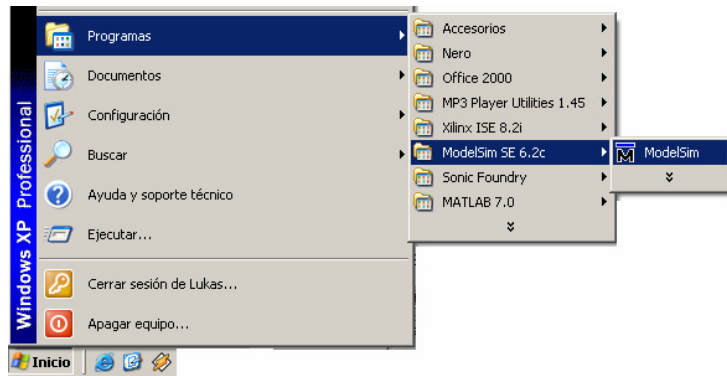


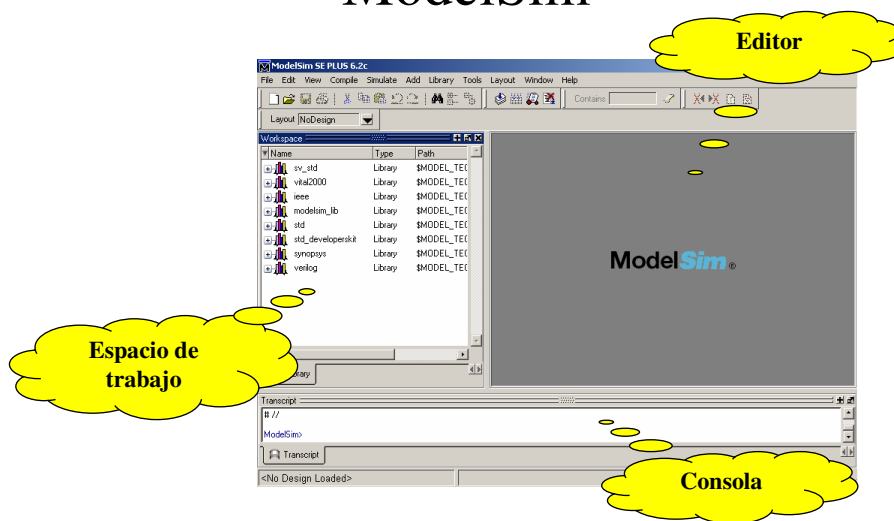
Tutorial ModelSim (I)

- **Diseño:**
 - Contador de 8 bits
- **Contenidos**
 - Utilización básica de la herramienta
 - Creación de Proyectos
 - Importación de archivos
 - Compilación de archivos
 - Simulación

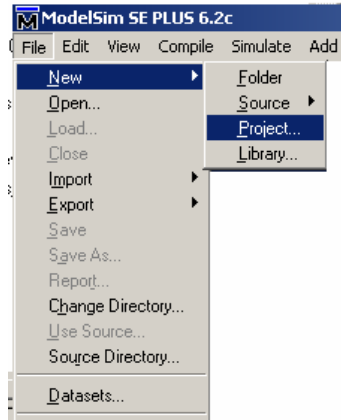
Inicio de la Aplicación



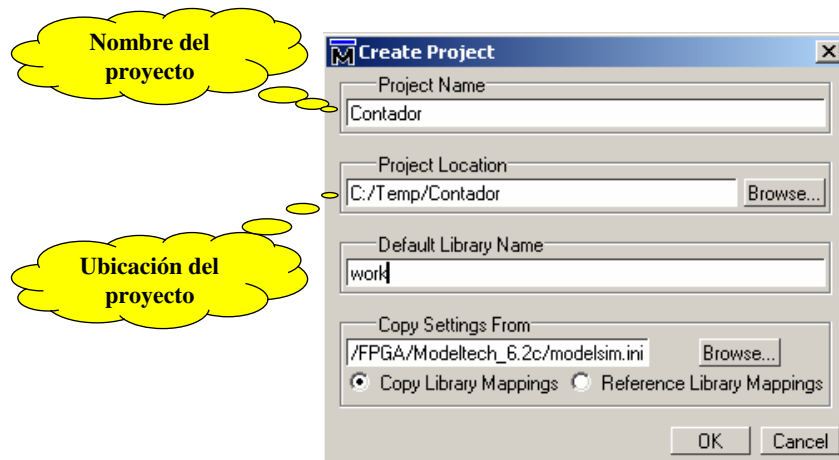
ModelSim



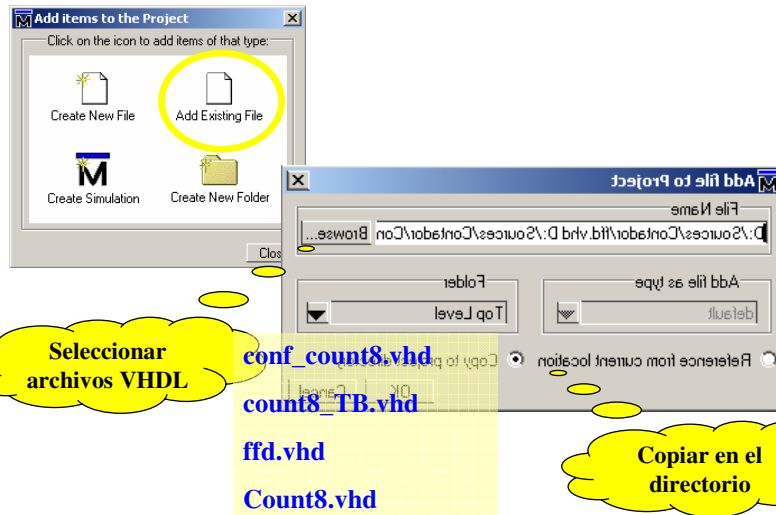
Creación de Proyecto



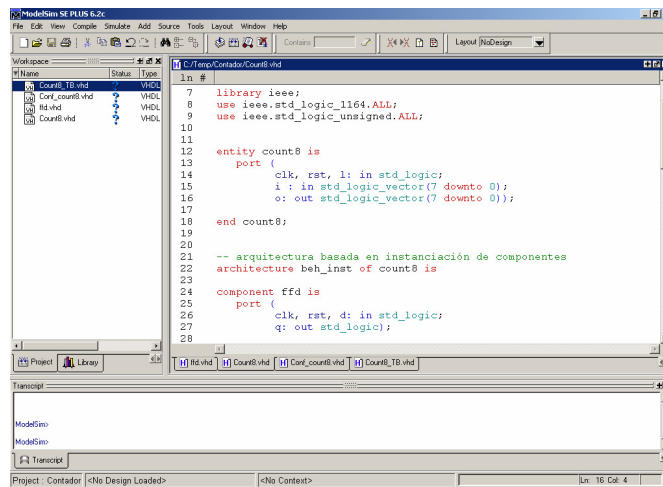
Creación de Proyecto



Importar Archivos



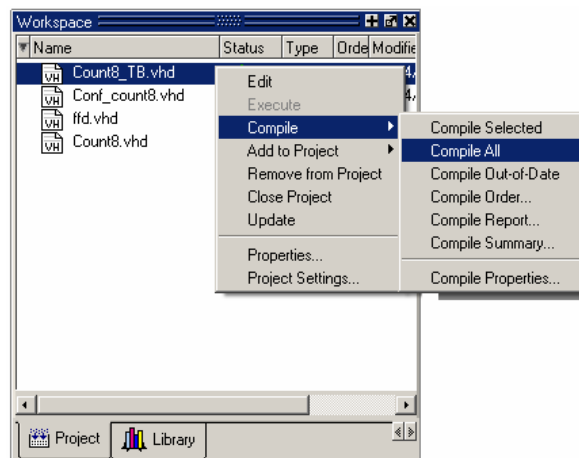
Edición de Archivos



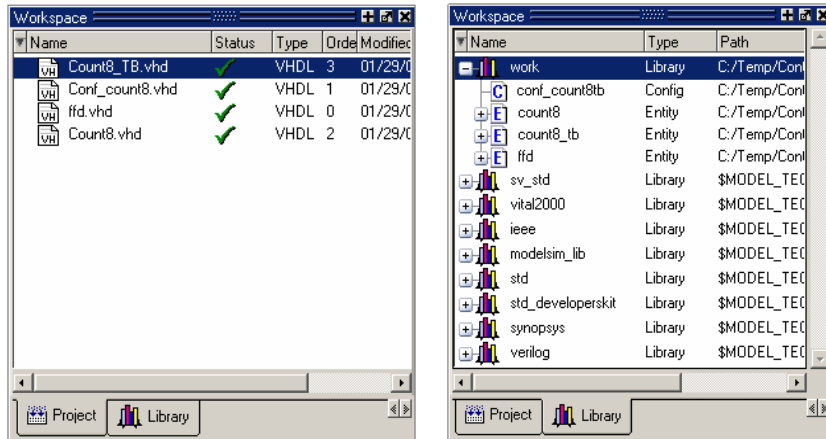
Archivos del proyecto

- **ffd.vhd**
 - implementación de Flip-Flop tipo D.
- **Count8.vhd**
 - Descripción de contador de 8 bits
 - Posee 2 arquitecturas
 - Basada en instanciación de componentes : **beh_inst**
 - Basada en procesos: **beh_proc**
- **count8_TB.vhd**
 - Testbench del contador de 8 bits.
- **conf_count8.vhd**
 - Configuración del Testbench

Compilación de Archivos



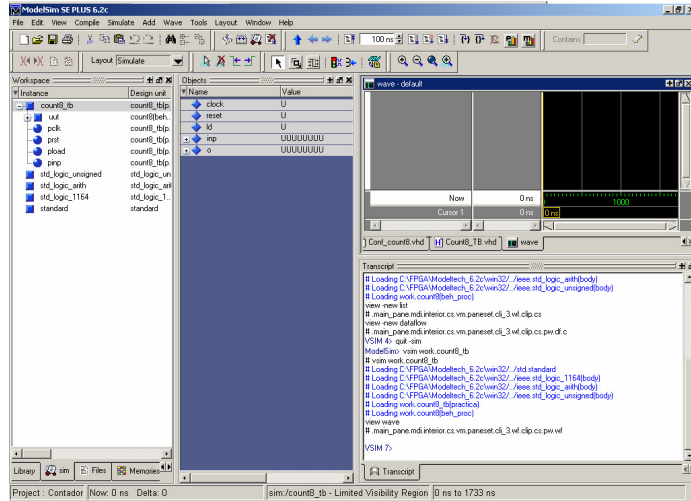
Compilación de Archivos



Simulación de arquitectura beh_inst



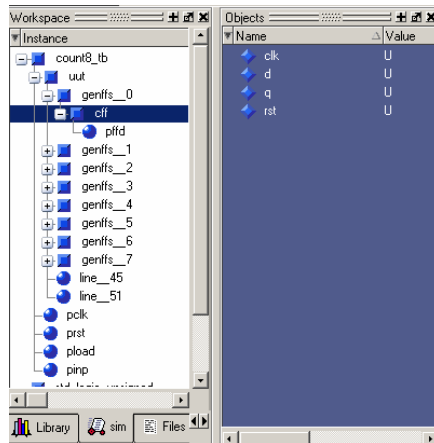
Simulación



Inserción de señales

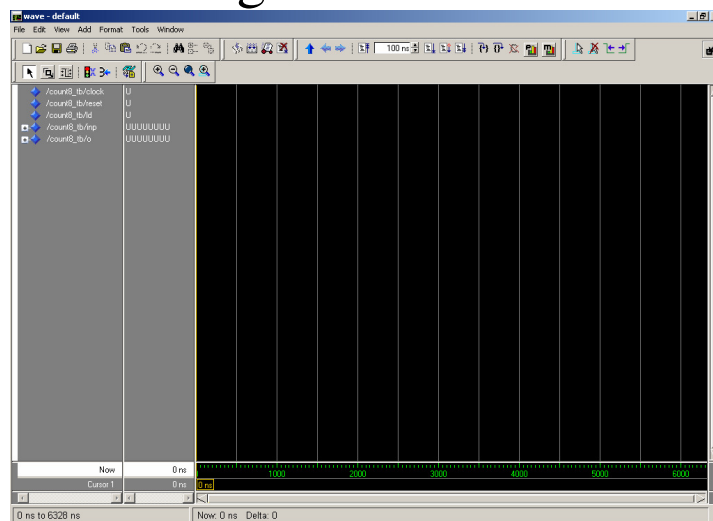


Inserción de señales



Es posible insertar señales internas a la unidad bajo testeo o de sus subcomponentes (Ej: señales de flip-flops)

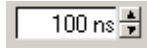
Ventana de gráfica de Simulación



Comandos de Simulación



Reiniciar ejecución



100 ns

Tiempo de paso de simulación



Avanzar un paso de simulación



Simula hasta finalizar tiempo por defecto

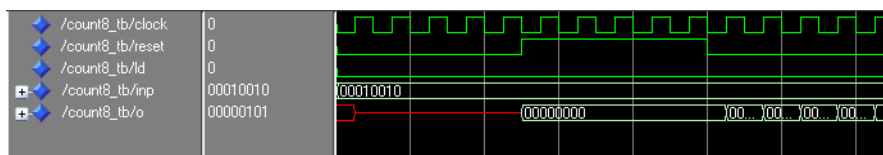


Simula indefinidamente



Detiene la simulación



Simulación (beh_inst)



- Para cambiar sistema de representación de las señales
 - 1) seleccionar la señal
 - 2) Menu -> Format -> Radix -> “Sistema elegido”

Simulación (beh_inst)

- Continuar simulando
- Verificar el comportamiento del contador

Nota: los comandos  y  avanzan o retroceden hasta la próxima transición de la señal seleccionada.

- Detener la simulación
 - En la ventana de ModelSim:
 - **Menu->Simulate->End Simulation**

Simulación (beh_proc)

- Modificar el archivo conf_count8tb para simular la arquitectura basada en procesos.

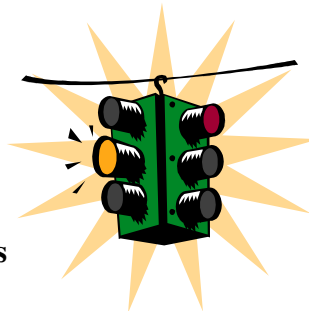
```
configuration conf_count8tb of count8_tb is
  for practica
    for UUT: count8
      --use entity work.count8(beh_inst);
      use entity work.count8(beh_proc);
    end for;
  end for;
end conf_count8tb;
```

Simulación (beh_proc)

- Guardar el archivo modificado
- Compilar
- Inicializar simulación
- Analizar el comportamiento

Tutorial ModelSim (II)

- **Diseño:**
controlador de semáforo.
- **Contenidos**
 - Edición de código VHDL
 - Lectura de reportes de errores
 - Creación de Testbenches



- 1 – Crear un nuevo proyecto “semaforo”
- 2 – Importar archivo “semaforo. vhd”

Descripción de la arquitectura (1)

- Puertos de entrada
 - clk : Reloj
 - rst: Reset asincrónico
- Puertos de salida
 - s1,s2: señales de 3 bits que corresponden con las luces del semáforo

S_i(2) **S_i(1)** **S_i(0)**

1 = encendido
0 = apagado

Descripción de la arquitectura (2)

- Señales
 - **count, next_count: establecen los tiempos del semáforo**
 - **state, next_state: estados actual y próximo**
 - **rst_count: reinicio de contador**

Descripción de la Arquitectura (3)

- Procesos explícitos
 - **sync**: actualiza los registros del diseño
 - **comb**: establecerá las transiciones
- Procesos implícitos
 - Asignación de salidas
 - $S1(2) \leq '1'$ when (state...

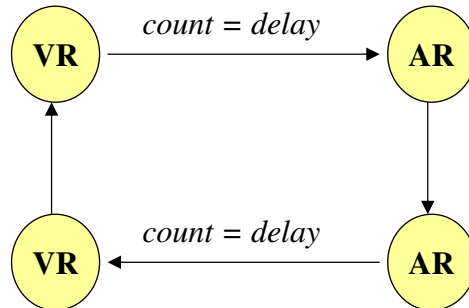
NOTA: Las señales *count* y *state* son registros. Sólo se actualizan en flancos ascendentes de reloj en el proceso *sync*

Comportamiento de la arquitectura

– Identificación de Estados:

Estado	S1	S2
VR	“001” ●●●	“100” ●●●
AR	“010” ●●●	“100” ●●●
RV	“100” ●●●	“001” ●●●
RA	“100” ●●●	“010” ●●●

Definición de la SM

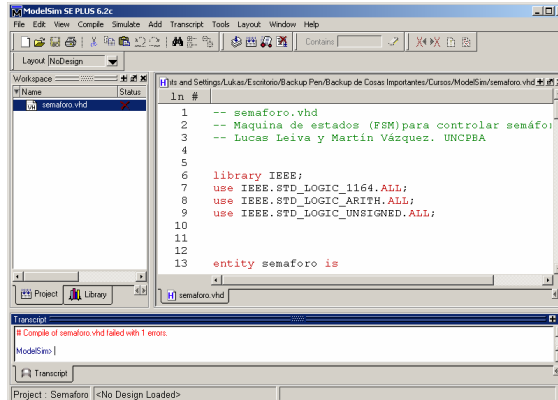


NOTA: Durante las transiciones se debe reiniciar el contador (`rst_count <= '1'`)

Edición de código VHDL

- Completar el proceso *comb*
 - transiciones de estado
 - Control de señal *rst_count*
- Compilar el proyecto
 - Errores ?

Identificación de Errores (1)

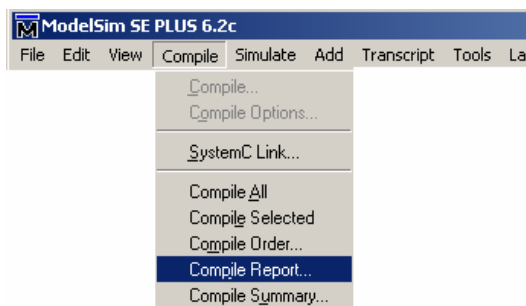


```
1 -- semaforo.vhd
2 -- Maquina de estados (FSM) para controlar semafo
3 -- Lucas Leiva y Martin Vázquez. UNCPBA
4
5
6 library IEEE;
7 use IEEE.STD_LOGIC_1164.ALL;
8 use IEEE.STD_LOGIC_ARITH.ALL;
9 use IEEE.STD_LOGIC_UNSIGNED.ALL;
10
11
12
13 entity semaforo is
```

- ¿Los errores no se muestran?!

- SI.

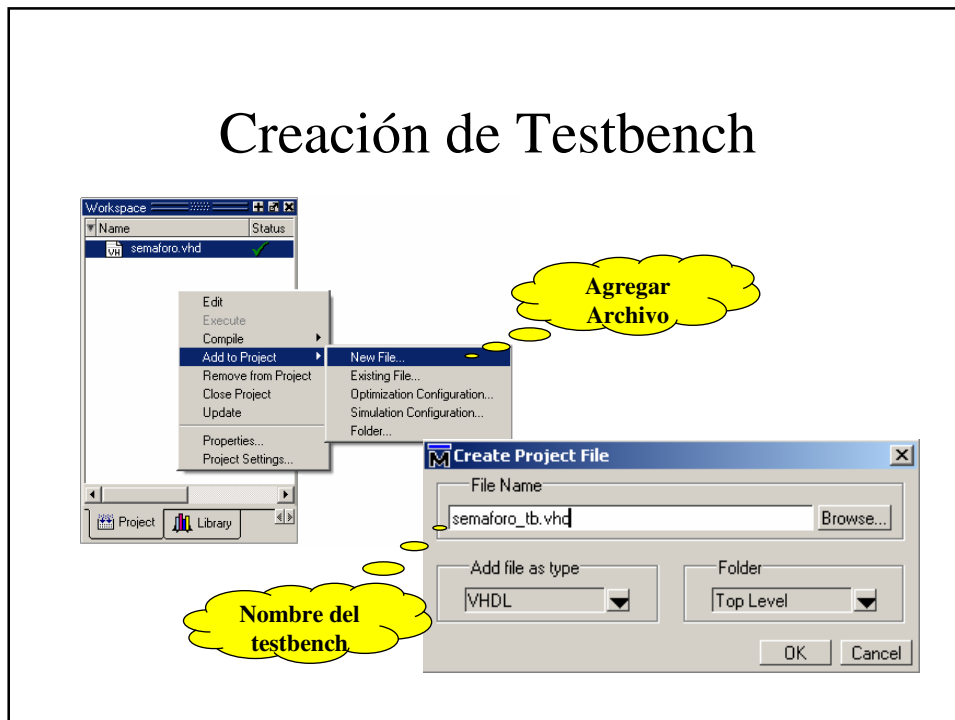
Identificación de Errores (2)



**** Error: C:/DIR/semaforo.vhd(97): (vcom-1152) Index value 3 is out of index range 2 downto 0 of ieee.std_logic_1164.std_logic_vector.**

s2(2) <= '1' when (state = VR) or (state = AR) else '0';

Creación de Testbench



Edición de Testbench

- Agregar el código necesario para que:
 - rst esté activa durante los primeros 100 ns.
 - clk posea una frecuencia de 10 MHz.
 - el valor de delay sea 15 ciclos de reloj.

Simulación

- Compilar los archivos
- Inicializar simulación del testbench
- Agregar señales del componente
- Agregar señales *state* y *count*
- Analizar comportamiento

Simulación (2)

/semaforo_tb/clk	0	
/semaforo_tb/rst	0	
/semaforo_tb/s1	001	
/semaforo_tb/s2	100	
/semaforo_tb/uut/state	vr	
/semaforo_tb/uut/count	UUU	

-Problema

La señal *count* no posee el valor esperado.

-Causa:

Dos procesos diferentes asignan valor a *count*

-Solución

next_count <= (others => '0') when rst_count...