

Guía de uso del entorno EDA playground (v1.10)

El entorno de desarrollo *EDA playground* permite escribir, compilar, depurar y simular diseños Verilog (y trabajar con otros lenguajes). Es un entorno integrado, ejecutable en navegador, que incluye editor de texto y visor de formas de onda e incorpora varios simuladores.

Su acceso se realiza a través de la siguiente dirección web

<https://www.edaplayground.com/>

En esta guía se mostrarán los pasos imprescindibles para usar el entorno. Se recomienda usar una cuenta registrada para poder usar todas las funcionalidades del entorno.

1. Aspectos básicos del entorno de desarrollo

Una vez iniciado el entorno, se mostrará la siguiente pantalla.

The screenshot shows the EDA Playground web interface. The browser address bar displays 'edaplayground.com'. The interface includes a top navigation bar with 'Run' and 'Save' buttons, and a sidebar on the left with 'Languages & Libraries' and 'Tools & Simulators' sections. The main workspace is divided into two code editors: 'testbench.sv' and 'design.sv'. Below the editors is a 'Share' section with a title input field, a visibility dropdown set to 'Public', and a 'Save' button. A text area for a description is also visible. Four orange callout boxes with numbers 1, 2, 3, and 4 point to specific elements: 1 points to the 'Collaborate' button in the sidebar; 2 points to the 'design.sv' code editor; 3 points to the 'Other Libraries' dropdown in the sidebar; and 4 points to the description text area.

En esta pantalla hay varias ventanas que sirven para configurar el entorno (1), para escribir el diseño del circuito (2), para escribir el test de simulación (3) y para recibir mensajes del sistema (4). Se especifican a continuación más detalles para cada ventana:

Ventana 1: En esta sección se debe seleccionar el lenguaje y el simulador que se desea usar. En el caso de Verilog:

Testbench + Design = System Verilog/Verilog

Tools & Simulators = Icarus Verilog

*Run Options = si vamos a simular seleccionar *Open EPWave after run**

Ventana 2: En esta parte hay una pestaña llamada *design.sv* correspondiente a un archivo Verilog del mismo nombre, en la que se puede escribir directamente, en Verilog, el diseño del circuito. También se puede pulsar el signo + que aparece arriba y en este caso, se subirá al entorno un archivo ya escrito o se creará uno nuevo con el nombre que se elija. En ese caso saldrá una nueva pestaña en esa zona y deberemos indicar que se incluya el contenido de ese archivo dentro del archivo *design.sv*. Esto se consigue usando la directiva de compilación ``include` de la forma que se muestra en esta imagen:



```
1 `include "dec16.v"
2 `include "compdec16.v"
3
4
```

Ventana 3: En esta parte se escribe el *testbench* del circuito en Verilog. Como en la ventana 2, puede hacerse directamente, en la única pestaña que hay, de nombre *testbench.sv* o se puede utilizar el signo + para subir más archivos al proyecto y luego añadir las directivas ``include` que se estimen convenientes en el archivo *testbench.sv*.

Ventana 4: Esta es la parte de diálogo, donde aparecerán los mensajes de error cuando los haya.

2. Escritura y compilación de un diseño.

Como se ha comentado en el apartado 1, se debe escribir el código Verilog del circuito en la ventana 2 de la forma que se ha indicado. A continuación, se pulsa sobre el comando Run (arriba a la izquierda) y se observa la ventana inferior (ventana 4) para ver si el código tenía errores de sintaxis:

```
testbench.sv  + SV/Verilog Testbench
design.sv voter.v * + SV/Verilog Design
1 `timescale 1ns / 1ps
2
3 module voter(
4 output z,
5 input a,
6 input b,
7 input c
8 );
9 assign z = a&b | a&c | b&c;
10 endmodule
11 // voter
12
```

error

```
Log Share
[2023-02-01 06:42:16 EST] iverilog '-Wall' design.sv testbench.sv && unbuffer vvp a.
./voterf.v:5: syntax error
I give up.
Exit code expected: 0, received: 1
Done
```

Una vez depurados los errores de sintaxis, habrá que validar el diseño, es posible que no haya ninguna incorrección en el lenguaje pero que el circuito descrito no se comporte según lo planificado. Para ello habrá que comenzar con la **simulación**.

3. Simulación de un diseño.

En este caso se debe escribir el código Verilog del testbench en la ventana 3 de la forma que se ha indicado en el apartado 1. Nuevamente se pulsa sobre el comando Run (arriba a la izquierda) y se observa la ventana inferior (ventana 4) para ver si el código tenía errores de sintaxis. También se debe seleccionar en la ventana 1 (configuración) la siguiente opción:

Run Options = seleccionar Open EPWave after run

Si no hay errores de sintaxis en el *testbench* se abrirá la ventana de ondas con el resultado de la simulación:

