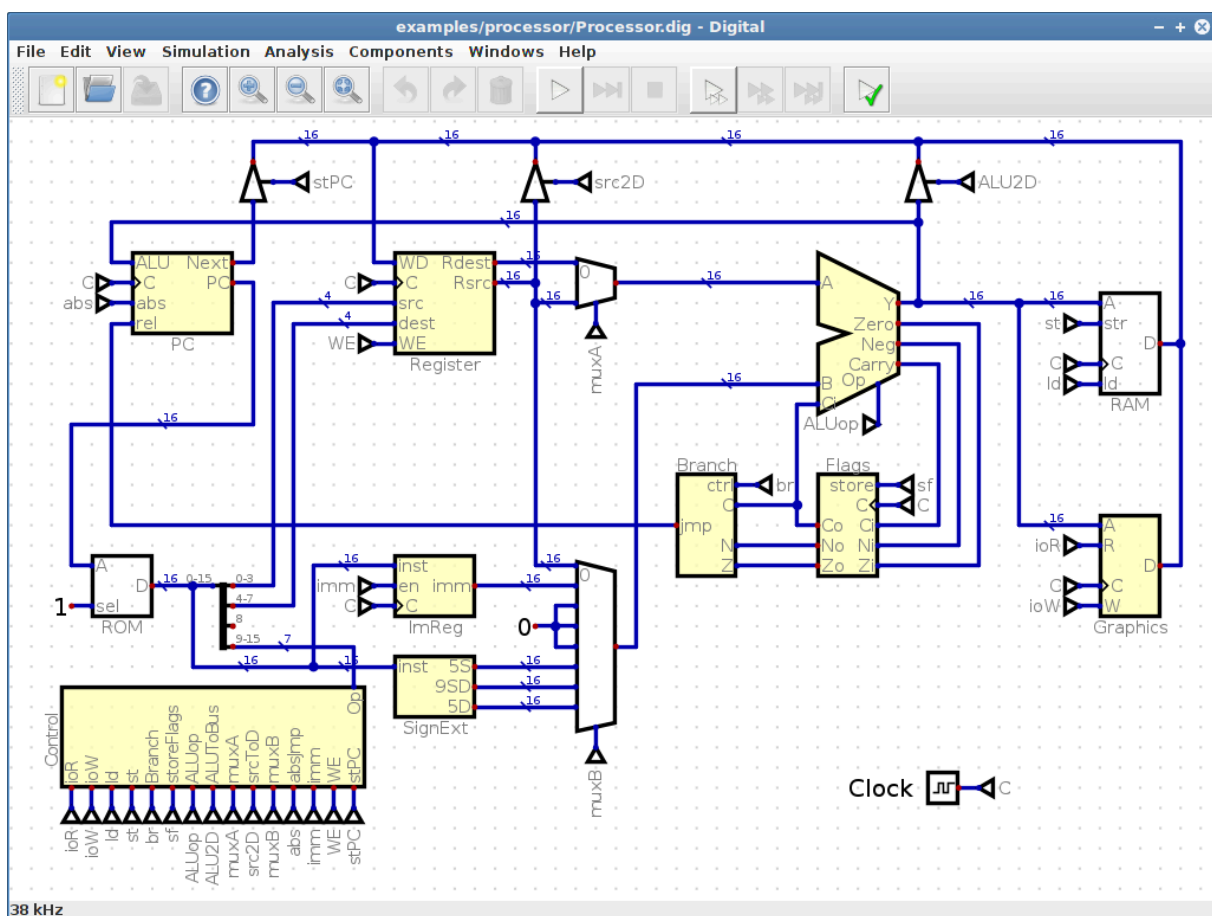


# Digital



Revisión: v0.31  
 Fecha: 2024-09-03 12:57

# Índice

## A General

1. Digital .....	6
1.1. Introducción .....	6
1.2. Primeros pasos .....	6
1.3. Cables .....	14
1.4. Diseño jerárquico .....	14
2. Simulación .....	18
2.1. Retardo de propagación .....	18
3. Análisis y síntesis de circuitos .....	18
4. Hardware .....	19
4.1. GAL16v8 y GAL22v10 .....	19
4.2. ATF150xAS .....	19
4.3. Exportación a VHDL o Verilog .....	19
5. Formas personalizadas .....	20
6. Preguntas habituales(FAQ) .....	20
7. Atajos de teclado .....	23

## B Ajustes

## C Interfaz de línea de comandos

## D Componentes

1. Lógica .....	
1.1. AND .....	29
1.2. NAND .....	29
1.3. OR .....	30
1.4. NOR .....	31
1.5. XOR .....	31
1.6. XNOR .....	32
1.7. NOT .....	33
1.8. Tabla de Consulta (LUT) .....	33
2. Entrada-Salida .....	
2.1. Salida .....	34
2.2. LED .....	35
2.3. Entrada .....	35
2.4. Señal de reloj .....	36
2.5. Pulsador .....	37
2.6. Interruptor DIP .....	37
2.7. Sonda .....	38
2.8. Gráfica de datos .....	38
2.9. Gráfica de datos activos .....	39
3. Entrada-Salida - Displays .....	
3.1. LED RGB .....	39
3.2. LED con dos conexiones .....	40
3.3. Pulsador con LED .....	40
3.4. Display de 7 segmentos .....	41
3.5. Display de 7 segmentos hexadecimal .....	42
3.6. Display de 16 segmentos .....	42

3.7. Bombilla .....	43
3.8. Matriz de ledes .....	43
4. Entrada-Salida - Mecánica .....	
4.1. Encoder rotatorio .....	44
4.2. Motor paso a paso, unipolar .....	45
4.3. Motor paso a paso, bipolar .....	45
5. Entrada-Salida - Periféricos .....	
5.1. Teclado .....	46
5.2. Terminal .....	47
5.3. Telnet .....	47
5.4. Monitor VGA .....	48
5.5. MIDI .....	49
6. Cables .....	
6.1. Tierra .....	50
6.2. Fuente de tensión .....	50
6.3. Valor constante .....	51
6.4. Túnel .....	51
6.5. Divisor/Agregador .....	51
6.6. Driver .....	52
6.7. Driver, selector invertido .....	53
6.8. Retardo .....	54
6.9. Resistencia pull-up .....	54
6.10. Resistencia pull-down .....	55
6.11. No conectado .....	55
7. Plexores .....	
7.1. Multiplexor .....	55
7.2. Demultiplexor .....	56
7.3. Decodificador .....	57
7.4. Selector de bit .....	57
7.5. Codificador con prioridad .....	58
8. Flip-Flops .....	
8.1. Flip-Flop RS .....	58
8.2. Flip-Flop RS con reloj .....	59
8.3. Flip-Flop JK .....	60
8.4. Flip-Flop tipo D .....	61
8.5. Flip-Flop tipo T .....	62
8.6. Flip-Flop JK asíncrono .....	63
8.7. Flip-Flop tipo D, asíncrono .....	64
8.8. Monoestable .....	65
9. Memorias - RAM .....	
9.1. RAM, puertos separados .....	66
9.2. Bloque de RAM, puertos separados .....	67
9.3. RAM, puerto bidireccional .....	67
9.4. RAM, selecciona chip .....	68
9.5. Bloque de registro .....	69
9.6. RAM, puerto dual .....	70
9.7. RAM asíncrona .....	71
9.8. RAM gráfica .....	72
10. Memorias - EEPROM .....	
10.1. EEPROM .....	73

10.2. EEPROM, puertos separados .....	74
11. Memorias .....	
11.1. Registro .....	75
11.2. ROM .....	76
11.3. Puerto dual ROM .....	77
11.4. Contador .....	78
11.5. Contador con preset .....	79
11.6. Generador de números aleatorios .....	81
12. Aritmética .....	
12.1. Sumador .....	81
12.2. Restar .....	82
12.3. Multiplicar .....	83
12.4. División .....	83
12.5. Registro de desplazamiento .....	84
12.6. Comparador .....	85
12.7. Negación .....	85
12.8. Extensor de signo .....	86
12.9. Contador de bits .....	86
13. Interruptores .....	
13.1. Conmutador .....	87
13.2. Conmutador de dos caminos .....	87
13.3. Relé .....	88
13.4. Relé de dos contactos .....	89
13.5. FET de canal P .....	90
13.6. FET de canal N .....	90
13.7. Fusible .....	91
13.8. Diodo polarizado .....	91
13.9. Diodo a masa .....	92
13.10. FET de canal P de compuerta flotante .....	93
13.11. FET de canal N de compuerta flotante .....	93
13.12. Puerta de transmisión .....	94
14. Varios .....	
14.1. Caso de prueba .....	94
15. Varios - Decoración .....	
15.1. Texto .....	95
15.2. Rectángulo .....	95
16. Varios - Genérica .....	
16.1. Inicialización genérica .....	96
16.2. Código .....	96
17. Varios - VHDL/Verilog .....	
17.1. Externo .....	97
17.2. Archivo externo .....	98
17.3. Control de pin .....	99
18. Varios .....	
18.1. Voltaje .....	99
18.2. Divisor bidireccional .....	100
18.3. Reset .....	100
18.4. Pausa .....	101
18.5. Parada .....	101
18.6. Temporizador asíncrono .....	102

## E Biblioteca

## A General

### 1. Digital

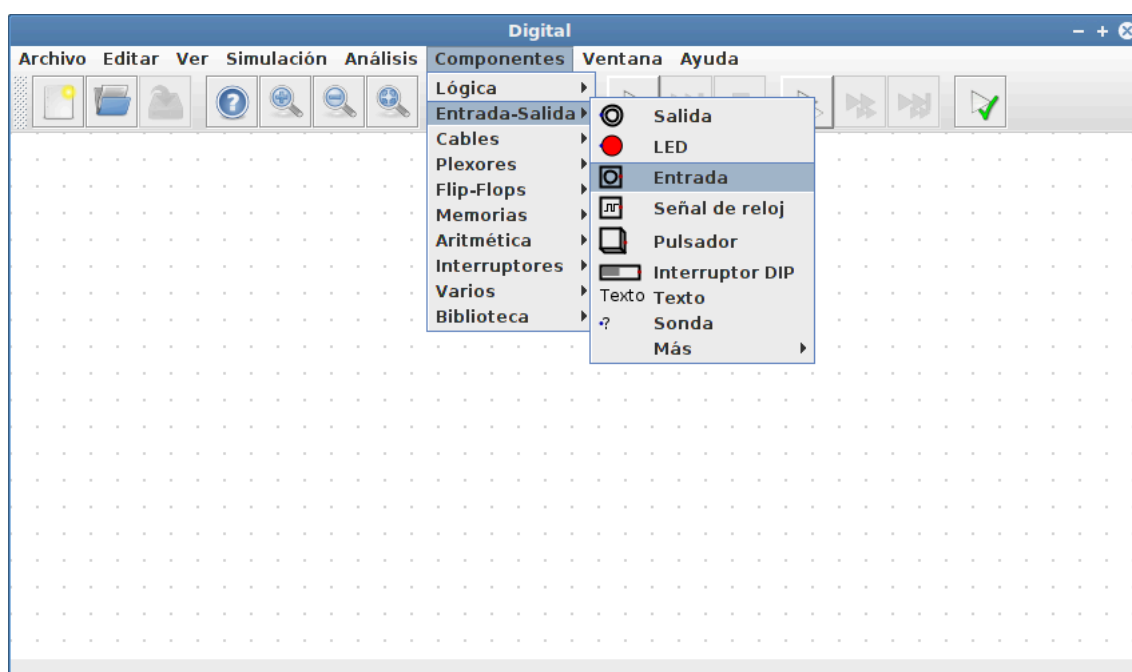
#### 1.1. Introducción

Digital es un simulador sencillo usado para simular circuitos digitales. Las puertas lógicas se conectan entre sí con cables, y se puede simular el comportamiento global del circuito. El usuario puede interactuar con la simulación presionando botones o fijando valores a las entradas del circuito.

Se pueden construir y simular la mayoría de los circuitos empleados en electrónica digital. En la carpeta *ejemplos*, el usuario puede navegar entre ejemplos que incluyen un procesador Harvard de 16 bits de ciclo sencillo.

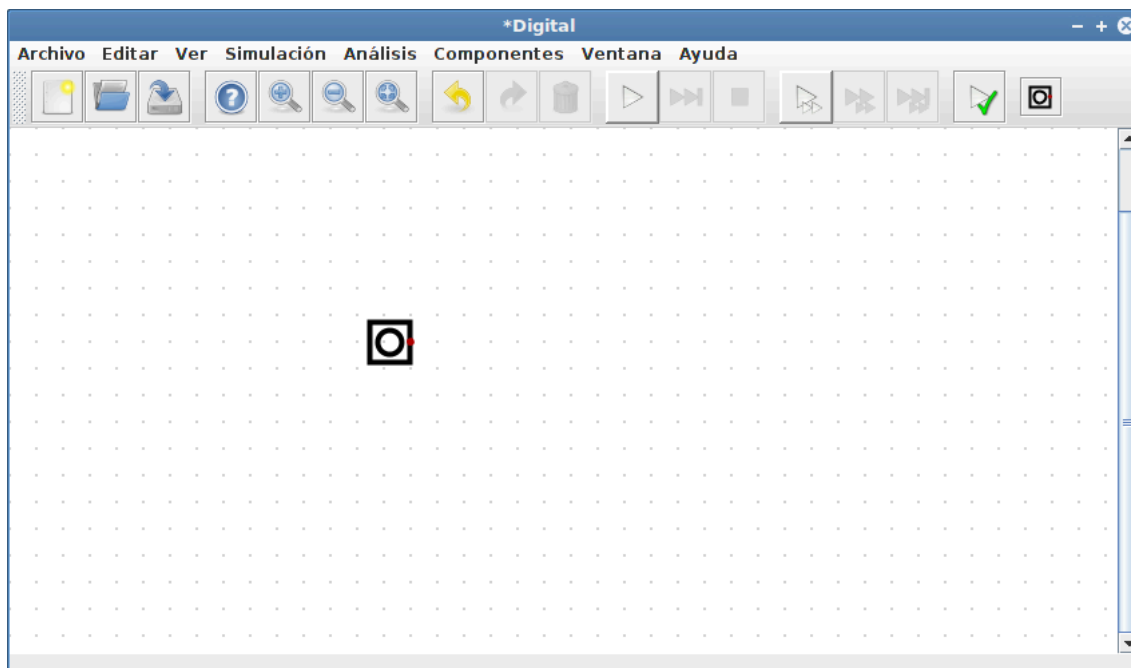
El simulador tiene dos modos de funcionamiento: Modo Edición y Modo Simulación. En el Modo Edición, pueden hacerse modificaciones al circuito. El usuario puede añadir o conectar componentes. En este modo, la simulación no es disponible. El Modo Simulación se activa presionando el botón *Inicio de la simulación* de la barra de herramientas. Cuando empieza la simulación, se prueba la consistencia del circuito. Si hay errores en el circuito, se mostrará un mensaje sobre el mismo, y los componentes afectados o cables se destacarán. Si el circuito no tiene errores, la simulación se habilitará. Entonces se podrá interactuar con la simulación que está en marcha. En el Modo Simulación no es posible modificar el circuito. Para eso, habrá que activar el Modo Edición de nuevo, parando la simulación.

#### 1.2. Primeros pasos

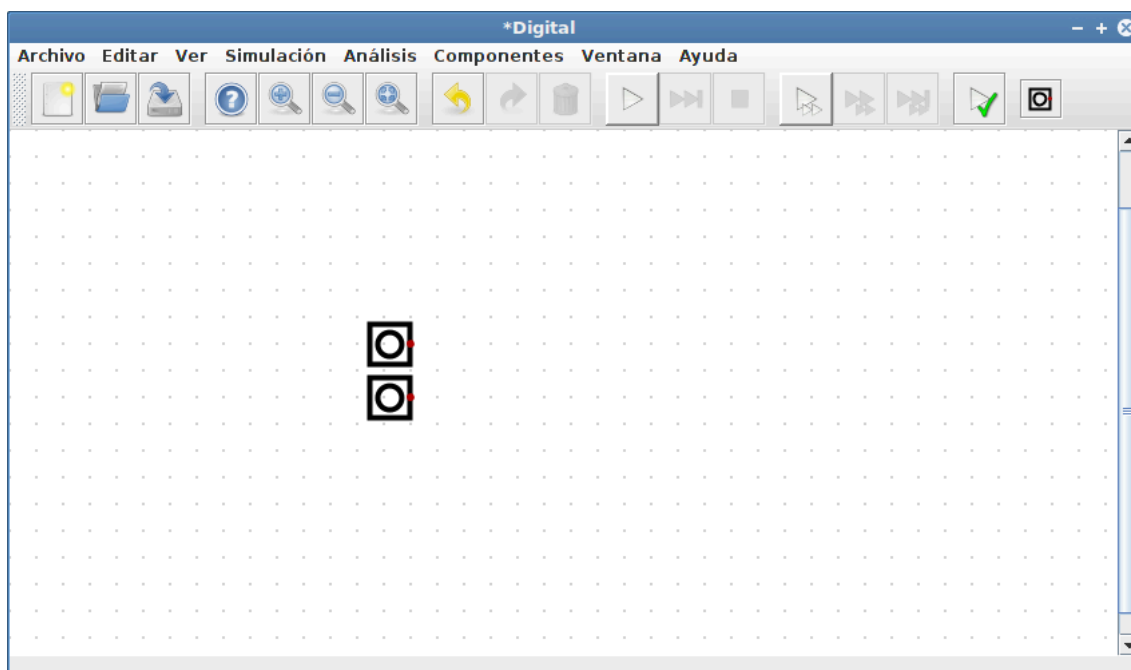


Como primer ejemplo, vamos a construir un circuito con una puerta OR-Exclusiva. De la ventana principal, el menú *Componentes* te permite seleccionar los diversos componentes. Una vez seleccionados, se colocan en el área de trabajo. Este proceso puede cancelarse presionando

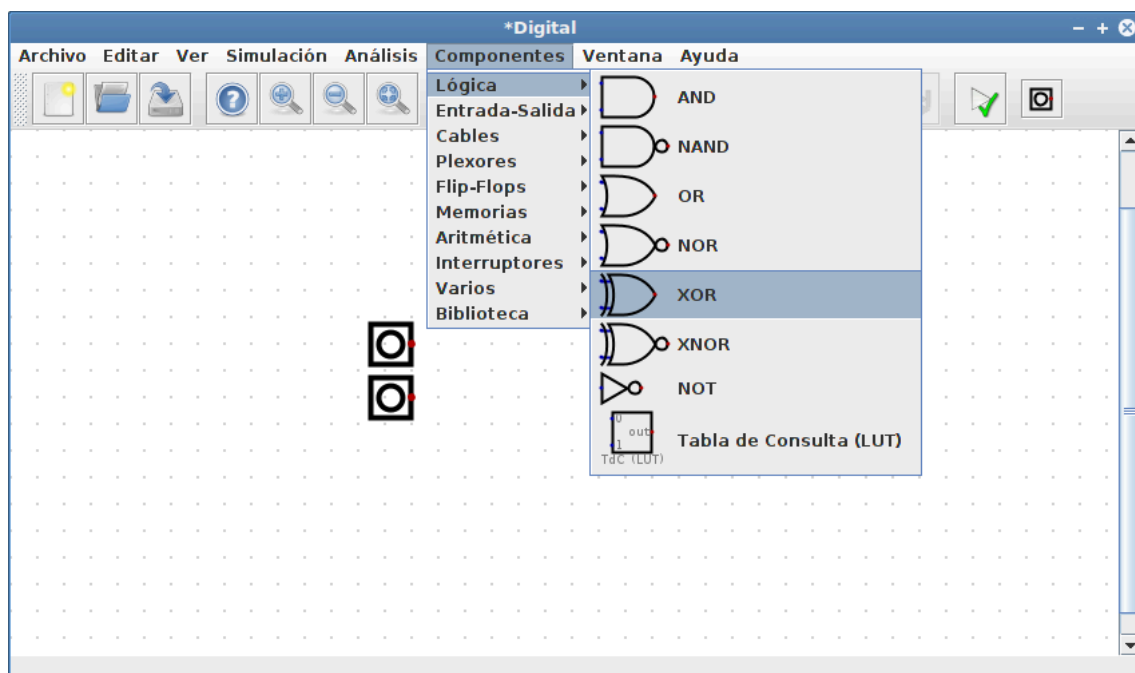
la tecla ESC en cualquier momento. Comienza por seleccionar un componente de entrada. Esto puede controlarse después interactivamente usando el ratón.



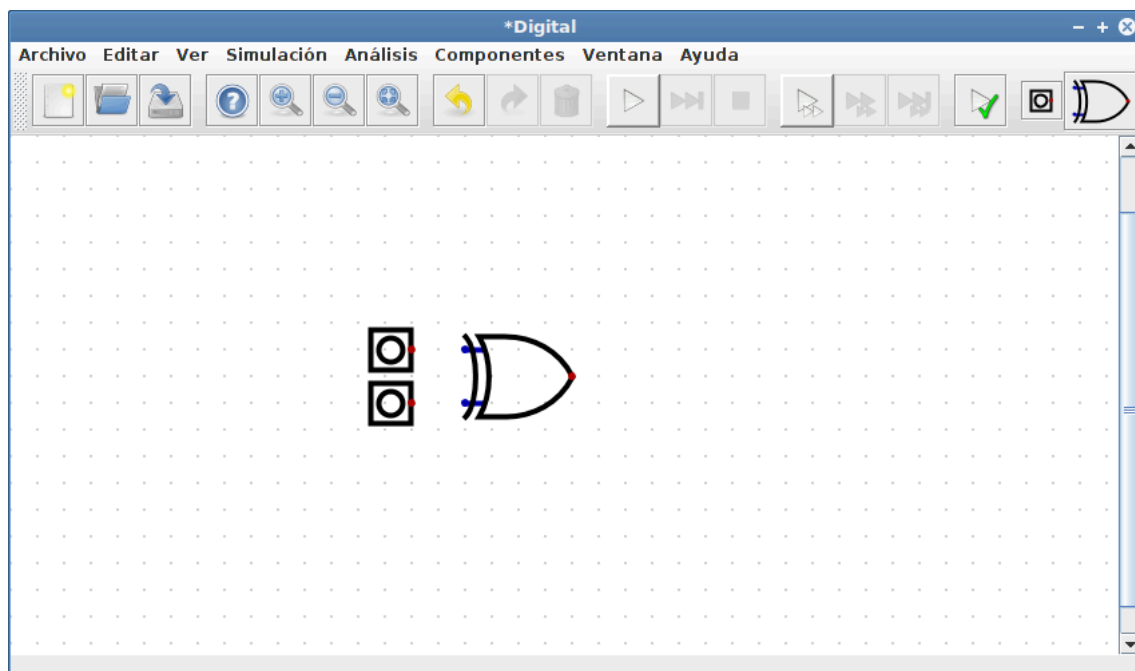
Después de la selección, la primera entrada puede ser colocada en el área de trabajo. El punto rojo del símbolo del componente de entrada es un punto de conexión entre el componente y un cable, que conectaremos luego. El color rojo indica una salida. Esto quiere decir que el puerto define un valor de señal o que contiene un cable.



Igualmente, añadimos una segunda entrada. Es mejor colocarla directamente debajo de la primera.

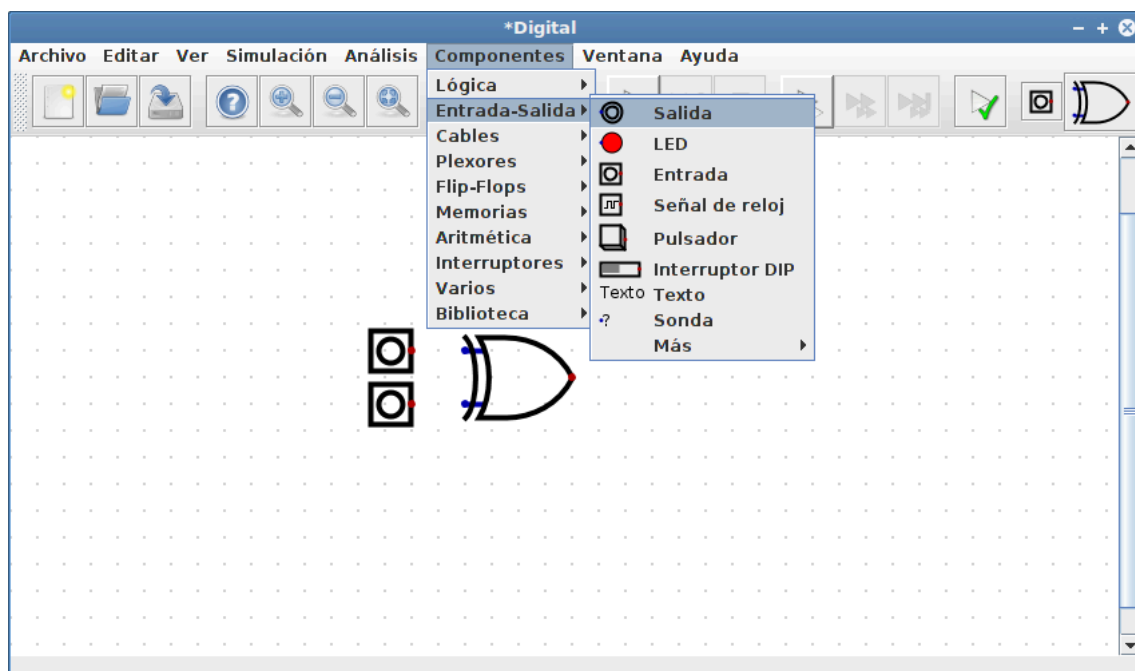


Después de añadir las entradas, selecciona la puerta OR-Exclusiva. Esta función representa la función lógica actual.

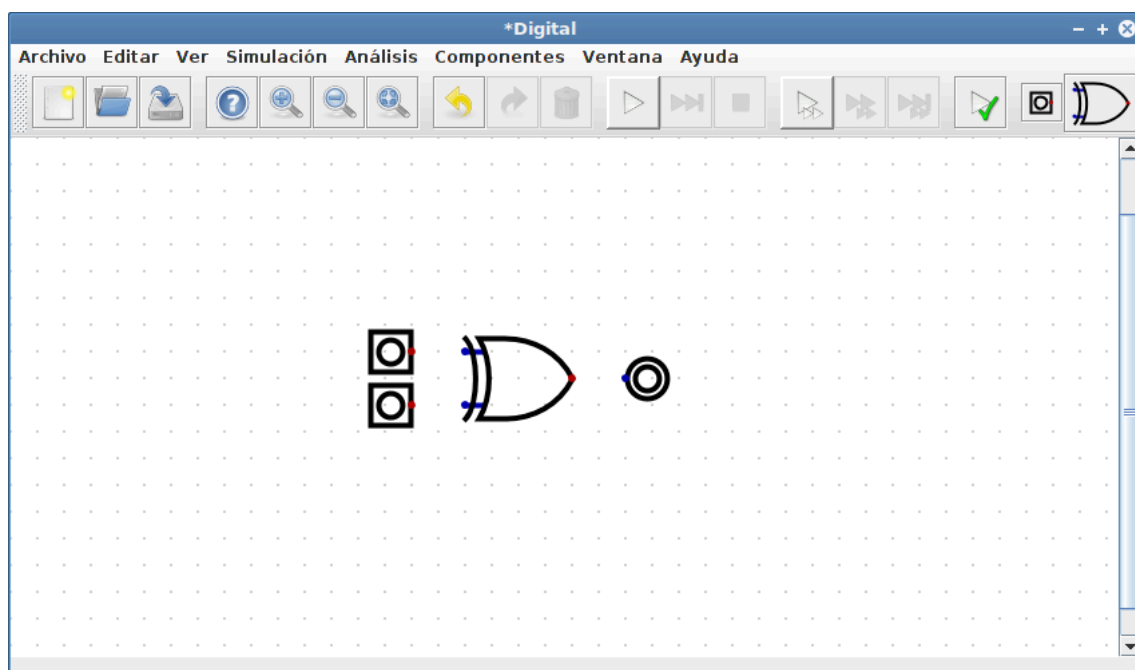


Ya se puede añadir esta puerta al circuito. Es mejor colocarla de forma que el cableado se haga lo más sencillo posible. Los puntos azules indican los terminales de entrada de la puerta.

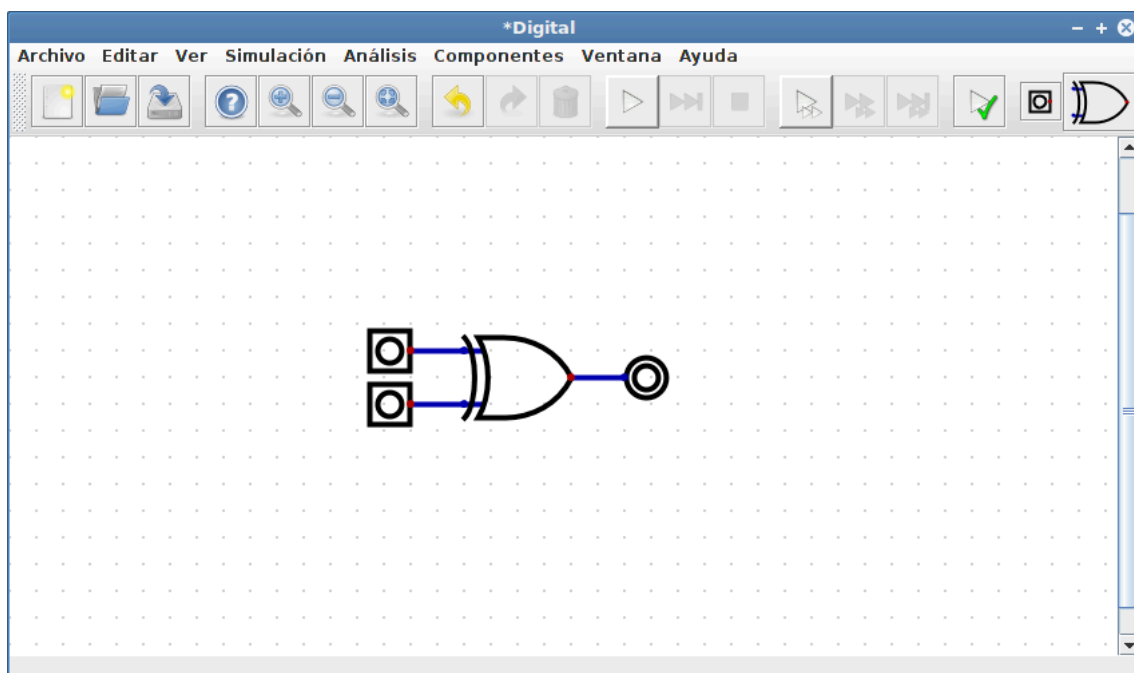




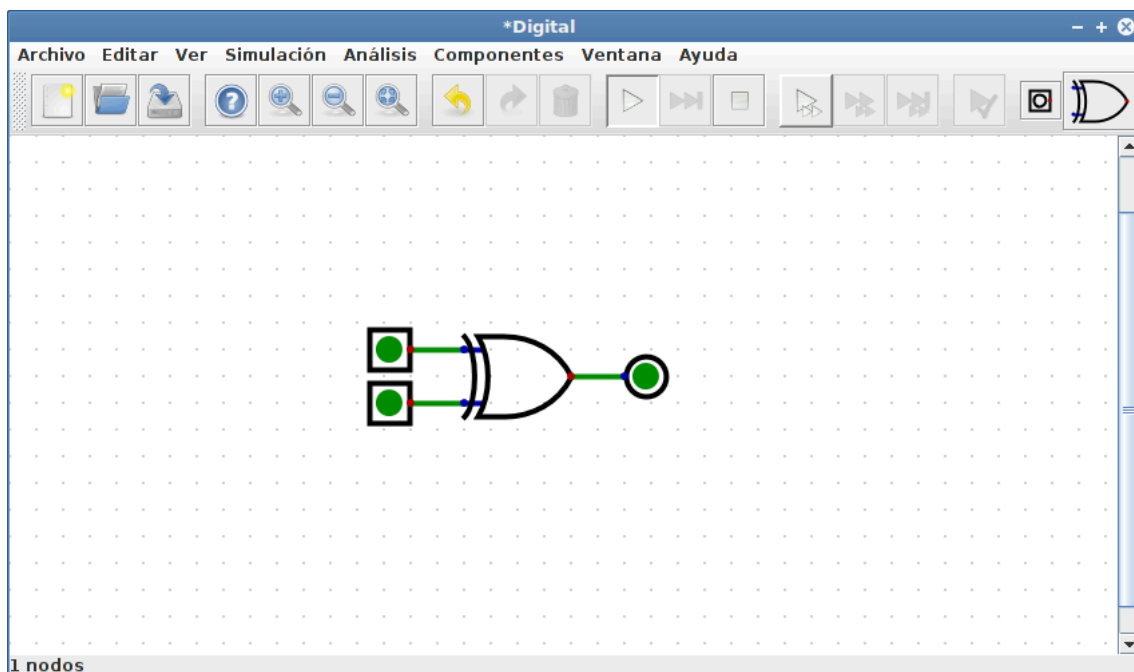
Ahora, selecciona una salida que pueda usarse para mostrar el estado de una señal o, en otro caso, pasar señales a un circuito incrustado.



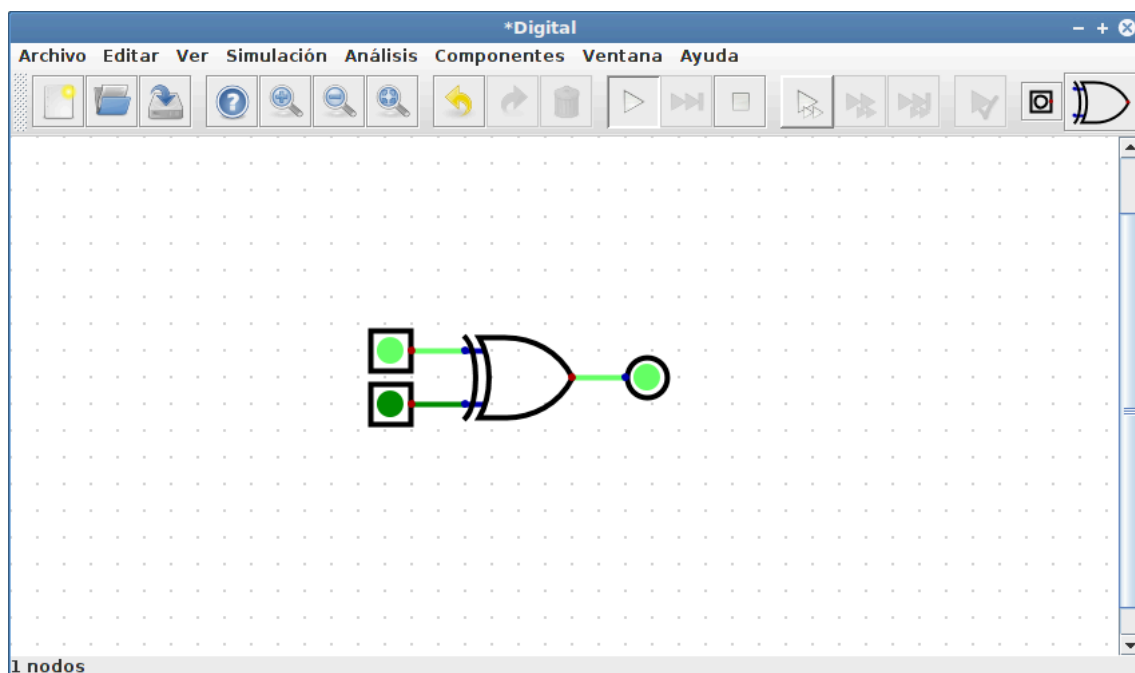
Se ha colocado de una forma en la que se puede cablear fácilmente. La salida tiene un punto azul, que indica un terminal de entrada. Aquí puedes alimentar el valor que se exportará.



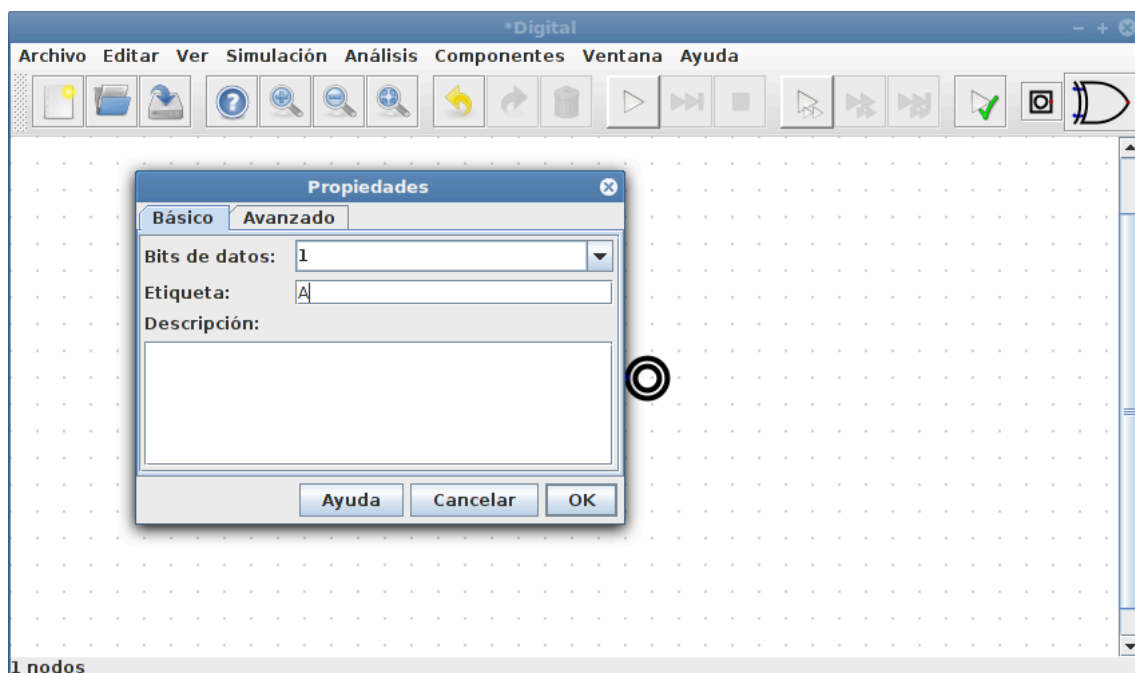
Después de que todos los componentes se haya seleccionado y colocado, usa el ratón para cablear una conexión entre los puntos azules y rojos. Asegúrate que exactamente un punto rojo se conecta con cualquier número de puntos azules. Sólo el uso de salidas de tres estados hace posible cambiar esta regla e interconectar varios puntos rojos. Si se han dibujado todos los cables, el circuito estará completo.



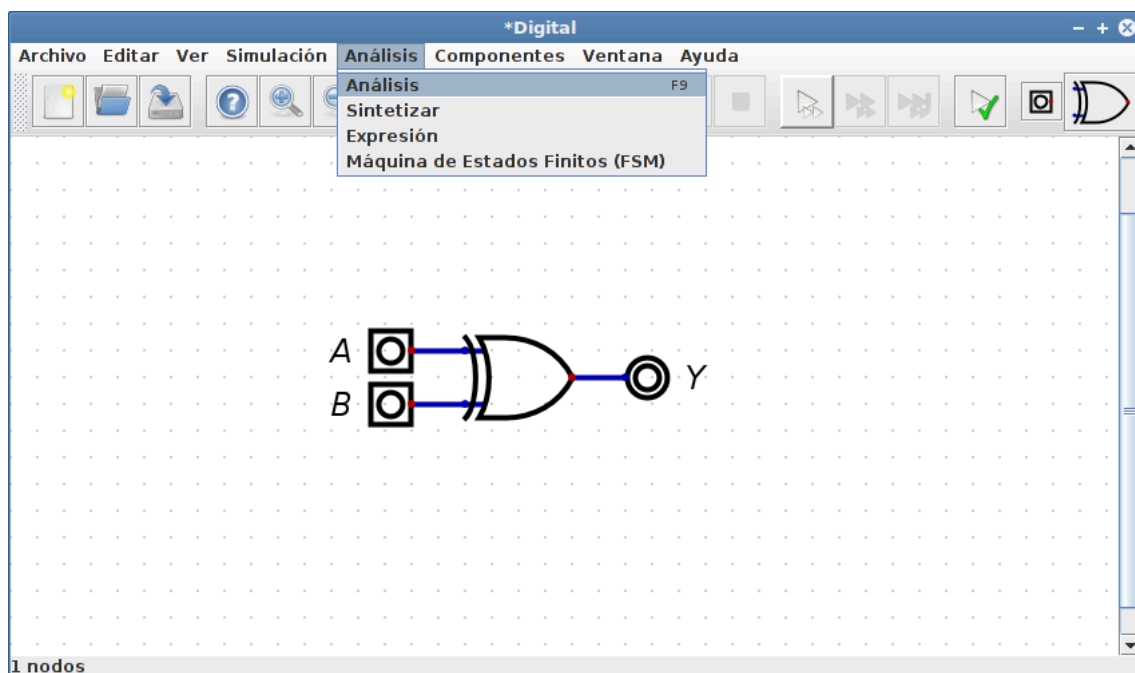
Cuando la simulación empieza, se puede interaccionar con el circuito. La simulación se pondrá en marcha con el botón correspondiente de la barra de herramientas. Después de empezar la simulación, el color de los cables cambiará, y las entradas y salidas se verán ahora rellenas. El color verde brillante indicará un '1' lógico y el verde oscuro un '0' lógico. En la figura de arriba, todos los cables tienen valor '0'.



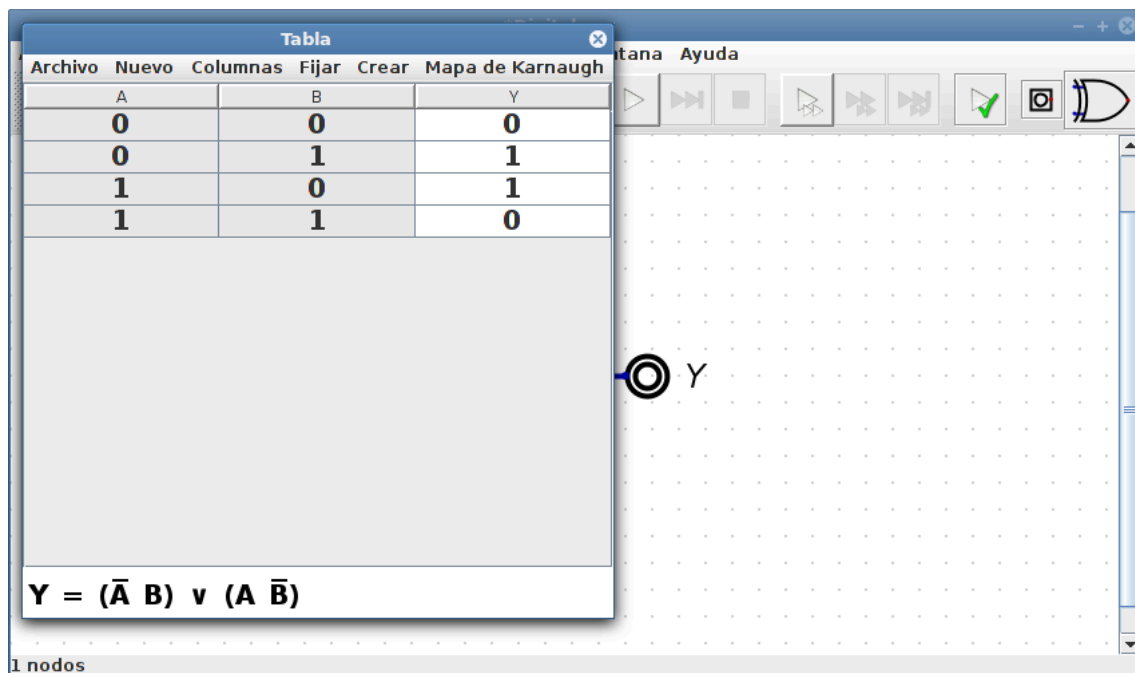
Haciendo clic con el ratón podremos cambiar los valores de las entradas. Puesto que la simulación está ahora activa, la salida cambia de acuerdo con lo que haya en las entradas. El circuito se comporta como una puerta OR-Exclusiva, como esperábamos.



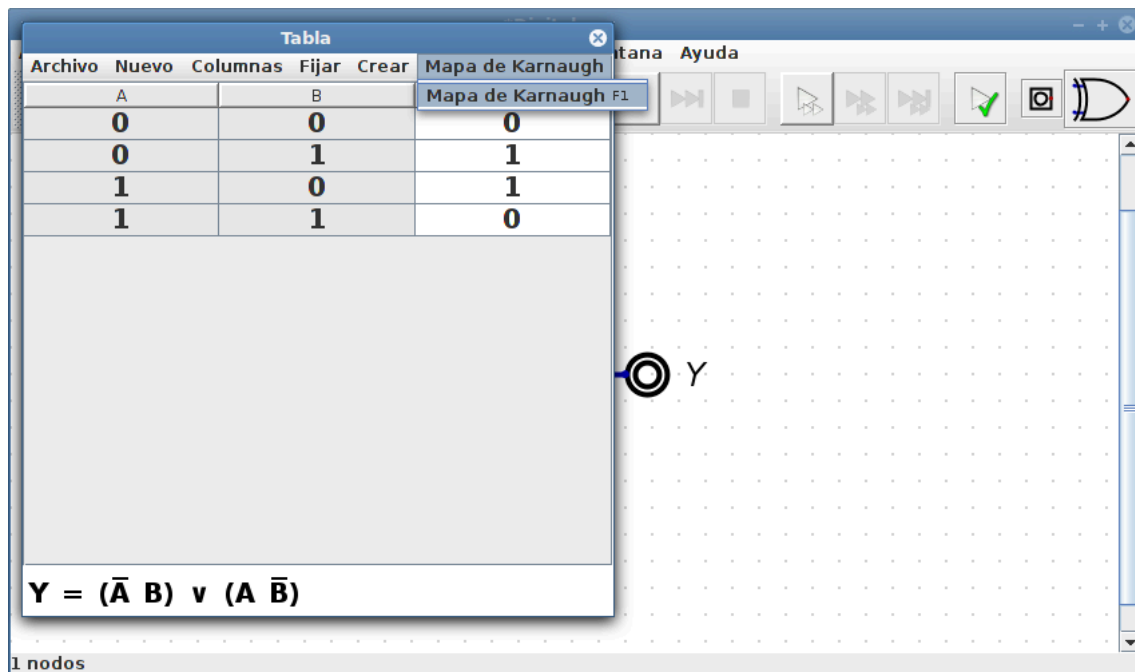
Para seguir trabajando con el circuito tendremos que parar la simulación. La forma más sencilla es con el botón "Para la simulación" que está en la barra de herramientas. Haciendo clic derecho con el ratón abriremos un diálogo que muestra las propiedades del componente. Con esto, por ejemplo, podremos fijar la etiqueta 'A' para la primera entrada.



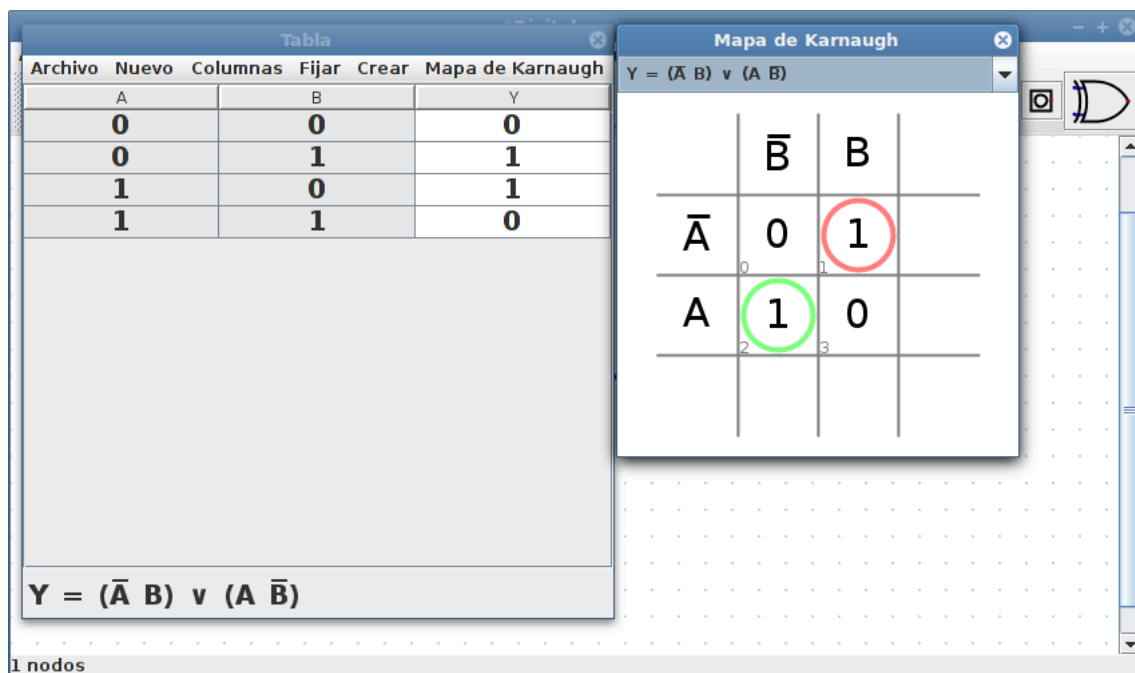
De este modo, las etiquetas que queden podrán configurarse adecuadamente. El menú *Análisis* contiene un comando (*Análisis*) que hace una análisis de nuestro circuito. Sin embargo, lo hará sólo si todas las entradas y salidas están correctamente etiquetadas.



La tabla de verdad del circuito simulado aparece en una nueva ventana. Bajo la tabla se puede encontrar la expresión algebraica asociada al circuito. Si hay varias expresiones algebraicas alternativas, se abrirá una ventana que muestra todas las expresiones.



El diálogo de tabla tiene la menú entrada *Mapa de Karnaugh* en su menú principal. Esto permite mostrar la tabla de verdad como un mapa de Karnaugh.



En la parte superior de este diálogo hay una lista desplegable que permite la selección de la expresión. De esta forma puedes, por ejemplo, mostrar cómo pueden aparecer diferentes expresiones algebraicas para la misma función. Sin embargo, en este ejemplo hay una sola expresión mínima. La tabla de verdad puede modificarse también haciendo clic en el mapa de Karnaugh.

### 1.3. Cables

Todos los componentes tienen que estar conectados mediante cables. No es posible conectar dos componentes poniendo uno al lado del otro.

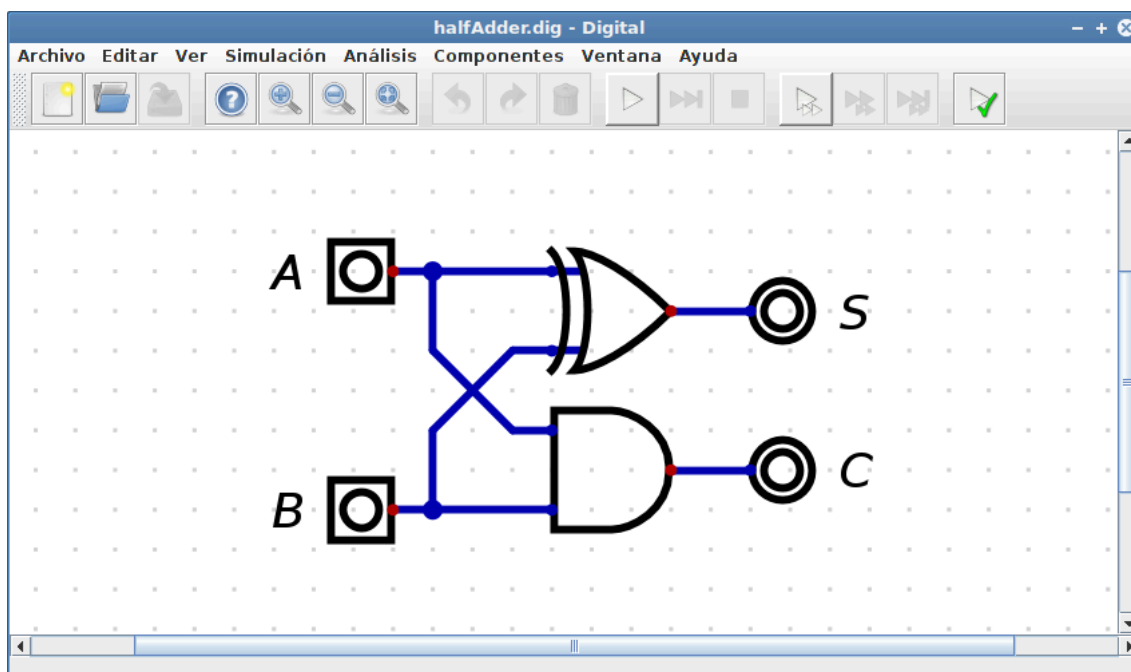
Además, sólo hay conexiones entre el punto final de un cable y un componente. Si un pin del componente se coloca en medio de un cable, no habrá conexión entre el componente y el cable. Por tanto, un cable debe terminar en cada pin que vaya a conectar. Incluso si se utiliza el componente "Túnel", tiene que haber un cable entre el pin y el componente "Túnel".

Para mover un componente, tiene que seleccionarse con la herramienta de selección rectangular, incluyendo los cables que tenga conectados. Para mover un componente sin los cables que tiene conectados, basta con seleccionar el componente haciendo clic sobre él (y quedará "pegado" al puntero del ratón).

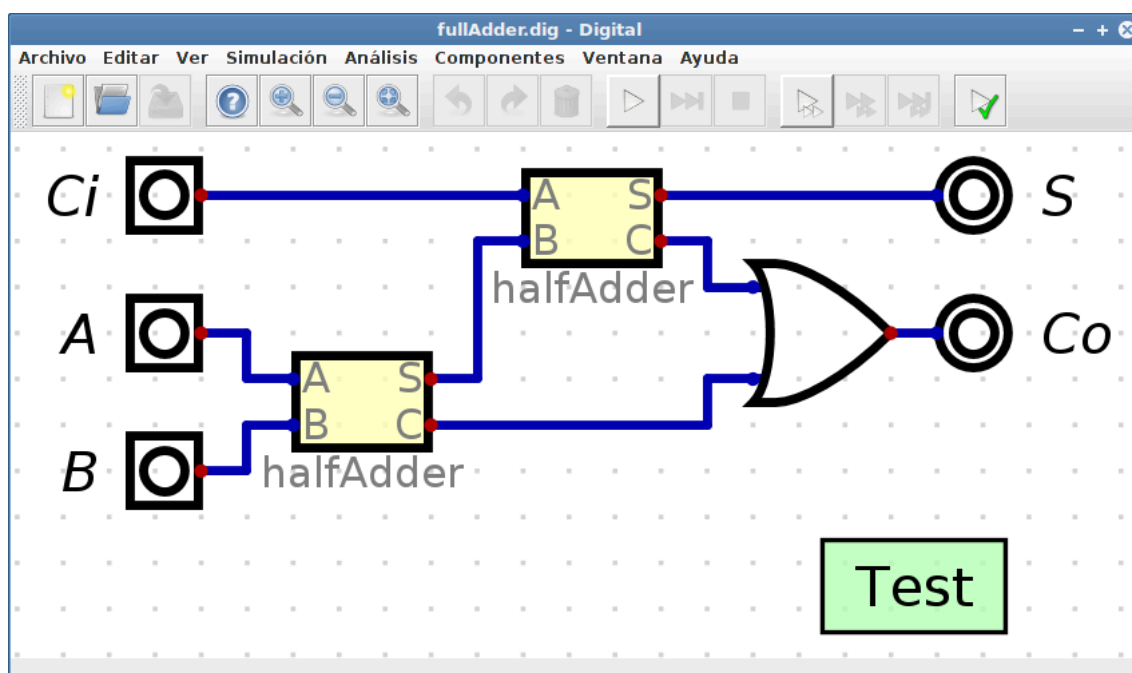
Con CTRL-clic, un trozo de cable puede seleccionarse para moverlo o borrarlo. Si la tecla D se presiona al dibujar un cable, podrá dibujarse un segmento diagonal. La tecla S permite dividir un segmento en dos. Para hacer esto, pon el puntero del ratón sobre dicho segmento y pulsa la letra. Desplázate hasta donde desees.

### 1.4. Diseño jerárquico

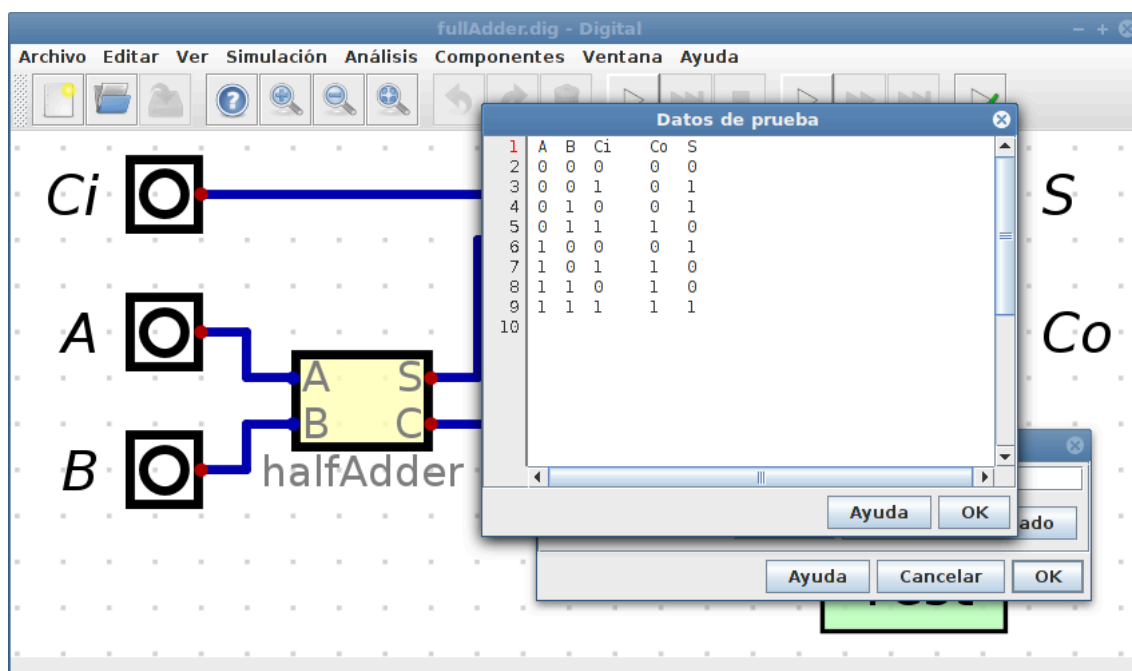
Si hay que diseñar un circuito complejo, esto puede hacerse bastante complicado. Para seguir la pista a las diferentes partes de un circuito, éstas pueden diseñarse en diferentes archivos. Esto permite también usar un "subcircuito", que ya se creó para una finalidad, varias veces en sucesivos circuitos. Esta forma de trabajar ofrece también la ventaja de que los archivos se pueden almacenar de forma independiente en un sistema de control de versiones y los cambios se pueden rastrear.



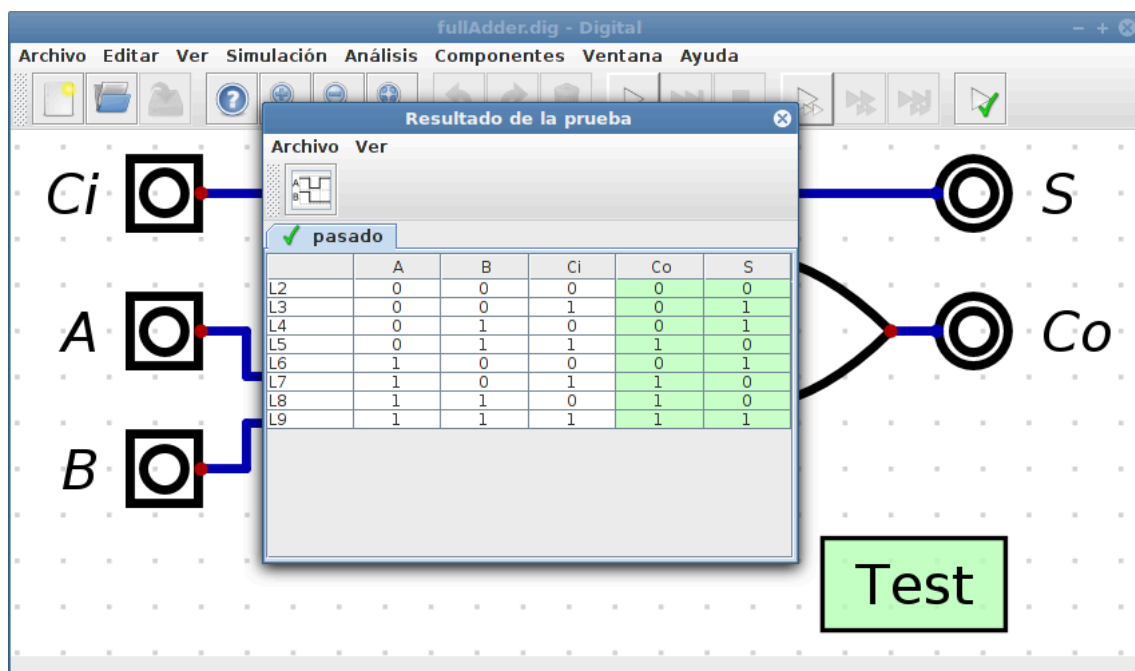
Como ejemplo, consideremos un sumador de 4 bits: primero construimos un semisumador. Éste consiste en una puerta XOR y una puerta AND. La suma de los dos bits 'A' y 'B' va a las salidas 'S' y 'C'. El circuito se almacena en el archivo *halfAdder.dig*.



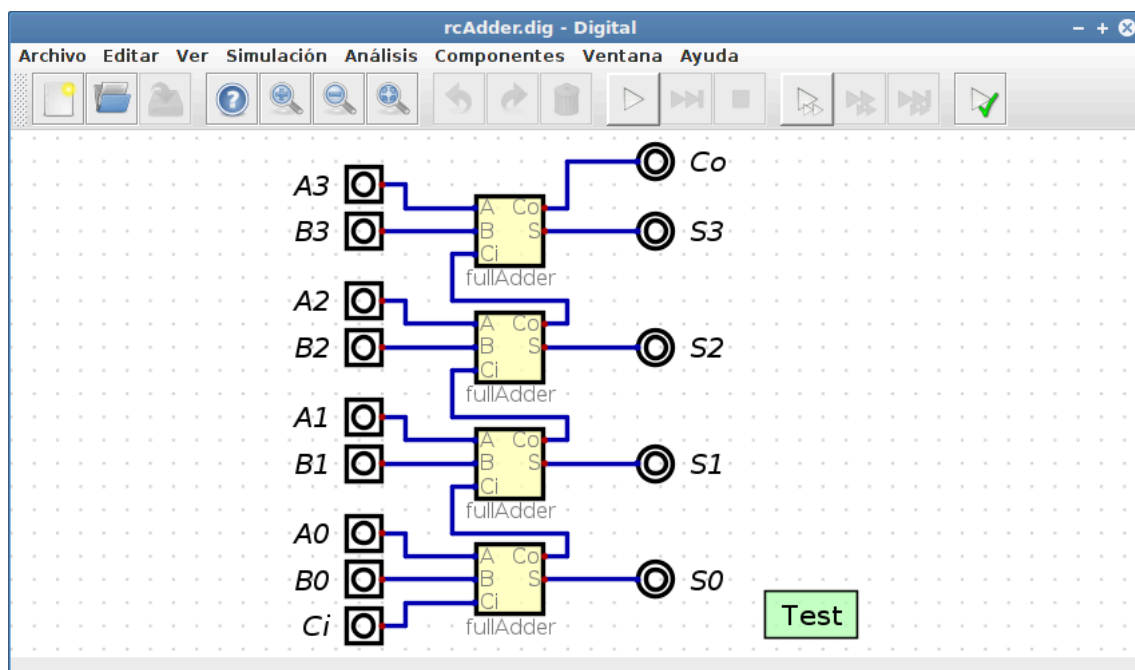
El sumador completo se puede construir a partir de dos semisumadores. Para ello, creamos un nuevo archivo vacío y lo guardamos como *fullAdder.dig* en la misma carpeta que el semisumador. Entonces el semisumador se puede añadir al nuevo circuito a través del menú *Componentes* → *Personalizado*. El orden de los pines en el encapsulado del semisumador puede ser reorganizado en el sumador en el menú *Editar* → *Ordenar las entradas* o *Editar* → *Ordenar las salidas*. El sumador completo suma los tres bits, 'A', 'B' y 'Ci', y da la suma por las salidas 'S' y 'Co'.



Para probar el correcto funcionamiento del sumador completo, deberíamos añadir un caso de prueba. En éste, se almacena una tabla de verdad que debería cubrir todos los posibles estados del circuito. Así puede automáticamente comprobarse si se cumple.

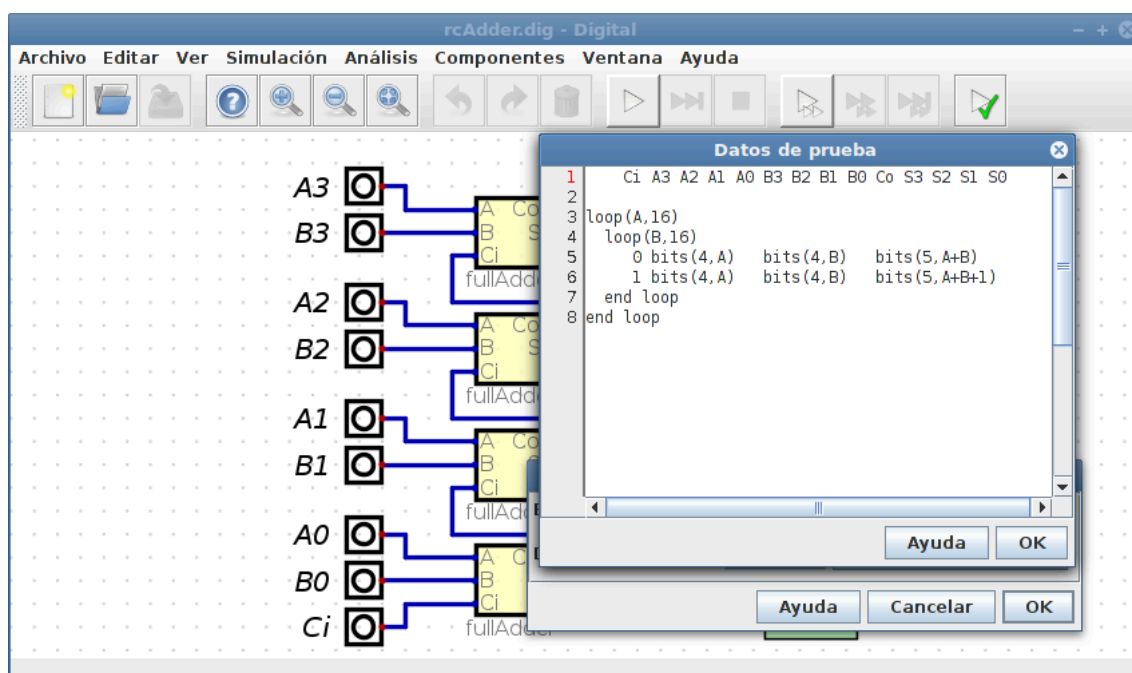


Los casos de prueba pueden ejecutarse a través del editor de casos de prueba o el botón casos de prueba de la barra de herramientas. Las celdas destacadas en verde indican que la salida del circuito encaja en la tabla de verdad que aparece en el caso de prueba.



Construido el sumador completo, pueden usarse varios para formar un sumador ripple-carry (sumador en cascada). En este caso, la salida de acarreo de una suma se usa como entrada de acarreo de la suma del siguiente bit, como hacemos habitualmente a mano. Este sumador de 4 bits debería ser probado, para lo que se introdujo el correspondiente caso de prueba.





Este caso de prueba realiza una comprobación al 100%, lo cual es posible sólo en circuitos relativamente sencillos: las 512 combinaciones de entrada se aplican al circuito, y se comprueba si la salida es la correcta. La primera línea muestra las señales de entrada y salida. Debajo se encuentran, en filas, los valores de entrada que se van a aplicar y los valores de salida que tienen que obtenerse, como en una tabla de verdad. En este ejemplo harían falta 512 líneas, lo cual convertiría la tarea en algo muy pesado y, muy probablemente, fuente de errores. Es más fácil y más seguro generar las líneas necesarias automáticamente. Para ello, se hace un barrido de las variables  $A$  y  $B$ , desde 0 hasta 15, y las respectivas parejas de valores se asignan a las entradas ' $A[n]$ ' y ' $B[n]$ '. Después se comprueba si el circuito da por salida  $A+B$ . Luego se hace la prueba con el bit de acarreo, en cuyo caso  $A+B+1$  debe resaltarse. Los detalles de la sintaxis del caso de prueba se dan en el diálogo de ayuda.

Si un circuito se incrusta en otro, sólo el nombre del archivo del subcircuito se almacena en el circuito, no el circuito incrustado. Los archivos correspondientes de los subcircuitos deben encontrarse en el sistema de archivos en el momento de la simulación. Para soportar las diferentes formas de trabajo de los usuarios y evitar una gestión compleja de las rutas de importación, etc., se ha implementado una estrategia de importación poco habitual.

Sólo los nombres de los archivos de los circuitos incrustados se almacenan en los archivos de los circuitos, no la ruta completa. Si un archivo necesita ser abierto, se busca en todas las subcarpetas el archivo del nombre correspondiente. Si se encuentra un archivo adecuado, se importa. Este procedimiento depende sólo del nombre del archivo que se quiere leer, no de su ruta. Como consecuencia, aparecerá un mensaje de error si hay varios archivos con el mismo nombre en diferentes subcarpetas, porque habrán aparecido ambigüedades.

Por tanto, una estructura de proyecto adecuada podría ser la siguiente: - El circuito básico se localiza en una carpeta aparte. - Todos los circuitos importados deberán estar en la misma carpeta o subcarpetas. Todos los circuitos deberán tener diferentes nombres, para que no hay circuitos del mismo nombre en diferentes carpetas.

## 2. Simulación

### 2.1. Retardo de propagación

Durante la simulación, cada puerta lógica tiene un retardo de propagación. Cada componente de la biblioteca tiene el mismo retardo de propagación independientemente de su complejidad. La puerta AND por tanto tiene el mismo retardo de propagación que el multiplicador. La única excepción son los diodos, conmutadores y separadores, que se emplean para crear buses de datos. Estos componentes no tienen retraso de propagación.

Si fuera necesario simular una puerta con un retardo de propagación mayor (por ejemplo, en un multiplicador), habría que introducir un retraso de puerta en el circuito, justo tras la salida del multiplicador.

Si se incluye un circuito en otro ("padre"), el circuito incluido mantiene su comportamiento temporal. Así, que si se incluye un circuito complejo que tiene un retardo de propagación elevado debido a que las señales de entrada tienen que pasar tres puertas antes de alcanzar la salida, este comportamiento se mantendrá al incluir el circuito. No hay retardos adicionales introducidos como resultado de incluir un circuito. Si no todas las salidas de un circuito tienen el mismo retardo de propagación, esto se trasladará al circuito "padre". En general, incluir un circuito en otro no modifica su comportamiento temporal. Un circuito incrustado se comporta exactamente igual que si todos los componentes se hubieran insertado en el mismo nivel de circuito.

## 3. Análisis y síntesis de circuitos

Se puede analizar un circuito a través de la entrada del menú *Análisis*. Para circuitos puramente combinacionales, se generará una tabla de verdad, que podrá modificarse como se quiera. Tras la edición, se generará un nuevo circuito.

También se pueden analizar o generar circuitos secuenciales. En lugar de una simple tabla de verdad, se creará la llamada "tabla de transición de estados". De este modo, cada flip-flop aparece como entrada y como salida en la tabla de transición de estados. En esta tabla, al lado derecho, puede encontrarse el "estado siguiente", que ocurrirá tras la siguiente señal de reloj. Dicho estado dependerá del estado actual de los flip-flops (lo cual aparece a la izquierda de la tabla). Para que el análisis pueda realizarse, deberán nombrarse todos los flip-flops.

Se suele aplicar el siguiente criterio para nombrar los flip-flops: el estado siguiente de un bit de la derecha de la tabla se indicará con un subíndice 'n+1', El correspondiente estado actual se indicará con 'n'. Si hay un estado 'A', 'An' indica el estado actual, y 'An+1' indicará el estado siguiente. Si, en la tabla de verdad de la izquierda o de la derecha, hay señales presentes que siguen este patrón, se asumirá que es una tabla de transición de estados, y se generará un circuito secuencial en lugar de uno combinacional.

Debería destacarse que el circuito a analizar puede contener sólo elementos puramente combinacionales, además de los flip-flops D y JK. Si un flip-flop está hecho con puertas NOR, este circuito no se reconocerá como flip-flop y, por tanto, no es posible analizarlo así.

## 4. Hardware

### 4.1. GAL16v8 y GAL22v10

En el menú de creación de circuitos de la tabla de verdad encontramos la posibilidad de generar los llamados archivos JEDEC. Éste es un formato de archivo especial que describe el 'fuse map' de un PLD (autómata programable). Este archivo JEDEC puede escribirse en el PLD usando un programador especial. De momento, están soportados circuitos del tipo *GAL16v8* y *GAL22v10* o dispositivos compatibles con 'fuse map'.

### 4.2. ATF150xAS

Los chips de la familia *ATF150x* son simples CPLD con hasta 128 macroceldas. Están disponibles en un encapsulado PLCC, que les hace adecuados para ejercicios de laboratorio: si un circuito se rompe durante los ejercicios, puede ser sustituido fácilmente. Además, está disponible el *ATDH1150USB* un programador de bajo coste, fácil de usar. Este programador puede programar los chips *ATF150x* del sistema, usando un interfaz JTAG. Una placa de pruebas adecuada (*ATF15XX-DK3-U*) también está disponible. El software *ATMISP*, disponible en el sitio web de ATMEL/Microchip, se necesita para programar los chips.

Por desgracia, los detalles del 'fuse map' no son públicos, así que no hay ningún adaptador para este chip que pueda integrarse en Digital, como sí se puede con los chips *GAL16v8* y *GAL22v10*.

Por tanto, los adaptadores *fit150[x].exe* que proporciona ATMEL deben usarse. Estos programas crean un archivo *JEDEC* a partir de un adecuado archivo *TT2* que puede programarse en el chip. Digital comienza el adaptador automáticamente cada vez que se crea un archivo *TT2*. Para ello, la ruta a los adaptadores *fit150[n].exe* debe especificarse en los ajustes. El archivo *JEDEC* puede entonces ser abierto y programado directamente con *ATMISP*.

Por cuestiones legales, el adaptador *fit1502.exe* no se puede distribuir con Digital. Sin embargo, se puede encontrar en la carpeta *WinCupl\Fitters* después de instalar *WinCupl*. *WinCupl* está disponible en el sitio web de ATMEL/Microchip. En sistemas Linux, los adaptadores pueden ser ejecutados por Digital si *Wine* está instalado.

### 4.3. Exportación a VHDL o Verilog

Se puede exportar un circuito a VHDL o a Verilog. Se generará un archivo que contiene la descripción completa del circuito. El código VHDL generado fue probado con Xilinx Vivado y el simulador VHDL open source ghdl. El código Verilog se ha probado con el simulador de Verilog Icarus Verilog.

Si un circuito contiene casos de prueba, los datos se usarán para generar un módulo de estímulo (test bench) HDL. Esto puede emplearse para probar el correcto funcionamiento del circuito en una simulación HDL.

Pueden crearse ficheros adicionales que son necesarios para algunas tarjetas para funcionar. De momento sólo están soportadas las tarjetas BASYS3 y las Mimas Mimas y Mimas V2. Se crea un archivo de restricciones, que contendrá la asignación de pines. La descripción de los pines puede encontrarse en las datasheets de la placas y debe ser introducida como los números de pin para las entradas y las salidas.

En la placa BASYS3, si la frecuencia del reloj del circuito es baja, se integrará en el código HDL un divisor de frecuencia para dividir la frecuencia de la placa adecuadamente. Si la fre-

cuencia de reloj seleccionada supera los 4.7MHz, la unidad MMCM de la Artix-7 se empleará para la generación del reloj. Esto asegura que los recursos de la FPGA proporcionados por la distribución del reloj se empleen. Esto permite al procesador incluido correr a 20 MHz y si sabe hacerlo sin el multiplicador, es posible hacerlo a 30 MHz.

## 5. Formas personalizadas

Aunque Digital tiene algunas opciones que determinan el aspecto de un circuito cuando se incrusta en otro, en algunos casos puede ser útil usar una forma especial para un subcircuito. Un ejemplo es la representación de la ALU en el procesador incluido en los ejemplos. Este capítulo explica cómo crear una forma especial para un circuito.

Digital no tiene un editor para crear una forma especial. En su lugar, hay que dar un pequeño rodeo. Primero, el circuito al que se le va a representar por una forma especial deberá estar abierto. Después, se generará una plantilla SVG para él. En esta plantilla, el circuito se representará por un sencillo rectángulo. También contendrá todos los pines del circuito, representados con círculos azules (entradas) y rojos (salidas). Para ver qué círculo pertenece a cada pin, puedes mirar en el ID del círculo en las propiedades del objeto. Este ID tiene la forma *pin:[nombre]* o *pin+:[nombre]*. En el último caso, el pin tendrá una etiqueta si se reimporta a Digital. Si no quieres esa etiqueta, el + puede eliminarse.

Este archivo SVG puede ahora editarse. El programa más adecuado es Inkscape, que es open source y disponible gratis. Los pines pueden moverse libremente, pero se mueven al siguiente punto de la rejilla durante la reimportación.

Si se van a usar archivos SVG que ya existen, es más fácil abrir la plantilla que se creó y pegar el gráfico existente a la plantilla a través de Copiar&Pegar.

Si el archivo se había guardado, puede ser importado con Digital. Se lee el archivo y toda la información necesaria se extrae y se almacena en el circuito. Para usar el circuito más adelante, ya no se requiere el archivo SVG.

Un detalle final: SVR es un formato de archivo muy potente y flexible. Puede usarse para describir gráficos extremadamente complejos. El importador de Digital no puede importar todos los archivos SVG sin errores. Si un archivo no puede ser importado, o no tiene el aspecto deseado, habrá que seguir haciendo pruebas hasta conseguir el resultado deseado.

## 6. Preguntas habituales(FAQ)

### ¿Cómo se mueve un cable?

Selecciona uno de los terminales del cable con la selección rectangular. Luego mueve este punto usando el ratón. También puedes seleccionar un cable con CTRL + clic izquierdo.

### ¿Cómo se borra un cable?

Selecciona uno de los terminales del cable y pulsa *DEL* o hacer clic en el botón de la papelera (barra de herramientas). También puedes seleccionar un cable con CTRL + clic izquierdo.

### ¿Cómo mover un componente y todos sus cables?

Selecciona el componente con la herramienta de selección rectangular. La selección debe incluir el componente completo. Luego mueve el componente, incluyendo los cables, usando el ratón.

**Hay un componente que no está conectado a un cable, incluso aunque los pines estén sobre el cable.**

Un pin sólo se conecta a un cable si el cable tiene un punto terminal en el pin.

**Si los nombres de los pines de un circuito son largos, los nombres no son legibles cuando el circuito se incrusta. ¿Qué puedo hacer?**

La anchura del bloque puede aumentarse usando la opción del menú *Editar* → *Editar las propiedades del circuito*.

**Los pines de un circuito incrustado tienen un orden que no es óptimo. ¿Cómo se modifica esto?**

La secuencia se puede cambiar usando la entrada del menú *Editar* → *Ordenar las entradas* o *Editar* → *Ordenar las salidas*.

**Cuando empieza la simulación, un cable se pone de color gris. ¿Qué significa esto?**

Los colores verde claro y oscuro indican que el cable está en estado alto o bajo (respectivamente). Un cable gris indica que está en estado de alta impedancia.

**Tengo una tabla de verdad. ¿Cómo calculo las expresiones booleanas mínimas?**

En el menú *Análisis* selecciona la entrada *Sintetizar*. Luego introduce la tabla de verdad. Al pie de la ventana puedes encontrar la ecuación booleana que la satisface. Si introduces más de una variable dependiente, una nueva ventana se abre, en la que se muestran todas las expresiones booleanas.

**He introducido una tabla de verdad, pero hay más de una expresión booleana. ¿Cuál es la correcta?**

Al minimizar una expresión booleana pueden resultar varias expresiones que describen la misma función. Digital las muestra todas si vienen de la misma tabla. Puede haber diferencias dependiente de las condiciones "indiferentes" de la tabla de verdad.

**Tengo una tabla de verdad. ¿Cómo creo un circuito que se corresponda con la tabla?**

En el menú *Análisis* selecciona la entrada *Sintetizar*. Luego introduce la tabla de verdad. Puedes editar la tabla usando los menús *Nuevo* o *Añadir columnas*. En el menú *Crear* puedes crear un circuito usando la entrada *Circuito*.

**¿Cómo se edita el nombre de una señal en la tabla de verdad?**

Haz clic derecho con el ratón sobre el encabezado de la tabla para editarlo.

**Tengo una expresión booleana. ¿Cómo se crea un circuito?**

En el menú *Análisis* selecciona la entrada *Expresión*. Luego introduce la expresión.

**¿Cómo se crea una tabla de verdad a partir de una expresión booleana?**

En el menú *Análisis* selecciona la entrada *Expresión*. Luego introduce la expresión. Luego crea el circuito en el menú *Análisis* usa la entrada *Análisis* para crear la tabla de verdad.

**¿Cómo se crea un archivo JEDEC a partir de un circuito dado?**

En el menú *Análisis* selecciona la entrada *Análisis*. Luego, en el menú *Crear* de la nueva ventana que aparece, escoge el dispositivo adecuado en el submenú *Dispositivo*.

**Cuando creo un archivo JEDEC, ¿cómo se asigna un número de pin a una señal concreta?**

En las correspondientes entradas y salidas puedes colocar un número de pin en el diálogo de propiedades del mismo.

**He creado un archivo JEDEC. ¿Cómo lo programo con un *GAL16v8* o *GAL22v10*?**

Para programar tales chips necesitamos un programador de hardware especial.

## 7. Atajos de teclado

<b>Space</b>	Inicia o para la simulación.
<b>F6</b>	Abre el diálogo de tabla de medidas.
<b>F7</b>	Ejecuta hasta un Break
<b>F8</b>	Ejecuta casos de prueba
<b>C</b>	Paso de reloj sencillo (funciona sólo en una simulación en marcha y sólo si hay un único componente de reloj).
<b>F9</b>	Análisis del circuito
<b>CTRL-A</b>	Selecciona todo.
<b>CTRL-X</b>	Corta los componentes seleccionados al portapapeles.
<b>CTRL-C</b>	Copia los componentes seleccionados al portapapeles.
<b>CTRL-V</b>	Pega los componentes que haya en el portapapeles.
<b>CTRL-D</b>	Duplica la selección actual sin modificar el portapapeles.
<b>R</b>	Gira el componente.
<b>L</b>	Inserta de nuevo el último componente que se insertó.
<b>CTRL-N</b>	Nuevo circuito.
<b>CTRL-O</b>	Abre un circuito.
<b>CTRL-S</b>	Guarda el circuito.
<b>CTRL-Z</b>	Deshaz la última modificación.
<b>CTRL-Y</b>	Vuelve a hacer la última modificación que no se ha deshecho.
<b>P</b>	Programa un diodo o un FG-FET.
<b>D</b>	Dibujando un cable, permite dibujar en diagonal.
<b>F</b>	Dibujando una línea, cambia la orientación.
<b>S</b>	Divide un cable en dos.
<b>ESC</b>	Interrumpe la acción que esté en ejecución.
<b>Del</b>	Elimina los componentes seleccionados.
<b>Backspace</b>	Elimina los componentes seleccionados.
<b>+</b>	Incrementa el número de entradas del componente al que apunta el ratón.
<b>-</b>	Reduce el número de entradas del componente al que apunta el ratón.
<b>CTRL +</b>	Zoom adentro
<b>CTRL -</b>	Zoom afuera
<b>F1</b>	Ajusta la pantalla al tamaño del circuito
<b>F5</b>	Muestra u oculta el árbol de componentes

## B Ajustes

Esto describe los ajustes del simulador disponibles

### Ajustes

Los ajustes globales del simulador especifican, entre otras cosas, el idioma, los símbolos que se van a usar o las rutas de las herramientas externas.

#### Propiedades

- Usa las formas IEEE 91-1984

- Usa las formas IEEE 91-1984 en lugar de formas rectangulares

- Idioma

- Idioma de la interfaz. Sólo tendrá efecto tras reiniciar.

- Formato

- Formato en pantalla de las expresiones.

- Esquema de color

- Esquema de color

- Colores definidos por el usuario

- Colores definidos por el usuario

- Vista árbol de componentes visible en el inicio

- Marcado, la vista de árbol de componentes es visible en el inicio

- Mostrar rejilla

- Muestra una rejilla en la ventana principal.

- Muestra el número de cables de un bus.

- ATENCIÓN: el valor se actualiza sólo cuando empieza la simulación.

- Consejos sobre la herramienta cable

- Si está seleccionado, las líneas de resaltan cuando el ratón para sobre ellas.

- Usa la tecla "Igual"

- Usa la tecla igual en lugar de la tecla más. Ésto es especialmente útil si el carácter "más" no es una tecla primaria, sino la segunda asignación de la tecla igual, por ejemplo, en un teclado americano o francés.

- Increase numbers in identifiers when copying.

- If set, in components whose label ends with a number, this number is incremented when copying.

- Muestra el cuadro de diálogo para renombrar los túneles

- Si se fija, se muestra un cuadro de diálogo para renombrar automáticamente todos los túneles que tienen el mismo nombre después de que se renombra un túnel.

- Biblioteca

- Carpeta que contiene la biblioteca de subcircuitos predefinida. Contiene, por ejemplo, los componentes de la serie 74xx. También puedes añadir tus circuitos guardándolos en esta ubicación.

- Biblioteca Java

- Archivo .jar con componentes adicionales implementado en Java.

- Adaptador ATF15xx

- Ruta al adaptador para el ATF15xx. Introduce la carpeta que contiene el archivo fit15xx.exe proporcionado por Microchip (antigua ATMEL).



**ATMISP**

Ruta al archivo ejecutable ATMISP.exe. Si se marca, el software ATMISP puede iniciarse automáticamente.

**GHDL**

Ruta al archivo ejecutable de GHDL. Sólo es necesario si utilizas GHDL para simular componentes definidos con VHDL.

**iVerilog**

Ruta a la carpeta de instalación de Icarus Verilog. Sólo es necesaria si quieres usar iVerilog para simular componentes definidos con Verilog.

**Configuración de la toolchain**

Se emplea para configurar la integración de una toolchain. Permite el arranque de herramientas externas, por ejemplo, programar una FPGA o similar.

**Tamaño de la fuente de los menús**

Tamaño de las fuentes usadas en el menú en % del tamaño por defecto

**Usa los clics de ratón de MacOS.**

Usa CTRL-clic en lugar de clic derecho.

**Permitir conexión remota**

Si se marca, se abre un puerto TCP/IP, a través del cual se puede controlar el simulador

**Número de puerto**

Puerto en el que se abre el servidor remoto

## Ajustes específicos del circuito

Los ajustes específicos del circuito afectan al comportamiento del circuito que está abierto. Así, por ejemplo, la forma que representa al circuito cuando se incrusta en otros. Estos ajustes se almacenan junto al circuito.

**Propiedades****Etiqueta**

Nombre de este elemento.

**Anchura**

Anchura del símbolo si el circuito se usa como componente en otro circuito.

**Color del fondo**

El color del fondo del circuito cuando se incrusta en otro circuito. No se emplea para empaquetados DIL.

**Descripción**

Breve descripción del elemento y su uso.

**Modificación bloqueada**

El circuito está bloqueado. Es posible configurar diodos y FGF-FETs.

**Forma**

La forma que se va a usar para la representación del circuito en un circuito incrustado. En el modo "Simple", las entradas se muestran a la izquierda y las salidas a la derecha de un rectángulo simple. Con "Diseño", la posición de las entradas y las salidas y su orientación en el circuito determinan la posición de los pines. En esta opción puede haber pines arriba o abajo también. Cuando se selecciona "Chip DIL", se empleará una máscara DIL para mostrar el circuito. El número de pines de las entradas y las salidas determinarán la posición de los pines en este caso.

**Forma personalizada**

Importación de un archivo SVG

**Altura**

Altura del símbolo si el circuito se emplea como componente en otro circuito.

**Número de pines DIL**

Número de pines. Cero indica que el número de pines se calcula automáticamente

**Contenido de la ROM**

Contenido de todas las ROMs empleadas.

**Muestra valores medidos al comenzar la simulación**

Cuando empieza la simulación, se muestra una tabla con los valores medidos

**Mostrar la barra de medida al comienzo de la simulación**

Cuando empieza la simulación, se muestra un gráfico con los valores medidos.

**Muestra gráfica de medidas en el modo puerta única al comienzo de la simulación**

Cuando comienza la simulación, se muestra un gráfico con los valores medidos en el modo paso de puerta

**Número máximo de pasos que mostrar**

Máximo número de valores almacenados. Si el máximo se alcanza, se descartan los valores más viejos.

**Precarga la memoria del programa al comienzo.**

Cuando se simula un procesador que emplea una RAM como memoria del programa, es difícil iniciar este procesador, porque el contenido de la RAM está siempre inicializado con ceros al comienzo de la simulación. Este ajuste permite cargar los datos en la memoria del programa al comienzo. La memoria del programa en la simulación debe también estar marcada.

**Archivo de programa**

Archivo que debería cargarse en la memoria del programa al comienzo de la simulación.

**Usa big endian en la importación**

Usa el orden del bite big endian en la importación

**Saltar la exportación en Verilog/VHDL**

Se salta la generación del interior del circuito en la exportación a Verilog/VHDL. Se mantienen las referencias al circuito, haciendo posible anular la implementación.

**El circuito es genérico**

Permite crear un circuito genérico

**Detección de oscilación**

Cantidad de tiempos de propagación de puerta en el que se detecta una oscilación si el circuito no ha sido estabilizado

**In case of oscillations, continue with random values.**

If this option is set, the behavior of the simulation becomes unpredictable and extremely slow if there is an oscillation!

## C Interfaz de línea de comandos

`java -cp Digital.jar CLI`

`test -circ [String] [-tests [String]] [-allowMissingInputs] [-verbose]:`

El primer nombre de archivo especifica el circuito que se va a probar. Si se especifica un segundo nombre de archivo, los casos de prueba se ejecutan para este archivo. Si no se especifica el segundo nombre de archivo, los casos de prueba se ejecutan para el primer archivo.

Opciones:

`-circ [String(def: )]`

Nombre del archivo que se va a probar.

`[-tests [String(def: )]]`

Nombre del archivo con casos de prueba.

`[-allowMissingInputs(def: false)]`

Permite que no haya entradas en el circuito que están definidas en el caso de prueba. Esto puede ser útil si hay varias posibles soluciones que dependen de diferentes entradas.

`[-verbose(def: false)]`

Si se activa, en caso de error se muestra el valor de la tabla.

`svg -dig [String] [-svg [String]] [-ieee] [-LaTeX] [-pinsInMathMode] [-hideTest] [-noShapeFilling] [-smallIO] [-noPinMarker] [-thinnerLines] [-highContrast] [-monochrome]:`

Puede usarse para crear un archivo SVG a partir de un circuito.

Opciones:

`-dig [String(def: )]`

Nombre del archivo del circuito.

`[-svg [String(def: )]]`

Nombre del archivo SVG que se va a escribir.

`[-ieee(def: false)]`

Uso de los símbolos IEEE.

`[-LaTeX(def: false)]`

El texto se introduce en notación LaTeX. Se necesita Inkscape para procesamiento posterior.

`[-pinsInMathMode(def: false)]`

Para etiquetas de pines, usar el modo math incluso si no contiene índices.

`[-hideTest(def: false)]`

Ocultar los Casos de Prueba

`[-noShapeFilling(def: false)]`

Los polígonos no están rellenos.

`[-smallIO(def: false)]`

Entradas y salidas se representan como pequeños círculos.

`[-noPinMarker(def: false)]`

Los marcadores de pines azul y rojo se omiten en los símbolos.

`[-thinnerLines(def: false)]`

Si se marca, las líneas se dibujan un poco más delgadas.

`[-highContrast(def: false)]`

Los cables y el texto de los pines se muestran en negro.

`[-monochrome(def: false)]`

Sólo se usan grises.

`stats -dig [String] [-csv [String]]:`

Crea un archivo CSV que contiene las estadísticas del circuito. Todos los componentes se listan en un archivo CSV.

Opciones:

`-dig [String(def: )]`

Nombre del archivo del circuito.

`[-csv [String(def: )]]`

Nombre del archivo CSV que se va a crear. Si no está indicada esta opción, se escribe la tabla en stdout.

`run -dig [String]:`

Runs a circuit headless.

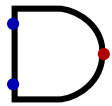
Opciones:

`-dig [String(def: )]`

File name of the circuit.

## D Componentes

### 1. Lógica



#### 1.1. AND

Puerta AND. Devuelve un 1 sólo si todas las entradas son 1. Es también posible usar buses con varios bits como entradas y salidas. En tal caso, la AND se ejecuta bit a bit. Esto quiere decir que los bits de menor peso de todas las entradas se operan con la AND y forman el bit de menor peso de la salida. Lo mismo sucede con el bit 1, el bit 2, etc. Exportable a VHDL/Verilog.

##### Entradas

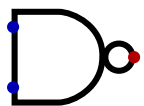
- In\_1  
1: valor de entrada para la operación lógica
- In\_2  
2: valor de entrada para la operación lógica

##### Salidas

- out  
Devuelve el resultado de la operación lógica

##### Propiedades

- Bits de datos
  - Número de bits de datos empleados.
- Número de entradas
  - Cantidad de entradas usada. Cada entrada tiene que estar conectada.
- Salidas invertidas
  - Puedes seleccionar las entradas que estarán invertidas
- Rotación
  - Orientación del elemento en el circuito.
- Anchura de la forma
  - Emplea una forma más ancha para visualizar la puerta.



#### 1.2. NAND

Combinación de AND y NOT. Devuelve 0 sólo si todas las entradas son 1. Si una de las entradas es 0, la salida es 1. Se pueden usar buses como entradas de varios bits. En este caso, la operación se aplica a cada bit de las entradas Exportable a VHDL/Verilog.

#### Entradas

In\_1

1: valor de entrada para la operación lógica

In\_2

2: valor de entrada para la operación lógica

#### Salidas

out

Devuelve el resultado de la operación lógica

#### Propiedades

Bits de datos

Número de bits de datos empleados.

Número de entradas

Cantidad de entradas usada. Cada entrada tiene que estar conectada.

Salidas invertidas

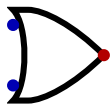
Puedes seleccionar las entradas que estarán invertidas

Rotación

Orientación del elemento en el circuito.

Anchura de la forma

Emplea una forma más ancha para visualizar la puerta.



### 1.3. OR

Puerta OR. Devuelve un 1 si por lo menos una de las entradas vale 1. Si todas las entradas son 0, la salida es también 0. Es también posible usar buses con varios bits como entradas y salida. En ese caso, la OR se ejecuta bit a bit. Esto quiere decir que los bits de menor peso de las entradas están operados con una OR y el resultado es el bit de menor peso de la salida. Lo mismo sucede con el bit 1, el bit 2, etc. Exportable a VHDL/Verilog.

#### Entradas

In\_1

1: valor de entrada para la operación lógica

In\_2

2: valor de entrada para la operación lógica

#### Salidas

out

Devuelve el resultado de la operación lógica

#### Propiedades

Bits de datos

Número de bits de datos empleados.

Número de entradas

Cantidad de entradas usada. Cada entrada tiene que estar conectada.

Salidas invertidas

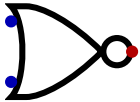
Puedes seleccionar las entradas que estarán invertidas

Rotación

Orientación del elemento en el circuito.

Anchura de la forma

Emplea una forma más ancha para visualizar la puerta.



#### 1.4. NOR

Combinación de OR y NOT. Devuelve un 0 si una de las entradas vale 1. Si todas las entradas son 0, la salida es 1. Es posible usar buses con varios bits por entrada. En tal caso, la operación se aplica a cada bit de las entradas. Exportable a VHDL/Verilog.

Entradas

In\_1

1: valor de entrada para la operación lógica

In\_2

2: valor de entrada para la operación lógica

Salidas

out

Devuelve el resultado de la operación lógica

Propiedades

Bits de datos

Número de bits de datos empleados.

Número de entradas

Cantidad de entradas usada. Cada entrada tiene que estar conectada.

Salidas invertidas

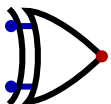
Puedes seleccionar las entradas que estarán invertidas

Rotación

Orientación del elemento en el circuito.

Anchura de la forma

Emplea una forma más ancha para visualizar la puerta.



#### 1.5. XOR

Si se usan dos entradas, la salida es 0 si las dos entradas son iguales. En otro caso, la salida estará a 1. Si se usan más de dos entradas, se comporta como XOR en cascada ( $A \text{ XOR } B \text{ XOR } C = ((A \text{ XOR } B) \text{ XOR } C)$ ). También es posible usar buses con varios bits por entrada. En tal caso, la operación se aplica a cada bit de las entradas. Exportable a VHDL/Verilog.

### Entradas

In\_1

1: valor de entrada para la operación lógica

In\_2

2: valor de entrada para la operación lógica

### Salidas

out

Devuelve el resultado de la operación lógica

### Propiedades

Bits de datos

Número de bits de datos empleados.

Número de entradas

Cantidad de entradas usada. Cada entrada tiene que estar conectada.

Salidas invertidas

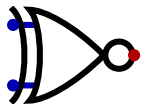
Puedes seleccionar las entradas que estarán invertidas

Rotación

Orientación del elemento en el circuito.

Anchura de la forma

Emplea una forma más ancha para visualizar la puerta.



## 1.6. XNOR

Combinación de XOR y NOT. A las entradas se les hace una operación XOR, y el resultado se invierte. También es posible usar buses con varios bits por entrada. En tal caso, la operación se aplica a cada uno de los bits de entrada. Exportable a VHDL/Verilog.

### Entradas

In\_1

1: valor de entrada para la operación lógica

In\_2

2: valor de entrada para la operación lógica

### Salidas

out

Devuelve el resultado de la operación lógica

### Propiedades

Bits de datos

Número de bits de datos empleados.

Número de entradas

Cantidad de entradas usada. Cada entrada tiene que estar conectada.

Salidas invertidas

Puedes seleccionar las entradas que estarán invertidas



Rotación

Orientación del elemento en el circuito.

Anchura de la forma

Emplea una forma más ancha para visualizar la puerta.



## 1.7. NOT

Invierte el valor de la entrada. Convierte un 1 en un 0 y viceversa. También es posible usar un bus con varios bits por entrada. En tal caso, la operación se aplica a cada bit de las entradas. Exportable a VHDL/Verilog.

Entradas

in

Entrada de la puerta NOT

Salidas

out

Valor de la entrada invertido

Propiedades

Bits de datos

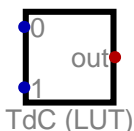
Número de bits de datos empleados.

Rotación

Orientación del elemento en el circuito.

Anchura de la forma

Emplea una forma más ancha para visualizar la puerta.



TdC (LUT)

## 1.8. Tabla de Consulta (LUT)

Extrae el valor de salida de una tabla de consulta. Esta puerta, por tanto, puede simular cualquier otra. Exportable a VHDL/Verilog.

Entradas

0

Entrada 0. Esta entrada en combinación con todas las demás define la dirección del valor almacenado que se va a devolver

1

Entrada 1. Esta entrada en combinación con todas las demás define la dirección del valor almacenado que se va a devolver

Salidas

out

Devuelve el valor almacenado en la posición que indican las entradas.

## Propiedades

### Bits de datos

Número de bits de datos empleados.

### Número de entradas

Cantidad de entradas usada. Cada entrada tiene que estar conectada.

### Etiqueta

Nombre de este elemento.

### Datos

Valores almacenados en este elemento.

### Rotación

Orientación del elemento en el circuito.

## 2. Entrada-Salida



### 2.1. Salida

Puede usarse para mostrar una señal de salida en un circuito. Este elemento también se usa para conectar un circuito con un circuito incrustado. En tal caso la conexión es bidireccional. También se usa para asignar un número de pin, si se genera código para un CPLD o una FPGA. Exportable a VHDL/Verilog.

#### Entradas

in

Este valor se usa para la conexión de salida.

#### Propiedades

##### Bits de datos

Número de bits de datos empleados.

##### Etiqueta

Nombre de este elemento.

##### Descripción

Breve descripción del elemento y su uso.

##### Rotación

Orientación del elemento en el circuito.

##### Formato de número

El formato se emplea para mostrar los números.

##### Dígitos de punto fijo

Número de dígitos binarios para fracciones

##### Número de pin

Número de este pin. Se usa para la representación de un circuito como encapsulado DIL y la asignación de pines cuando se programa un CPLD. Si hay varios bits, todos los números de pines deben especificarse en una lista separada por comas.

##### Mostrar en gráfico de medida

Muestra el valor en el gráfico de medida.

##### Forma pequeña

Si se selecciona, se usará una forma más pequeña.



## 2.2. LED

Un LED puede usarse para visualizar un valor de salida de un bit. Se enciende si la entrada vale 1.

### Entradas

in

Entrada del LED. El LED se enciende si la entrada está a 1.

### Propiedades

Etiqueta

Nombre de este elemento.

Color

Color del elemento.

Rotación

Orientación del elemento en el circuito.

Tamaño

Tamaño de la forma en el circuito.

Mostrar en gráfico de medida

Muestra el valor en el gráfico de medida.



## 2.3. Entrada

Puede usarse para controlar manualmente una señal de entrada de un circuito con el ratón. Este componente también se usa para conectar un circuito con otro incrustado. En este caso la conexión es bidireccional. Exportable a VHDL/Verilog.

### Salidas

out

Proporciona el valor de esta entrada

### Propiedades

Bits de datos

Número de bits de datos empleados.

Etiqueta

Nombre de este elemento.

Descripción

Breve descripción del elemento y su uso.

Rotación

Orientación del elemento en el circuito.

Por defecto

Este valor se fija si el circuito arranca. Una "Z" significa "estado de alta impedancia".

Es una entrada de tres estados

Si se marca, la entrada tendrá permitido estar en estado de alta impedancia. En un elemento de entrada, esto se permite si el estado de alta impedancia se fija como valor por defecto.

Salida no cero.

Evita la salida cero. Esto es especialmente útil cuando se configura circuitos de relés. Se puede activar sólo si se permite una salida de alta impedancia.

Formato de número

El formato se emplea para mostrar los números.

Dígitos de punto fijo

Número de dígitos binarios para fracciones

Número de pin

Número de este pin. Se usa para la representación de un circuito como encapsulado DIL y la asignación de pines cuando se programa un CPLD. Si hay varios bits, todos los números de pines deben especificarse en una lista separada por comas.

Mostrar en gráfico de medida

Muestra el valor en el gráfico de medida.

Forma pequeña

Si se selecciona, se usará una forma más pequeña.



## 2.4. Señal de reloj

Señal de reloj. Es posible controlarla con un reloj en tiempo real. Dependiendo de la complejidad del circuito, la frecuencia que muestre el reloj puede ser menor que la seleccionada. Si la frecuencia supera los 50 Hz, la representación gráfica del circuito no se actualizará en cada ciclo de reloj por lo que los colores de los cables no se actualizarán. Si el reloj en tiempo real no se activa, el reloj puede ser controlado por clics de ratón. También se usa para asignar un número de pin, si se va a generar código para un CPLD o una FPGA. Exportable a VHDL/Verilog.

Salidas

C

Conmuta entre 0 y 1 con la frecuencia de reloj seleccionada.

Propiedades

Etiqueta

Nombre de este elemento.

Iniciar el reloj de tiempo real

Si está habilitado, el reloj de simulación comienza cuando el circuito arranca

Frecuencia/Hz

La frecuencia en tiempo real empleada para el reloj real.

Rotación

Orientación del elemento en el circuito.

Número de pin

Número de este pin. Se usa para la representación de un circuito como encapsulado DIL y la asignación de pines cuando se programa un CPLD. Si hay varios bits, todos los números de pines deben especificarse en una lista separada por comas.

Forma pequeña

Si se selecciona, se usará una forma más pequeña.



## 2.5. Pulsador

Pulsador simple que vuelve a su posición cuando se libera

Salidas

out

Señal de salida del botón

Propiedades

Etiqueta

Nombre de este elemento.

Activo a 0

Si se marca, la salida es baja si el componente está activo.

Controlar con el teclado

El botón se controla con el teclado. Para usar las flechas del teclado, usa ARRIBA, ABAJO, IZQUIERDA o DERECHA como etiquetas.

Rotación

Orientación del elemento en el circuito.

Mostrar en gráfico de medida

Muestra el valor en el gráfico de medida.



## 2.6. Interruptor DIP

Interruptor DIP simple, que puede dar 0 o 1 como salida.

Salidas

out

Valor de salida del interruptor

Propiedades

Etiqueta

Nombre de este elemento.

Descripción

Breve descripción del elemento y su uso.

Rotación

Orientación del elemento en el circuito.

Salida alta

El valor de salida por defecto del conmutador DIP cuando empieza la simulación.



## 2.7. Sonda

Un valor medido puede verse en una gráfico o en una tabla de medidas. Este componente puede usarse para observar fácilmente valores de circuitos incrustados. No afecta a la simulación.

Entradas

in

El valor de la medida

Propiedades

Etiqueta

Nombre de este elemento.

Modo display

Define si se muestra el valor o un contador.

Rotación

Orientación del elemento en el circuito.

Formato de número

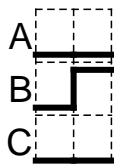
El formato se emplea para mostrar los números.

Dígitos de punto fijo

Número de dígitos binarios para fracciones

Mostrar en gráfico de medida

Muestra el valor en el gráfico de medida.



## 2.8. Gráfica de datos

Muestra un dibujo de los datos dentro del panel del circuito. Pueden dibujarse ciclos de reloj completos o cambios en puertas simples. No afecta a la simulación.

Propiedades

Mostrar pasos de puerta individual

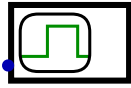
Muestra todos los pasos de puerta simple en el gráfico.

Número máximo de pasos que mostrar

Máximo número de valores almacenados. Si el máximo se alcanza, se descartan los valores más viejos.

Ajustarse a la rejilla

Si se fija, el componente se alinea con la rejilla.



## 2.9. Gráfica de datos activos

Muestra una gráfica de los valores medidos, por lo que los valores medidos se guardan solo si la señal de entrada cambia. El guardado tiene lugar cuando el circuito se ha estabilizado. La activación no comienza la medida en un rango real, pero cada evento de activación almacena un simple valor de medida para cada una de las señales mostradas. Como entrada directa solo está el activador. Las entradas y salidas de los circuitos, flip-flops y registros y el componente sonda pueden usarse como señales. Esto puede activarse en los respectivos componentes.

Entradas

T

Un cambio en esta entrada provoca el almacenamiento de los valores medidos

Propiedades

Etiqueta

Nombre de este elemento.

Disparador

Condición de disparo para la grabación de datos.

Número máximo de pasos que mostrar

Máximo número de valores almacenados. Si el máximo se alcanza, se descartan los valores más viejos.

## 3. Entrada-Salida - Displays



### 3.1. LED RGB

Un LED RGB cuyo color se puede controlar con tres entradas. En cada una de ellas se conecta un canal para el color.

Entradas

R

Canal del color rojo

G

Canal del color verde

B

Canal del color azul

Propiedades

Bits de datos

Número de bits de datos empleados.

Etiqueta

Nombre de este elemento.

Rotación

Orientación del elemento en el circuito.

Tamaño

Tamaño de la forma en el circuito.



### 3.2. LED con dos conexiones

LED con conexiones para el cátodo y el ánodo. El LED se enciende si el ánodo está conectado a nivel alto y el cátodo a nivel bajo. Este LED no se puede usar como resistencia pull-down. Funciona únicamente como elemento que indica funcionamiento. La resistencia que se ve está destinada a representar la resistencia serie que se precisa para limitar la corriente.

Entradas

A

Ánodo del LED

C

Cátodo del LED

Propiedades

Etiqueta

Nombre de este elemento.

Color

Color del elemento.

Rotación

Orientación del elemento en el circuito.

Mostrar en gráfico de medida

Muestra el valor en el gráfico de medida.



### 3.3. Pulsador con LED

Un pulsador que vuelve a su estado original cuando se presiona. El pulsador tiene un led que puede ser conmutado con una señal de entrada.

Entradas

in

Entrada para controlar el LED



## Salidas

out

Señal de salida del botón

## Propiedades

Etiqueta

Nombre de este elemento.

Activo a 0

Si se marca, la salida es baja si el componente está activo.

Controlar con el teclado

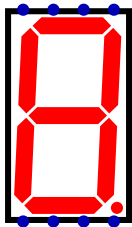
El botón se controla con el teclado. Para usar las flechas del teclado, usa ARRIBA, ABAJO, IZQUIERDA o DERECHA como etiquetas.

Color

Color del elemento.

Rotación

Orientación del elemento en el circuito.



### 3.4. Display de 7 segmentos

Display de siete segmentos. Cada segmento tiene su propia entrada de control. Se numeran con las letras 'a', 'b', 'c', 'd', 'e', 'f', 'g', en sentido dextrógiro, empezando por el segmento horizontal superior.

#### Entradas

a

Esta entrada controla el segmento 'a'.

b

Esta entrada controla el segmento 'b'.

c

Esta entrada controla el segmento 'c'.

d

Esta entrada controla el segmento 'd'.

e

Esta entrada controla el segmento 'e'.

f

Esta entrada controla el segmento 'f'.

g

Esta entrada controla el segmento 'g'.

dp

Esta entrada controla el punto decimal.

#### Propiedades

Color

Color del elemento.

Conexión común

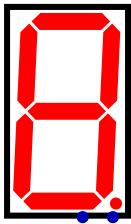
Si está seleccionado, se simula la entrada de ánodo o cátodo común

Común

Tipo de conexión común.

Persistencia de la visión

Especifica la duración del resplandor. A mayor valor, mayor la duración del resplandor



### 3.5. Display de 7 segmentos hexadecimal

Display de 7 segmentos con una entrada hexadecimal de 4 bits.

Entradas

d

El valor en esta entrada se visualiza en el display.

dp

Esta entrada controla el punto decimal.

Propiedades

Color

Color del elemento.

Tamaño

Tamaño de la forma en el circuito.



### 3.6. Display de 16 segmentos

La entrada de LED tiene 16 bits, que controlan los segmentos. La segunda entrada controla el punto decimal.

#### Entradas

led

Bus de 16 bits para controlar los ledes.

dp

Esta entrada controla el punto decimal

#### Propiedades

Color

Color del elemento.

Tamaño

Tamaño de la forma en el circuito.



### 3.7. Bombilla

La bombilla tiene dos conexiones. Si la atraviesa corriente, se enciende. La dirección de la corriente no importa. La bombilla se enciende cuando las entradas tienen diferente valor. Por tanto, se comporta igual que una puerta XOR

#### Entradas

A

Conector A

B

Conector B

#### Propiedades

Etiqueta

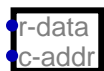
Nombre de este elemento.

Color

Color del elemento.

Rotación

Orientación del elemento en el circuito.



Matriz de ledes

### 3.8. Matriz de ledes

Matriz de ledes. Los ledes se muestran en una ventana aparte. Los ledes de una columna del display se controlan mediante una expresión de datos. En la otra entrada, se selecciona la columna actual. Por tanto, se hace una representación multiplexada. Los ledes se pueden encender indefinidamente en la simulación para evitar el parpadeo.

## Entradas

### r-data

El estado de la fila de una columna de ledes. Cada bit de esta expresión representa el estado de una fila de la columna actual.

### c-addr

El número de la columna actual cuyo estado actual está visible en la otra entrada.

## Propiedades

### Etiqueta

Nombre de este elemento.

### Filas

Especifica el número de filas indicando el número de bits de la palabra que la designa.

### Bits de dirección de las columnas

Direcciona columnas individuales. Tres bits indican ocho columnas.

### Color

Color del elemento.

### Evitar el parpadeo

No se puede incrementar la frecuencia tanto como para que desaparezca el parpadeo. Sin embargo, para suprimir el parpadeo, se puede activar "afterglow" para los ledes con esta opción. Si está activado, los ledes permanecen encendidos, incluso si uno de los pines cambia a "alta impedancia". Esto simula una frecuencia por encima de la frecuencia crítica de parpadeo.

### Rotación

Orientación del elemento en el circuito.

## 4. Entrada-Salida - Mecánica



### 4.1. Encoder rotatorio

Mando rotatorio con un encoder rotatorio. Se usa para detectar giros.

## Salidas

### A

Señal A del encoder

### B

Señal B del encoder

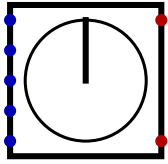
## Propiedades

### Etiqueta

Nombre de este elemento.

### Rotación

Orientación del elemento en el circuito.



#### 4.2. Motor paso a paso, unipolar

Motor paso a paso unipolar, con dos interruptores limitadores de posición. Están soportados el control normal (full step), medio paso (half step) o modo wave.

##### Entradas

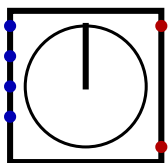
- P0  
Fase 0
- P1  
Fase 1
- P2  
Fase 2
- P3  
Fase 3
- com  
Conexión de la bobina central común

##### Salidas

- S0  
Switch de posición límite 0. Vale 1 cuando el ángulo del motor es 0°
- S1  
Switch de posición límite 1. Vale 1 cuando el ángulo del motor es 180°

##### Propiedades

- Etiqueta  
Nombre de este elemento.
- salida invertida  
Si se marca, la salida estará invertida.
- Rotación  
Orientación del elemento en el circuito.



#### 4.3. Motor paso a paso, bipolar

Motor paso a paso bipolar, con dos interruptores de posición límite. Están soportados el control normal (full step), medio paso (half step) o modo wave.

### Entradas

- A+  
Bobina A, positivo
- A-  
Bobina A, negativo
- B+  
Bobina B, positivo
- B-  
Bobina B, negativo

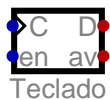
### Salidas

- S0  
Switch de posición límite 0. Vale 1 cuando el ángulo del motor es 0°
- S1  
Switch de posición límite 1. Vale 1 cuando el ángulo del motor es 180°

### Propiedades

- Etiqueta  
Nombre de este elemento.
- salida invertida  
Si se marca, la salida estará invertida.
- Rotación  
Orientación del elemento en el circuito.

## 5. Entrada-Salida - Periféricos



### 5.1. Teclado

Teclado que puede usarse para introducir texto. Este componente almacena la entrada, que puede leerse posteriormente. Una ventana aparte se abre para escribir el texto

#### Entradas

- C  
Reloj. Un flanco de subida elimina el último carácter de la memoria.
- en  
Si está alto, la salida D está activa y sale un carácter. También habilita la entrada de reloj.

#### Salidas

- D  
El último carácter escrito, o cero si no hay ningún carácter disponible. La salida es el carácter Java de 16 bits
- av  
Esta salida indica que hay caracteres disponibles. Se puede usar para disparar una interrupción.

## Propiedades

### Etiqueta

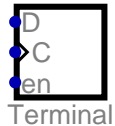
Nombre de este elemento.

### Salidas invertidas

Puedes seleccionar las entradas que estarán invertidas

### Rotación

Orientación del elemento en el circuito.



## 5.2. Terminal

Puedes escribir caracteres ASCII en el terminal. El terminal abre su propia ventana para visualizar la salida.

### Entradas

D

El dato que se va a escribir en la terminal.

C

Reloj. Un flanco de subida escribe el valor de la entrada en la terminal.

en

Un 1 en esta entrada habilita la entrada de reloj.

### Propiedades

#### Caracteres por línea

Número de caracteres mostrados en una línea.

#### Líneas

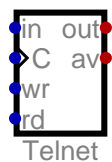
Número de líneas que se mostrarán.

#### Etiqueta

Nombre de este elemento.

#### Rotación

Orientación del elemento en el circuito.



## 5.3. Telnet

Permite una conexión Telnet al circuito. Es posible recibir y enviar caracteres vía Telnet.

### Entradas

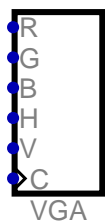
- in  
El dato que se va a enviar.
- C  
Entrada de reloj.
- wr  
Si se activa, se envía el byte de datos de entrada.
- rd  
Si se activa, se muestra un byte recibido.

### Salidas

- out  
Salida de datos
- av  
Muestra un 1 si hay algún dato presente.

### Propiedades

- Etiqueta  
Nombre de este elemento.
- Modo Telnet  
Si se activa, se evalúan los comandos de control Telnet. Además, el servidor envía los comandos SGA y ECHO. Si esta opción se desactiva, el servidor es un simple servidor TCP.
- Puerto  
Puerto que va a ser abierto por el servidor.
- Rotación  
Orientación del elemento en el circuito.



## 5.4. Monitor VGA

Analiza las señales de vídeo entrantes y muestra el gráfico correspondiente. Puesto que la simulación no se puede ejecutar en tiempo real, se necesita el reloj de pixel además de las señales de vídeo.

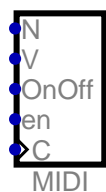


## Entradas

- R  
Componente de color rojo
- G  
Componente de color verde
- B  
Componente de color azul
- H  
Señal de sincronización horizontal
- V  
Señal de sincronización vertical
- C  
Reloj de píxel

## Propiedades

- Etiqueta  
Nombre de este elemento.
- Rotación  
Orientación del elemento en el circuito.

**5.5. MIDI**

Usa el MIDI del sistema para tocar notas.

## Entradas

- N  
Nota
- V  
Volumen
- OnOff  
Si está activo, funcionará al presionar una tecla del teclado. Si no está activo, funcionará al liberar la tecla.
- en  
Habilita el componente
- C  
Reloj

## Propiedades

- Etiqueta  
Nombre de este elemento.
- Canal MIDI  
Selecciona el canal MIDI que se va a usar.

Instrumento MIDI

Instrumento MIDI que se va a usar.

Permitir cambio en el programa

Añade una nueva entrada. Si esta entrada está a alto, el valor en la entrada N se usará para cambiar el programa (instrumento).

Rotación

Orientación del elemento en el circuito.

## 6. Cables



### 6.1. Tierra

Conexión a tierra. La salida es siempre cero. Exportable a VHDL/Verilog.

Salidas

out

La salida siempre devuelve 0.

Propiedades

Bits de datos

Número de bits de datos empleados.

Etiqueta

Nombre de este elemento.

Rotación

Orientación del elemento en el circuito.



### 6.2. Fuente de tensión

Conexión a la fuente de tensión. La salida es siempre 1. Exportable a VHDL/Verilog.

Salidas

out

Esta salida siempre devuelve 1.

Propiedades

Bits de datos

Número de bits de datos empleados.

Etiqueta

Nombre de este elemento.

Rotación

Orientación del elemento en el circuito.

1•

### 6.3. Valor constante

Componente que devuelve un valor dado como simple constante. El valor puede fijarse en el diálogo de propiedades. Exportable a VHDL/Verilog.

Salidas

out  
Devuelve el valor dado como constante.

Propiedades

Bits de datos  
Número de bits de datos empleados.  
Valor  
Valor de la constante.  
Rotación  
Orientación del elemento en el circuito.  
Formato de número  
El formato se emplea para mostrar los números.  
Dígitos de punto fijo  
Número de dígitos binarios para fracciones



### 6.4. Túnel

Conecta componentes sin cable. Todos los elementos "túnel" que tengan el mismo nombre de red están conectados entre sí. Funciona sólo localmente, por lo que no es posible conectar diferentes circuitos. Los túneles sin nombre se ignoran en silencio. Exportable a VHDL/Verilog.

Entradas

in  
Conexión al túnel

Propiedades

Nombre de la red  
Todas las redes con el mismo nombre están conectadas entre sí.  
Rotación  
Orientación del elemento en el circuito.



### 6.5. Divisor/Agregador

Divide o crea un haz de cables o un bus de datos con más de un bit. Con un bus es posible, por ejemplo, generar conexiones de 16 bits sin tener que generar 16 cables individuales. Las

16 conexiones pueden agruparse en un cable. El divisor tiene una dirección, es decir, sólo puede transmitir señales en una dirección. Exportable a VHDL/Verilog.

#### Entradas

0-3

Los bits de entrada 0-3

4-7

Los bits de entrada 4-7

#### Salidas

0-7

Los bits de salida 0-7

#### Propiedades

##### División de las entradas

Si, por ejemplo, tenemos cuatro bits, dos bits y otros dos más que se van a usar como entradas, esto puede configurarse con "4,2,2". El número indica la cantidad de bits. Por conveniencia, puede emplearse el asterisco: 16 bits puede configurarse con "[Bits]\*[Número]" como "1\*16". También es posible especificar los bits que se van a usar directamente en cualquier orden. Por ejemplo, "4-7,0-3" configura los bits 4-7 y 0-3. Esta notación permite cualquier disposición de los bits.

##### División de la salida

Si, por ejemplo, tenemos cuatro bits, dos bits y otros dos más van a emplearse como salidas, esto puede configurarse con "4,2,2". El número indica la cantidad de bits. Por conveniencia, puede usarse también el asterisco: 16 bits puede configurarse con "[Bits]\*[Número]" como "1\*16". También es posible indicar los bits que se van a usar en cualquier orden. Por ejemplo, "4-7,0-3" configura los bits 4-7 y 0-3. Esta notación permite cualquier disposición de los bits. Los bits de salida pueden usarse varias veces: "0-7,1-6,4-7".

##### Rotación

Orientación del elemento en el circuito.

##### Reflejo

Refleja el componente del circuito

##### Propagación

Configura la propagación de las entradas y salidas del circuito.



## 6.6. Driver

Un driver puede usarse para conectar un valor de señal a otro cable. El driver se controla por la entrada "sel". Si "sel" está bajo, la salida está en alta impedancia. Si "sel" está alto, la salida tomará el valor de la entrada. Exportable a VHDL/Verilog.

#### Entradas

in

Valor de entrada del driver

sel

Pin para controlar el driver. Si su valor es 1, la entrada se propaga a la salida. Si el valor es 0, la salida estará en estado de alta impedancia.

## Salidas

out

Si la entrada "sel" es 1, la entrada pasa a esta salida. Si la entrada "sel" es 0, esta salida estará en estado de alta impedancia.

## Propiedades

Bits de datos

Número de bits de datos empleados.

salida invertida

Si se marca, la salida estará invertida.

Cambia la posición del selector

Esta opción te permite mover el pin del selector al otro lado del plexor.

Rotación

Orientación del elemento en el circuito.



## 6.7. Driver, selector invertido

Un driver puede usarse para conectar una palabra de datos a otra línea. El driver se controla por la entrada "sel". Si "sel" está a 1, la salida estará en alta impedancia. Si "sel" está a 0, la salida contendrá el valor de la entrada. Exportable a VHDL/Verilog.

## Entradas

in

Valor de entrada del driver.

sel

Pin para controlar el driver. Si su valor es 0, la entrada se propaga a la salida. Si su valor es 1, la salida pasará a estado de alta impedancia.

## Salidas

out

Si la entrada "sel" es 1, la entrada pasa a esta salida. Si la entrada "sel" es 0, esta salida estará en estado de alta impedancia.

## Propiedades

Bits de datos

Número de bits de datos empleados.

salida invertida

Si se marca, la salida estará invertida.

Cambia la posición del selector

Esta opción te permite mover el pin del selector al otro lado del plexor.

Rotación

Orientación del elemento en el circuito.



## 6.8. Retardo

Retarda la señal en función de un tiempo de propagación. Retarda una señal por un número ajustable de retardos de puertas. Todos los demás componentes de Digital tienen un retardo de puerta como retardo de propagación

Entradas

in

Entrada de la señal que se va a retardar

Salidas

out

La señal de entrada retrasada un tiempo de retardo de puerta

Propiedades

Bits de datos

Número de bits de datos empleados.

Duración

Tiempo de demora en unidades de retraso de propagación de puertas.

Rotación

Orientación del elemento en el circuito.



## 6.9. Resistencia pull-up

Si una red está en estado de alta impedancia, esta resistencia pone la red a alto. En otro caso, este componente no hace nada.

Salidas

out

Un "alto débil"

Propiedades

Bits de datos

Número de bits de datos empleados.

Rotación

Orientación del elemento en el circuito.



### 6.10. Resistencia pull-down

Si la red está en estado de alta impedancia, esta resistencia pone la red a masa. En otro caso, este componente no tiene efecto.

Salidas

out

Un "bajo débil"

Propiedades

Bits de datos

Número de bits de datos empleados.

Rotación

Orientación del elemento en el circuito.



### 6.11. No conectado

Este componente puede emplearse para poner un cable en alta impedancia. Si una entrada de una puerta lógica está en alta impedancia, el valor leído en ella estará indeterminado. Observa que en realidad en muchos casos, puede ocurrir un excesivo consumo de corriente e incluso un daño si una entrada no está fijada a alta o baja, sino que permanece desconectada.

Salidas

out

Esta salida siempre proporciona alta impedancia.

Propiedades

Bits de datos

Número de bits de datos empleados.

## 7. Plexores



### 7.1. Multiplexor

Componente que usa el valor del pin de selección (sel) para decidir qué valor de entrada se pone en la salida. Exportable a VHDL/Verilog.

### Entradas

sel

Esta entrada se emplea para seleccionar el dato de entrada que pasará a la salida.

in\_0

El 0 dato de entrada del multiplexor.

in\_1

El 1 dato de entrada del multiplexor.

### Salidas

out

El valor de la entrada seleccionada.

### Propiedades

Bits de datos

Número de bits de datos empleados.

Número de los bits del selector

Número de bits empleados para el selector de entrada.

Cambia la posición del selector

Esta opción te permite mover el pin del selector al otro lado del plexor.

Rotación

Orientación del elemento en el circuito.



## 7.2. Demultiplexor

Componente que puede enviar el valor de entrada a una cualquiera de varias salidas. Las otras salidas estarán fijadas al valor por defecto. Exportable a VHDL/Verilog.

### Entradas

sel

Este pin selecciona la salida que se va a emplear.

in

El valor de esta entrada se propagará a la salida de datos seleccionada.

### Salidas

out\_0

Salida de datos 0.

out\_1

Salida de datos 1.

### Propiedades

Bits de datos

Número de bits de datos empleados.

Número de los bits del selector

Número de bits empleados para el selector de entrada.

Cambia la posición del selector

Esta opción te permite mover el pin del selector al otro lado del plexor.



Rotación

Orientación del elemento en el circuito.

Por defecto

Este valor se fija si se inicia el circuito. En el demultiplexor, este valor se fija para las salidas no seleccionadas.



### 7.3. Decodificador

Uno de los pines seleccionables es 1. Todas las demás salidas son 0. Exportable a VHDL/Verilog.

Entradas

sel

Esta entrada selecciona la salida que está habilitada. La salida seleccionada está a 1. Todas las demás salidas están a 0.

Salidas

out\_0

Salida 0. Esta salida es 1 si se selecciona mediante la entrada "sel"

out\_1

Salida 1. Esta salida es 1 si se selecciona mediante la entrada "sel"

Propiedades

Número de los bits del selector

Número de bits empleados para el selector de entrada.

Cambia la posición del selector

Esta opción te permite mover el pin del selector al otro lado del plexor.

Rotación

Orientación del elemento en el circuito.



### 7.4. Selector de bit

Selecciona un bit de un bus de datos. Exportable a VHDL/Verilog.

Entradas

in

Bus de entrada

sel

Esta entrada selecciona el bit

Salidas

out

Bit seleccionado.

## Propiedades

Número de los bits del selector

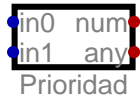
Número de bits empleados para el selector de entrada.

Cambia la posición del selector

Esta opción te permite mover el pin del selector al otro lado del plexor.

Rotación

Orientación del elemento en el circuito.



## 7.5. Codificador con prioridad

Si una de las entradas está activa, su número pasa a la salida. Si varias entradas están activas a la vez, el número más alto se pasa a la salida. Exportable a VHDL/Verilog.

### Entradas

in0

La entrada 0 del codificador con prioridad.

in1

La entrada 1 del codificador con prioridad.

### Salidas

num

Número de la entrada fijada

any

Si esta entrada está activa, al menos una de las entradas está activa.

### Propiedades

Etiqueta

Nombre de este elemento.

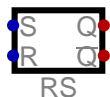
Número de los bits del selector

Número de bits empleados para el selector de entrada.

Rotación

Orientación del elemento en el circuito.

## 8. Flip-Flops



### 8.1. Flip-Flop RS

Componente para almacenar un bit. Dispone de las funciones "set" y "reset" para fijar o borrar el bit almacenado. Si ambas entradas valen 1, ambas salidas se pondrán a uno. Si ambas entradas se ponen a 0 a la vez, el estado final será indeterminado.

## Entradas

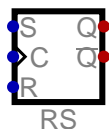
- S  
Entrada de Set
- R  
Entrada de Reset.

## Salidas

- Q  
Devuelve el valor almacenado.
- $\neg Q$   
Devuelve el inverso del valor almacenado.

## Propiedades

- Etiqueta  
Nombre de este elemento.
- Salidas invertidas  
Puedes seleccionar las entradas que estarán invertidas
- Rotación  
Orientación del elemento en el circuito.
- Reflejo  
Refleja el componente del circuito
- Por defecto  
Este valor se fija si se inicia el circuito. En el demultiplexor, este valor se fija para las salidas no seleccionadas.
- Usar como valor de medida  
Si se marca, el valor es un valor de medida, y aparece en la tabla de datos y el gráfico. Además, debe especificarse una etiqueta que sirva como identificación del valor.



## 8.2. Flip-Flop RS con reloj

Componente para almacenar un bit. Permite a las funciones "set" y "reset" fijar o borrar el bote almacenado. Si ambas entradas (S,R) están a 1 en el flanco de subida del reloj, el estado final es indeterminado

## Entradas

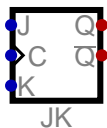
- S  
La entrada de Set
- C  
Entrada de reloj. Un flanco de subida inicia un cambio de estado.
- R  
Entrada de Reset

## Salidas

- Q  
Devuelve el valor almacenado.
- $\neg Q$   
Devuelve el ingreso del valor almacenado.

## Propiedades

- Etiqueta  
Nombre de este elemento.
- Salidas invertidas  
Puedes seleccionar las entradas que estarán invertidas
- Rotación  
Orientación del elemento en el circuito.
- Reflejo  
Refleja el componente del circuito
- Por defecto  
Este valor se fija si se inicia el circuito. En el demultiplexor, este valor se fija para las salidas no seleccionadas.
- Usar como valor de medida  
Si se marca, el valor es un valor de medida, y aparece en la tabla de datos y el gráfico. Además, debe especificarse una etiqueta que sirva como identificación del valor.



### 8.3. Flip-Flop JK

Puede almacenar ( $J=K=0$ ), fijar ( $J=1, K=0$ ), reiniciar ( $J=0, K=1$ ) o invertir ( $J=K=1$ ) el valor almacenado. El cambio de estado se produce sólo en un flanco de subida en la entrada de reloj. Exportable a VHDL/Verilog.

## Entradas

- J  
Entrada Set del Flip-Flop
- C  
Entrada de reloj. Un flanco de subida provoca un cambio de estado.
- K  
Entrada de Reset del Flip-Flop

## Salidas

- Q  
Devuelve el valor almacenado.
- $\neg Q$   
Devuelve el inverso del valor almacenado.

## Propiedades

**Etiqueta**

Nombre de este elemento.

**Salidas invertidas**

Puedes seleccionar las entradas que estarán invertidas

**Rotación**

Orientación del elemento en el circuito.

**Reflejo**

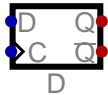
Refleja el componente del circuito

**Por defecto**

Este valor se fija si se inicia el circuito. En el demultiplexor, este valor se fija para las salidas no seleccionadas.

**Usar como valor de medida**

Si se marca, el valor es un valor de medida, y aparece en la tabla de datos y el gráfico. Además, debe especificarse una etiqueta que sirva como identificación del valor.

**8.4. Flip-Flop tipo D**

Componente empleado para almacenar un valor. El valor en el pin D se almacena en un flanco de subida del reloj en el pin C. La anchura del bit puede seleccionarse, permitiéndose almacenar múltiples bits. Exportable a VHDL/Verilog.

**Entradas**

D

Entrada del bit que se va a almacenar.

C

Pin de reloj para almacenar un valor. El valor en la entrada D se almacena en un flanco de subida de este pin.

**Salidas**

Q

Devuelve el valor almacenado.

$\neg Q$

Devuelve el inverso del valor almacenado.

**Propiedades****Bits de datos**

Número de bits de datos empleados.

**Etiqueta**

Nombre de este elemento.

**Salidas invertidas**

Puedes seleccionar las entradas que estarán invertidas

**Rotación**

Orientación del elemento en el circuito.

**Reflejo**

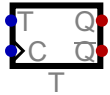
Refleja el componente del circuito

Por defecto

Este valor se fija si se inicia el circuito. En el demultiplexor, este valor se fija para las salidas no seleccionadas.

Usar como valor de medida

Si se marca, el valor es un valor de medida, y aparece en la tabla de datos y el gráfico. Además, debe especificarse una etiqueta que sirva como identificación del valor.



## 8.5. Flip-Flop tipo T

Almacena un solo bit. Invierte el estado en un flanco de subida introducido por la entrada C.

Entradas

T

Habilita la función de conmutación.

C

Entrada de reloj. Un flanco de subida invierte la salida, si la entrada T está a 1.

Salidas

Q

Devuelve el valor almacenado.

$\neg Q$

Devuelve el inverso del valor almacenado.

Propiedades

Etiqueta

Nombre de este elemento.

Habilita la entrada

Si se marca, se dispondrá de una entrada de habilitación (T).

Salidas invertidas

Puedes seleccionar las entradas que estarán invertidas

Rotación

Orientación del elemento en el circuito.

Reflejo

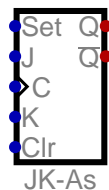
Refleja el componente del circuito

Por defecto

Este valor se fija si se inicia el circuito. En el demultiplexor, este valor se fija para las salidas no seleccionadas.

Usar como valor de medida

Si se marca, el valor es un valor de medida, y aparece en la tabla de datos y el gráfico. Además, debe especificarse una etiqueta que sirva como identificación del valor.



## 8.6. Flip-Flop JK asíncrono

Tiene la posibilidad de almacenar ( $J=K=0$ ), fijar ( $J=1, K=0$ ), borrar ( $J=0, K=1$ ) o conmutar ( $J=K=1$ ) el valor almacenado. El cambio de estado tiene lugar sólo en un flanco de subida en la entrada de reloj C. Hay dos entradas adicionales que fijan el estado o lo reinician sin señal de reloj. Exportable a VHDL/Verilog.

### Entradas

#### Set

Set asíncrono. Un valor alto en esta entrada fija el Flip-Flop.

#### J

Entrada de Set del Flip-Flop

#### C

Entrada de reloj. Un flanco de subida provoca un cambio de estado.

#### K

Entrada de Reset del Flip-Flop.

#### Clr

Reset asíncrono. Un valor alto en esta entrada borra el Flip-Flop.

### Salidas

#### Q

Devuelve el valor almacenado.

#### $\neg Q$

Devuelve el inverso del valor almacenado.

### Propiedades

#### Etiqueta

Nombre de este elemento.

#### Salidas invertidas

Puedes seleccionar las entradas que estarán invertidas

#### Rotación

Orientación del elemento en el circuito.

#### Reflejo

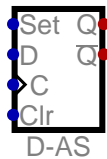
Refleja el componente del circuito

#### Por defecto

Este valor se fija si se inicia el circuito. En el demultiplexor, este valor se fija para las salidas no seleccionadas.

#### Usar como valor de medida

Si se marca, el valor es un valor de medida, y aparece en la tabla de datos y el gráfico. Además, debe especificarse una etiqueta que sirva como identificación del valor.



### 8.7. Flip-Flop tipo D, asíncrono

Componente usado para almacenar un valor. El valor del pin D se almacena con un flanco de subida por el pin de reloj, C. Se puede seleccionar el tamaño de bit, lo cual permite almacenar varios bits. Exportable a VHDL/Verilog.

#### Entradas

##### Set

Set asíncrono. Si está a 1, todos los bits almacenados valdrán 1.

##### D

Entrada del bit que se va a almacenar.

##### C

Pin de control para almacenar un bit. El bit de la entrada D se almacena en un flanco de subida de este pin.

##### Clr

Clear asíncrono. Si está a uno, todos los bits se ponen a 0.

#### Salidas

##### Q

Devuelve el valor almacenado.

##### $\neg Q$

Devuelve el inverso del valor almacenado.

#### Propiedades

##### Bits de datos

Número de bits de datos empleados.

##### Etiqueta

Nombre de este elemento.

##### Salidas invertidas

Puedes seleccionar las entradas que estarán invertidas

##### Rotación

Orientación del elemento en el circuito.

##### Reflejo

Refleja el componente del circuito

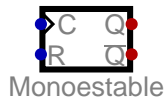
##### Por defecto

Este valor se fija si se inicia el circuito. En el demultiplexor, este valor se fija para las salidas no seleccionadas.

##### Usar como valor de medida

Si se marca, el valor es un valor de medida, y aparece en la tabla de datos y el gráfico. Además, debe especificarse una etiqueta que sirva como identificación del valor.





## 8.8. Monoestable

El monoestable se activa con un flanco de subida en la entrada de reloj. Después de un tiempo de retardo que es configurable, el monoestable se borrará automáticamente. El monoestable se puede disparar varias veces. Se puede usar sólo si hay exactamente un componente reloj presente en el circuito. Este componente de reloj se usa como unidad de medida del tiempo de retardo.

### Entradas

- C  
Entrada de reloj. Un flanco de subida activa el monoestable.
- R  
Entrada de Reset. Un valor alto borra el monoestable.

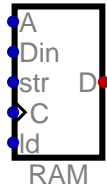
### Salidas

- Q  
salida
- $\neg Q$   
salida invertida

### Propiedades

- Etiqueta  
Nombre de este elemento.
- Anchura de pulso  
La anchura de pulso se mide en ciclos de reloj.
- Salidas invertidas  
Puedes seleccionar las entradas que estarán invertidas
- Rotación  
Orientación del elemento en el circuito.
- Reflejo  
Refleja el componente del circuito
- Por defecto  
Este valor se fija si se inicia el circuito. En el demultiplexor, este valor se fija para las salidas no seleccionadas.
- Usar como valor de medida  
Si se marca, el valor es un valor de medida, y aparece en la tabla de datos y el gráfico. Además, debe especificarse una etiqueta que sirva como identificación del valor.

## 9. Memorias - RAM



### 9.1. RAM, puertos separados

Módulo de RAM con entradas separadas para almacenamiento y salida para leer los datos almacenados. Exportable a VHDL/Verilog.

#### Entradas

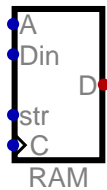
- A Dirección desde la que leer o en la que escribir.
- Din Datos que se van a almacenar en la RAM.
- str Si esta entrada está a 1 y hay un flanco de subida del reloj, los datos se almacenan.
- C Entrada de reloj
- Id Si esta entrada está a 1, la salida está activa y los datos son visibles en la salida.

#### Salidas

- D Pin de salida de datos.

#### Propiedades

- Bits de datos  
Número de bits de datos empleados.
- Bits de dirección  
Número de bits de dirección empleados.
- Etiqueta  
Nombre de este elemento.
- Rotación  
Orientación del elemento en el circuito.
- Formato de número  
El formato se emplea para mostrar los números.
- Dígitos de punto fijo  
Número de dígitos binarios para fracciones
- Memoria del programa  
Convierte a esta ROM a memoria de programa. Por ello podrá accederse a ella desde un IDE externo.



## 9.2. Bloque de RAM, puertos separados

Módulo de RAM con entradas separadas para almacenamiento y salida para lectura de los datos almacenados. Esta RAM actualiza la salida sólo en un flanco de subida de la entrada de reloj. Esto permite el uso del bloque de RAM en una FPGA. Exportable a VHDL/Verilog.

### Entradas

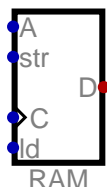
- A  
Dirección desde la que leer o en la que escribir.
- Din  
Datos que van a almacenarse en la RAM.
- str  
Los datos se guardan si esta salida está alta y hay un flanco de subida del reloj.
- C  
Reloj de entrada.

### Salidas

- D  
Pin de salida de los datos.

### Propiedades

- Bits de datos  
Número de bits de datos empleados.
- Bits de dirección  
Número de bits de dirección empleados.
- Etiqueta  
Nombre de este elemento.
- Rotación  
Orientación del elemento en el circuito.
- Memoria del programa  
Convierte a esta ROM a memoria de programa. Por ello podrá accederse a ella desde un IDE externo.



## 9.3. RAM, puerto bidireccional

Módulo de RAM con un pin bidireccional para lectura y escritura de datos.

## Entradas

- A  
Dirección en la que leer y escribir.
- str  
Si esta entrada está alta y hay un flanco de subida del reloj, el dato se almacena.
- C  
Reloj
- ld  
Si esta entrada está alta, la salida estará activa y el dato es visible en ella.

## Salidas

- D  
Conexión bidireccional de datos.

## Propiedades

- Bits de datos  
Número de bits de datos empleados.
- Bits de dirección  
Número de bits de dirección empleados.
- Etiqueta  
Nombre de este elemento.
- Rotación  
Orientación del elemento en el circuito.
- Formato de número  
El formato se emplea para mostrar los números.
- Dígitos de punto fijo  
Número de dígitos binarios para fracciones
- Memoria del programa  
Convierte a esta ROM a memoria de programa. Por ello podrá accederse a ella desde un IDE externo.



## 9.4. RAM, selecciona chip

Módulo de RAM con conexión bidireccional para lectura y escritura de datos. Si la entrada CS está baja, el componente está deshabilitado. Esto permite construir una RAM más grande con RAMs más pequeñas y un decodificador de direcciones. El ciclo de escritura trabaja así: poniendo CS alto, el componente se selecciona. Un flanco de subida en WE bloquea la dirección, y el siguiente flanco de bajada en WE almacena el dato.

## Entradas

A

Dirección en la que leer y escribir.

CS

Si esta entrada está alta, la RAM está habilitada. En otro caso la salida estará siempre en alta impedancia.

WE

Si está en alto, el dato se escribe en la RAM.

OE

Si esta entrada está en alto, el valor almacenado va a la salida.

## Salidas

D

Conexión de datos bidireccional.

## Propiedades

Bits de datos

Número de bits de datos empleados.

Bits de dirección

Número de bits de dirección empleados.

Etiqueta

Nombre de este elemento.

Salidas invertidas

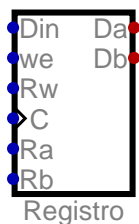
Puedes seleccionar las entradas que estarán invertidas

Rotación

Orientación del elemento en el circuito.

Memoria del programa

Convierte a esta ROM a memoria de programa. Por ello podrá accederse a ella desde un IDE externo.



## 9.5. Bloque de registro

Memoria con un puerto que permite escribir y dos puertos que permiten leer de la memoria simultáneamente. Puede usarse para implementar registros de procesadores. Dos registros pueden leerse simultáneamente y un tercero puede ser escrito. Exportable a VHDL/Verilog.

## Entradas

Din

Dato que se va a almacenar en el registro Rw.

we

Si esta entrada está alta y hay un flanco de subida del reloj, se almacena el dato.

Rw

Registro en el que se escriben los datos.

C

Reloj

Ra

Registro que es visible en el puerto a.

Rb

Registro que es visible en el puerto b.

## Salidas

Da

Puerto de salida a

Db

Puerto de salida b

## Propiedades

Bits de datos

Número de bits de datos empleados.

Bits de dirección

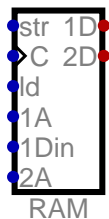
Número de bits de dirección empleados.

Etiqueta

Nombre de este elemento.

Rotación

Orientación del elemento en el circuito.



## 9.6. RAM, puerto dual

RAM con un puerto que permite escribir y leer desde ella, y un segundo puerto de sólo lectura. Este segundo puerto puede usarse para dar acceso a alguna lógica de gráficos al contenido de la memoria. De este modo, un procesador puede escribir en la RAM, y una lógica de gráficos puede leer de la RAM. Exportable a VHDL/Verilog.

## Entradas

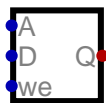
- str  
Si esta entrada está alta y hay un flanco de subida del reloj, el dato se almacena.
- C  
Reloj
- ld  
Si esta entrada está alta, la salida estará activada, y el dato será visible en la salida 1D.
- 1A  
Dirección en la que el puerto 1 es leído o escrito.
- 1Din  
Dato que se va a almacenar en la RAM.
- 2A  
Dirección empleada para leer por el puerto 2.

## Salidas

- 1D  
Puerto de salida 1
- 2D  
Puerto de salida 2

## Propiedades

- Bits de datos  
Número de bits de datos empleados.
- Bits de dirección  
Número de bits de dirección empleados.
- Etiqueta  
Nombre de este elemento.
- Rotación  
Orientación del elemento en el circuito.
- Memoria del programa  
Convierte a esta ROM a memoria de programa. Por ello podrá accederse a ella desde un IDE externo.



RANm asíncrona

## 9.7. RANm asíncrona

Siempre y cuando se habilite, se almacena. Corresponde a una RAM muy simple, donde las líneas de direcciones y de datos están directamente conectadas a los decodificadores de las celdas de memoria. Exportable a VHDL/Verilog.

## Entradas

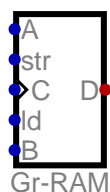
- A  
Dirección en la que se produce la lectura o la escritura
- D  
Datos que se van a almacenar
- we  
Habilitación de escritura. Siempre y cuando esta entrada esté a 1, el valor aplicado a D se almacena en la dirección aplicada a A, tanto si A o D han cambiado o no.

## Salidas

- Q  
Salida de los datos almacenados

## Propiedades

- Bits de datos  
Número de bits de datos empleados.
- Bits de dirección  
Número de bits de dirección empleados.
- Salidas invertidas  
Puedes seleccionar las entradas que estarán invertidas
- Etiqueta  
Nombre de este elemento.
- Rotación  
Orientación del elemento en el circuito.
- Memoria del programa  
Convierte a esta ROM a memoria de programa. Por ello podrá accederse a ella desde un IDE externo.



## 9.8. RAM gráfica

Se emplea para mostrar un gráfico mapa de bits. Este componente se comporta como una RAM. Adicionalmente, muestra el contenido en una ventana gráfica. Cada píxel se representa por una dirección de memoria. El valor almacenado define el color del píxel, usando una paleta de color fija. Hay dos buffers de pantalla para soportar el paso de página. La entrada B selecciona el buffer que se muestra. Así, el tamaño total de memoria es  $dx * dy * 2$  palabras. La paleta usada se organiza así: los índices 0-9 corresponden a los colores blanco, negro, rojo, verde, azul, amarillo, cian, magenta, naranja y rosa. Los índices 32-63 mapean valores de gris y los índices 64-127 representan 64 valores de color, cada uno con dos bits por canal de color. Esto resulta en una paleta simple que puede ser direccionada con sólo 7 bits. Si la arquitectura soporta índices de 16 bits, desde el índice 0x8000, puede usarse un modo de alto color con 5 bits por canal de color lo cual permitirá 32768 colores.



## Entradas

- A Dirección en la que leer y escribir.
- str Si esta entrada está alta y hay un flanco de subida del reloj, el dato se almacena.
- C Reloj
- ld Si esta entrada está alta, la salida se activa y el dato es visible en la salida.
- B Selecciona el buffer de pantalla que se va a mostrar.

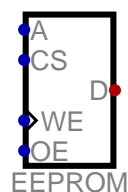
## Salidas

- D Conexión de datos bidireccional.

## Propiedades

- Bits de datos
  - Número de bits de datos empleados.
- Etiqueta
  - Nombre de este elemento.
- Anchura en píxeles
  - Anchura de la pantalla en píxeles.
- Altura en píxeles
  - Altura de la pantalla en píxeles.
- Rotación
  - Orientación del elemento en el circuito.

# 10. Memorias - EEPROM



## 10.1. EEPROM

Módulo EEPROM con conexión bidireccional para lectura y escritura de datos. Si la entrada CS está a 0, el componente está deshabilitado. Los datos se almacenan como en una ROM. Así, se mantienen cuando la simulación finaliza y se reinicia. El ciclo de escritura trabaja así: colocamos CS a alto, y se selecciona el componente. Un flanco de subida en WE bloquea la dirección, y el siguiente flanco de bajada en WE almacena los datos.

## Entradas

A

Dirección en la que leer y escribir.

CS

Si esta entrada está alta, la EEPROM está habilitada. En otro caso, la salida está siempre en alta impedancia.

WE

Si está alta, el dato se escribe en la EEPROM.

OE

Si esta entrada está alta, extrae el valor almacenado.

## Salidas

D

Entrada de datos bidireccional.

## Propiedades

Bits de datos

Número de bits de datos empleados.

Bits de dirección

Número de bits de dirección empleados.

Etiqueta

Nombre de este elemento.

Salidas invertidas

Puedes seleccionar las entradas que estarán invertidas

Datos

Valores almacenados en este elemento.

Rotación

Orientación del elemento en el circuito.

Formato de número

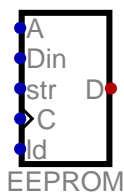
El formato se emplea para mostrar los números.

Dígitos de punto fijo

Número de dígitos binarios para fracciones

Memoria del programa

Convierte a esta ROM a memoria de programa. Por ello podrá accederse a ella desde un IDE externo.



## 10.2. EEPROM, puertos separados

Módulo EEPROM con entradas separadas para almacenamiento y salida para lectura de los datos almacenados.

## Entradas

- A Dirección desde la que leer o en la que escribir
- Din Datos que se van a almacenar en la EEPROM.
- str Si esta entrada está alta y hay un flanco de subida del reloj el dato se almacena.
- C Entrada de reloj
- ld Si esta entrada está alta la salida estará activa y el dato es visible en la salida.

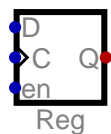
## Salidas

- D Pin de salida de datos.

## Propiedades

- Bits de datos  
Número de bits de datos empleados.
- Bits de dirección  
Número de bits de dirección empleados.
- Etiqueta  
Nombre de este elemento.
- Datos  
Valores almacenados en este elemento.
- Rotación  
Orientación del elemento en el circuito.
- Formato de número  
El formato se emplea para mostrar los números.
- Dígitos de punto fijo  
Número de dígitos binarios para fracciones
- Memoria del programa  
Convierte a esta ROM a memoria de programa. Por ello podrá accederse a ella desde un IDE externo.

# 11. Memorias



## 11.1. Registro

Componente para almacenar valores. El tamaño de bits de la palabra de datos puede escogerse. A diferencia de un FF tipo D, el registro proporciona una entrada que habilita al reloj. Exportable a VHDL/Verilog.

### Entradas

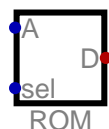
- D  
Pin de entrada de la expresión que se va a almacenar
- C  
Entrada de reloj. Un flanco de subida almacena el valor en el pin D.
- en  
Pin de habilitación. Sólo se almacena un valor si este pin está a 1.

### Salidas

- Q  
Devuelve el valor almacenado.

### Propiedades

- Bits de datos  
Número de bits de datos empleados.
- Etiqueta  
Nombre de este elemento.
- Salidas invertidas  
Puedes seleccionar las entradas que estarán invertidas
- Rotación  
Orientación del elemento en el circuito.
- Contador de programa  
Hace de este registro un contador de programa El valor de este registro es devuelto al IDE ensamblador externo para marcar la línea actual de código durante la depuración.
- Usar como valor de medida  
Si se marca, el valor es un valor de medida, y aparece en la tabla de datos y el gráfico. Además, debe especificarse una etiqueta que sirva como identificación del valor.



## 11.2. ROM

Elemento de memoria no volátil. Los datos almacenados pueden editarse en el diálogo de propiedades. Exportable a VHDL/Verilog.

### Entradas

- A  
Este pin define la dirección de la expresión que va a la salida.
- sel  
Si la entrada es alta, la salida está activada. Si está baja, la salida de datos está en estado de alta impedancia.

### Salidas

- D  
La expresión seleccionada si la entrada "sel" está alta.

## Propiedades

### Bits de datos

Número de bits de datos empleados.

### Bits de dirección

Número de bits de dirección empleados.

### Etiqueta

Nombre de este elemento.

### Datos

Valores almacenados en este elemento.

### Rotación

Orientación del elemento en el circuito.

### Formato de número

El formato se emplea para mostrar los números.

### Dígitos de punto fijo

Número de dígitos binarios para fracciones

### Memoria del programa

Convierte a esta ROM a memoria de programa. Por ello podrá accederse a ella desde un IDE externo.

### Recargar al iniciar el modelo

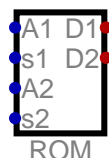
Recarga el archivo HEX cada vez que el modelo se ejecuta.

### Archivo

Archivo a cargar en la ROM.

### Usa big endian en la importación

Usa el orden del bite big endian en la importación



## 11.3. Puerto dual ROM

Componente de memoria no volátil. Los datos almacenados pueden ser editados en el diálogo de propiedades.

### Entradas

#### A1

Este pin define la dirección de la palabra de datos que va a mostrarse en D1.

#### s1

Si la entrada está en alto, la salida D1 está activada. Si está en bajo, la salida de datos estará en estado de alta impedancia.

#### A2

Este pin define la dirección de la palabra de datos que va a mostrarse en D2.

#### s2

Si la entrada está en alto, la salida D2 estará activada. Si está en bajo, la salida de datos estará en estado de alta impedancia.

## Salidas

D1

La palabra de datos seleccionada si la entrada s1 está en valor alto.

D2

La palabra de datos seleccionada si la entrada s2 está en valor alto.

## Propiedades

Bits de datos

Número de bits de datos empleados.

Bits de dirección

Número de bits de dirección empleados.

Etiqueta

Nombre de este elemento.

Datos

Valores almacenados en este elemento.

Rotación

Orientación del elemento en el circuito.

Formato de número

El formato se emplea para mostrar los números.

Dígitos de punto fijo

Número de dígitos binarios para fracciones

Memoria del programa

Convierte a esta ROM a memoria de programa. Por ello podrá accederse a ella desde un IDE externo.

Recargar al iniciar el modelo

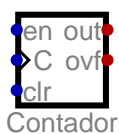
Recarga el archivo HEX cada vez que el modelo se ejecuta.

Archivo

Archivo a cargar en la ROM.

Usa big endian en la importación

Usa el orden del bite big endian en la importación



## 11.4. Contador

Contador simple. La entrada de reloj incrementa el contador. Puede ser puesta a 0 con la entrada "clr". El número de bits del contador puede fijarse en el diálogo de propiedades. Exportable a VHDL/Verilog.

## Entradas

en

Si se fija a 1, el contador está activo.

C

Entrada de reloj. Un flanco de subida incrementa el contador.

clr

Reset síncrono del contador si se fija a 1.

## Salidas

out

Devuelve el valor contado.

ovf

Salida de overflow (desbordamiento). Este pin vale 1 si el contador llega a su valor máximo y en la entrada hay un 1.

## Propiedades

Bits de datos

Número de bits de datos empleados.

Salidas invertidas

Puedes seleccionar las entradas que estarán invertidas

Etiqueta

Nombre de este elemento.

Rotación

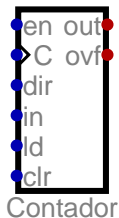
Orientación del elemento en el circuito.

Usar como valor de medida

Si se marca, el valor es un valor de medida, y aparece en la tabla de datos y el gráfico. Además, debe especificarse una etiqueta que sirva como identificación del valor.

Contador de programa

Hace de este registro un contador de programa El valor de este registro es devuelto al IDE ensamblador externo para marcar la línea actual de código durante la depuración.



## 11.5. Contador con preset

Contador cuyo valor puede fijarse. Además, se pueden indicar el valor máximo y una dirección de conteo. Exportable a VHDL/Verilog.

## Entradas

- en
  - Si se fija a 1, el contador se habilita.
- C
  - Entrada de reloj. Un flanco de subida incrementa el contador.
- dir
  - Indica la dirección de conteo. Un "0" indica ascendente.
- in
  - Esta palabra de datos se almacena en el contador cuando "ld" está activo.
- ld
  - Si se fija, el valor que haya en "in" se almacenará en el contador en la siguiente señal de reloj.
- clr
  - Reset síncrono del contador si se fija a 1.

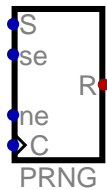
## Salidas

- out
  - Devuelve el valor contado.
- ovf
  - Pin de overflow (desbordamiento). Se pone a 1 si la entrada "en" vale 1 y el contador alcanza el máximo valor contando ascendente o alcanza el 0 en cuenta descendente.

## Propiedades

- Bits de datos
  - Número de bits de datos empleados.
- Valor máximo
  - Si se introduce un ero, se utilizará el valor máximo (todos los bits serán uno).
- Salidas invertidas
  - Puedes seleccionar las entradas que estarán invertidas
- Etiqueta
  - Nombre de este elemento.
- Rotación
  - Orientación del elemento en el circuito.
- Usar como valor de medida
  - Si se marca, el valor es un valor de medida, y aparece en la tabla de datos y el gráfico. Además, debe especificarse una etiqueta que sirva como identificación del valor.
- Contador de programa
  - Hace de este registro un contador de programa El valor de este registro es devuelto al IDE ensamblador externo para marcar la línea actual de código durante la depuración.





## 11.6. Generador de números aleatorios

Puede emplearse para generar números aleatorios. Cuando se inicia la simulación, el generador se reinicia, de forma que una nueva secuencia de números pseudoaleatorios se genera en cada inicio. El generador puede lanzarse en la simulación en marcha con un valor de inicio para generar una secuencia de números pseudoaleatorios definida.

### Entradas

- S  
Nuevo valor de inicio del generador
- se  
Activo, se reinicia el generador de números aleatorios con el valor de inicio en el siguiente flanco de subida del reloj.
- ne  
Activo, sale un nuevo número aleatorio en el siguiente flanco de subida del reloj.
- C  
Entrada de reloj.

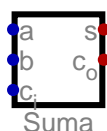
### Salidas

- R  
Salida del número pseudoaleatorio

### Propiedades

- Bits de datos  
Número de bits de datos empleados.
- Etiqueta  
Nombre de este elemento.
- Rotación  
Orientación del elemento en el circuito.

## 12. Aritmética



### 12.1. Sumador

Componente que hace sumas simples. Suma los dos valores enteros de las entradas "a" y "b" ( $a+b$ ). El resultado será incrementado en uno si la entrada de acarreo (carry) está activa. Exportable a VHDL/Verilog.

### Entradas

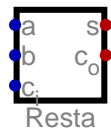
- a  
Primera entrada de la suma.
- b  
Segunda entrada de la suma.
- c\_i  
Entrada de acarreo: si se fija, el resultado se incrementa en uno.

### Salidas

- s  
Resultado de la suma
- c\_o  
Salida de acarreo. Si se fija ha habido un desbordamiento.

### Propiedades

- Etiqueta  
Nombre de este elemento.
- Bits de datos  
Número de bits de datos empleados.
- Rotación  
Orientación del elemento en el circuito.



## 12.2. Restar

Componente para restas sencillas. Resta los números binarios de las entradas "a" y "b" (a-b). Si la entrada de acarreo vale 1, el resultado se disminuye en 1. Exportable a VHDL/Verilog.

### Entradas

- a  
Entrada "a" para la resta.
- b  
Entrada "b" para la resta.
- c\_i  
Entrada de acarreo: si está fijada el resultado se reduce en 1.

### Salidas

- s  
La salida devuelve el resultado de la diferencia.
- c\_o  
La salida devuelve 1 si ocurrió un desbordamiento.

### Propiedades

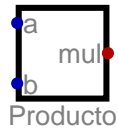
- Etiqueta  
Nombre de este elemento.

Bits de datos

Número de bits de datos empleados.

Rotación

Orientación del elemento en el circuito.



### 12.3. Multiplicar

Componente que multiplica. Multiplica los números enteros de las entradas "a" y "b".  
Exportable a VHDL/Verilog.

Entradas

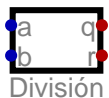
- a  
Entrada del factor "a"
- b  
Entrada del factor "b"

Salidas

- mul  
Salida para el resultado del producto.

Propiedades

- Etiqueta  
Nombre de este elemento.
- Operación con signo  
Si está seleccionado, la operación se hace con valores con signo (complemento a 2).
- Bits de datos  
Número de bits de datos empleados.
- Rotación  
Orientación del elemento en el circuito.



### 12.4. División

Componente que divide. Divide el entero de la entrada "a" entre el entero de la entrada "b".  
Si el divisor es cero, se divide por uno. Si la división lleva signo, el resto es siempre positivo.

Entradas

- a  
Dividendo
- b  
Divisor

## Salidas

q  
Cociente

r  
Resto

## Propiedades

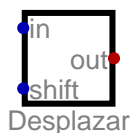
Etiqueta  
Nombre de este elemento.

Bits de datos  
Número de bits de datos empleados.

Operación con signo  
Si está seleccionado, la operación se hace con valores con signo (complemento a 2).

Resto siempre positivo  
Si está activado, el resto de una división con signo será siempre positivo.

Rotación  
Orientación del elemento en el circuito.



## 12.5. Registro de desplazamiento

Componente que desplaza bits. Desplaza el valor de entrada un número de bits dado por la entrada de "shift"

## Entradas

in  
Bits de entrada que se van a desplazar.

shift  
Entrada con la anchura del desplazamiento

## Salidas

out  
Salida con el valor desplazado

## Propiedades

Etiqueta  
Nombre de este elemento.

Bits de datos  
Número de bits de datos empleados.

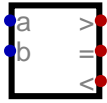
La entrada de desplazamiento tiene signo  
La entrada de desplazamiento tiene formato complemento a 2

Dirección  
Fijar dirección.

Modo  
Modo del registro de desplazamiento

Rotación

Orientación del elemento en el circuito.



## 12.6. Comparador

Componente que compara valores de bits. Compara los números binarios de los pines "a" y "b" y fija las correspondientes salidas. Exportable a VHDL/Verilog.

Entradas

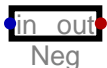
- a  
Entrada del valor "a"
- b  
Entrada del valor "b"

Salidas

- >  
Esta salida es 1 si la entrada "a" es mayor que la "b".
- =  
Esta salida es 1 si la entrada "a" y "b" son iguales.
- <  
Esta salida es 1 si la entrada "a" es menor que la "b"

Propiedades

- Etiqueta  
Nombre de este elemento.
- Bits de datos  
Número de bits de datos empleados.
- Operación con signo  
Si está seleccionado, la operación se hace con valores con signo (complemento a 2).
- Rotación  
Orientación del elemento en el circuito.



## 12.7. Negación

Negación en complemento a 2. Exportable a VHDL/Verilog.

Entradas

- in  
Entrada de la palabra de datos que se va a negar con complemento a 2

Salidas

- out  
Devuelve el resultado de la negación con complemento a 2.

## Propiedades

### Bits de datos

Número de bits de datos empleados.

### Rotación

Orientación del elemento en el circuito.



## 12.8. Extensor de signo

Incrementa la anchura de bits de un valor manteniendo el signo. Si la entrada es un solo bit, este bit será la salida de todos los bits de salida. Exportable a VHDL/Verilog.

### Entradas

in

Valor de entrada. La amplitud del bit de entrada debe ser menor que la del bit de salida.

### Salidas

out

Valor de entrada extendido. La amplitud del bit de entrada debe ser menor que la del bit de salida.

## Propiedades

### Etiqueta

Nombre de este elemento.

### Introduce anchura de bit

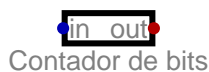
El número de bits de salida debe ser mayor que el de bits de entrada.

### Anchura del bit de salida

El número de bits de salida debe ser mayor que el de bits de entrada.

### Rotación

Orientación del elemento en el circuito.



## 12.9. Contador de bits

Devuelve la cantidad de bits en el valor de entrada.

### Entradas

in

Entrada en la que los bits se cuentan.

### Salidas

out

Devuelve el número de bits.

## Propiedades

### Bits de datos

Número de bits de datos empleados.

### Rotación

Orientación del elemento en el circuito.

## 13. Interruptores



### 13.1. Conmutador

Conmutador simple. No hay retraso por puertas: un cambio de señal se propaga inmediatamente.

## Salidas

### A1

Una de las conexiones del conmutador.

### B1

Una de las conexiones del conmutador.

## Propiedades

### Bits de datos

Número de bits de datos empleados.

### Etiqueta

Nombre de este elemento.

### Contar contactos

Cantidad de contactos disponibles

### Cerrado

Fija el estado inicial del conmutador.

### Rotación

Orientación del elemento en el circuito.

### Reflejo

Refleja el componente del circuito

### El switch se comporta como una entrada

Si se analiza el modelo, el switch se comporta como una entrada, en la que "abierto" corresponde a '0' y "cerrado" a '1'.



### 13.2. Conmutador de dos caminos

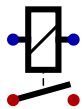
Conmutador de dos caminos. No hay retraso por puertas: un cambio de señal se propaga inmediatamente.

### Salidas

- A1  
Una de las conexiones del conmutador.
- B1  
Una de las conexiones del conmutador.
- C1  
Una de las conexiones del conmutador.

### Propiedades

- Bits de datos  
Número de bits de datos empleados.
- Etiqueta  
Nombre de este elemento.
- Contar contactos  
Cantidad de contactos disponibles
- Rotación  
Orientación del elemento en el circuito.
- Reflejo  
Refleja el componente del circuito



## 13.3. Relé

Un relé es un conmutador que puede ser controlado por un electroimán. Si la corriente pasa por la bobina del electroimán, el conmutador se cierra o se abre. No hay diodo en antiparalelo, por lo que la dirección de la corriente no importa. El conmutador se activa si las entradas tienen diferentes valores. El relé se comporta igual a una puerta XOR.

### Entradas

- in1  
Una de las entradas de control del relé.
- in2  
Una de las entradas de control del relé.

### Salidas

- A1  
Una de las conexiones del conmutador.
- B1  
Una de las conexiones del conmutador.

### Propiedades

- Bits de datos  
Número de bits de datos empleados.
- Etiqueta  
Nombre de este elemento.
- Contar contactos  
Cantidad de contactos disponibles



El relé está normalmente cerrado.

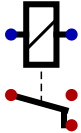
Si se marca, el relé estará cerrado si la entrada es baja.

Rotación

Orientación del elemento en el circuito.

Reflejo

Refleja el componente del circuito



### 13.4. Relé de dos contactos

Un relé es un conmutador que puede ser controlado por un electroimán. Si la corriente pasa por la bobina del electroimán, el conmutador se cierra o se abre. No hay diodo en antiparalelo, por lo que la dirección de la corriente no importa. El conmutador se activa si las entradas tienen diferentes valores. El relé se comporta igual a una puerta XOR.

Entradas

in1

Una de las entradas para controlar el relé.

in2

Una de las entradas para controlar el relé.

Salidas

A1

Una de las conexiones del conmutador.

B1

Una de las conexiones del conmutador.

C1

Una de las conexiones del conmutador.

Propiedades

Bits de datos

Número de bits de datos empleados.

Etiqueta

Nombre de este elemento.

Contar contactos

Cantidad de contactos disponibles

Rotación

Orientación del elemento en el circuito.

Reflejo

Refleja el componente del circuito



### 13.5. FET de canal P

FET de canal P. La base está conectada a la tensión positiva y el transistor es simulado sin diodo interno.

#### Entradas

G  
Puerta

#### Salidas

S  
Fuente  
D  
Drenaje

#### Propiedades

Bits de datos  
Número de bits de datos empleados.

Unidireccional  
Los transistores unidireccionales propagan una señal sólo de la fuente al drenaje. Son mucho más rápidos de simular que los bidireccionales. Puesto que no hay realimentación del drenaje a la fuente, los transistores no cortocircuitan cables al conducir. Este modo es necesario en algunos circuitos CMOS.

Etiqueta  
Nombre de este elemento.

Rotación  
Orientación del elemento en el circuito.

Reflejo  
Refleja el componente del circuito



### 13.6. FET de canal N

FET de canal N. La base está conectada a masa y el transistor se simula sin diodo interno.

#### Entradas

G  
Puerta

#### Salidas

D  
Drenaje  
S  
Fuente

## Propiedades

### Bits de datos

Número de bits de datos empleados.

### Unidireccional

Los transistores unidireccionales propagan una señal sólo de la fuente al drenaje. Son mucho más rápidos de simular que los bidireccionales. Puesto que no hay realimentación del drenaje a la fuente, los transistores no cortocircuitan cables al conducir. Este modo es necesario en algunos circuitos CMOS.

### Etiqueta

Nombre de este elemento.

### Rotación

Orientación del elemento en el circuito.

### Reflejo

Refleja el componente del circuito



## 13.7. Fusible

Fusible usado para construir memorias programables una sola vez.

### Salidas

#### out1

Una de las conexiones del conmutador.

#### out2

Una de las conexiones del conmutador.

### Propiedades

#### Programado

Si se marca, el diodo estará "flotante" o "programado". En un FET de puerta flotante, ésta estará cargada. Puedes cambiar este ajuste con la tecla "p".

#### Rotación

Orientación del elemento en el circuito.



## 13.8. Diodo polarizado

Diodo unidireccional simplificado, usado para poner un cable a VDD. Se usa para implementar una puerta OR. Es necesario conectar una resistencia pull-down a la salida del diodo. En la simulación el diodo se comporta como una puerta con tres valores: Si la entrada está alta, la salida también está alta. En todos los otros casos (la entrada está baja o en alta impedancia) la salida está en alta impedancia. Así, dos diodos conectados en antiparalelo pueden mantenerse uno a otro en estado alto, lo cual no es posible con diodos reales. Éste es un diodo real: no hay pérdida de voltaje a través del diodo en polarización directa.

**Entradas**

in

Si la entrada es alta, la salida es también alta. En todos los otros casos la salida estará en alta impedancia.

**Salidas**

out

Si la entrada está en alto, la salida también está en alto. En todos los demás casos la salida estará en alta impedancia.

**Propiedades****Programado**

Si se marca, el diodo estará "flotante" o "programado". En un FET de puerta flotante, ésta estará cargada. Puedes cambiar este ajuste con la tecla "p".

**Rotación**

Orientación del elemento en el circuito.

**13.9. Diodo a masa**

Diodo unidireccional simplificado, usado para poner un cable a masa. Se usa para implementar una puerta AND. Es necesario conectar una resistencia pull-up a la salida de los diodos. Si la entrada está baja, la salida estará baja. En los demás casos (entrada alta o alta impedancia) la salida estará en alta impedancia. Así dos diodos conectados en antiparalelo pueden mantenerse en estado bajo uno a otro, lo cual no es posible con diodos reales. Es un diodo ideal: no hay pérdida de voltaje en polarización directa.

**Entradas**

in

Si la entrada está baja, la salida también estará baja. En todos los demás casos la salida estará en alta impedancia.

**Salidas**

out

Si la entrada está baja, la salida también estará baja. En todos los demás casos la salida estará en alta impedancia.

**Propiedades****Programado**

Si se marca, el diodo estará "flotante" o "programado". En un FET de puerta flotante, ésta estará cargada. Puedes cambiar este ajuste con la tecla "p".

**Rotación**

Orientación del elemento en el circuito.



### 13.10. FET de canal P de compuerta flotante

FET de canal P de compuerta flotante. La base está conectada a tierra y el transistor se simula sin diodo interno. Si hay carga almacenada en la puerta flotante, el FET no conduce, incluso si la puerta está a 0.

#### Entradas

G  
Puerta

#### Salidas

S  
Fuente  
D  
Drenaje

#### Propiedades

Bits de datos  
Número de bits de datos empleados.

Etiqueta  
Nombre de este elemento.

Programado  
Si se marca, el diodo estará "flotante" o "programado". En un FET de puerta flotante, ésta estará cargada. Puedes cambiar este ajuste con la tecla "p".

Rotación  
Orientación del elemento en el circuito.

Reflejo  
Refleja el componente del circuito



### 13.11. FET de canal N de compuerta flotante

FET de canal N de compuerta flotante.

#### Entradas

G  
Puerta

#### Salidas

D  
Drenaje  
S  
Fuente

### Propiedades

#### Bits de datos

Número de bits de datos empleados.

#### Etiqueta

Nombre de este elemento.

#### Programado

Si se marca, el diodo estará "flotante" o "programado". En un FET de puerta flotante, ésta estará cargada. Puedes cambiar este ajuste con la tecla "p".

#### Rotación

Orientación del elemento en el circuito.

#### Reflejo

Refleja el componente del circuito



## 13.12. Puerta de transmisión

Un puerta de transmisión real se construye sólo con dos transistores. Por tanto, se usa a menudo para ahorrar transistores durante la implementación en silicio.

### Entradas

S

Entrada de control

$\neg S$

Entrada de control invertida

### Salidas

A

Entrada A

B

Entrada B

### Propiedades

#### Bits de datos

Número de bits de datos empleados.

#### Rotación

Orientación del elemento en el circuito.

## 14. Varios

Test

### 14.1. Caso de prueba

Describe un caso de prueba. En un caso de prueba (test case) de un circuito, se puede describir cómo éste debería comportarse. Podrá entonces automáticamente comprobarse

si el comportamiento del circuito corresponde a su descripción. Si no es así, se muestra un mensaje de error. El texto de ayuda del editor de casos de prueba describe detalladamente cómo puede crearse dicho caso de prueba. Exportable a VHDL/Verilog.

#### Propiedades

##### Etiqueta

Nombre de este elemento.

##### Datos de prueba

Descripción de los valores de prueba. Pueden encontrarse detalles de la sintaxis en el diálogo de ayuda del editor de datos de prueba.

##### Habilitado

Habilita o deshabilita este componente.

## 15. Varios - Decoración

### Texto

#### 15.1. Texto

Muestra un texto en el circuito. No afecta a la simulación. El texto puede cambiarse en el diálogo de Propiedades

#### Propiedades

##### Descripción

Breve descripción del elemento y su uso.

##### Tamaño de fuente

Fija el tamaño de fuente que se usará para el texto.

##### Rotación

Orientación del elemento en el circuito.

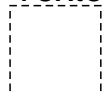
##### Orientación

Posición del texto en coordenadas relativas.

##### Ajustarse a la rejilla

Si se fija, el componente se alinea con la rejilla.

### Texto



#### 15.2. Rectángulo

Muestra un rectángulo en el circuito. No afecta a la simulación. Si se usa un signo menos en el encabezado, éste se omite.

#### Propiedades

##### Etiqueta

Nombre de este elemento.

##### Anchura

Anchura en unidades de rejilla

Altura

Altura en unidades de rejilla

Tamaño de fuente

Fija el tamaño de fuente que se usará para el texto.

Texto interior

Coloca texto dentro del rectángulo.

Texto al pie

Coloca texto al pie del rectángulo.

Texto a la derecha

Coloca texto a la derecha del rectángulo.

Ajustarse a la rejilla

Si se fija, el componente se alinea con la rejilla.

## 16. Varios - Genérica



init

### 16.1. Inicialización genérica

Código que se ejecuta para iniciar un circuito genérico. Si se va a iniciar un circuito genérico, este componente tiene que estar presente. Exportable a VHDL/Verilog.

Propiedades

Etiqueta

Nombre de este elemento.

Habilitado

Habilita o deshabilita este componente.

Parametrización genérica

Declaraciones hechas para generar un circuito

Código

### 16.2. Código

Código que se ejecuta cuando se concreta un circuito genérico. Puede usarse, por ejemplo, para añadir componentes o cables a un circuito. Exportable a VHDL/Verilog.

Propiedades

Parametrización genérica

Declaraciones hechas para generar un circuito



## 17. Varios - VHDL/Verilog



### 17.1. Externo

Componente para ejecutar un proceso externo para calcular la función lógica. Se usa para especificar el comportamiento de un componente con VHDL o Verilog. La simulación real del comportamiento debe hacerse con un simulador externo. De momento sólo están soportados el simulador GHDL para VHDL, e Icarus Verilog para Verilog. ¡La etiqueta del componente debe coincidir con el nombre de la entidad o módulo! Exportable a VHDL/Verilog.

#### Entradas

in

#### Salidas

out

#### Propiedades

##### Etiqueta

Nombre de este elemento.

##### Anchura

Anchura del símbolo si el circuito se usa como componente en otro circuito.

##### Entradas

Entradas del proceso externo. Es una lista de nombres de señales separadas por comas. Para cada nombre de señal, usando ":", puede indicarse el número de bits. Las entradas de un sumador de 8 bits podrían, por ejemplo, describirse como "a:8,b:8;C\_in".

##### Salidas

Salidas de un proceso externo. Es una lista de nombres de señales separadas por comas. Para cada nombre de señal, usando ":", puede indicarse el número de bits. Las salidas de un sumador de 8 bits podrían describirse, por ejemplo, como "s:8,C\_out".

##### Código del programa

Código del programa que va a ejecutar la aplicación externa.

##### Aplicación

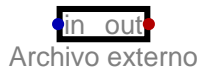
Indica qué aplicación usar.

##### Opciones GHDL

Opciones empleadas para procesar todos los pasos en GHDL

##### Opciones de IVerilog

Opciones que se usan para todos los pasos de procesamiento con IVerilog.



## 17.2. Archivo externo

Componente para ejecutar un proceso externo para calcular la función lógica. Se usa para especificar el comportamiento de un componente en VHDL o en Verilog. La simulación real del comportamiento debe hacerse con un simulador externo. De momento, sólo están soportados el simulador ghdl para VHDL y el Icarus Verilog para Verilog. ¡La etiqueta del componente debe coincidir con el nombre de la entidad o módulo! Exportable a VHDL/Verilog.

### Entradas

in

### Salidas

out

### Propiedades

#### Etiqueta

Nombre de este elemento.

#### Anchura

Anchura del símbolo si el circuito se usa como componente en otro circuito.

#### Entradas

Entradas del proceso externo. Es una lista de nombres de señales separadas por comas. Para cada nombre de señal, usando ":", puede indicarse el número de bits. Las entradas de un sumador de 8 bits podrían, por ejemplo, describirse como "a:8,b:8;C\_in".

#### Salidas

Salidas de un proceso externo. Es una lista de nombres de señales separadas por comas. Para cada nombre de señal, usando ":", puede indicarse el número de bits. Las salidas de un sumador de 8 bits podrían describirse, por ejemplo, como "s:8,C\_out".

#### Código del programa

El archivo que contiene el código del programa a ejecutar por la aplicación externa.

#### Aplicación

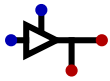
Indica qué aplicación usar.

#### Opciones GHDL

Opciones empleadas para procesar todos los pasos en GHDL

#### Opciones de IVerilog

Opciones que se usan para todos los pasos de procesamiento con IVerilog.



### 17.3. Control de pin

Lógica de control para un pin bidireccional. ¡Se precisa este componente sólo en el contexto de generación de VHDL o Verilog, con el fin de crear un puerto HDL bidireccional! Exportable a VHDL/Verilog.

#### Entradas

wr  
Dato que se va a mostrar

oe  
Activa la salida.

#### Salidas

rd  
Datos que se van a leer.

pin  
Conector para el pin real. Sólo se debería conectar aquí una salida simple.

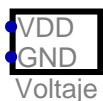
#### Propiedades

Bits de datos  
Número de bits de datos empleados.

Rotación  
Orientación del elemento en el circuito.

Reflejo  
Refleja el componente del circuito

## 18. Varios



### 18.1. Voltaje

No tiene función. Asegurar que VDD y GND estén conectados. Puede emplearse en circuitos 74xx para generar los pines para el suministro de corriente y comprobar el cableado correcto.

#### Entradas

VDD  
¡Debe estar conectado a VDD!

GND  
¡Debe estar conectado a GND!

#### Propiedades

Etiqueta

Nombre de este elemento.

Rotación

Orientación del elemento en el circuito.



## 18.2. Divisor bidireccional

Puede usarse para buses de datos, y simplifica la construcción de módulos de memoria en encapsulado DIL, por lo que la implementación del bus de datos se simplifica.

Entradas

OE

Activo, el valor del terminal de datos D común se propaga a la salida de bits D[i]. En caso contrario, los bits D[i] se propagan a la salida común D.

Salidas

D

Conexión común de datos.

D0

Bit de datos 0 del divisor.

Propiedades

Bits de datos

Número de bits de datos empleados.

Rotación

Orientación del elemento en el circuito.

Reflejo

Refleja el componente del circuito

Propagación

Configura la propagación de las entradas y salidas del circuito.



## 18.3. Reset

La salida de este componente está alta durante la inicialización del circuito. Después de que el circuito se ha estabilizado, la salida pasa a bajo. Si la salida está invertida, se comporta al contrario. Exportable a VHDL/Verilog.

Salidas

Reset

Salida de Reset.

Propiedades

Etiqueta

Nombre de este elemento.

salida invertida

Si se marca, la salida estará invertida.

Rotación

Orientación del elemento en el circuito.



#### 18.4. Pausa

Si se usa este componente en el circuito, se habilita el botón "Ejecutar hasta un Break", entre "Iniciar" y "Parar". Este botón detiene el circuito hasta que se detecta un flanco de subida en la entrada de este componente. Este elemento puede usarse para depurar por el reloj el circuito hasta un punto de Break. También se puede implementar un comando BRK ensamblador. Esto permite ejecutar un programa hasta el siguiente comando BRK. Esta función sólo se puede usar si se desactiva el reloj en tiempo real.

Entradas

brk

Detiene la simulación rápida si se detecta un flanco de subida.

Propiedades

Etiqueta

Nombre de este elemento.

Habilitado

Habilita o deshabilita este componente.

Ciclos de tiempos muertos

Si dicha cantidad de ciclos se alcanza sin señal de pausa, se crea un error.

Rotación

Orientación del elemento en el circuito.



#### 18.5. Parada

Un flanco de subida en la entrada para la simulación. Tiene el mismo efecto que presionar el botón "Stop" de la barra de herramientas.

Entradas

stop

Un flanco de subida para la simulación.

Propiedades

Etiqueta

Nombre de este elemento.

Salidas invertidas

Puedes seleccionar las entradas que estarán invertidas

Rotación

Orientación del elemento en el circuito.

Async

## 18.6. Temporizador asíncrono

Permite la configuración del temporizado de un circuito secuencial asíncrono, como una tubería de Muller. El circuito debe empezar en un modo de paso de puerta única y debe poder alcanzar un estado estable en el inicio. El circuito secuencial puede ser iniciado interactivamente o con una puerta de reset. No se permite el uso de un componente de reloj en este modo.

### Propiedades

- Iniciar el reloj de tiempo real

  - Si está habilitado, el reloj de simulación comienza cuando el circuito arranca

- Frecuencia/Hz

  - La frecuencia en tiempo real empleada para el reloj real.

## E Biblioteca

- 27c801:** 8 Mbit (1Mb x 8) UV EPROM
- 28c010:** 1-Megabit (128K x 8) Paged Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c16:** 16K (2K x 8) Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c64:** 64K (8K x 8) Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c256:** 256K (32K x 8) Paged Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c512:** 512K-Bit (64K x 8) CMOS Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 7400:** quad 2-input NAND gate
- 7401:** quad 2-input NAND gate with open-collector outputs
- 7402:** quad 2-input NOR gate
- 7403:** quad 2-input NAND gate with open-collector outputs, different pinout than 7401
- 7404:** hex inverter
- 7405:** hex inverter, open-collector output
- 7406:** hex inverter buffer, open-collector output
- 7407:** hex buffer, open-collector output
- 7408:** quad 2-input AND gate
- 7409:** quad 2-input AND gate with open-collector outputs
- 7410:** triple 3-input NAND gate
- 7411:** triple 3-input AND gate
- 7412:** triple 3-input NAND gate with open-collector outputs
- 7413:** dual 4-input NAND gate, Schmitt trigger
- 7414:** hex inverter, Schmitt trigger
- 7415:** triple 3-input AND gate with open-collector outputs
- 7416:** hex inverter buffer, open-collector output, same as 7406
- 7417:** hex buffer, open-collector output, same as 7407
- 7420:** dual 4-input NAND gate
- 7421:** dual 4-input AND gate
- 7425:** dual 4-input NOR gate with strobe
- 7427:** triple 3-input NOR gate
- 7428:** quad 2-input NOR buffer
- 7430:** 8-input NAND gate
- 7432:** quad 2-input OR gate
- 7434:** hex buffer
- 7440:** dual 4-input NAND buffer
- 7442:** 4-line BCD to 10-line decimal decoder
- 7447:** BCD to 7-segment decoder, active low
- 7448:** BCD to 7-segment decoder, active high
- 7451:** 2-input/3-input AND-NOR gate
- 7454:** 2-3-2-3-line AND NOR gate
- 7455:** 2 wide 4-input AND-NOR gate
- 7458:** dual AND OR gate
- 7474:** dual D-flip-flop
- 7476:** dual J-K flip-flops with preset and clear
- 7480:** Gated Full Adder with Complementary Inputs and Complementary Sum Outputs

**7482:** 2-bit binary full adder  
**7483:** 4-bit binary full adder  
**7483Real:** 4-bit binary full adder, real gates  
**7485:** 4-bit comparator  
**7486:** quad 2-input XOR gate  
**7489:** 64-bit RAM  
**7490:** asynchronous two - five - decimal addition counter  
**7493:** 4-bit Binary Counter. Connect QA to CKB and clock on CKA for full 4-bit counter.  
Connect QB to R1 and QD to R2 for a BCD counter.  
**74107:** dual J-K flip-flops with clear  
**74109:** Dual J-NOT-K flip-flop with set and reset; positive-edge-trigger  
**74112:** Dual J-K negative-edge-triggered flip-flop, clear and preset  
**74116:** dual 4-bit D-type latches  
**74125:** Quadruple bus buffer gates with 3-state outputs (active low output enable)  
**74126:** Quadruple bus buffer gates with 3-state outputs (active high output enable)  
**74133:** 13-input NAND gate  
**74138:** 3-line to 8-line decoder/demultiplexer, inverted out  
**74139:** dual 2-line to 4-line decoder/demultiplexer  
**74147:** 10-line to 4-line priority encoder  
**74148:** 8-line to 3-Line priority encoder  
**74150:** 4-line to 16-line data selectors/multiplexers  
**74151:** 3-line to 8-line data selectors/multiplexers  
**74153:** dual 4-line to 1-line data selectors/multiplexers  
**74154:** 4-line to 16-line decoders/demultiplexers  
**74157:** quad 2-line to 1-line data selectors/multiplexers  
**74160:** decimal synchronous counter, async clear  
**74161:** hex synchronous counter, async clear  
**74162:** decimal synchronous counter  
**74162Real:** decimal synchronous counter, real gates  
**74163:** hex synchronous counter  
**74164:** 8-bit parallel-out serial shift register, asynchronous clear  
**74165:** parallel-load 8-bit shift register  
**74166:** 8-Bit Parallel-In/Serial-Out Shift Register  
**74173:** quad 3-state D flip-flop with common clock and reset  
**74174:** hex D-flip-flop  
**74175:** quad D-flip-flop  
**74181:** 4-bit arithmetic logic unit  
**74182:** look-ahead carry generator  
**74189:** 64-Bit Random Access Memory with 3-STATE Outputs  
**74190:** Presettable synchronous 4-bit bcd up/down counter  
**74191:** Presettable synchronous 4-bit binary up/down counter  
**74193:** Synchronous 4-Bit Up/Down Binary Counter with Dual Clock  
**74194:** 4-Bit Bidirectional Universal Shift Register  
**74194real:** 4-Bit Bidirectional Universal Shift Register, Databook implementation.  
**74198:** 8-bit shift register  
**74238:** 3-line to 8-line decoder/demultiplexer  
**74244:** octal 3-state buffer/line driver/line receiver  
**74245:** octal bus transceivers with 3-state outputs  
**74247:** BCD to 7-segment decoder, active low, tails on 6 and 9  
**74248:** BCD to 7-segment decoder, active high, tails on 6 and 9



**74253:** dual tri state 4-line to 1-line data selectors/multiplexers  
**74257:** quad 2-line to 1-line data selectors/multiplexers (3-state output)  
**74260:** dual 5-input NOR gate  
**74266:** quad 2-input XNOR gate with open collector outputs  
**74273:** octal D-type flip-flop with clear  
**74280:** 9 bit Odd-Even Parity Generator-Checker  
**74283:** 4-bit binary full adder, alternative pinning  
**74299:** 8-Input Universal Shift/Storage Register with Common Parallel I/O Pins  
**74373:** octal transparent latches  
**74374:** octal positive-edge-triggered flip-flops  
**74377:** Octal D Flip-Flop with enable  
**74381:** 4-Bit Arithmetic Logic Unit with high-speed expansion  
**74382:** 4-Bit Arithmetic Logic Unit with ripple carry output  
**74540:** octal buffer/line driver, inverted  
**74541:** octal buffer/line driver  
**74573:** octal transparent latches, different pinout compared to 74373  
**74574:** octal positive-edge-triggered flip-flops, different pinout compared to 74374  
**74590:** 8-bit binary counter with tri-state output registers  
**74595:** 8-Bit Shift Registers with 3-State Output Registers  
**74670:** 3-state 4-by-4 Register File  
**74682:** 8-bit digital comparator  
**74688:** 8-bit identity comparator  
**74779:** 8-Bit Bidirectional Binary Counter with 3-STATE Outputs  
**74804:** hex 2-input NAND gate <https://www.ti.com/lit/ds/symlink/sn74as804b.pdf>  
**74805:** hex 2-input NOR gate <http://www.ti.com/lit/ds/symlink/sn54as805b.pdf>  
**74808:** hex 2-input AND gate <http://www.ti.com/lit/ds/symlink/sn54as808b.pdf>  
**74832:** hex 2-input OR gate <http://www.ti.com/lit/ds/symlink/sn54as832b.pdf>  
**744002:** dual 4-input NOR gate  
**744017:** Johnson decade counter with 10 decoded outputs  
**744075:** triple 3-input OR gate  
**747266:** quad 2-input XNOR gate  
**7440105:** 4-Bit x 16-Word FIFO Register  
**A623308A:** 8K X 8 BIT CMOS SRAM  
**RAM32Bit:** A 32-bit memory that allows byte access and can handle non-aligned memory addresses.