Imagen que contiene Icono

Descripción generada automáticamenteLogotipo

Descripción generada automáticamente con confianza media

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Arquitectura de Computadoras

**“Multiplexores en HDL (Verilog)”**

Alumno:

Malagón Baeza Alan Adrian

Profesor:

Alemán Arce Miguel Ángel

Grupo: 5CV1

**Introducción**

Un multiplexor es un circuito que pasa una de sus múltiples entradas a una sola salida en función de una entrada seleccionada. Esto se puede considerar como un interruptor digital. El multiplexor tiene n líneas de selección, 2n entradas y una salida. El siguiente ejemplo muestra el proceso de diseño manual de un multiplexor 2 a 1 (es decir, utilizando el enfoque de diseño digital clásico).

**Ejemplo: Multiplexor 2 a 1 - Síntesis lógica manual**

Mux\_2\_To\_1

|  |  |
| --- | --- |
| i\_Select | o\_Data |
| 0 | i\_Data1 |
| 1 | i\_Data2 |

i\_Data1

o\_Data

i\_Data2

i\_Select

Para diseñar el multiplexor, es útil enumerar todos los valores posibles para i\_Data1, i\_Data2 e i\_Select en forma de tabla de verdad

|  |  |  |  |
| --- | --- | --- | --- |
| i\_Select | i\_Data1 | i\_Data2 | o\_Data |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

i\_Select i\_Data1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |

Cuando i\_Select=0, la salida es i\_Data1

i\_Data2

Cuando i\_Select=1, la salida es i\_Data2

o\_Data = i\_Select’.i\_Data1 + i\_Select’.i\_Data2

Mux\_2\_To\_1

**Diagrama

Descripción generada automáticamente con confianza media**

i\_Select

o\_Data

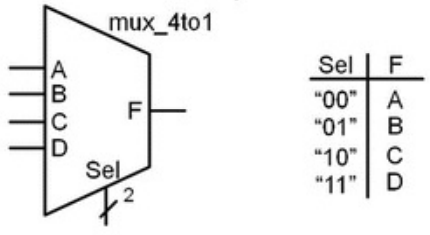
i\_Data1

i\_Data2

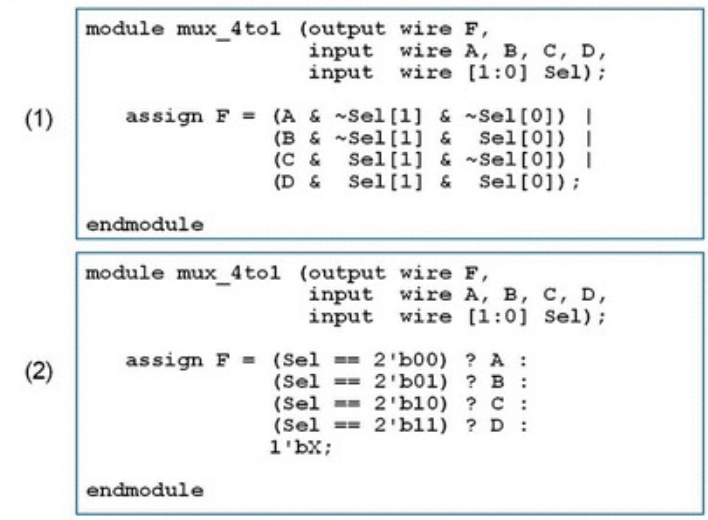
En Verilog, se puede implementar un multiplexor mediante asignación continua con operadores lógicos o condicionales. El siguiente ejemplo muestra cómo modelar el multiplexor en Verilog utilizando estas técnicas.

**Ejemplo: Multiplexor 4 a 1 - Modelado de Verilog mediante Asignación Continua**

El símbolo y la tabla de verdad para el multiplexor 4 a 1 son los siguientes:

****

Las siguientes son dos formas diferentes de implementar el comportamiento del multiplexor con asignación continua: (1) con operadores lógicos: y (2) con operador condicional.



**Desarrollo**

**Propuesta 1**

**Código Verilog**

Texto

Descripción generada automáticamente

**Código Verilog Testbench**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Implementación RTL en Vivado 2022.2**

Gráfico, Diagrama

Descripción generada automáticamente

Podemos observar cómo utilizó un multiplexor para la implementación del multiplexor 2 a 1.

**Resultado de la simulación:**

**Interfaz de usuario gráfica

Descripción generada automáticamente**

Cuando i\_Select = 00, la salida o\_Data es i\_Data2

Cuando i\_Select = 10, la salida o\_Data es i\_Data1

Como se observa en la simulación, la salida o\_Data recibe el valor de entrada i\_DataX según la expresión asignada en el selector i\_Select.

o\_Data = i\_Select ? i\_Data1 : i\_Data2;

o\_Data = 00 ? i\_Data1 (true) : i\_Data2 (false); 00 -> false

o\_Data = i\_Data2

**Propuesta 2, Parametrización**

**Código Verilog**

Texto

Descripción generada automáticamente

**Código Verilog Testbench**

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

**Descripción RTL en Vivado 2022.2**

**Gráfico, Diagrama

Descripción generada automáticamente** Podemos observar cómo utilizó un multiplexor para la implementación del multiplexor 2 a 1.

**Resultado de la simulación:**

**Interfaz de usuario gráfica

Descripción generada automáticamente**

Cuando i\_Select = 00, la salida o\_Data es i\_Data2

Cuando i\_Select = 01, la salida o\_Data es i\_Data1

Como se observa en la simulación, la salida o\_Data recibe el valor de entrada i\_DataX según la expresión asignada en el selector i\_Select.

o\_Data = i\_Select ? i\_Data1 : i\_Data2;

o\_Data = 00 ? i\_Data1 (true) : i\_Data2 (false); 00 -> false

o\_Data = i\_Data2

**Propuesta 3**

**Código Verilog**

Imagen que contiene Texto

Descripción generada automáticamente

**Código Verilog Testbench (Con función $random)**

Interfaz de usuario gráfica, Texto, Correo electrónico

Descripción generada automáticamente

Escala de tiempo

Descripción generada automáticamente

**Descripción RTL en Vivado 2022.2**Diagrama, Esquemático

Descripción generada automáticamente

Podemos observar cómo utilizó multiplexores para la implementación del multiplexor 4 a 1.

**Resultado de la simulación:**

Pantalla de un video juego

Descripción generada automáticamente con confianza media

Cuando i\_Select = 10, la salida o\_Data es i\_Data3

Cuando i\_Select = 00, la salida o\_Data es i\_Data1

Como se observa en la simulación, la salida o\_Data recibe el valor de entrada i\_DataX según el valor indicado en el selector i\_Select.

**Propuesta 4 Operadores Lógicos**

**Código Verilog**

**Tabla

Descripción generada automáticamente**

**Código Verilog Testbench**

Interfaz de usuario gráfica, Texto, Correo electrónico

Descripción generada automáticamente Texto

Descripción generada automáticamente

**Descripción RTL en Vivado 2022.2**

Diagrama, Esquemático

Descripción generada automáticamente

Podemos observar cómo utilizó las compuertas lógicas AND y OR para la implementación del multiplexor 4 a 1.

**Resultado de la simulación:**

Pantalla de un video juego

Descripción generada automáticamente con confianza media

Cuando i\_Select = 10, la salida o\_Data es i\_Data3

Cuando i\_Select = 00, la salida o\_Data es i\_Data1

Como se observa en la simulación, la salida o\_Data recibe el valor de entrada i\_DataX según el valor indicado en el selector i\_Select.

**Conclusión**

El término lógica de circuito integrado de escala media (MSI) se refiere a un conjunto de circuitos lógicos combinacionales básicos que implementan funciones simples y de uso común, como multiplexores. La lógica MSI también puede incluir operaciones como comparadores y circuitos aritméticos simples.

Si bien un circuito lógico MSI puede tener múltiples salidas, cada salida requiere su propia expresión lógica única que se basa en las entradas del sistema.

Un multiplexor es un sistema que tiene **una salida** y múltiples entradas. En un momento dado, una y solo una entrada se enruta a la salida en función del valor en un conjunto de *líneas seleccionadas*. Para n líneas seleccionadas, un multiplexor puede admitir 2n entradas.

Los HDL son particularmente útiles para describir la lógica MSI debido a su capacidad de modelado abstracto. Mediante el uso de condiciones booleanas y asignaciones de vectores, el comportamiento de la lógica MSI se puede modelar de forma compacta e intuitiva.

**Referencia**

1. Brock J. LaMeres, Introduction to Logic Circuits & Logic Design with Verilog, Springer, 1st Edition, USA, 2017.