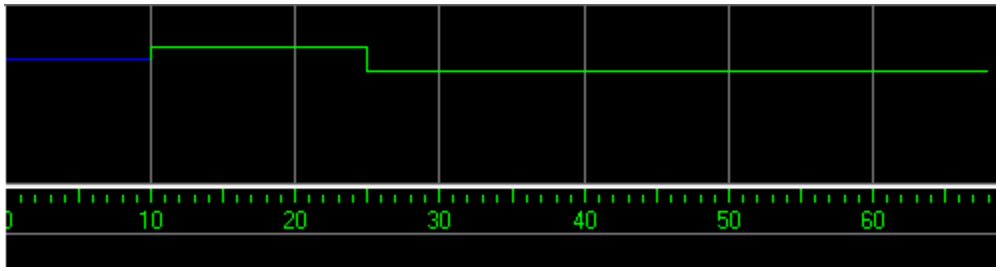
VHDL: Asignación de señales

La asignación puede realizarse de acuerdo a **dos tipos de retardo**: **inercial y transporte**.

Retardo de transporte

Ejemplo:

Se considera declarada la señal a (***signal*** *a: std_logic :=* *'Z';*)



'1' 10 ns	'0' 25 ns	'1' 40 ns	'0' 50 ns

Primera asignación

'1' 10 ns	'0' 25 ns	'0' 30 ns	

Segunda asignación

VHDL: Asignación de señales

La asignación puede realizarse de acuerdo a **dos tipos de retardo**: **inercial y transporte**.

Retardo inercial

- Todas las transacciones posteriores a la nueva se eliminan del driver (del mismo modo que en transporte).

- Todas las transacciones anteriores a la nueva, a partir del primer valor distinto, se eliminan.

VHDL: Asignación de señales

La asignación puede realizarse de acuerdo a **dos tipos de retardo**: **inercial y transporte**.

Retardo inercial

Ejemplo:

Se considera declarada la señal a (***signal*** *a: std_logic :=* *'Z'*;)

```
pp: process
begin
    a <= '1' after 10 ns, '0' after 25 ns, '1' after 40 ns, '0' after 50 ns;
    a <= '1' after 45, '0' after 70 ns;
    wait;
end process;
```

'1' 10 ns	'0' 25 ns	'1' 40 ns	'0' 50 ns

Primera asignación

'1' 40 ns	'1' 45 ns	'0' 70 ns	

Segunda asignación

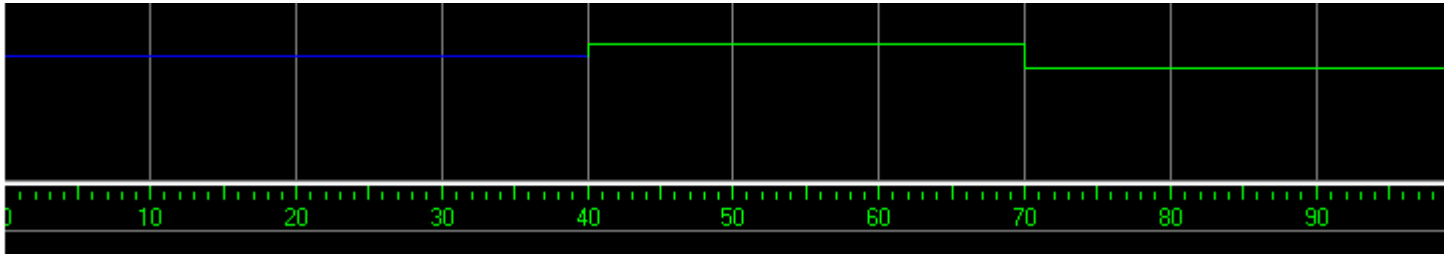
VHDL: Asignación de señales

La asignación puede realizarse de acuerdo a **dos tipos de retardo**: **inercial y transporte**.

Retardo inercial

Ejemplo:

Se considera declarada la señal a (***signal*** *a: std_logic :=* *'Z';*)



'1' 10 ns	'0' 25 ns	'1' 40 ns	'0' 50 ns

Primera asignación

'1' 40 ns	'1' 45 ns	'0' 70 ns	

Segunda asignación

VHDL: Asignación de señales

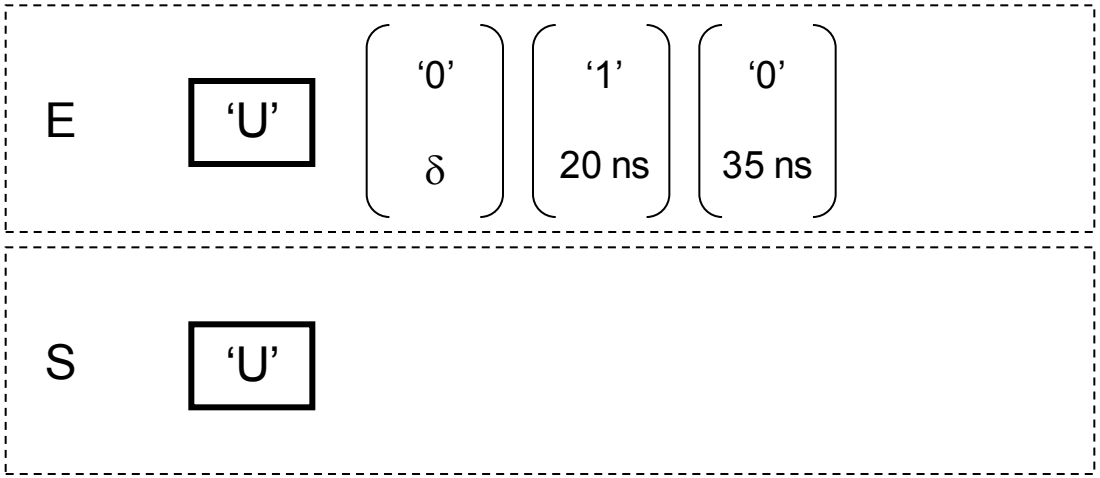
- **Ejemplo 1**

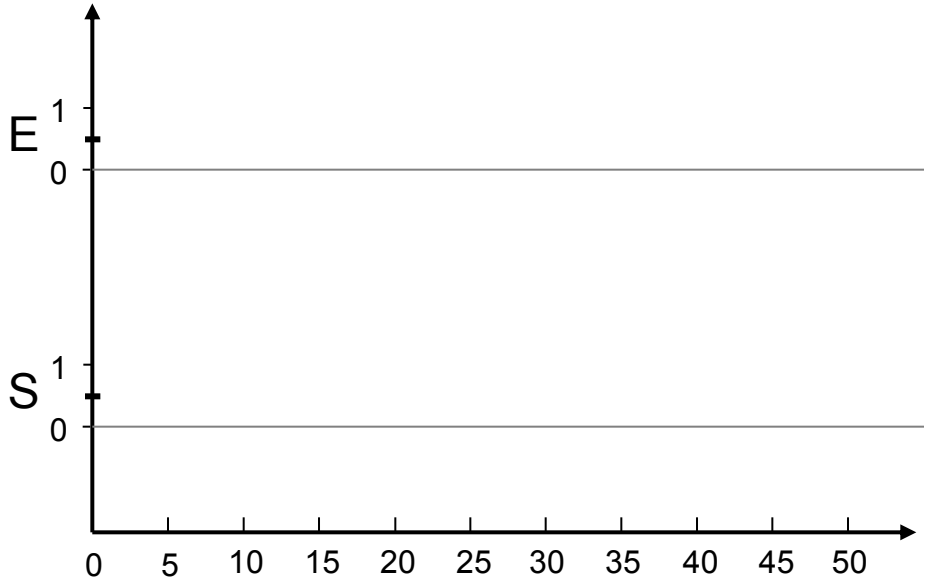
```
entity pru_inercial is
    port (
        E: in std_logic;
        S: out std_logic
    );
end;

architecture beh of pru_inercial is
begin
    process(E)
    begin
        S <= E after 10 ns;
    end process;
end;
```

```
entity pulso is
end;

architecture beh of pulso is
    signal E: std_logic;
    signal S: std_logic;
    component pru_inercial is
        port (
            E: in std_logic;
            S: out std_logic
        );
    end component;
begin
    E <= '0', '1' after 20 ns, '0' after 35 ns;
    aa: pru_inercial port map(E => E, S => S);
end;
```

VHDL: Asignación de señales

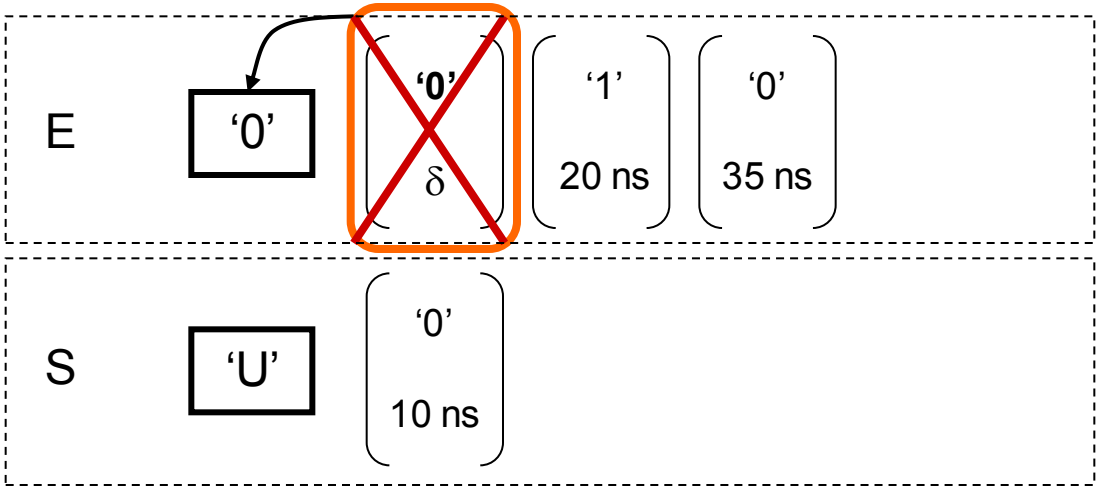
t = 0

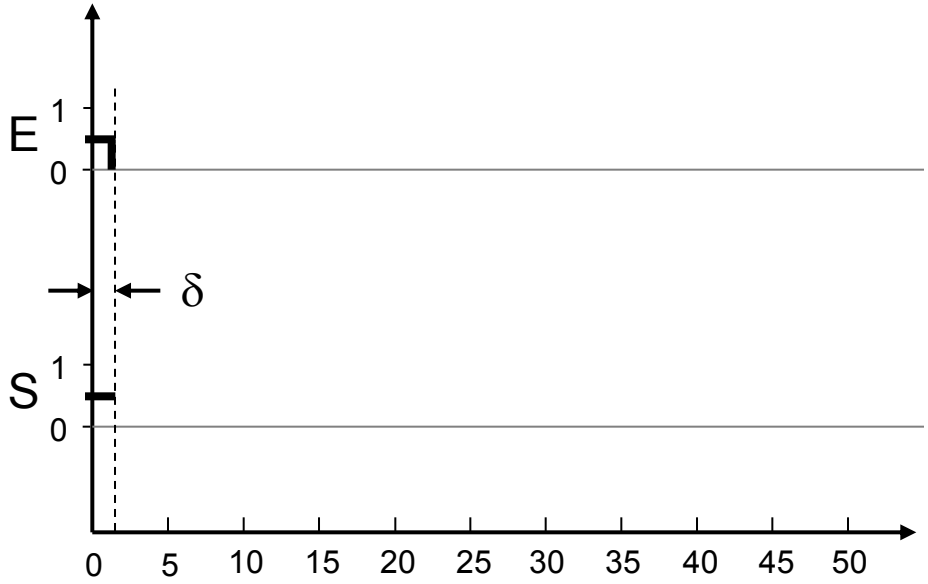




VHDL: Asignación de señales

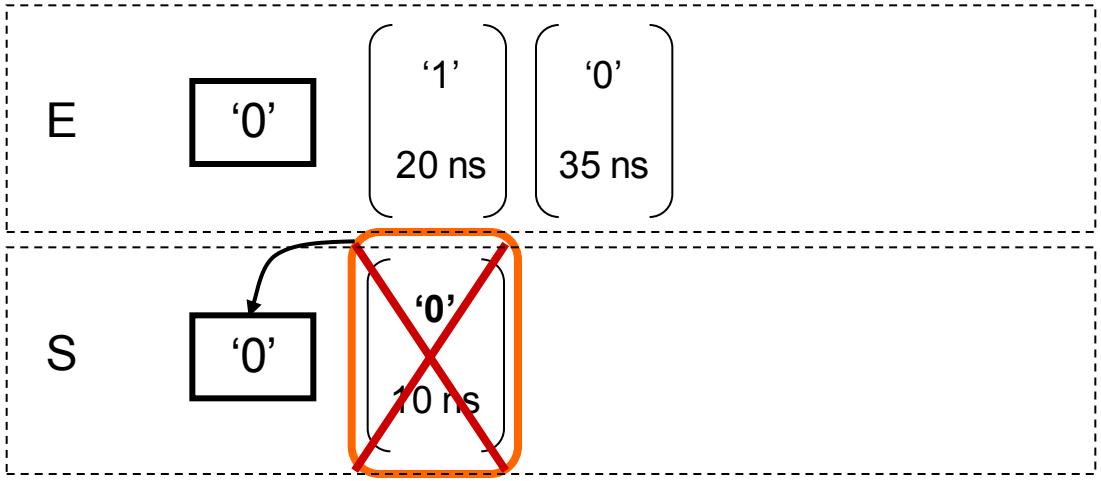
t = δ

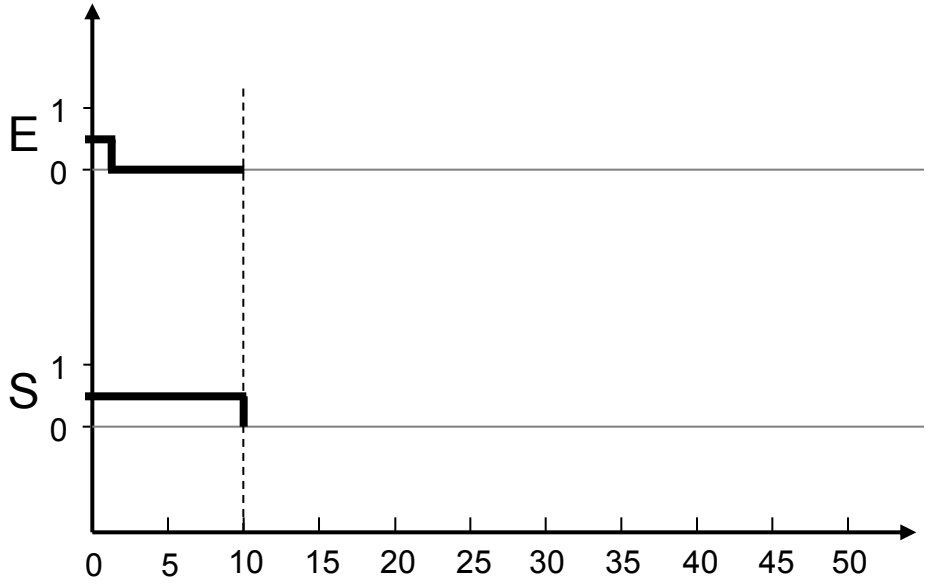




VHDL: Asignación de señales

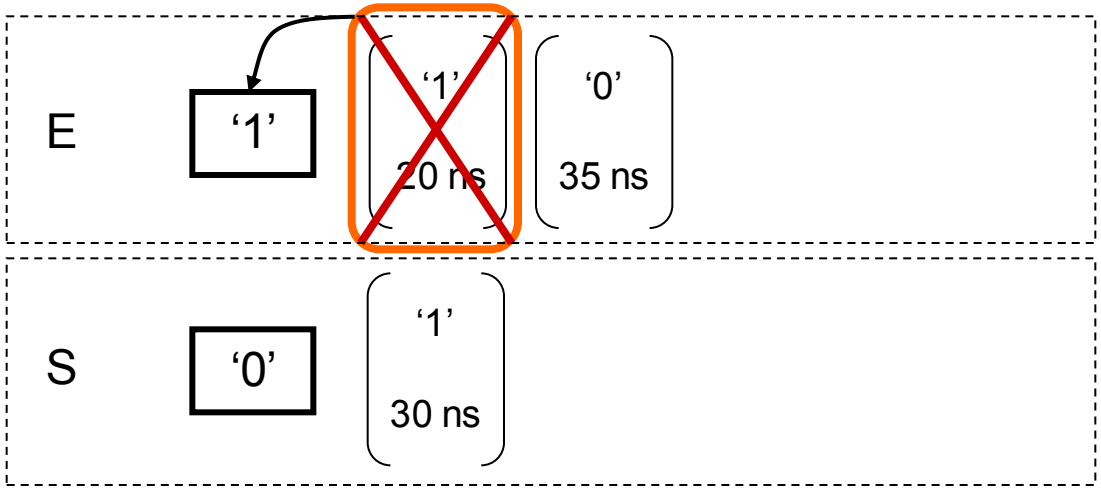
t = 10

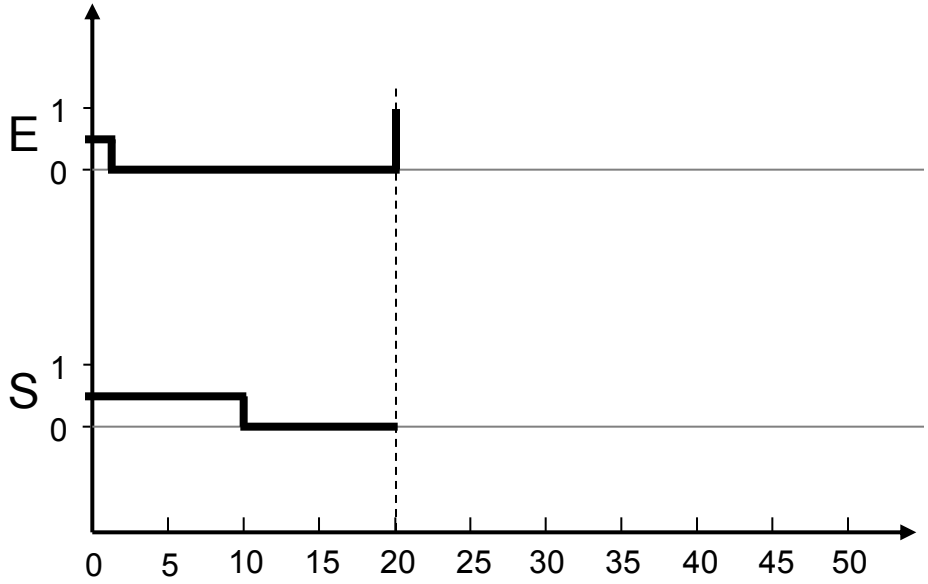




VHDL: Asignación de señales

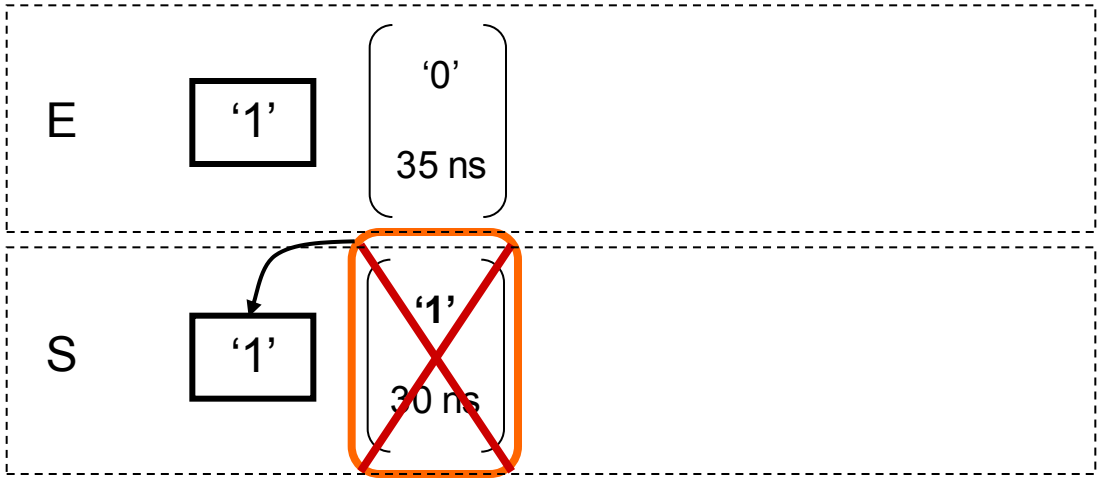
t = 20

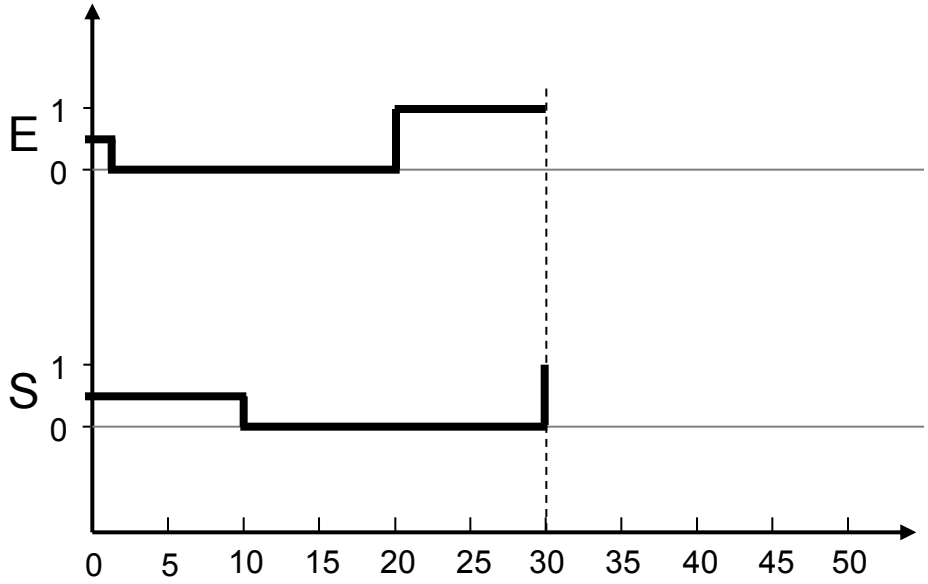




VHDL: Asignación de señales

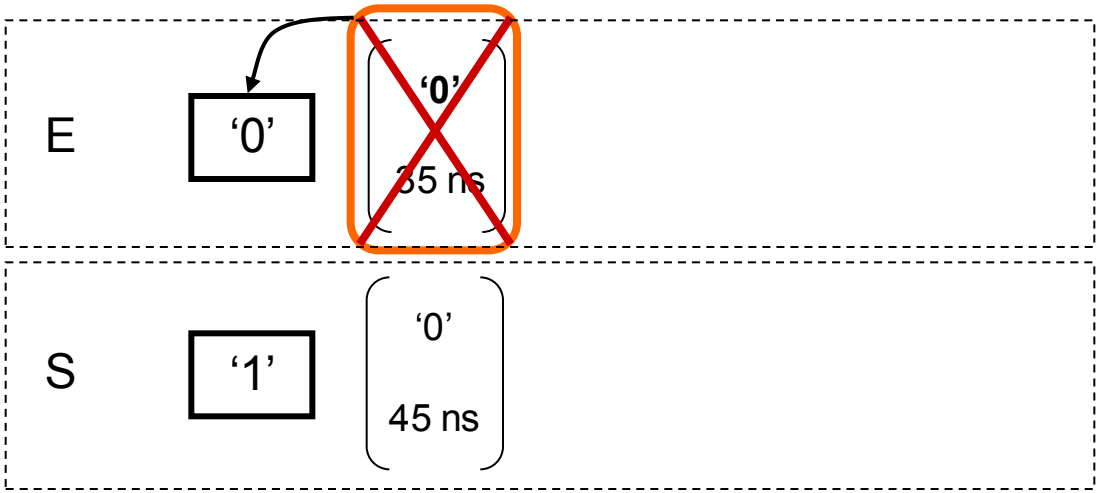
t = 30

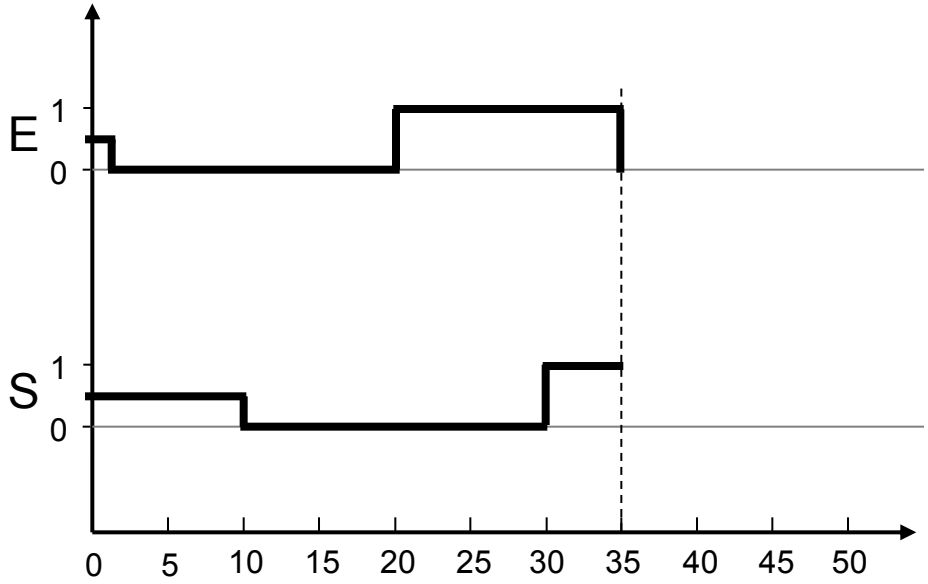




VHDL: Asignación de señales

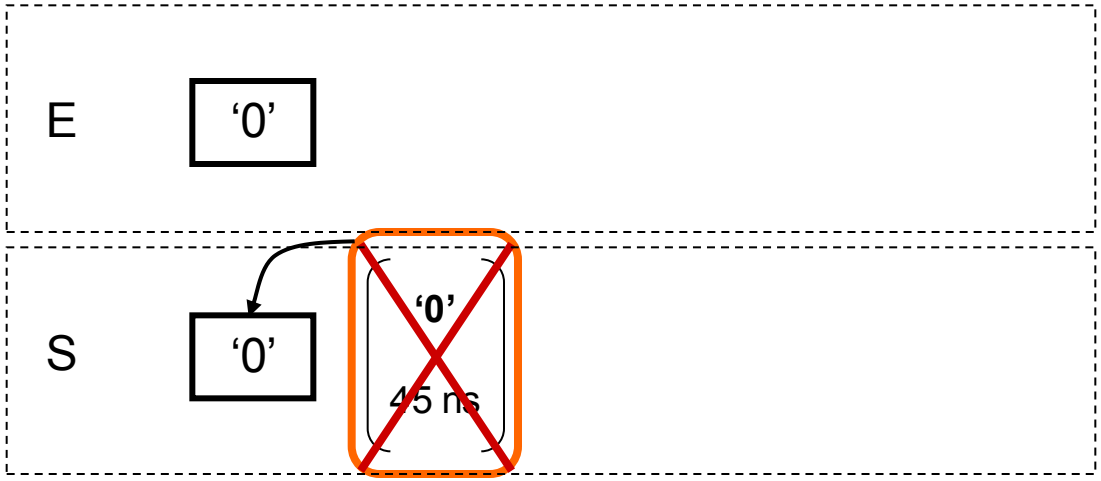
t = 35

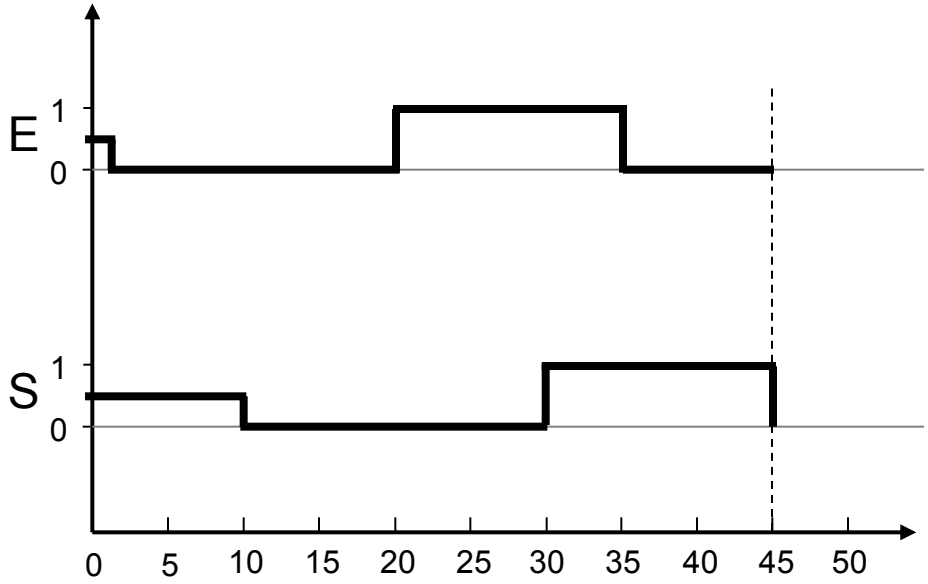




VHDL: Asignación de señales

t = 45



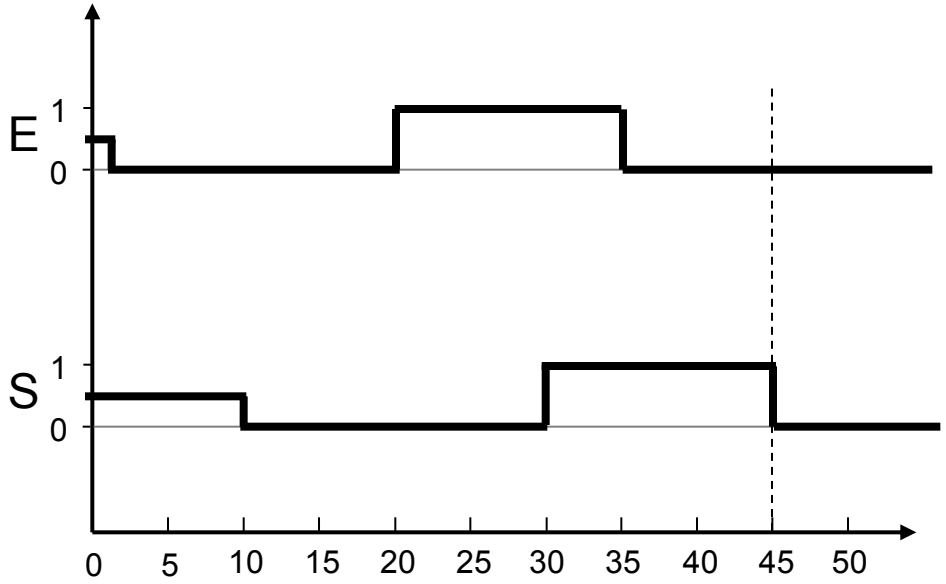


VHDL: Asignación de señales

t > 45

E ‘0’

S ‘0’

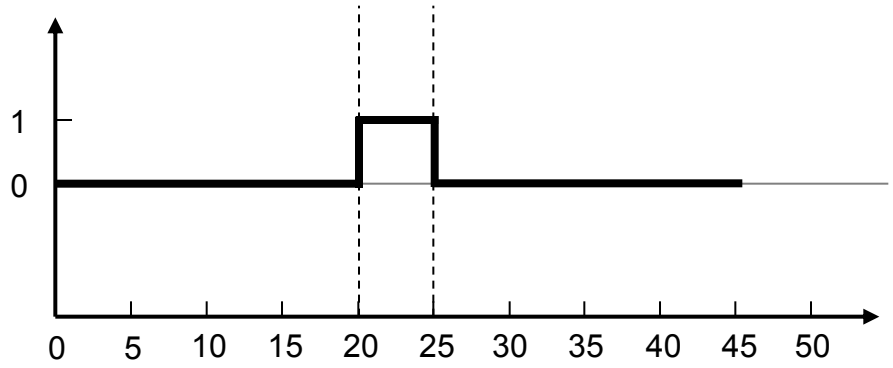


VHDL: Asignación de señales

- **Ejemplo 2**

¿Qué ocurriría si la entrada fuera un pulso de 5 ns?

E <= ‘0’, ‘1’ after 20 ns, ‘0’ after 25 ns;



VHDL: Asignación de señales

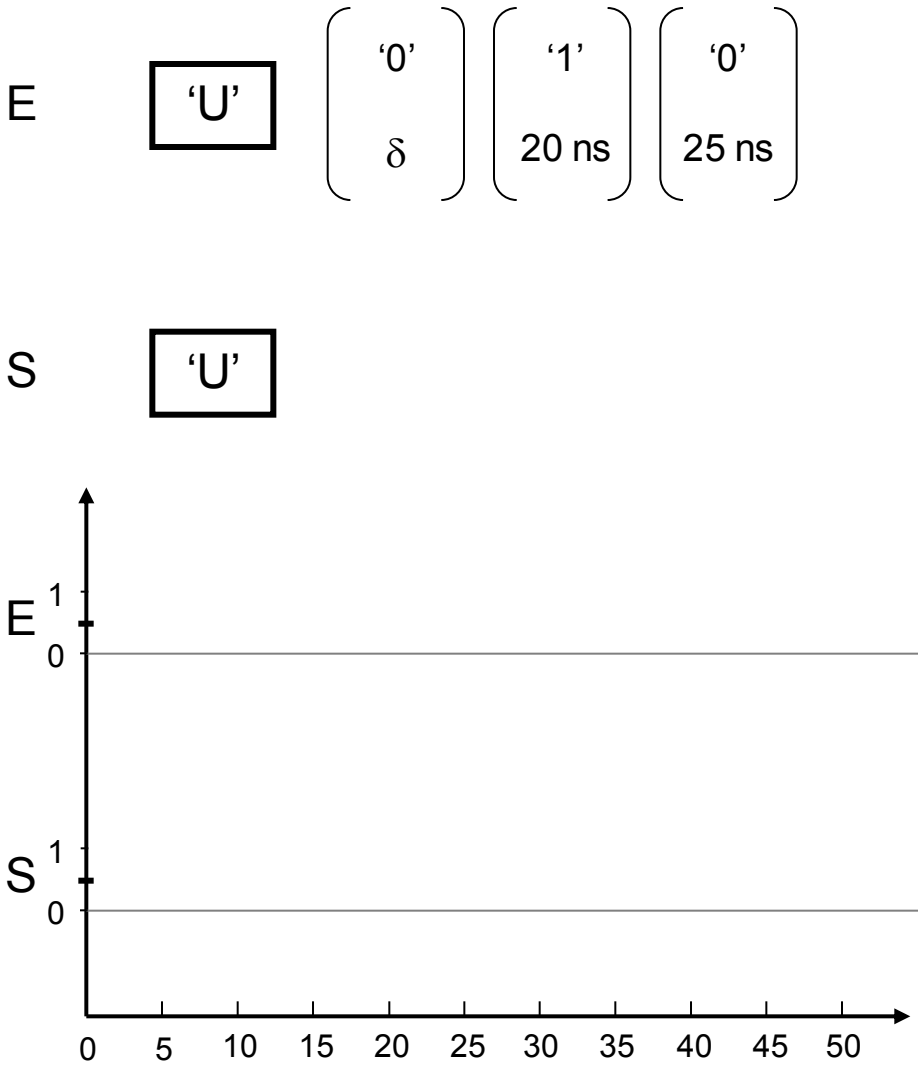
- **Ejemplo 2**

```
entity pru_inercial is
    port (
        E: in std_logic;
        S: out std_logic
    );
end;

architecture beh of pru_inercial is
begin
    process(E)
    begin
        S <= E after 10 ns;
    end process;
end;
```

```
entity pulso is
end;

architecture beh of pulso is
    signal E: std_logic;
    signal S: std_logic;
    component pru_inercial is
        port (
            E: in std_logic;
            S: out std_logic
        );
    end component;
begin
    E <= '0', '1' after 20 ns, '0' after 25 ns;
    aa: pru_inercial port map(E => E, S => S);
end;
```

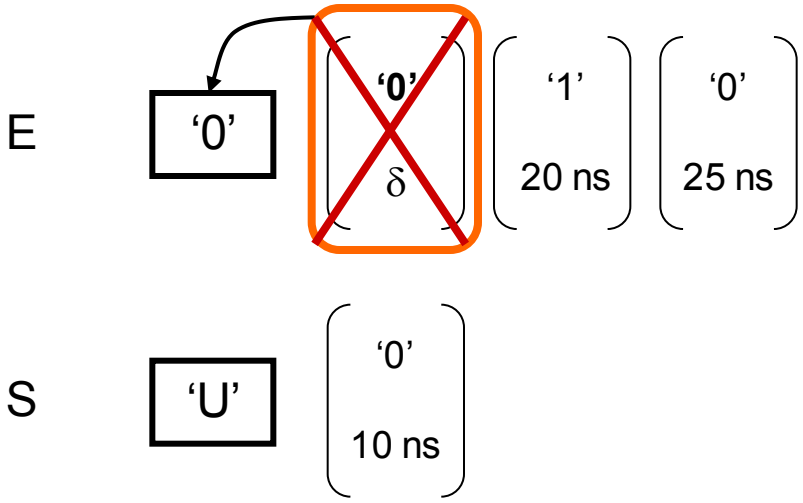
VHDL: Asignación de señales

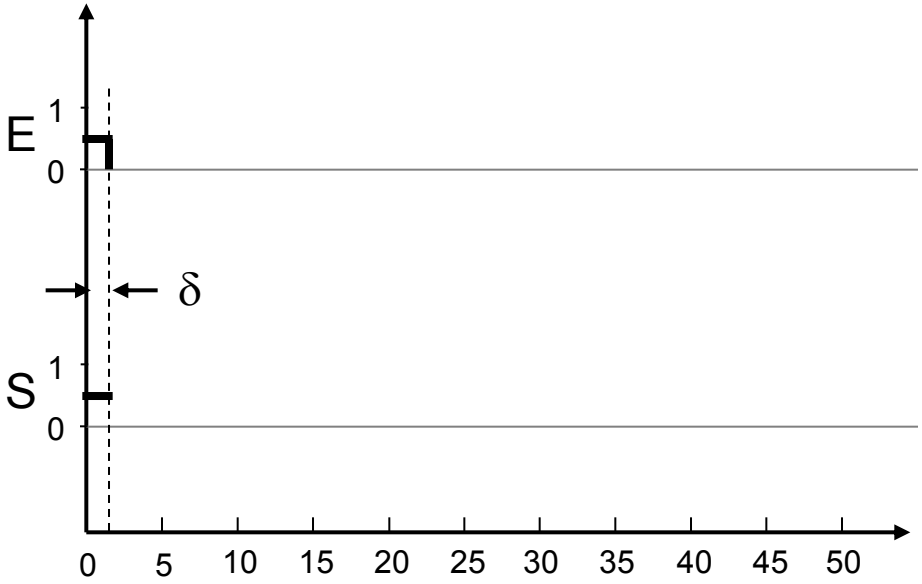
t = 0



VHDL: Asignación de señales

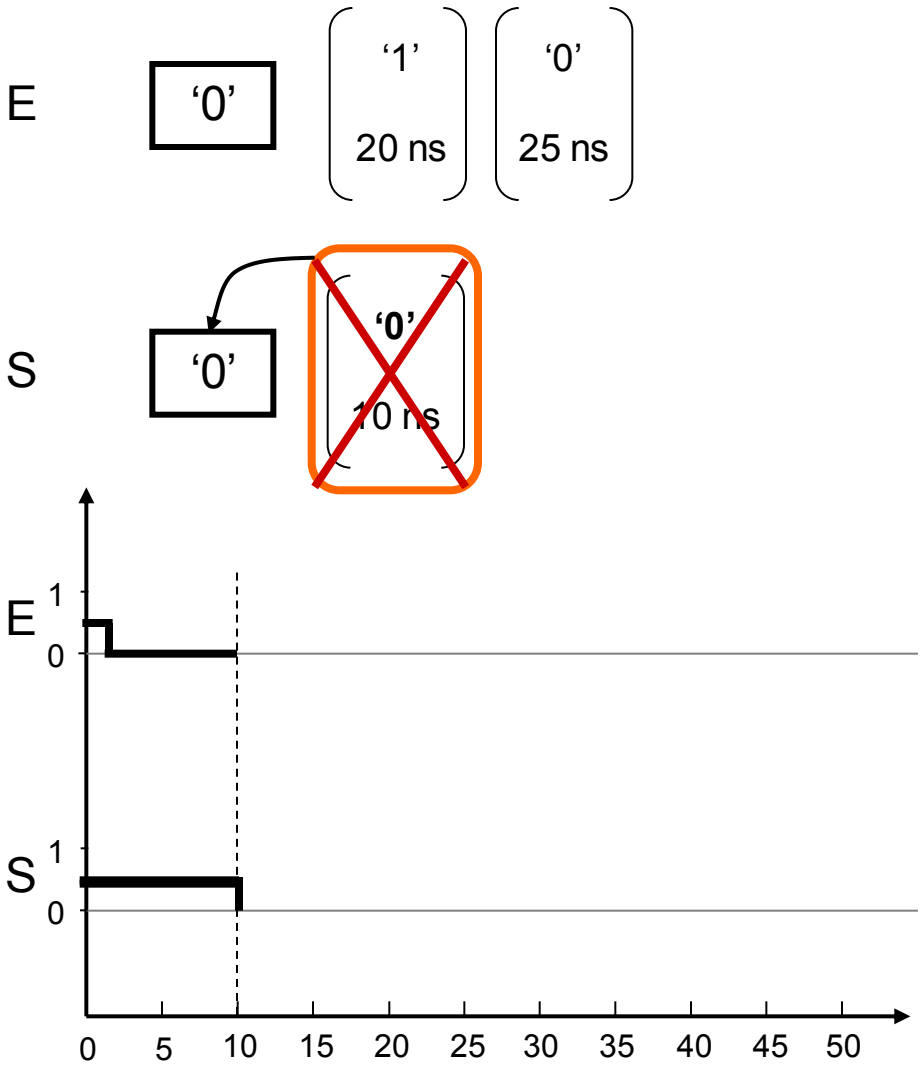
t = δ





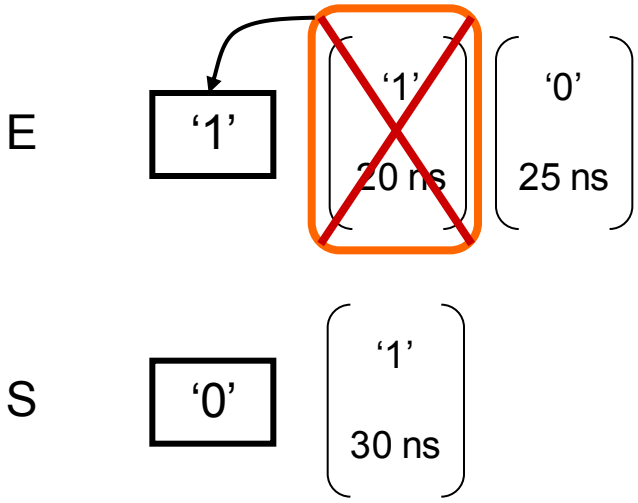
VHDL: Asignación de señales

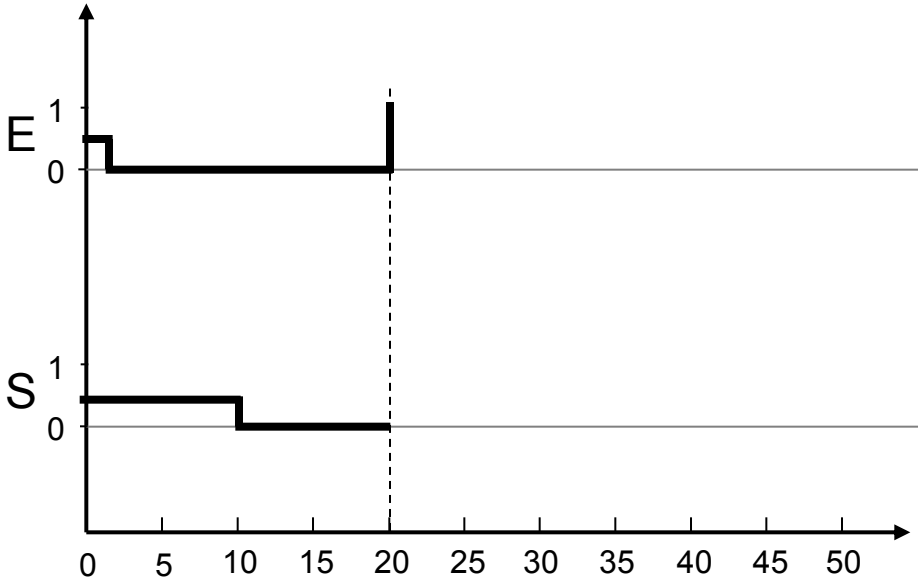
t = 10



VHDL: Asignación de señales

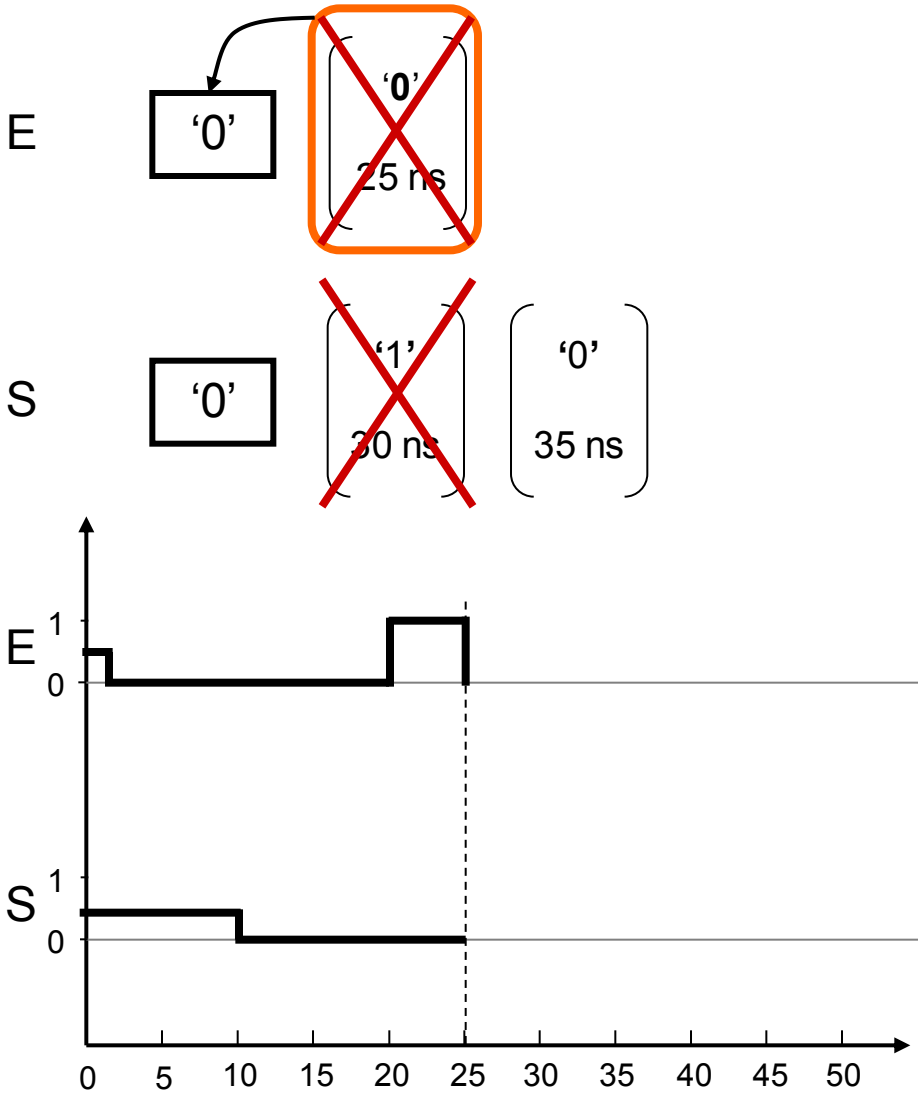
t = 20





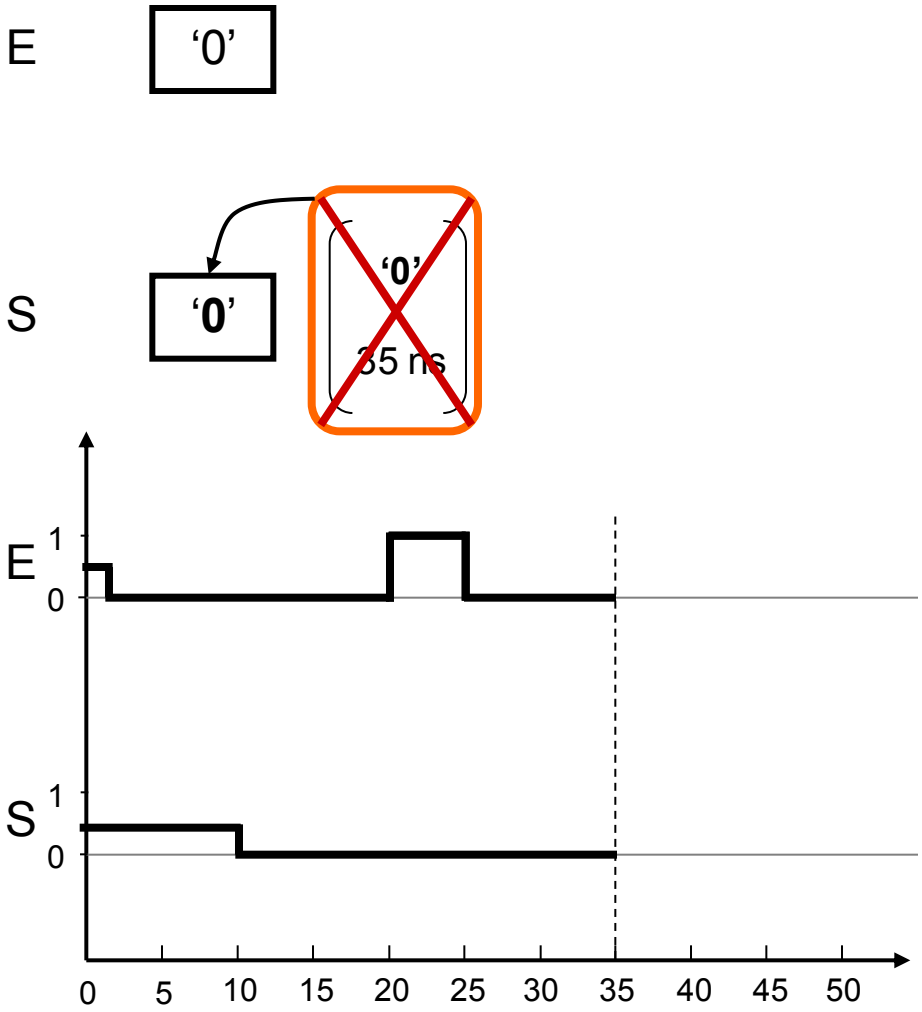
VHDL: Asignación de señales

t = 25



VHDL: Asignación de señales

t = 35

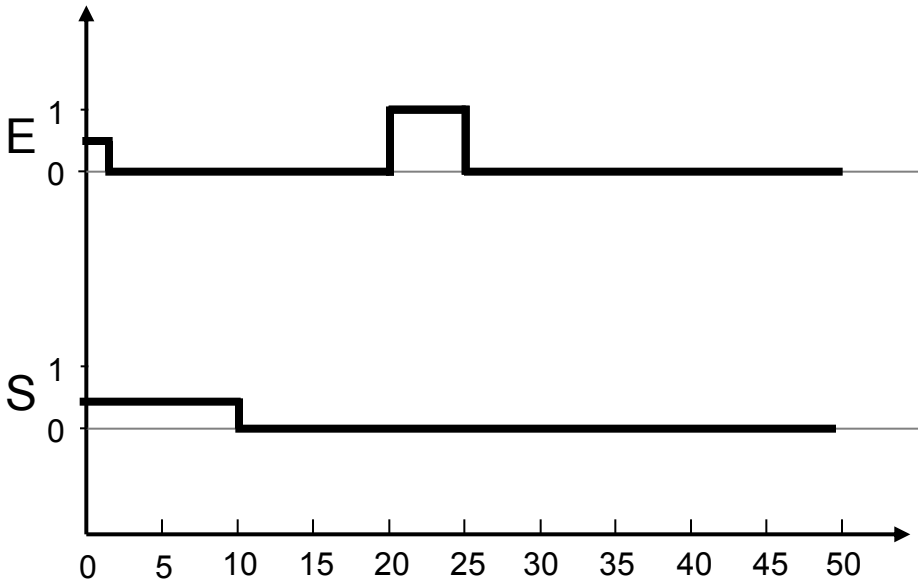


VHDL: Asignación de señales

t > 35

E ‘0’

S ‘0’



VHDL: Asignación de señales

- **Ejemplo 3**

```
entity asigna_03 is
end;

architecture asigna_03_arq of asigna_03 is
    signal a, b, c: std_logic := '0';
begin
    a <= '1' after 2 ns;
    b <= not a after 3 ns;
    c <= not b after 10 ns;
end;
```

VHDL: Asignación de señales

- Ejemplo 4

```
entity asigna_04 is
end;

architecture asigna_04_arq of asigna_04 is
    signal x: std_logic := 'Z';
begin
    process
    begin
        x <= '1' after 5 ns;
        x <= '0' after 8 ns;
        x <= '1' after 6 ns;
        wait;
    end process;
end;
```

VHDL: Asignación de señales

- **Ejemplo 5**

```
entity asigna_05 is
end;

architecture asigna_05_arq of asigna_05 is
    signal x: std_logic := 'Z';
begin
    process
    begin
        x <= '1' after 5 ns, '0' after 10 ns, '1' after 20 ns;
        x <= '0' after 12 ns, '1' after 16 ns, '0' after 25 ns;
        wait;
    end process;
end;
```

VHDL: Asignación de señales

- **Ejemplo 6**

```
entity asigna_06 is
end;

architecture asigna_06_arq of asigna_06 is
    constant N: integer := 4;
    constant TA: bit_vector(0 to N-1) := "1100";
    constant TB: bit_vector(0 to N-1) := "1010";
    signal a, b, c: bit := '0';
begin
    c <= a xor b after 10 ns;
    process
    begin
        for i in 0 to N-1 loop
            a <= TA(i);
            b <= TB(i);
            wait on c for 20 ns;
        end loop;
        wait;
    end process;
end;
```

VHDL: Asignación de señales