**MEMORY ARRAYS – LOGIC ARRAYS**

Felipe Moreno, Christian Pinzón

*Universidad Nacional De Colombia*

*Bogotá, Colombia*

efmorenor@unal.edu.co

ccpinzong@unal.edu.co

1. INTRODUCCIÓN

Este corto informe explicará de manera breve y puntual el diseño las pautas que se tuvieron en cuenta para la construcción del código de una memoria RAM y memoria ROM de N x N bits, parametrizable y un arreglo de compuertas parametrizable para ser AND o OR.

1. DESCRIPCIÓN DE LOS CODIGOS

Antes de diseñar cada uno de los módulos era necesario diferenciar las funciones del ROM y la RAM, para comenzar a definir variables e identificar las funciones que nos permitieran diseñar o crear el código, de acuerdo a lo anterior la memoria RAM puede leer/escribir sobre sí misma y la memoria ROM como caso contrario, sólo puede leer por lo tanto no tiene acceso a la escritura.

Se necesitaba que las memorias fueran de tamaño N x N bits donde N es máximo de 128 bits y es parametrizable.

Por otra parte se necesita un arreglo de compuertas configurable para ser AND o OR, de N bits, donde N es parametrizable. Siendo así, este módulo se comportaría como un dispositivo de lógica combinacional.



* Memoria RAM y ROM

De acuerdo al código implementado las funciones que se tuvieron en cuenta para su funcionamiento fueron:

* *Parameter:* Se usó para la parametrización de cada uno de los módulos, donde se asigna el nombre y su respectivo valor, que en nuestro caso es N.
* *Inputs, Outputs*: Se especifica el nombre de las entradas y salidas respectivamente con las cuales se van a trabajar.
* *Reg*: Esta variable se usó para modelar las memorias, donde se accede mediante un determinado bit, con esta variable se declara el tamaño de cada memoria.
* *Always:* Este proceso se ejecuta continuamente a modo de bucle, y está continuamente ejecutándose. Que se activa cuando detecta un flanco de subida (posedge).
* *Case:* Esta variable se usó para elaborar el código de la memoria ROM, donde evalúa una expresión y en función de su valor ejecutará la sentencia o grupo de sentencias agrupadas en el primer caso que coincida.

Después de haber creado el código para cada módulo, se procedió a la simulación mediante Testbench, el cual permitió comprobar el adecuado funcionamiento del código, analizando los tiempos configurados, aplicando entradas determinadas para obtener salidas esperadas

* Logic Arrays

Además de algunas funciones usadas anteriormente, para este módulo se hicieron uso de estas:

* *Generate: P*roporciona datos para producir una determinada cantidad de variables de hardware en función del valor de un parámetro.
* *Wire:* Se lleva a cabo esta declaración a la variable que realiza asignación continua, en otras palabras este comando se usa cuando se modela lógica combinacional, que en este caso aplica.

Cabe resaltar que para declarar los módulos se usó *begin*  y *end,*  los cuales sirven para delimitarlos.