

Pre-Informe 3

Tomás Catalán
201773110-9

1 Investigacion de conceptos 1:

- a) ¿Qué es un lenguaje de descripción de hardware? ¿Qué lo diferencia de un lenguaje de programación?
R: Un lenguaje de descripción de hardware (HDL, hardware description language) es un lenguaje de programación especializado que se utiliza para definir la estructura, diseño y operación de circuitos electrónicos, y más comúnmente, de circuitos electrónicos digitales, como el convertidor analógico-digital o cualquier antena satelital. Así, los lenguajes de descripción de hardware hacen posible una descripción formal de un circuito electrónico, y posibilitan su análisis automático y su simulación. Los lenguajes de descripción de hardware se parecen mucho a otros lenguajes de programación de ordenadores tales como el C o Java: básicamente consisten en una descripción textual con expresiones, declaraciones y estructuras de control. Sin embargo, una importante diferencia entre los HDL y otros lenguajes de programación está en que el HDL incluye explícitamente la noción de tiempo.
- b) Explique la funcionalidad de los tipos de variable wire, reg y logic, y cuales son sus diferencias.
R:
- reg: Representan variables con capacidad de almacenar información.
 - wire: Representan conexiones estructurales entre componentes. No tienen capacidad de almacenamiento.
 - logic: Representan valores booleanos con 1 ó 0 siendo verdadero o falso respectivamente.
- c) assign: todas las sentencias se ejecutan de manera concurrente.
<=: Puede ejecutar concurrentemente N sentencias dentro de un proceso always. Es recomendado para descripción de lógica secuencial.
=: Este se ejecuta secuencialmente dentro de un always y es usado en la descripción de circuitos combinacionales
- d) reg [15:0]g = 16'hA6B2;
reg: esto define el tipo de dato de nuestra variable.
[15 : 0]= Esto define la cantidad de bits que tendrá nuestra variable.
16: esto nos dice el número de bits del valor.
h: esto nos dice que la notación que sigue es en hexadecimal.
A6B2: notación hexadecimal.
Binario: 1010011010110010
- e) R: esto ocurre ya que second no tiene capacidad de almacenamiento así que al momento de necesitar su valor se ejecuta en el momento. En el momento en que se ejecuto el valor de first había cambiado así que también afectara en el resultado de second, todo esto ocurrió dentro del always_ff.

2 Analisis de codigo 2:

Se toma como $r=000$

- $Y=01110111$ $X=00000000$ $Z=000$ $N=01110111$ $M=001$
- $Y=01110110$ $X=00000000$ $Z=001$ $N=01110110$ $M=010$
- $Y=01110100$ $X=00000000$ $Z=010$ $N=01110100$ $M=011$
- $Y=01110100$ $X=00000000$ $Z=011$ $N=01110100$ $M=100$
- $Y=01110100$ $X=00010000$ $Z=100$ $N=10000100$ $M=101$
- $Y=01100100$ $X=00010000$ $Z=101$ $N=01110100$ $M=110$
- $Y=01000100$ $X=0010000$ $Z=110$ $N=01000100$ $M=111$
- $Y=11000100$ $X=$
- $N=11011100$