

TECNOLOGÍA DE COMPUTADORES

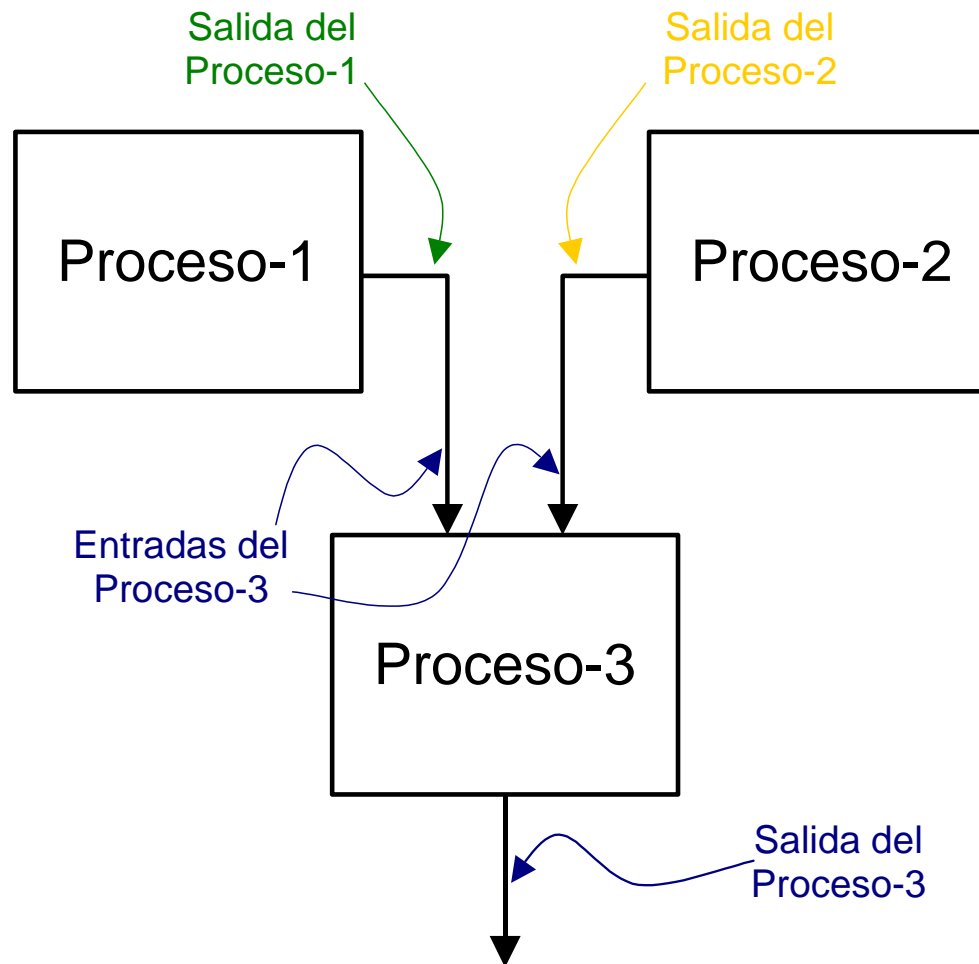
Tema 1

"Panorámica del lenguaje de descripción hardware VHDL" *(2/2)*

Agustín Álvarez Marquina

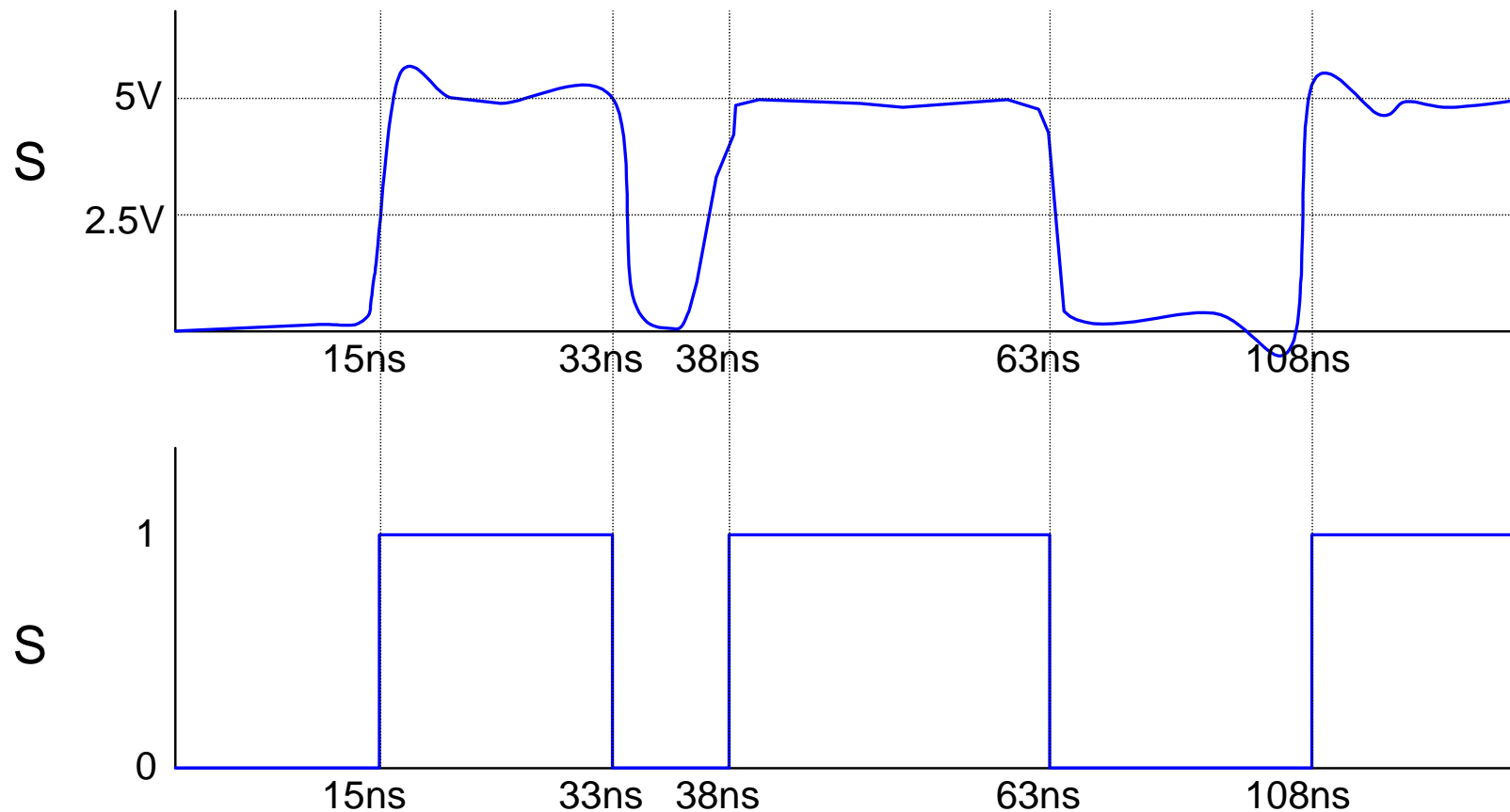
Señales

○ Los procesos se comunican a través de señales.



Caracterización de señales. Modelado (I)

- La respuesta transitoria de las señales se ignora.



Caracterización de señales. Modelado (II)

○ La respuesta transitoria de las señales se ignora.

- ❑ El paquete standard define un tipo **BIT** que puede tomar solamente los valores '0' y '1'.
- ❑ El paquete std_logic_1164 sin embargo define un tipo **STD_ULOGIC**, donde una señal binaria puede tomar hasta 8 posibles valores:

```
TYPE std_ulogic IS ('U', -- Uninitialized
                    'X', -- Forcing  Unknown
                    '0', -- Forcing  0
                    '1', -- Forcing  1
                    'Z', -- High Impedance
                    'W', -- Weak      Unknown
                    'L', -- Weak      0
                    'H', -- Weak      1
                    '- ' -- Don't care
                    );
```

Caracterización de señales.

Transacciones, eventos y propagación (I)

○ **Transacciones (valor, tiempo). A la señal se le asigna un valor para un tiempo determinado.**

□ Ejemplo anterior:

(0, 0ns)

(0, 7ns)

(1, 15ns)

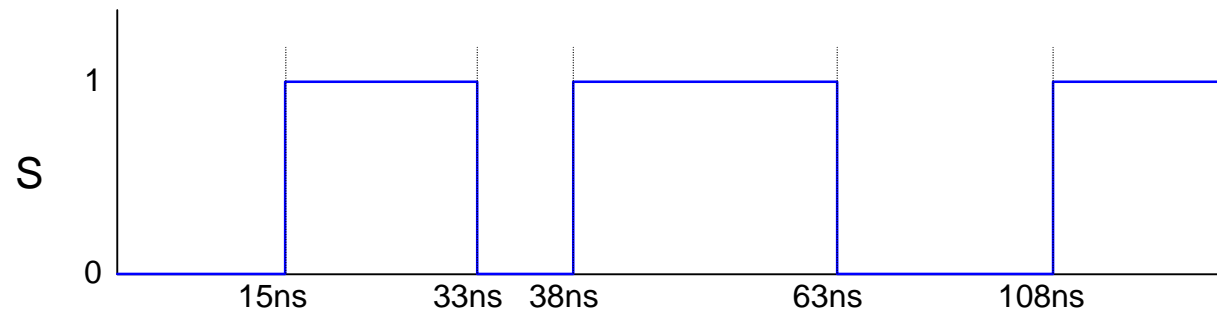
(1, 30ns)

(0, 33ns)

(1, 38ns)

(0, 63ns)

(1, 108ns)



Caracterización de señales. Transacciones, eventos y propagación (II)

○ **Eventos.** Un evento se produce cuando en una asignación se produce un cambio en el valor de la señal.

□ Hay transacciones que no produce eventos.

□ Ejemplo anterior:

(0, 0ns)

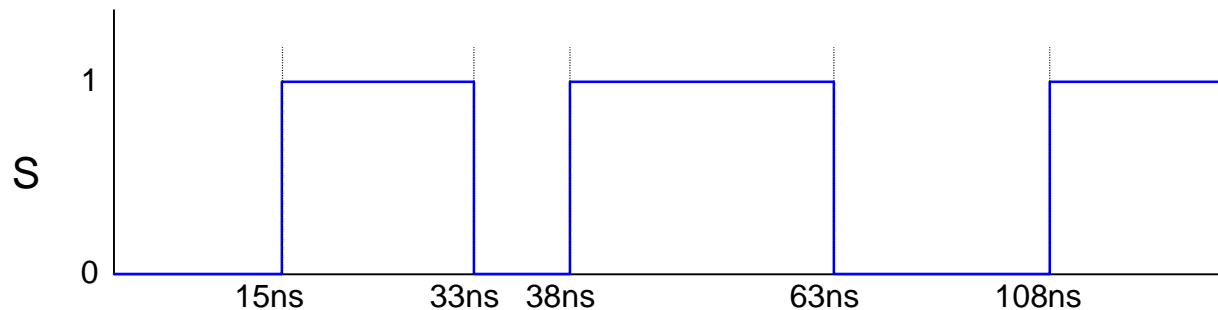
(1, 15ns)

(0, 33ns)

(1, 38ns)

(0, 63ns)

(1, 108ns)



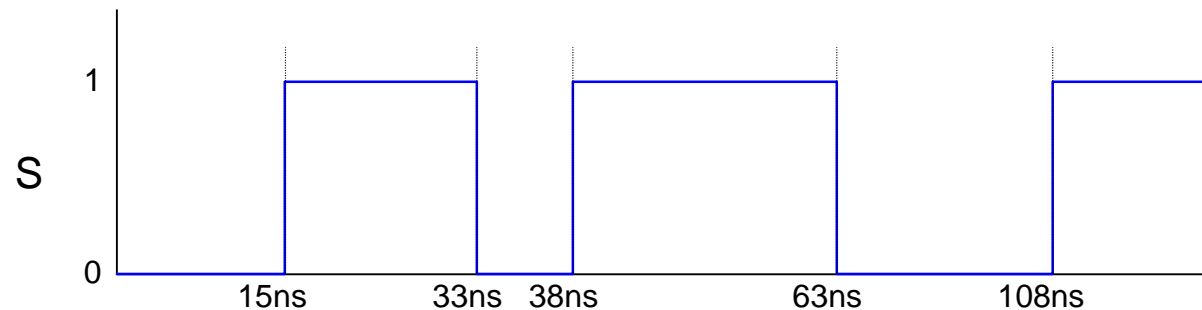
Caracterización de señales.

Transacciones, eventos y propagación (III)

○ Propagación de señales.

□ Ejemplo anterior:

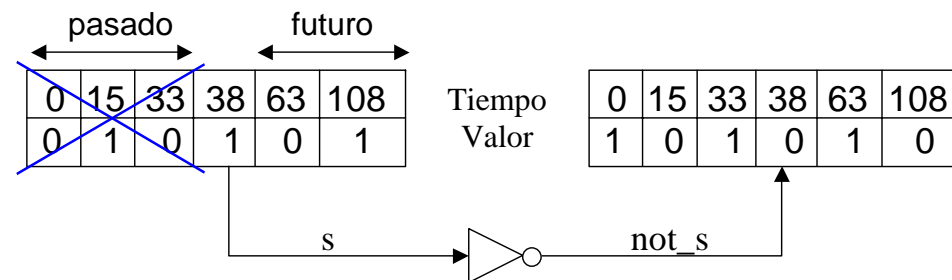
s <= '0', '1' AFTER 15 NS, '0' AFTER 33 NS,
'1' AFTER 38 NS, '0' AFTER 63 NS, '1' AFTER 108 NS;



Caracterización de señales. Drivers

○ Asociado a cada señal hay un driver.

- ❑ Es una cola (vector) empleada para almacenar la forma de onda, que proyecta los valores futuros de la señal.
- Asignaciones distintas de la señal comparten un mismo driver y ocurren en el mismo proceso.
- Cuando avanza el tiempo de simulación las transacciones se consumen en la cola.
- Un driver está activo cada vez que se elimina una transacción de la cola.



Modelo temporal en VHDL

○ La simulación está conducida por los eventos.

- Los eventos de las señales disparan la simulación.

○ La simulación del modelo solamente tiene en cuenta el instante en el que continúa la ejecución de un proceso después del periodo de inactividad de una sentencia *WAIT*.

- El driver asociado a `not_s` cambiará cada vez que haya un cambio en `s`.

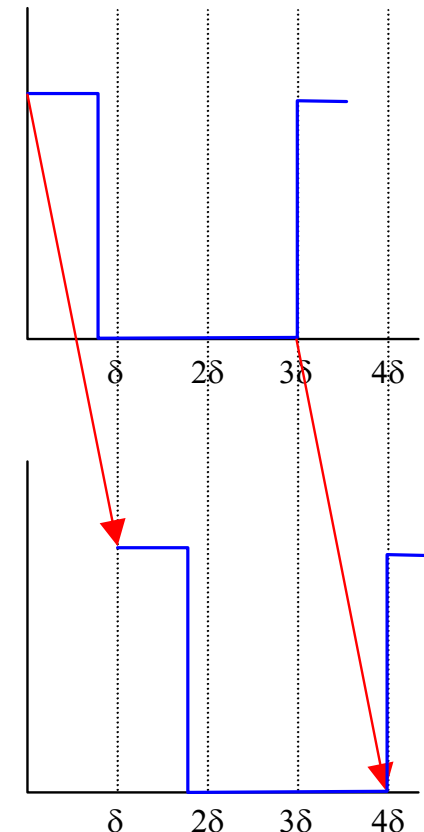
```
PROCESS
BEGIN
    not_s<= NOT s;
    WAIT ON s;
END PROCESS;
```

El retardo delta (δ)

- El retardo delta se asume por defecto cuando no se especifica un retardo concreto en la asignación de una señal (causalidad).

```
ARCHITECTURE --- OF --- IS  
---  
BEGIN  
    salida<= entrada;  
---  
END ---;
```

El valor de la señal **salida** tomará el valor negado al de la señal entrada después de un retardo δ (infinitesimal y cuyo valor por defecto son femtosegundos).

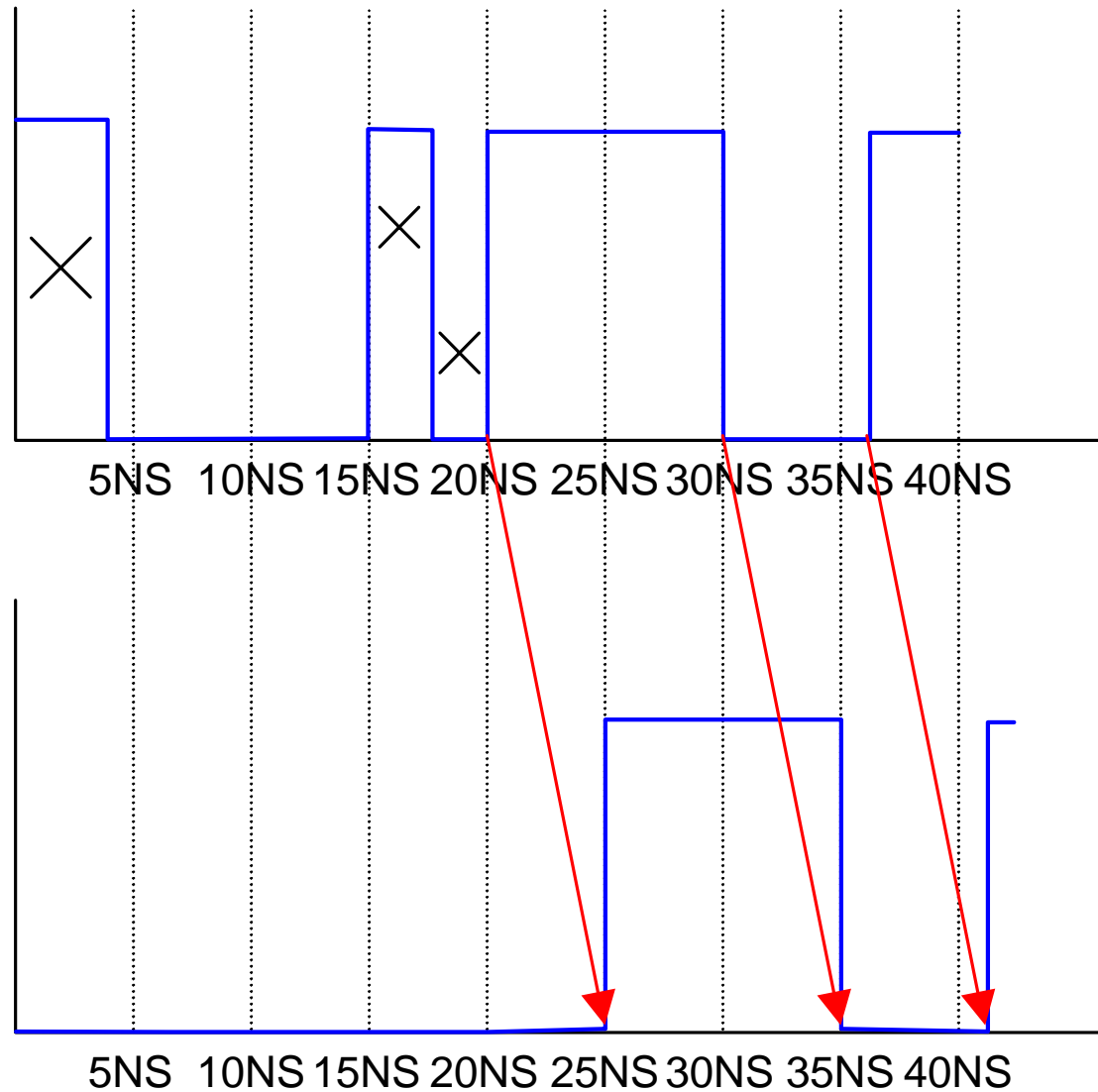


Retardo inercial (I)

- Se especifica con la palabra reservada **AFTER**, y es el adecuado para modelar la conmutación de los circuitos digitales.
 - Un pulso de duración menor del retardo (tiempo de conmutación del circuito) no se transmite.

```
ARCHITECTURE --- OF --- IS
---
BEGIN
    salida<= entrada AFTER 5 NS;
---
END ---;
```

Retardo inercial (II)

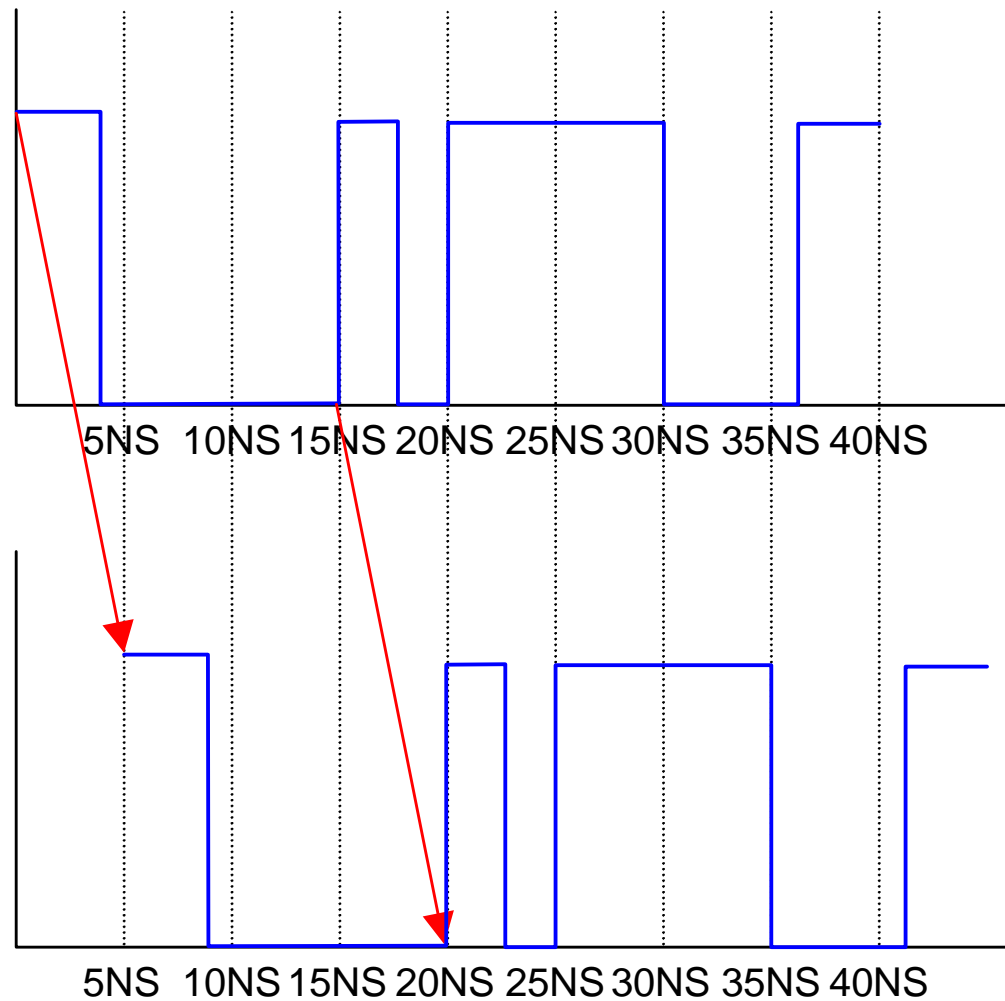


Retardo transporte (I)

- Se especifica con la palabra reservada ***TRANSPORT***, y es el adecuado para modelar líneas de transmisión.
 - Se transmite la señal íntegra independientemente de cuál sean la duración de los pulsos.

```
ARCHITECTURE --- OF --- IS
---
BEGIN
    salida<= TRANSPORT entrada AFTER 5 NS;
---
END ---;
```

Retardo transporte (II)



Fases de ejecución de un programa VHDL

○ Programa fuente VHDL

Analizador VHDL

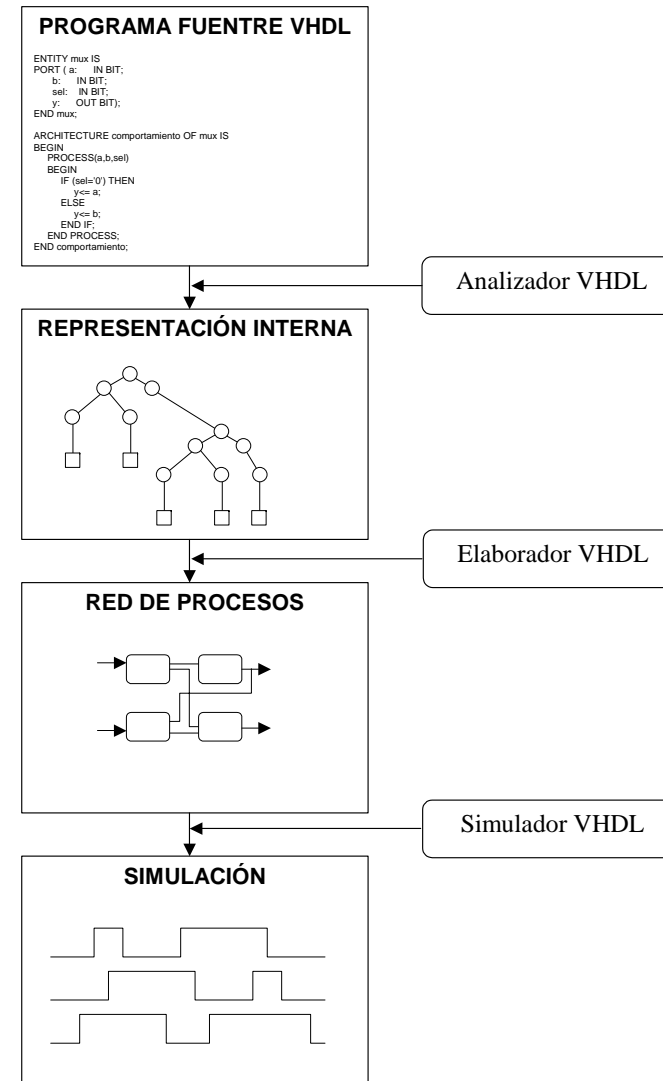
○ Representación interna

Elaborador VHDL

○ Red de procesos

Simulador VHDL

○ Simulación



Realización del test (I)

○ El test de un diseño se puede realizar escribiendo una entidad y una arquitectura.

○ La entidad no tiene puertos.

□ Declaración:

```
ENTITY nombreEntidadTest IS  
END nombreEntidadTest;
```

□ Ejemplo:

```
ENTITY testInversor IS  
END testInversor;
```


Realización del test (II)

- La arquitectura se escribe en estilo estructural, instanciando el componente que se va a testar.

```
ARCHITECTURE nombreArquitecturaTest
OF nombreEntidad IS
    <Declaración de señales>
    <Declaración de componentes>
    <Configuración de la arquitectura>
BEGIN
    <Instanciación del componente a testear>
    <Driver de las señales de test>
END nombreArquitecturaTest;
```

Realización del test (III)

○ Ejemplo inicial.

```
ARCHITECTURE una OF testInversor IS
    SIGNAL s, not_s: BIT;
    COMPONENT inversor
        PORT(s: IN BIT; not_s: OUT BIT);
    END COMPONENT;
    FOR com: inversor USE
        ENTITY WORK.inversor(argInversor);
    BEGIN
        com: inversor PORT MAP(s, not_s);
        s<= '0', '1' AFTER 15 NS,
            '0' AFTER 33 NS, '1' AFTER 38 NS,
            '0' AFTER 63 NS, '1' AFTER 108 NS;
    END una;
```

Modelo de simulación (I)

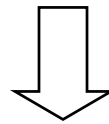
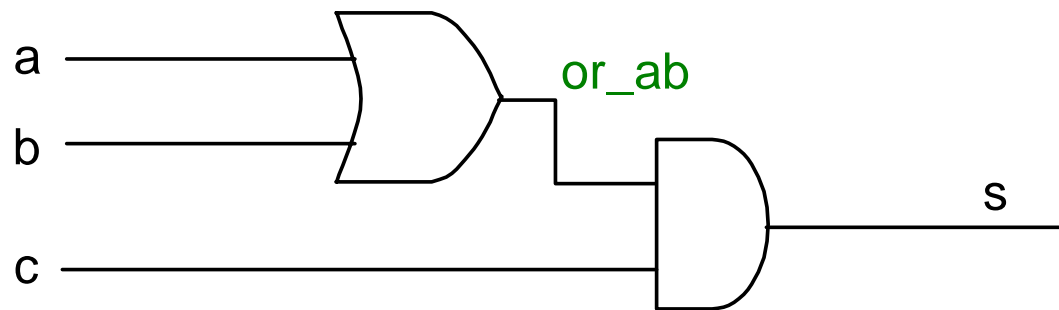
○ El modelo de simulación ha de tener en cuenta que:

- ❑ El funcionamiento del hardware es paralelo.
- ❑ La salida de un sistema puede cambiar solamente cuando cambien sus entrada (eventos).

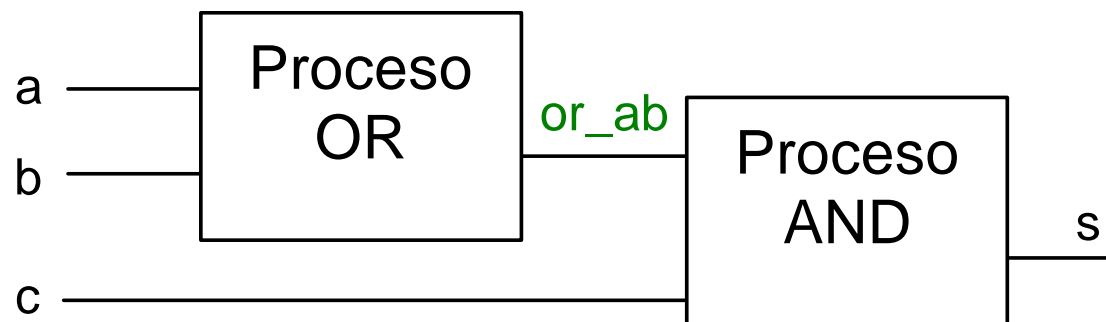
○ VHDL entiende cada sentencia concurrente como un proceso y los procesos se comunican entre si mediante señales.

- ❑ Sentencia *PROCESS*.
- ❑ Asignación concurrente de señal \leftarrow .
- ❑ Instancia de componente (modelo estructural).

Modelo de simulación (II)



•Señal interna.



Modelo de simulación (III)

○ Consideraciones previas:

- ❑ Cuando se pone en funcionamiento un sistema todos sus componentes pueden activarse simultáneamente.
- ❑ Si tuviésemos una máquina paralela cada *proceso* se podría ejecutar en un procesador distinto.
 - Normalmente solo disponemos de un único procesador y por tanto la ejecución de los procesos tiene que ser secuencial.
- ❑ ¿Cómo se ejecuta un modelo paralelo en una máquina secuencial, teniendo en cuenta que el resultado de la simulación tiene que ser independiente del orden en el que se escriban/ejecuten los procesos?

Modelo de simulación (IV)

○ Definiciones de tiempos.

- ❑ El modelo de simulación está controlado por **eventos**.
 - Solamente se van a evaluar los procesos cuando haya cambios en las señales que los activan.
- ❑ Las señales están caracterizadas por sus correspondientes drivers.
 - Los drivers se actualizan al encontrarse con una sentencia *WAIT* en un *PROCESS* o, después de ejecutar una sentencia concurrente de otro tipo.

Modelo de simulación (IV)

○ Tipos de tiempos.

□ Tiempo total de simulación (T)

- Tiempo durante el cual vamos a observar el funcionamiento del circuito (ej. 100 ns).

□ Tiempo de simulación (ΔT)

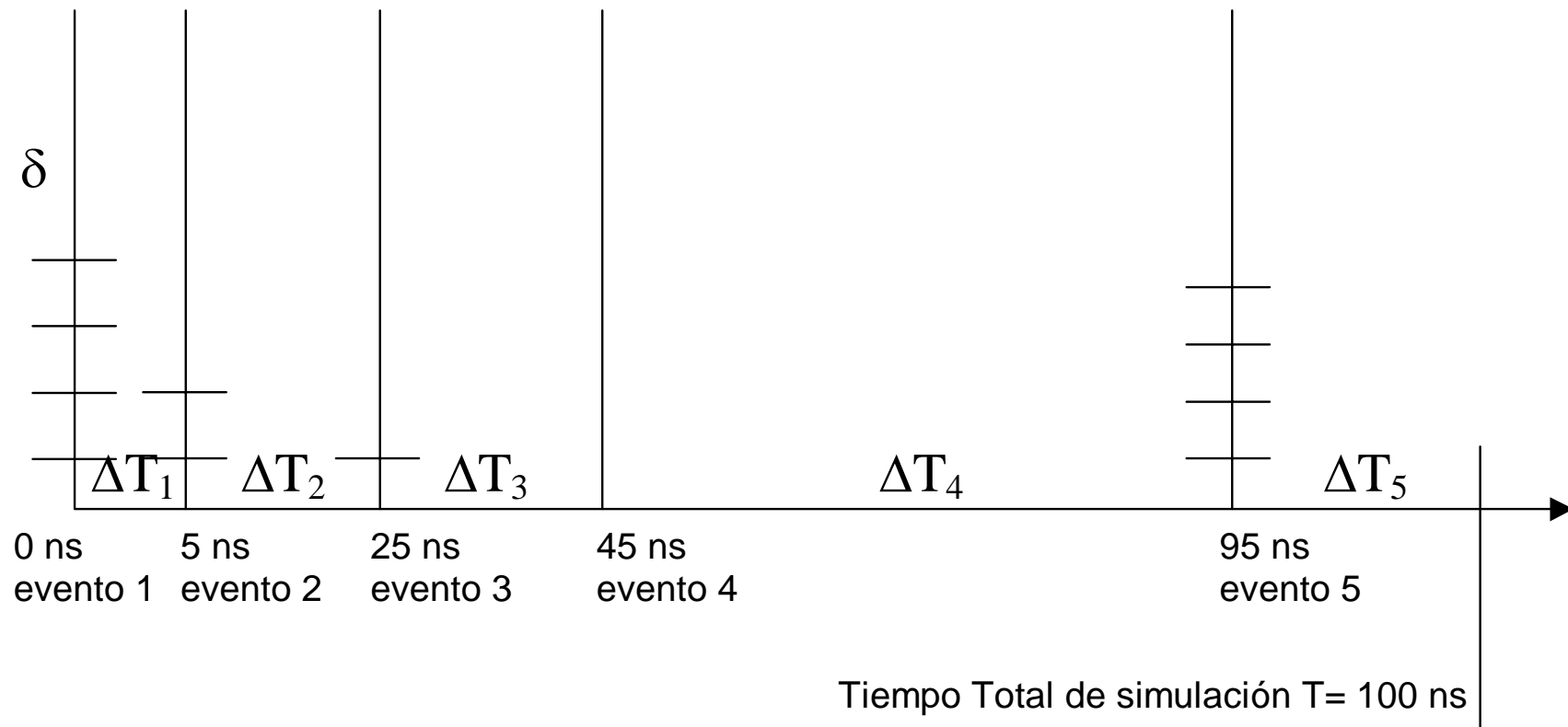
- Tiempo específico para el cual estamos realizando la simulación por haberse producido un evento.
 - Supone un avance de tiempo físico en el tiempo total de simulación.
- Este tiempo avanza en una cantidad de ΔT , **no fija**, según se van consumiendo los eventos en la cola (driver).

□ Ciclo o paso de simulación (δ)

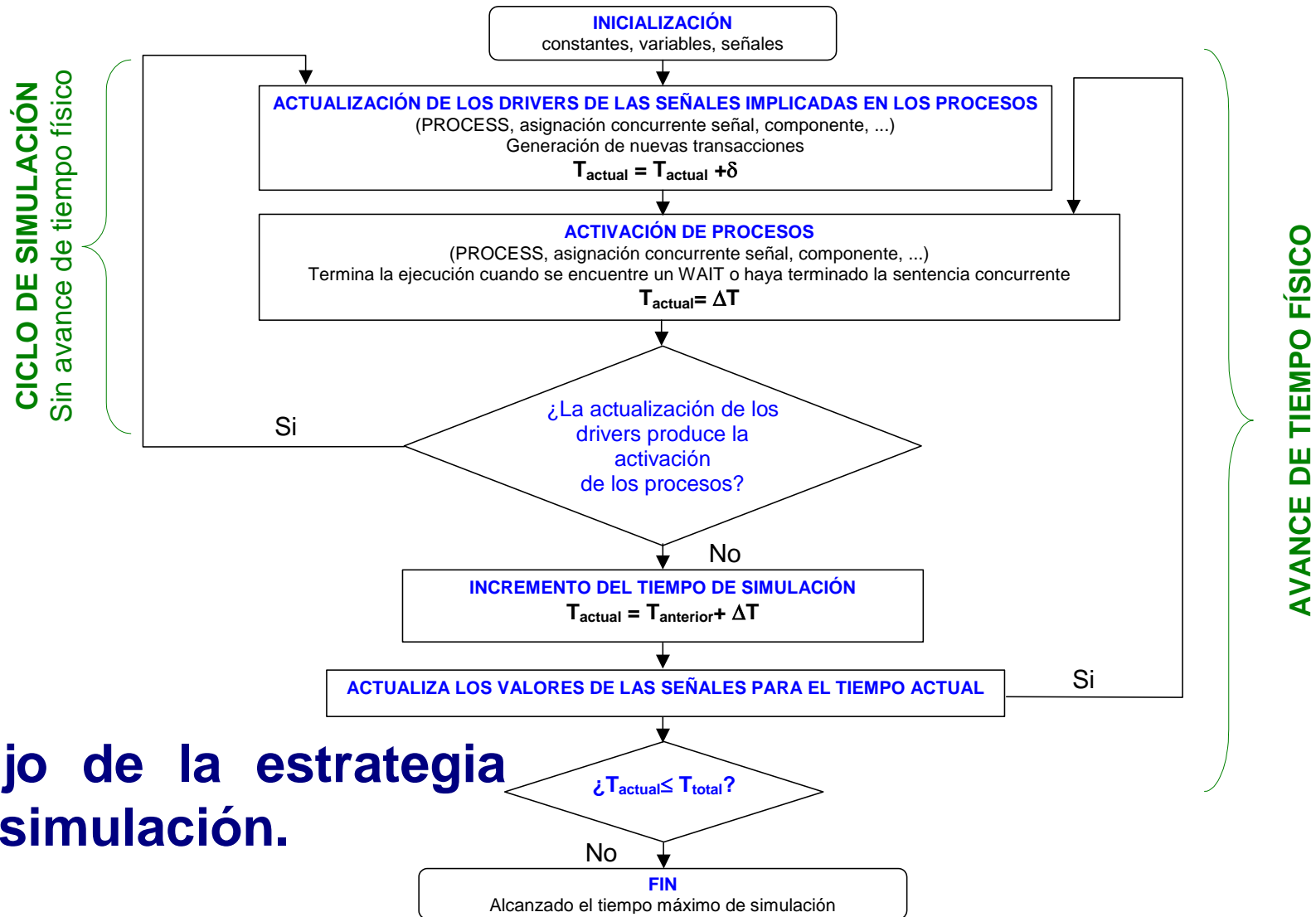
- No supone un avance de tiempo físico sino un retardo de tipo δ .

Modelo de simulación (V)

○ Tipos de tiempos.



Modelo de simulación (VI)



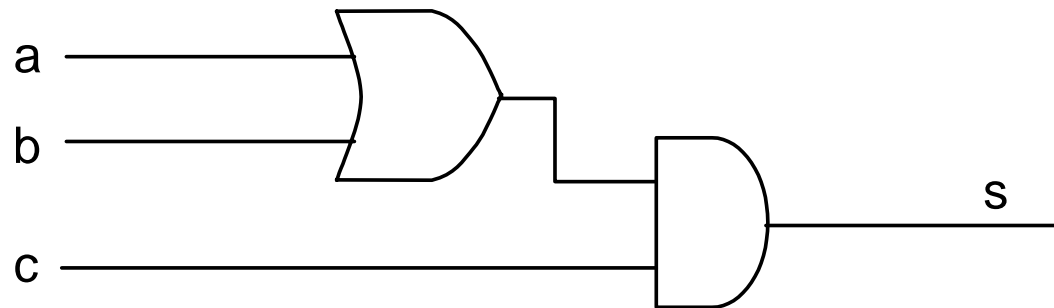
○ Flujo de la estrategia de simulación.

Modelo de simulación (VII)

○ Visión simplificada de la simulación.

- ❑ La simulación puede verse como un bloque, que consta de dos etapas:
 - Actualización de señales.
 - Ejecución de procesos.
- ❑ Termina al agotarse el tiempo total de simulación.

Modelo de simulación. Ejemplo A (I)



```
-- Definición de la entidad del diseño  
ENTITY circuitoCombinacional IS  
    PORT (a,b,c: IN BIT; s: OUT BIT);  
END circuitoCombinacional;
```

Modelo de simulación. Ejemplo A (II)

```
-- Arquitectura descrita en estilo de comportamiento
ARCHITECTURE comportamiento OF circuitoCombinacional IS
    SIGNAL or_ab: BIT;
BEGIN
    OR2: PROCESS
    BEGIN
        or_ab<= a OR b;
        WAIT ON a,b;
    END PROCESS;

    AND2: PROCESS
    BEGIN
        s<= or_ab AND c;
        WAIT ON or_ab,c;
    END PROCESS;
END comportamiento;
```

Modelo de simulación. Ejemplo A (III)

```
-- Arquitectura descrita en estilo de flujo
ARCHITECTURE flujo OF circuitoCombinacional
IS
    SIGNAL or_ab: BIT;
BEGIN
    or_ab<= a OR b;
    s<= or_ab AND c;
END flujo;
```

Modelo de simulación. Ejemplo A (IV)

```
-- Arquitectura descrita en estilo estructural
ARCHITECTURE estructural OF circuitoCombinacional IS

    COMPONENT or2
        PORT(a,b: IN BIT; s: OUT BIT);
    END COMPONENT;

    COMPONENT and2
        PORT(a,b: IN BIT; s: OUT BIT);
    END COMPONENT;

    SIGNAL or_ab: BIT;

    FOR all: or2 USE ENTITY WORK.or2(or2_arq);
    FOR all: and2 USE ENTITY WORK.and2(and2_arq);

BEGIN
    U0: or2 PORT MAP(a, b, or_ab);
    U1: and2 PORT MAP(or_ab, c, s);
END estructural;
```

Modelo de simulación. Ejemplo A (V)

```
-- Definición de la entidad de test
ENTITY test IS
END test;

-- Arquitectura que prueba implementación estilo comportamiento
ARCHITECTURE test_arq OF test IS

    COMPONENT circuitoCombinacional
        PORT (a,b,c: IN BIT; s: OUT BIT);
    END COMPONENT;

    SIGNAL a,b,c,s: BIT;

    FOR all: circuitoCombinacional USE

        ENTITY WORK.circuitoCombinacional(comportamiento);

BEGIN
```

•Sigue en la siguiente transparencia.

Modelo de simulación. Ejemplo A (VI)

```
U0: circuitoCombinacional PORT MAP(a,b,c,s);
```

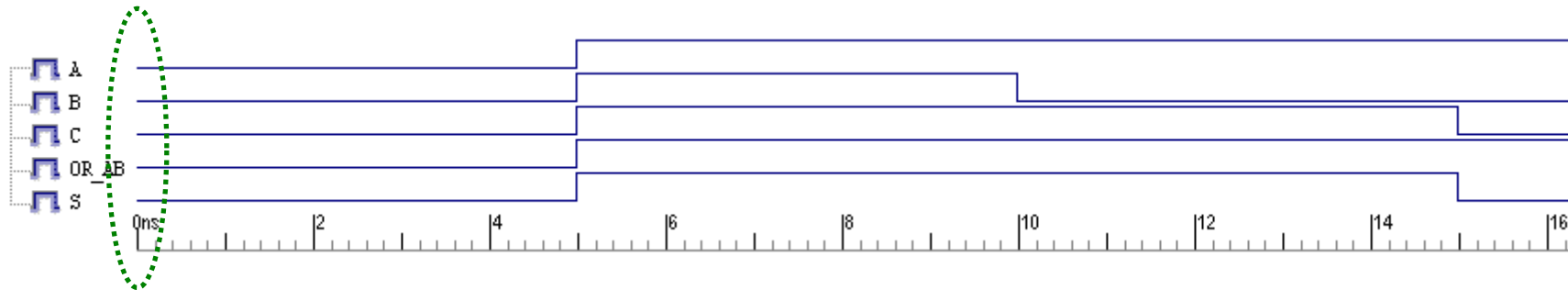
```
a<=    '0',  
        '1' AFTER 5 NS,  
        '1' AFTER 10 NS;
```

```
b<=    '0',  
        '1' AFTER 5 NS,  
        '0' AFTER 10 NS;
```

```
c<=    '0',  
        '1' AFTER 5 NS,  
        '1' AFTER 10 NS,  
        '0' AFTER 15 NS;
```

```
END test_arq;
```


Modelo de simulación. Ejemplo A (VII)



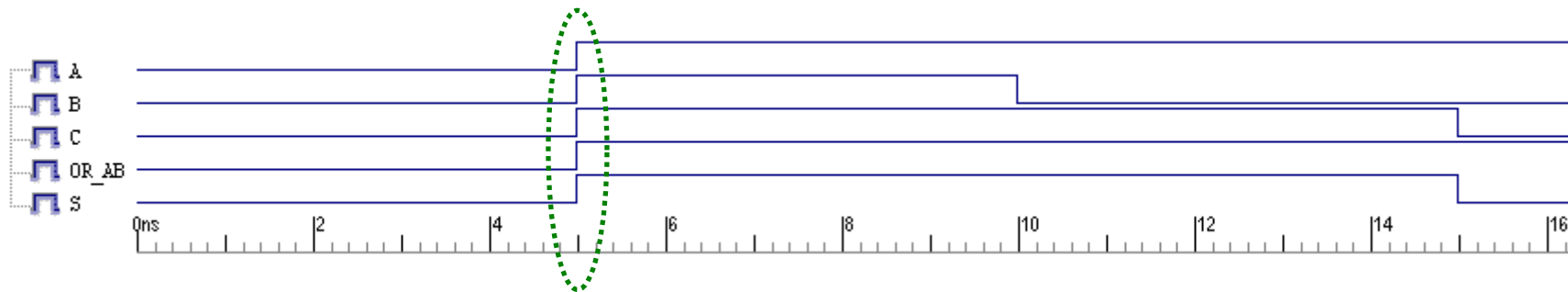
○ Instante 0ns.

□ Se produce un evento sobre **a**, **b**, **c** (ciclo $\delta=0$).

- Señales **a**, **b**, **c** reciben valores iniciales.
- Valor por defecto de todas las señales es '0'.
- Por tanto, no hay cambios en ninguna señal.

✓ **Avance en el tiempo físico hasta siguiente evento.**

Modelo de simulación. Ejemplo A (VIII)



○ Instante 5ns.

- Se produce un evento sobre **a**, **b**, **c** (ciclo $\delta = 0$).
 - Al final del ciclo las 3 señales cambian a su nuevo valor, en concreto pasan de '0' a '1'.
 - Señal **or_ab** recibe la suma booleana de los valores de origen de **a** ('0') y **b** ('0').
 - Por tanto, no cambia de valor en este paso.
 - Señal **s** recibe el producto booleano de los valores originales de **or_ab** ('0') y **c** ('0').
 - No cambia de valor en este paso.

Modelo de simulación. Ejemplo A (IX)

○ Instante 5ns.

- Cálculo de valores futuros de (ciclo $\delta = 1$):
 - Señal **or_ab** recibe la suma booleana de los actuales valores de **a** ('1') y **b** ('1').
 - Cambia de valor en este paso.
 - Al final del ciclo se actualiza esta señal al nuevo valor ('1').
 - Señal **s** recibe el producto booleano de los actuales valores de **or_ab** ('0') y **c** ('1').
 - No cambia de valor en este paso.

Modelo de simulación. Ejemplo A (X)

○ Instante 5ns.

- Cálculo de valores futuros de (ciclo $\delta = 2$):
 - Señal **or_ab** recibe la suma booleana de los actuales valores de **a** ('1') y **b** ('1').
 - No cambia de valor en este paso (valor al inicio del ciclo ya es '1').
 - Señal **s** recibe el producto booleano de los actuales valores de **or_ab** ('1') y **c** ('1').
 - Cambia de valor en este paso.
 - Al final del ciclo se actualiza esta señal al nuevo valor ('1').

Modelo de simulación. Ejemplo A (XI)

○ Instante 5ns.

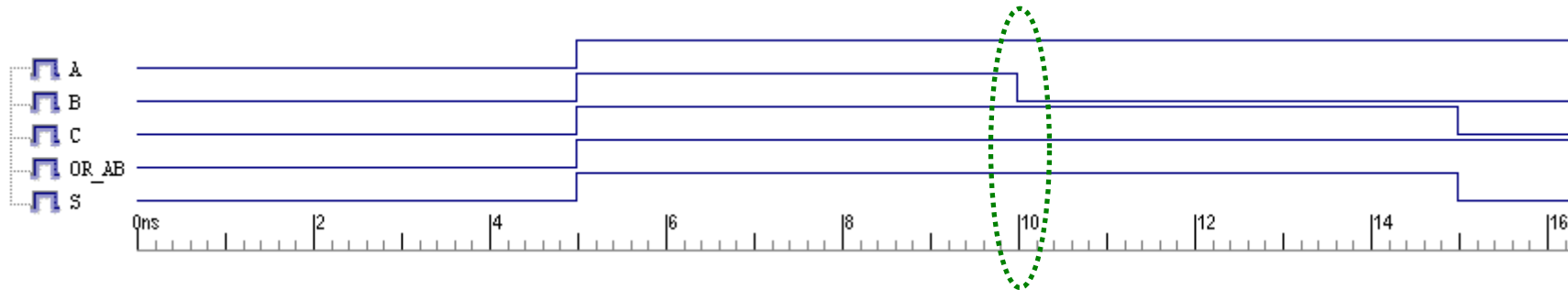
□ Cálculo de valores futuros de (ciclo $\delta=3$):

- Señal **or_ab** recibe la suma booleana de los actuales valores de **a** ('1') y **b** ('1').
 - No cambia de valor en este paso (valor al inicio del ciclo ya es '1').
- Señal **s** recibe el producto booleano de los actuales valores de **or_ab** ('1') y **c** ('1').
 - No cambia de valor en este paso (valor al inicio del ciclo ya es '1').

✓ En este ciclo ninguna señal ha variado → Este ciclo es innecesario.

✓ Al final del instante $\delta=2$ se produce un avance en el tiempo físico hasta la llegada del siguiente evento.

Modelo de simulación. Ejemplo A (XII)



○ Instante 10ns.

- ❑ Se produce un evento sobre **b** (ciclo $\delta = 0$).
 - Al final del ciclo la señal **b** cambia a su nuevo valor; en concreto pasa de '1' a '0'.
 - Señal **or_ab** recibe la suma booleana de los actuales valores de **a** ('1') y **b** ('1').
 - No cambia de valor en este paso.
 - Señal **s** recibe el producto booleano de los actuales valores de **or_ab** ('1') y **c** ('1').
 - No cambia de valor en este paso.

Modelo de simulación. Ejemplo A (XIII)

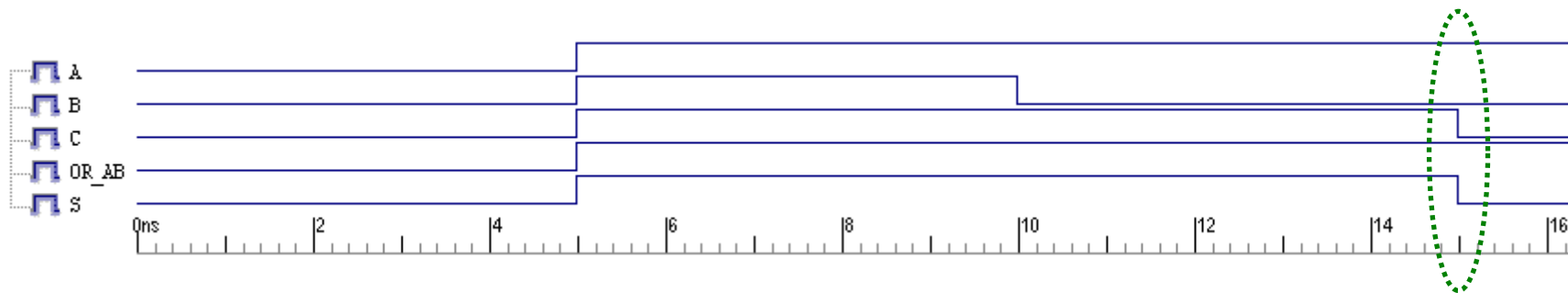
○ Instante 10ns.

□ Cálculo de valores futuros de (ciclo $\delta=1$).

- Señal **or_ab** recibe la suma booleana de los actuales valores de **a** ('1') y **b** ('0').
 - No cambia de valor en este paso.
- Señal **s** recibe el producto booleano de los actuales valores de **or_ab** ('1') y **c** ('1').
 - No cambia de valor en este paso.

- ✓ En este ciclo ninguna señal ha variado → Este ciclo es innecesario.
- ✓ Al final del instante $\delta=0$ se produce un avance en el tiempo físico hasta la llegada del siguiente evento.

Modelo de simulación. Ejemplo A (XIV)



○ Instante 15ns.

- Se produce un evento sobre **c** (ciclo $\delta = 0$).
 - Al final del ciclo la señal **c** cambia a su nuevo valor; en concreto pasa de '1' a '0'.
 - Señal **or_ab** recibe la suma booleana de los actuales valores de **a** ('1') y **b** ('0').
 - No se produce cambio puesto que estas entradas no han variado.
 - Señal **s** recibe el producto booleano de los actuales valores de **or_ab** ('1') y **c** ('1').
 - No cambia de valor en este paso.

Modelo de simulación. Ejemplo A (XV)

○ Instante 15ns.

□ Cálculo de valores futuros de (ciclo $\delta=1$):

- Señal **or_ab** recibe la suma booleana de los actuales valores de **a** ('1') y **b** ('0').
 - No se produce cambio puesto que las entradas no han variado.
- Señal **s** recibe el producto booleano de los actuales valores de **or_ab** ('1') y **c** ('0').
 - Al final del ciclo se produce un cambio en esta señal, que pasa de valer '1' a valer '0'.

Modelo de simulación. Ejemplo A (XVI)

○ Instante 15ns.

□ Cálculo de valores futuros de (ciclo $\delta = 2$):

- Señal **or_ab** recibe la suma booleana de los actuales valores de **a** ('1') y **b** ('0').
 - No se produce cambio puesto que las entradas no han variado.
- Señal **s** recibe el producto booleano de los actuales valores de **or_ab** ('1') y **c** ('0').
 - No se produce cambio puesto que las entradas no han variado.

✓ En este ciclo ninguna señal ha variado → Este ciclo es innecesario.

✓ Al final del instante $\delta = 1$ se produce un avance en el tiempo físico hasta la llegada del siguiente evento.

Modelo de simulación. Ejemplo A (XVII)

- ✓ A partir de este punto, no existen nuevos eventos en las señales de entrada *a*, *b*, *c*.
- ✓ Se alcanza el final de la simulación.

Modelo de simulación. Ejemplo A (XVIII)

Tiempo (ns)	Delta (δ)	a	b	c	or_ab	s
0	0	0	0	0	0	0
5	0	1	1	1	0	0
	1				1	
	2					1
10	0	1	0	1	1	1
15	0	1	0	0	1	1
	1					0

Modelo de simulación. Ejemplo B (I)

- **Vamos ahora a definir una nueva arquitectura de estilo comportamiento** (`comportamientoRetardo`) **asociada a la misma entidad** (`circuitoCombinacional`).
- Esta arquitectura establece dos retardos:
 - Elemento OR: 4ns
 - Elemento AND: 2ns
- **Realizaremos entonces una nueva simulación manteniendo los valores de entrada para las señales *a*, *b*, *c*.**

Modelo de simulación. Ejemplo B (II)

```
-- Arquitectura descrita en estilo de comportamiento  
ARCHITECTURE comportamientoRetardo OF circuitoCombinacional  
IS
```

```
    SIGNAL or_ab: BIT;  
BEGIN  
    OR2: PROCESS  
    BEGIN  
        or_ab<= a OR b AFTER 4 NS;  
        WAIT ON a,b;  
    END PROCESS;  
  
    AND2: PROCESS  
    BEGIN  
        s<= or_ab AND c AFTER 2 NS;  
        WAIT ON or_ab,c;  
    END PROCESS;  
END comportamientoRetardo;
```

Añadido



Modelo de simulación. Ejemplo B (III)

-- Definición de la entidad de test

ENTITY test **IS**

END test;

-- Arquitectura que prueba implementación estilo comportamiento

ARCHITECTURE test_arq **OF** test **IS**

COMPONENT circuitoCombinacional

PORT (a,b,c: **IN** **BIT**; s: **OUT** **BIT**);

END COMPONENT;

SIGNAL a,b,c,s: **BIT**;

FOR all: circuitoCombinacional **USE**

ENTITY WORK.circuitoCombinacional(**comportamientoRetardo**);

Único cambio



BEGIN

•Sigue en la siguiente transparencia.

Modelo de simulación. Ejemplo B (IV)

```
U0: circuitoCombinacional PORT MAP(a,b,c,s);
```

```
a<=    '0',  
        '1' AFTER 5 NS,  
        '1' AFTER 10 NS;
```

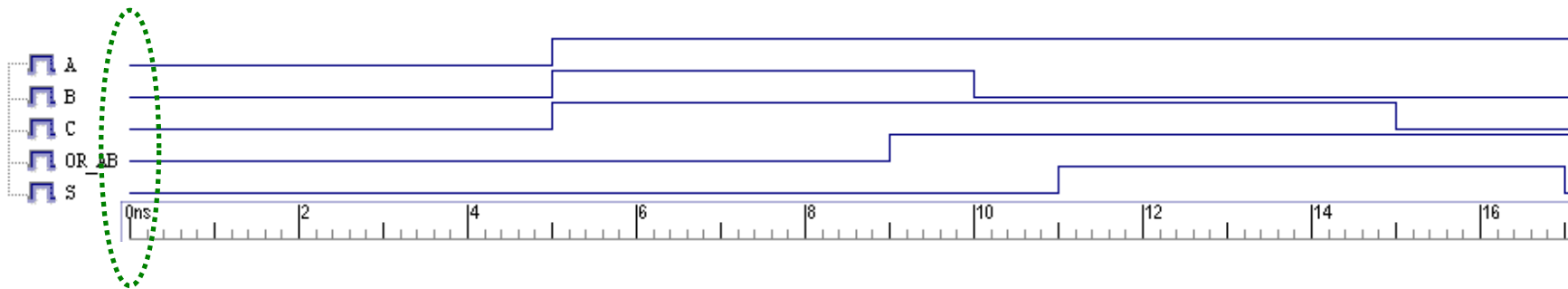
```
b<=    '0',  
        '1' AFTER 5 NS,  
        '0' AFTER 10 NS;
```

Sin modificaciones

```
c<=    '0',  
        '1' AFTER 5 NS,  
        '1' AFTER 10 NS,  
        '0' AFTER 15 NS;
```

```
END test_arq;
```


Modelo de simulación. Ejemplo B (V)



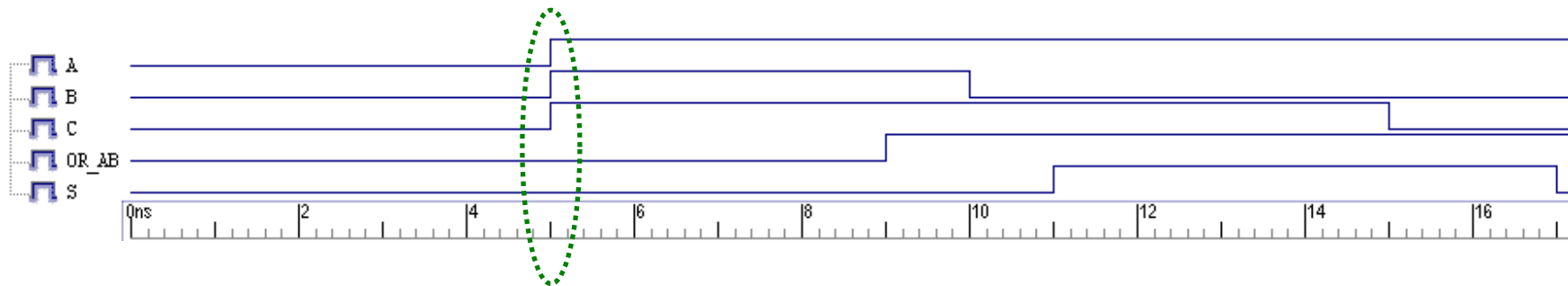
○ Instante 0ns.

□ Se produce un evento sobre **a**, **b**, **c** (ciclo $\delta = 0$).

- Señales **a**, **b**, **c** reciben valores iniciales.
- Valores por defecto de todas las señales es '0'.
- Por tanto, no hay cambios en ninguna señal.

✓ **Avance en el tiempo físico hasta siguiente evento.**

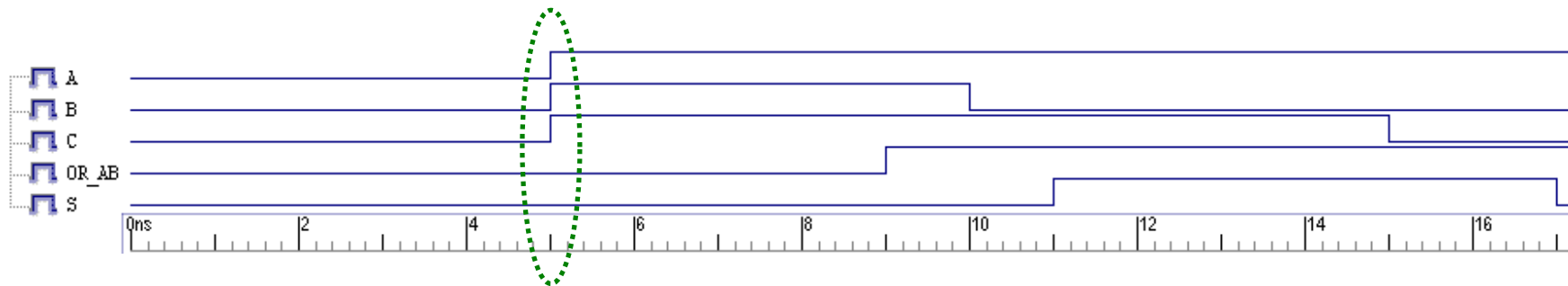
Modelo de simulación. Ejemplo B (VI)



○ Instante 5ns.

- Se produce un evento sobre **a**, **b**, **c** (ciclo $\delta = 0$).
 - Al final del ciclo las 3 señales cambian a su nuevo valor, en concreto pasan de '0' a '1'.
 - Señal **or_ab** recibe la suma booleana de los valores **a** y **b**, pero no hasta pasados 4ns.
 - Señal **or_ab** sufrirá una modificación en el instante 9ns.
 - Por tanto, no cambia de valor en este paso δ ni en los siguientes.

Modelo de simulación. Ejemplo B (VII)

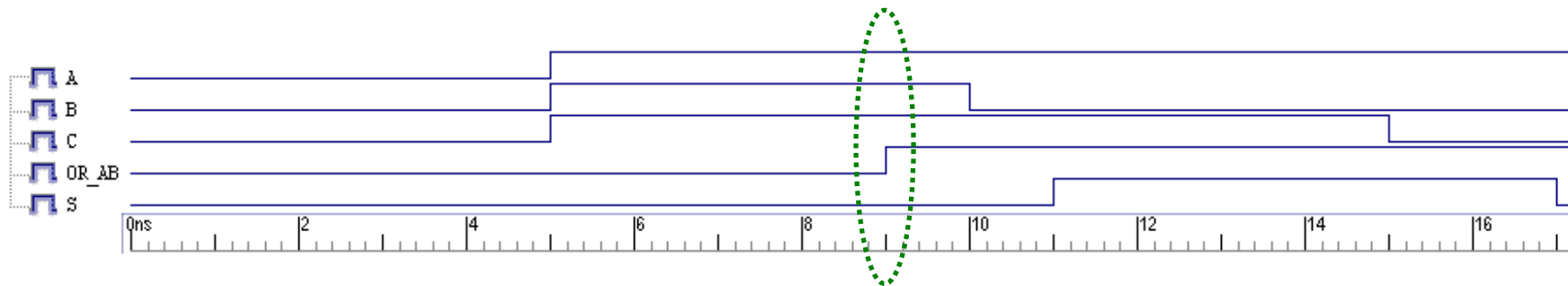


○ Instante 5ns (continuación).

- Señal **s** recibe el producto booleano de los valores de **or_ab** y **c**, pero no hasta pasados 2ns.
 - Además, la señal **or_ab** no recibe su valor hasta el instante 9ns. Por tanto, en el instante 7ns la señal **s** no puede recibir su valor (depende de **or_ab**).
 - Por tanto, no cambia de valor en este paso δ ni en los siguientes.

✓ **Avance en el tiempo físico hasta siguiente evento.**

Modelo de simulación. Ejemplo B (VIII)



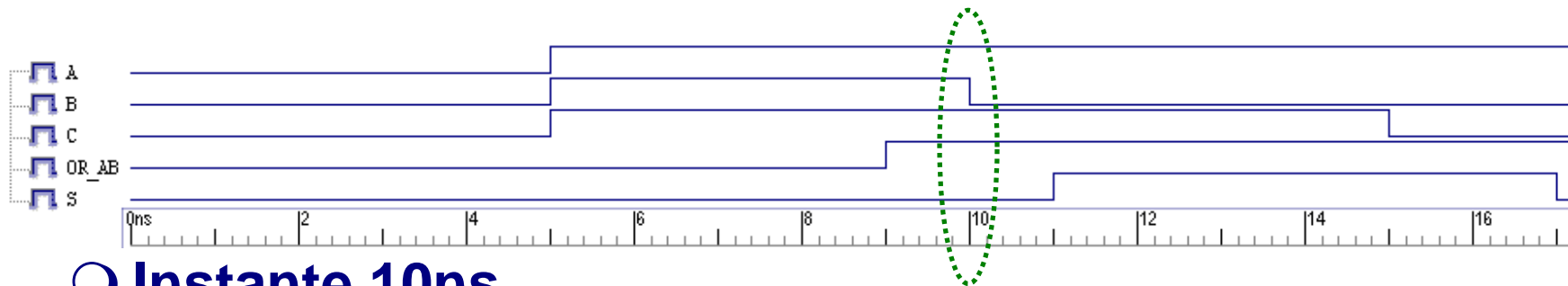
○ Instante 9ns.

❑ Se produce un evento sobre la señal **or_ab** (ciclo $\delta = 0$).

- Señal **s** recibe el producto booleano de los valores de **or_ab** y **c**, pero no hasta pasados 2ns.
 - Señal **s** sufrirá una modificación en el instante 11ns.
 - Por tanto, no cambia de valor en este paso δ ni en los siguientes.

✓ **Avance en el tiempo físico hasta siguiente evento.**

Modelo de simulación. Ejemplo B (IX)

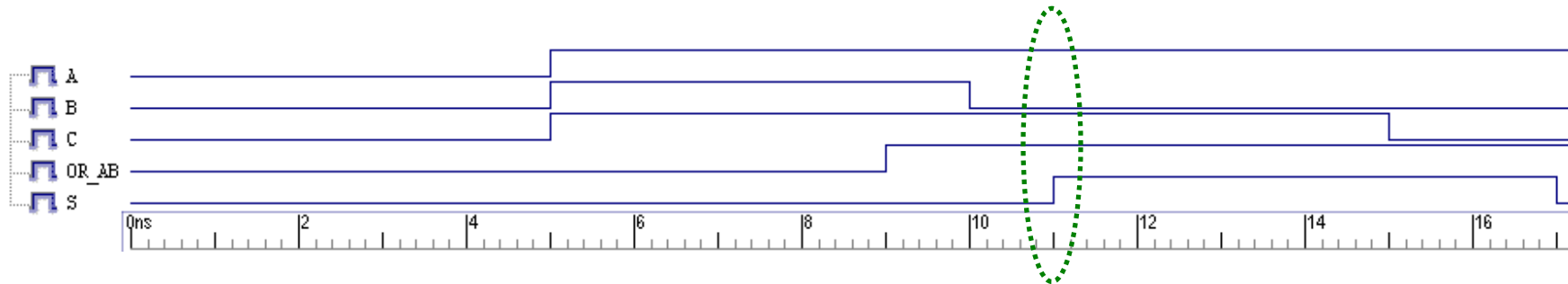


○ Instante 10ns.

- ❑ Se produce un evento sobre **b** (ciclo $\delta = 0$).
 - Al final del ciclo la señal **b** cambia a su nuevo valor; en concreto pasa de '1' a '0'.
 - Señal **or_ab** recibe la suma booleana de los valores de **a** ('1') y **b** ('0'), pero no hasta pasados 4ns.
 - Por tanto, no cambia de valor en este paso δ ni en los siguientes.
 - Además, el valor de **or_ab** no va a cambiar porque seguirá valiendo '1'. Esto quiere decir que no se producirá un evento en el instante 14ns.

✓ Avance en el tiempo físico hasta siguiente evento.

Modelo de simulación. Ejemplo B (X)

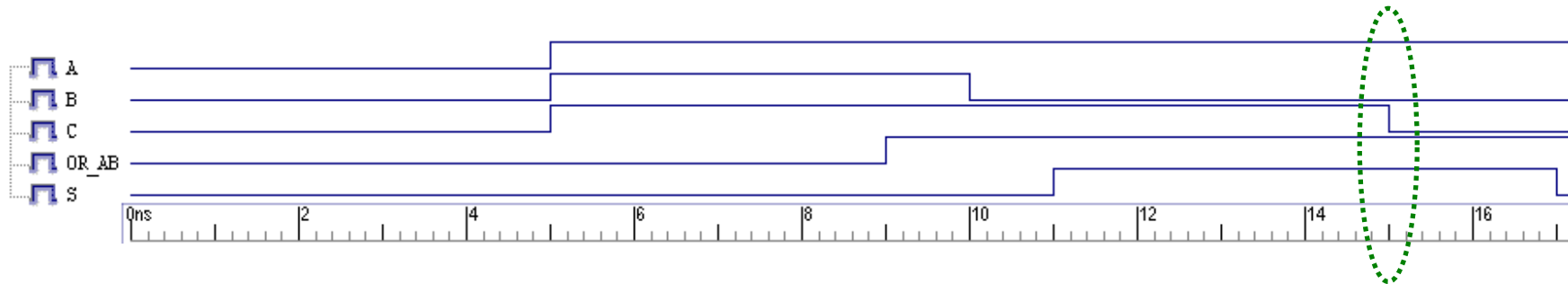


○ Instante 11ns.

- Se produce un evento sobre **s** (ciclo $\delta = 0$).
 - Señal **s** recibe ahora el producto booleano de los valores de **or_ab** y **c**, puesto que ya han pasado los 2ns desde el instante 9ns.
 - No se producirán nuevos cambios en los siguientes pasos δ .

✓ **Avance en el tiempo físico hasta siguiente evento.**

Modelo de simulación. Ejemplo B (XI)

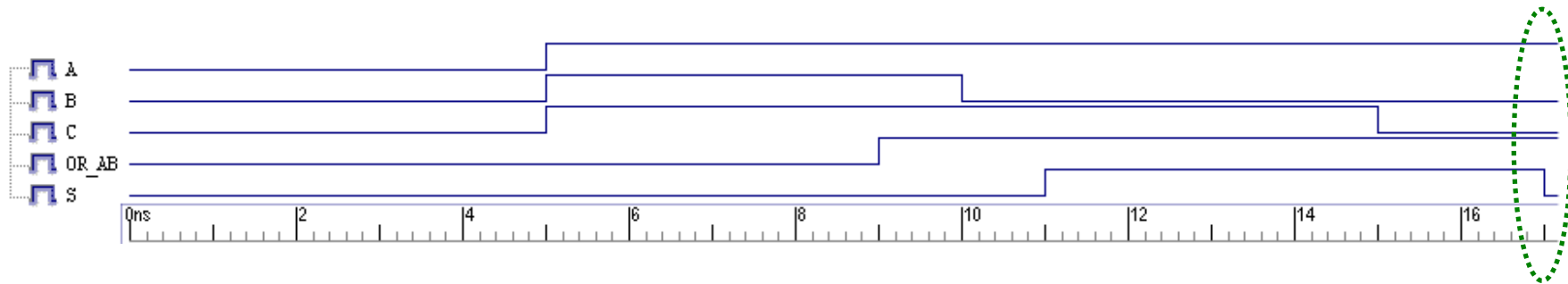


○ Instante 15ns.

- Se produce un evento sobre **c** (ciclo $\delta = 0$).
 - Al final del ciclo las señal **c** cambia a su nuevo valor; en concreto pasa de '1' a '0'.
 - Señal **s** recibe el producto booleano de los valores de **or_ab** y **c**, pero no se actualiza hasta pasados 2ns.
 - El cambio se producirá en el instante 17ns.
 - Por tanto, no cambia de valor en este paso δ ni en los siguientes.

✓ **Avance en el tiempo físico hasta siguiente evento.**

Modelo de simulación. Ejemplo B (XII)



○ Instante 17ns.

□ Cálculo de valores futuros de (ciclo $\delta = 0$):

- Señal *s* recibe el producto booleano de los valores de *or_ab* y *c*, después de haber transcurrido 2ns.
 - No se producirán nuevos cambios en los siguientes pasos δ .

✓ **Avance en el tiempo físico hasta siguiente evento.**

Modelo de simulación. Ejemplo B (XIII)

- ✓ A partir de este punto, no existen nuevos eventos en las señales de entrada *a*, *b*, *c*.
- ✓ Se alcanza el final de la simulación.

Modelo de simulación. Ejemplo B (XIV)

Tiempo (ns)	Delta (δ)	a	b	c	or_ab	s
0	0	0	0	0	0	0
5	0	1	1	1	0	0
9	0	1	1	1	1	0
10	0	1	0	1	1	0
11	0	1	0	1	1	1
15	0	1	0	0	1	1
17	0	1	0	0	1	0