

Laboratorio 3

Arquitectura y Organización de Computadores

Profesores: Mauricio Solar y Viktor Tapia
Ayudantes de cátedra: Javier Rojas y Mauricio Cortés
Ayudantes de tareas: Joaquín Montes y Benjamín López

5 de mayo de 2023

1 Reglas generales

Para la siguiente tarea se debe utilizar la plataforma EDA Playground¹ para diseñar un circuito y un testbench que cumplan con los requerimientos de la sección 2 utilizando el lenguaje de descripción de hardware SystemVerilog. Se exigirá que el formato del diseño y el testbench se presente de la forma más limpia y ordenada posible. Deberá incluir un README con la identificación de los estudiantes que desarrollaron la tarea, además de cualquier supuesto utilizado.

Para utilizar EDA Playground deben logearse en la página, y se recomienda que usen su correo institucional para esto. Luego deben configurar la plataforma usando la ventana de la izquierda con los valores de la tabla 1.

Testbench + Design	SystemVerilog/Verilog	UVM/OVM	None
Other Libraries	None	Enable TL-Verilog	Desactivado
Enable Easier UVM	Desactivado	Enable VUnit	Desactivado
Tools and Simulators	Aldec Riviera Pro 2022.04	Open EPWave after run	Activado
Download files after run	Desactivado		

Cuadro 1: Configuración de EDA Playground

2 Tarea

El laboratorio 3 consiste en el uso de lenguajes de descripción de hardware para el diseño de un circuito capaz de cumplir una serie de requisitos, y el desarrollo de un testbench que evalúe el rendimiento del circuito. El problema a resolver y los requerimientos de la solución se presentan a continuación.

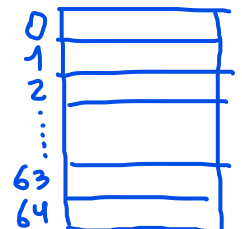
2.1 Requerimientos

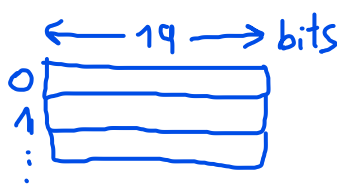
Se solicita diseñar un circuito combinado y un testbench que permitan satisfacer los siguientes requisitos.

- El diseño del circuito debe separar los componentes en módulos distintos, al menos uno por cada requerimiento.
- El diseño del circuito debe incluir un único módulo central que contenga un bus que conecte una instancia de cada uno de los otros módulos.
- El circuito debe leer las instrucciones y datos desde una memoria ROM de 64 espacios, la que se debe indexar usando un contador binario de 6 bits

read only memory

¹Disponible en <https://edaplayground.com/>





- Cada instrucción de la memoria consiste en un número de 19 bits, donde los primeros 3 bits corresponden al código de la operación a realizar, los siguientes 8 bits corresponden al valor inmediato A, y los últimos 8 bits corresponden al valor inmediato B.
- El circuito debe implementar una ALU que ejecute las 8 operaciones de la tabla 2.

Código	Operación	Descripción
000	Suma	$A + B$
001	Resta	$A - B$
010	Bitshift Izquierda	$A \ll B$
011	Bitshift Derecha	$A \gg B$
100	Composición	$A \& B$
101	Disyunción	$A B$
110	Exclusión	$A \oplus B$
111	Negación	$\sim A$

000 00000000 00000000
OP A B

Cuadro 2: Operaciones de la ALU

- El resultado de la operación debe entregarse por pantalla usando el formato

<INSTRUCCION> <OPERACION> <INMEDIATO A> <INMEDIATO B> <RESULTADO>

donde cada término entre llaves anguladas corresponde al valor del concepto.

- La suma y la resta deben implementarse usando un Prefix Adder capaz de manejar ambos valores inmediatos. Los bitshift deben rellenar el espacio generado tras el desplazamiento con un 0. Las operaciones lógicas deben aplicarse sobre cada bit (bitwise). El inmediato B se ignora en el caso de la negación.

2.2 Formato de los circuitos

El circuito debe dividirse en varios subcircuitos, cada uno implementado en un módulo distinto. El subcircuito principal debe llamarse `main`, y el resto de los subcircuitos deben tener nombres descriptivos. El subcircuito principal debe tener una disposición similar a la figura 1.

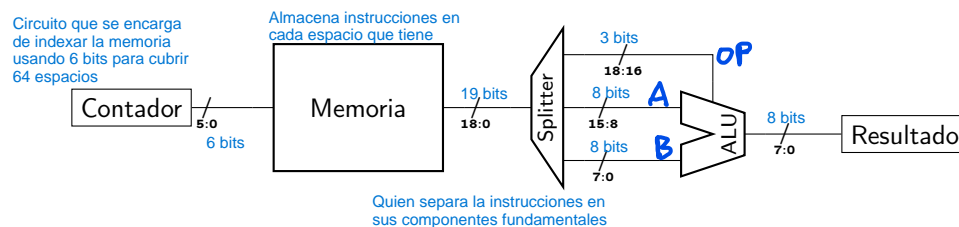


Figura 1: Ejemplo de cómo se podría ver el módulo `main`

2.3 Consideraciones

- La entrada de datos se debe hacer mediante una memoria ROM indexada por un contador.
- La salida de datos debe hacerse a través de la terminal de EDA Playground usando el macro `$display` y siguiendo el formato del enunciado. También puede hacer uso de la ventana de EPWave para visualizar los valores del circuito.
- Se debe crear un solo archivo de diseño y un solo archivo de testbench. El archivo de diseño debe contener todos los módulos de la tarea, y el archivo de testbench debe contener un único módulo llamado `testbench` que implemente el diseño en un ambiente donde se pueda comprobar su funcionalidad.

1 dígito hexadecimal = 4 bits
5 dígitos hex = 20 bits, se descarta el último

2.4 Ejemplos

Entrada: 19 bits	3 bits	8 bits	8 bits	8 bits
Instrucción (Hex)	Operación	Inmediato A	Inmediato B	Resultado
01713	000 (Suma)	00010111 (23)	00010011 (19)	00101010 (42)
1034C	001 (Resta)	00000111 (7)	01001100 (76)	10111011 (-69)
21F05	010 (Bitshift Izq.)	00011111	00000101 (5)	11100000
31F02	011 (Bitshift Der.)	00011111	00000010 (2)	00000111
45D52	100 (Composición)	01011101	01010010	01010000
55D52	101 (Disyunción)	01011101	01010010	01011111
65D52	110 (Exclusión)	01011101	01010010	00001111
75D52	111 (Negación)	01011101	Ignorado	10100010

Nota: El valor que debe entregarse por pantalla debiera verse similar a una de las filas de la tabla de ejemplos.

3 README

El archivo README debe contener como mínimo el nombre, rol y paralelo de los estudiantes que desarrollaron la tarea, además de cualquier supuesto utilizado e indicaciones que encuentren relevantes para la ejecución de la solución.

4 Consideraciones generales

- Se debe trabajar en parejas. La solución al laboratorio debe entregarse en Aula a más tardar el día 19 de mayo de 2023 a las 23:59 horas. Se descontarán 5 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de tareas.
- La tarea debe realizarse en EDA Playground. Se recomienda que se familiarice rápidamente con la plataforma, y ante cualquier duda consulte con sus compañeros o directamente con los ayudantes lo antes posible. El único responsable si no acude a alguien para resolver sus dudas a tiempo es usted.
- Puede utilizar una cantidad arbitraria de módulos, siempre y cuando cada uno tenga un nombre descriptivo e incluya como mínimo el subcircuito main.
- La entrega considera solo dos archivos `design.sv` y `testbench.sv` junto con el README. Los archivos deberán ser comprimidos y enviados juntos en un archivo .zip de nombre LAB3_R0L1_R0L2.
- Todas las preguntas respecto al laboratorio deben hacerse a través del foro de consultas en Aula para que todos los estudiantes puedan beneficiarse de cualquier respuesta. **Se responderán consultas hasta 48 horas antes de la fecha de entrega.**
- Si no se entrega un README, o si su circuito no funciona, la nota del laboratorio es 0 hasta la corrección.
- Se descontarán 50 puntos por no respetar el formato de entrega. Se descontarán 25 puntos por día de atraso o fracción.