



Universidad Autónoma de Querétaro

Facultad de Ingeniería
Ingeniería en Automatización



SDLR - switch case, if y when

M en C. Marcos Romo Avilés

Practica 3: switch case, if y when

- **Introducción**

- Al diseñar sistemas digitales con FPGAs, es esencial manejar diferentes flujos de control y decisiones lógicas basadas en señales de entrada variables. Las estructuras de control switch-case e if en lenguajes de descripción de hardware como VHDL son fundamentales para implementar esta lógica condicional. En la Práctica 3, nos enfocaremos en cómo estas estructuras nos permiten interpretar y responder a las señales de los switches de entrada en una placa FPGA.

Practica 3: switch case, if y when

- En VHDL, la construcción **process** es un elemento fundamental que permite definir un conjunto de instrucciones que se ejecutarán secuencialmente. Dentro de un process, se pueden realizar operaciones lógicas, aritméticas, manipulación de señales, y mucho más. Los process son especialmente útiles para describir comportamientos síncronos y asíncronos en circuitos digitales, permitiendo simular la lógica secuencial y combinacional.
- **Sensibilidad:** Cada process en VHDL tiene una lista de sensibilidad, la cual define a qué señales reacciona el process.
- **Variables y señales:** Dentro de un process, se pueden declarar variables locales que existen y retienen su valor solo dentro del contexto del process. A diferencia de las variables, las señales (signal) tienen un alcance que trasciende el process y pueden ser utilizadas para comunicarse entre diferentes process o con la entidad que engloba el process.

Practica 3: switch case, if y when

- **Secuencialidad:** Aunque VHDL se utiliza para describir hardware que opera en paralelo, el código dentro de un process se ejecuta de manera secuencial. Esto permite describir algoritmos o comportamientos que dependen de una secuencia específica de operaciones.
- **Implementación de estados:** Los process son fundamentales para implementar máquinas de estado finito (FSM), donde el estado actual del sistema determina su comportamiento en respuesta a las entradas y puede cambiar basándose en estas.

Practica 3: switch case, if y when

- Mientras que if es efectivo para decisiones simples, switch-case es más adecuado para cuando hay múltiples condiciones a evaluar, como interpretar una entrada codificada o seleccionar entre varios estados o modos:

```
process(entrada_num)
```

```
begin
```

```
  case entrada_num is
```

```
    when "0000" => display <= "0000001";
```

```
    when "0001" => display <= "1001111";
```

```
    when others => display <= "1111111"; end case;
```

```
end process;
```

Practica 3: switch case, if y when

- Mostrar el abecedario en la pantalla de 7 segmentos utilizando switch cases
 - Escribir la tabla de verdad
 - Simulación
 - E implementación

Practica 3: switch case, if y when

- La instrucción **if** es una de las estructuras de control más básicas y se utiliza para ejecutar código basado en condiciones booleanas:

```
process(clk, reset)
begin
    if reset = '1' then
        -- Inicializa o resetea algo cuando reset está activo
    elsif rising_edge(clk) then
        -- Realiza operaciones en el flanco ascendente del reloj
    end if;
end process;
```

Practica 3: switch case, if y when

- Mostrar 8 de al menos 6 letras en las pantallas de 7 segmentos utilizando if
 - Escribir la tabla de verdad
 - Simulación
 - E implementación

Practica 3: switch case, if y when

- En VHDL, when se utiliza comúnmente dentro de las expresiones condicionales y en asignaciones seleccionadas o condicionales. La construcción when permite especificar una acción o un valor basado en una condición.

```
resultado <= "1000" when entrada = "000" else  
    "0100" when entrada = "001" else  
    "0010" when entrada = "010" else  
    "0001";
```