

A dark blue vertical bar is positioned on the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the date '10-2-2024'. Below the banner, several thin, curved lines in shades of blue and grey sweep upwards from the bottom left corner.

10-2-2024

Programación de sistemas reconfigurables

Tarea 4. Arquitectura y estilos

Ingeniería en electrónica y computación

ANDRADE SALAZAR, IGNACIO

CENTRO UNIVERSITARIO DE LOS VALLES, UNIVERSIDAD DE
GUADALAJARA

Arquitectura y estilos de programación VHDL

1-¿Defina una arquitectura?

Una arquitectura define el algoritmo o la estructura de solución de una entidad, en esta se describen las instrucciones o los procedimientos “programa” que deben llevarse a cabo para obtener la solución deseada por el diseñador.

2-Hay tres tipos de estilos de programación, ¿cuáles son?

De manera general y con base en como se presenta la entidad(entity), a primera vista, en VHDL se pueden distinguir los siguientes estilos de programación:

- Estilo por flujo de datos
- Estilo funcional
- Estilo estructural

3-Comente sobre la descripción por flujo de datos, primer párrafo

La descripción por flujo de datos muestra con detalle la transferencia de información entre entradas y las salidas de una entidad. Este estilo, totalmente comprensible para el usuario, se recomienda para quienes recién se inician en la programación en VHDL, en materias como diseño lógico, donde las tablas de verdad y las ecuaciones lógicas son parte fundamental en la descripción de un circuito lógico.

4-Ponga el listado 1.9 (programa) como ejemplo – transcribirlo

```
--Declaración flujo de datos

library ieee;

use ieee.std-logic_1164.all;

entity tabla ios

port (A,B: in std_logic;

      F0, F1: out std_logic);

end table;
```

architecture flujo of tabla is

begin - - comenzar

F0 <= '1' when (A='0' and B='0') else

'1' when (A='0' and B='1') else

'0' ;

F1 <= '0' when (A='1' and B='1') else

'1' ;

end flujo;

5-La segunda opción de flujo de datos es apoyándose en los operadores lógicos

Los operadores lógicos que se utilizan en la descripción con ecuaciones booleanas y que están definidos dentro de los diferentes tipos de datos -bit, boolean o std-logic- son los operadores: and, or, nand, xor, xnor y not. Las operaciones que se efectúen entre estos (excepto not) deben realizarse con datos que tengan la misma longitud o palabra de bits.

Ponga las ecuaciones que están antes del listado 1.10, y ponga el listado 1.10 -transcribirlo

$$F0 = \neg A * \neg B + \neg A * B$$

$$F1 = \neg A * \neg B + \neg A * B + A * \neg B$$

$$F0 <= ((\text{NOT } A \text{ AND NOT } B) \text{ OR } (\text{NOT } A \text{ AND } B)) ;$$

$$F1 <= ((\text{NOT } A \text{ AND NOT } B) \text{ OR } (\text{NOT } A \text{ AND } B) \text{ OR } (A \text{ AND NOT } B)) ;$$

--Declaración flujo de datos

library ieee;

```

use ieee.std_logic_1164.all;

entity table is

port (A, B: out std_logic ;

      F0, F1 : out std_logic) ;

End table ;

architecture flujo of table is

begin - - comenzar

      F0 <= ( (NOT A AND NOT B) OR (NOT A AND B) ) ;

      F1 <= ( (NOT A AND NOT B) OR (NOT A AND B) OR (A AND NOT B) ) ;

End flujo;

```

6-Operadores lógicos ¿cuáles son ¿y ¿cuándo se utilizan?

Los operadores lógicos presentan el siguiente orden y prioridad al momento de ser compilados:

1. Expresiones entre paréntesis
2. Complementos
3. Función and
4. Función or

Se utiliza ‘_’ una comilla para expresar un valor individual

Se utiliza “-“ doble comilla para describir un grupo de bits “vector”

7-¿Qué es lo más importante en una descripción funcional?

En una descripción funcional lo mas importante es el conocimiento global del sistema, razón por la cual las entidades diseñadas bajo este estilo son programadas como una caja negra; es decir, no importa la organización o la estructura interna de la entidad, solo se requiere que el programador conozca lo que espera obtener en la salida y la forma en que operan las pins de entrada.

ponga el listado 1.13 - transcribirlo

--Ejemplo de una descripción funcional

```
library ieee;

use ieee.std_logic_1164.all;

entity functional is

port (A,B: in std_logic;

      c: out std_logic);

end functional ;

architecture caja of funcional is

begin

    if A = B then    --Si A=B entonces

        C<='1';

    Else            -- sino

        C<='0';

    end if;

end process;

end caja;
```

8-Una declaración funcional utiliza una nueva sentencia deno..... complete el párrafo, pág 18

...denominada process, la cual se aprecia en la línea 10. La delaración proceso(process) está acompañada de una lista sensitiva entre paréntesis(A,B), que hace referencia a las señales que determinan el funcionamiento del proceso - el cual debe entenderse como el hecho de que el algoritmo es sensible al cambio de valor de estas variables-.

9-Descripción estructural basa su comportamiento en ... complete

...modelos lógicos ya establecidos (compuertas, sumadores, contadores, proyectos especiales, etc.). Es importante destacar que estas estructuras pueden ser diseñadas por el usuario y guardadas para su posterior utilización o extraídas de los paquetes contenidos en las librerías de diseño del software que se este utilizando, como se verá más adelante.

Ponga el listado 1.15 - transcribirlo

```
library ieee;

use ieee.std_logic_1164.all;

entity estructura is
    port (a, b: in bit_vector (0 to 1);
          c: out bit);
end estructura;

use work.gates.all;

architecture structural of estructura is
    signal x: bit_vector (0 to 1);
begin

    U0:      xnor2 port map (a(0), b(0), x(0) );
    U1:      xnor2 port map (a(1), b(1), x(1) );
    U2:      and2  port map (x(0), x(1), c);

End structural;
```

10-Transcriba tal cual el párrafo del punto 1.4 Comparación entre los estilos de diseño

El estilo de diseño utilizado en la programación depende de manera exclusiva del diseñador y de la complejidad del proyecto. Por ejemplo, un diseño puede describirse mediante ecuaciones booleanas, pero si este es muy extenso quizá sea mas apropiado hacerlo a través de estructuras jerárquicas para dividirlo. Ahora bien, si se requiere diseñar un sistema cuyo funcionamiento dependa solo de sus entradas y sus salidas, es conveniente emplear la descripción funcional, la cual tiene ventaja de requerir un número menor de instrucciones, además de que el diseñador no necesita un conocimiento previo de la función de cada componente que forma el circuito.