

Funciones System Verilog: Assert.

Vargas Meza Aldo Alexandro, 213495653

System Verilog es un paquete de extensiones de Verilog como lenguaje descriptivo de hardware, estándar de IEEE.

I. ASSERTIONS

Las aserciones se usan principalmente para validar comportamiento de un diseño, y saber si está fluyendo de manera correcta.

También nos dan cobertura funcional para el testeo de un diseño. También pueden ser usadas como una especificación formal del lenguaje, haciendo los requerimientos claros y con un propósito específico.

Una aserción es una instrucción para una herramienta de verificación, y checar así una propiedad.

Las propiedades pueden ser checadas dinámicamente por simuladores como VCS o estáticas analizados por otra herramienta de chequeo de propiedad como Magellan.

En SystemVerilog hay dos tipos de aserciones: inmediata (**assert**) y concurrentes (**assert property**). Declaraciones de cobertura (**cover property**) de propiedad son concurrentes y tienen la misma sintaxis que las aserciones concurrentes, como lo hacen las aserciones de asunción, que se utilizan principalmente por las herramientas formales.

Por último, **expect** es una declaración de procedimiento que comprueba que se produce alguna actividad específica (propiedad). Los tres tipos de aserciones concurrente y la declaración **expect** hacen uso de secuencias que describen el diseño "comportamiento temporal" es decir, el comportamiento en el tiempo, tal como se define por uno o más `clk's`.

Las **aserciones inmediatas** son declaraciones procedimentales utilizadas principalmente en la simulación. Son básicamente entradas que necesitan que algo sea "verdadero" similar a la función **if**. La diferencia es que el **if** no afirma que una función sea verdadera, simplemente checa que así lo sea. Para ejemplificar:

La sintaxis es la siguiente

```
[ name : ] assert ( expression ) [ pass_statement ] [ else fail_statement ]
```

La aserción lleva a cabo un chequeo booleano, y evalúa si el procedimiento es el correcto.

```
always @(negedge reset)
  a_fsm_reset: assert (state == LOAD)
    $display("FSM reset in %m passed");
  else
    $display("FSM reset in %m failed");
```

if (A == B) ... // Simply checks if A equals B
assert (A == B); // Asserts that A equals B; if not, an error is generated

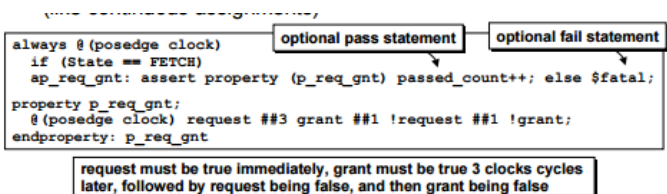
Assert se encarga de que la condición se esté cumpliendo, si la aserción evalúa que no se cumple la condición, el simulador escribe un mensaje de error y falla.

Son utilizadas en debuggin atrapando a los errores fatales por ejemplo, los can't happen.

Son utilizadas como monitores para malos comportamientos y para alertar de posibles malos comportamientos.

Las **aserciones concurrentes** pueden llevar a cabo un test de diferentes secuencias de eventos dependientes de diferentes `clks` y describe la secuencia de los eventos. La sintaxis es:

```
[ name : ] assert property ( property_specification ) pass_statement [ else fail_statement ]
```



II. IMPORTANCIA DE LAS ASERCIONES

Le da una visibilidad de caja blanca al diseño, y permite especificar los requerimientos así como hacer la detección de errores una tarea más sencilla.

En una simulación, los errores de diseño pueden ser detectados automáticamente. Análisis formales pueden detectar si la aserción funciona correctamente.

III. DISPLAY %P SYSTEMVERILOG

Una nueva función agregada a la LRM del 2009 es la impresión con el símbolo `%p`. Este, imprime un patrón de asignación que es equivalente a los datos del valor del objeto.

Puedes imprimir cualquier tipo de dato, incluidos arreglos, estructuras, clases y más.

```
initial begin
    ascend = '{0,1,2,3};
    $display("%p", ascend);      // '{0, 1, 2, 3}
    ascend = '{4{8}};
    $display("%p", ascend);      // '{8, 8, 8, 8}
end
```

IV. BIBLIOGRAFÍA

<http://stackoverflow.com/questions/20326307/class-object-print-in-system-verilog>

http://www.sutherland-hdl.com/papers/2006-DesignCon_Getting_Started_with_SVA_presentation.pdf

<https://www.doulos.com/knowhow/sysverilog/tutorial/assertions/>

<http://www.asic-world.com/systemverilog/assertions1.html>
Wikibooks.org/wiki/Programaci%C3%B3n_en_Verilog/