

Introducción a la simulación con Quartus II

Introducción al Diseño Lógico (E0301)
Ing. en Computación

14/05/2025

Introducción

- En esta clase se presentará la utilización del software Quartus II para la simulación y verificación de circuitos lógicos.
- A partir de un ejemplo de la práctica aprenderemos a:
 - **Crear** un proyecto.
 - **Representar** un circuito lógico.
 - **Verificar** el comportamiento temporal del mismo.
 - **Generar** un nuevo componente para ser reutilizado en otros diseños.

Software Quartus II

- Este software fue creado por Altera Corp. (luego parte de Intel).
- Quartus II permite realizar el análisis y síntesis de diseños lógicos, con el objetivo final de programar el resultado en un dispositivo FPGA.
- Utilizaremos la versión 13.1.0 Web edition, disponible en forma gratuita en el [sitio web](#) de Intel:
 - Para sist. op. Windows: multiple download -> `Quartus-web-13.1.0.162-windows.tar`
 - Para Linux ([link](#)): multiple download -> `Quartus-web-13.1.0.162-linux.tar`

Circuito multiplicador binario de dos bits

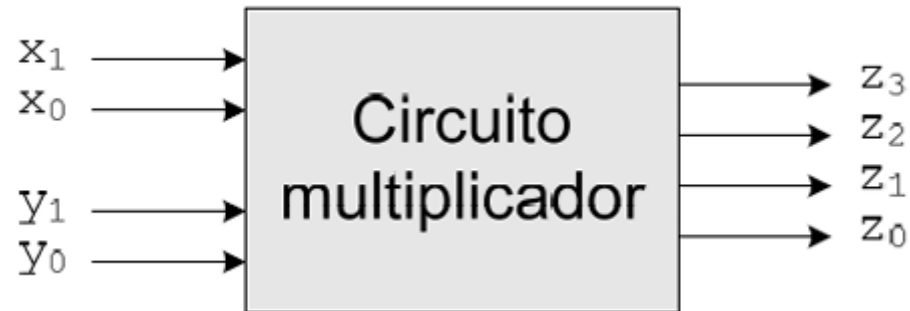
- Tomaremos como ejemplo el ejercicio 5 del TP 1b:

Diseñe un circuito multiplicador binario de números de dos bits

$$X = x_1 \ x_0$$

$$Y = y_1 \ y_0$$

El resultado debe expresarse en 4 bits: $Z = z_3 \ z_2 \ z_1 \ z_0$



Circuito multiplicador binario de dos bits (cont.)

Con números de dos bits, los datos de entrada serán 0, 1, 2, 3. El resultado de la multiplicación puede tomar los valores 0, 1, 2, 3, ..., 9 (la salida requiere 4 bits).

A partir de las tablas de verdad, sintetizar las salidas: z_3 z_2 z_1 z_0

Circuito multiplicador binario de dos bits (cont.)

x1	x0	y1	y0	z3	z2	z1	z0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

z3 = ...

z2 = ...

z1 = ...

z0 = ...

Simulación en Quartus II: hoja de ruta

1. Crear un Proyecto
2. Implementar el multiplicador de dos bits
3. Simular y verificar el funcionamiento del circuito
4. Crear un nuevo componente (mult2bits)
5. Implementar un multiplicador de números enteros de dos bits X e Y representados en complemento a 2 aprovechando el componente creado (ejercicio 6b).
6. Simular el circuito final.

Simulación en Quartus II: algunos tips

- Para verificar un circuito (que compila sin errores): **File->New->Verification / Debugging files -> University Program VWF.**
- Para agregar las señales: **Edit-> Insert -> Insert node or bus... -> Node finder -> List** y agregar las señales buscadas. Luego para agrupar entradas, seleccionarlas en la lista y hacer click con botón derecho **Grouping -> Group**
- Para crear un nuevo componente a partir de un diseño existente: **File-> Create/Update -> Create Symbol files for current file.** Se creará un archivo con extension .bsf. El nuevo componente estará disponible en **Symbol Tool -> Libraries: Project**
- Luego de realizar correcciones en un circuito se debe recompilar el proyecto. Si se tiene más de un archivo esquemático, se debe seleccionar el que se quiere compilar como Top-level Entity: en la Ventana **Project navigator -> solapa Files -> botón derecho sobre el archivo .bdf deseado -> Set as Top-Level Entity**

Simulación en Quartus II: algunos errores comunes

- La ruta completa donde se aloja el proyecto y los nombres de los archivos no puede contener espacios.
- Las instancias de los elementos en el esquemático no pueden tener nombres repetidos (en la versión 13.1 puede ocurrir que el programa erróneamente asigne el mismo número de instancia a dos objetos distintos).
- Los pines no pueden tener espacio en el nombre.
- No dejar pines sin conectar.

Circuito multiplicador binario de dos bits en complemento a 2

x1	x0	y1	y0	z3'	z2'	z1'	z0'
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	1	1	1	0
0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0
1	0	0	1	1	1	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	0	1	0
1	1	0	0	0	0	0	0
1	1	0	1	1	1	1	1
1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1

Ahora los números de entrada pueden ser -2,-1,0,1.

- Una opción es realizar la tabla de verdad y sintetizar las salidas del circuito $z3' z2' z1' z0'$ (ver ejercicio 6a).

- En el ejercicio 6b se propone reutilizar el diseño del multiplicador de números sin signo del ejercicio 5. Para esto:

¿Que similitudes hay entre las salidas de los multiplicadores? ¿Cómo podríamos aprovechar los resultados anteriores ($z3 z2 z1 z0$) y el componente que ya creamos? Ver siguiente filmina.

Circuito multiplicador binario de dos bits en complemento a 2

Comparación de las
tablas de verdad:

x1	x0	y1	y0	z3	z2	z1	z0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Multiplicador sin signo

x1	x0	y1	y0	z3'	z2'	z1'	z0'
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	1	1	1	0
0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0
1	0	0	1	1	1	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	0	1	0
1	1	0	0	0	0	0	0
1	1	0	1	1	1	1	1
1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1

Multiplicador con signo