



GRADO EN INGENIERÍA SOFTWARE

Arquitectura de Computadores

2024-2025

PROYECTO I Logisim

Este es el enunciado de la Práctica de Logisim I relacionada con circuitos combinacionales

Evaluación y condiciones: Se recuerda a los alumnos que esta práctica se realiza por parejas que deben comunicadas a los respectivos profesores. La nota de esta práctica cuenta la mitad del 30% de la nota final que está asignado a prácticas/proyectos. Es decir cuenta un 15% de la nota final.

Es necesario obtener mínimo un 4 mínimo por separado en cada uno de los proyectos y para aprobar la asignatura y 4,5 de media entre esta práctica y la segunda evaluable

ENUNCIADO:

En esta práctica se pide desarrollar una pequeña ALU (Unidad Aritmético Lógica).

Se empleará el software logisim, y la forma de construirlo será modular (construyendo subcircuitos y probándolos). Es decir, se pedirán realizar una serie de subcircuitos con unas condiciones concretas, los cuales agruparemos en una librería logisim. Finalmente generaremos un circuito principal que emplee los subcircuitos anteriores con el fin de obtener una pequeña ALU.

A CONTINUACIÓN, SE DETALLAN LOS SUBCIRCUITOS QUE HAY QUE CREAR:

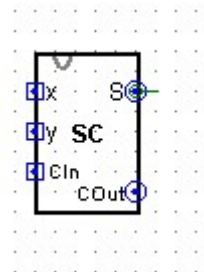
Subcircuito 1: Sumador completo de 1 bit (SumCom1bit):

Este circuito es el visto detalladamente en clase el cual realiza la suma de dos números de un bit.

Entradas	<ul style="list-style-type: none">• X: primer bit a sumar• Y: segundo bit a sumar• CIn: Acarreo / Carry de entrada
----------	--

Salidas	<ul style="list-style-type: none"> S : suma de ambos bits COut: el Acarreo / Carry de salida tras efectura la suma

Cuando el circuito esté construido habrá que salvarlo dentro del proyecto como un subcircuito. Se pide a los alumnos que también configuren de forma adecuada el aspecto del subcircuito de tal manera que se presente algo similar a:

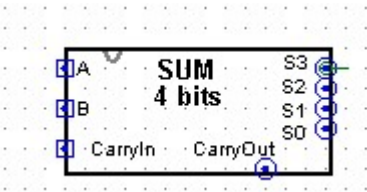


Subcircuito 2: Sumador completo de 4 bits (Sum4bits):

Este circuito hará uso del circuito anterior *SumCom1bit*. Básicamente lo que hace es sumar 2 números de 4 bits.

Entradas	<ul style="list-style-type: none"> A: primer número de 4 bits a sumar B: segundo número de 4 bits a sumar CarryIn: Acarreo / Carry de entrada (1 bit)
Salidas	<ul style="list-style-type: none"> S3 : bit más significativo del número resultante de la suma de 4 bits S2 :Siguiente bit al más significativo S3 S1: Siguiente bit a S2 S0: Bit menos significativo del número de 4 bits resultante tras la suma CarryOut: el Acarreo / Carry de salida (1 bit) tras efectuar la suma

Cuando el circuito esté construido habrá que salvarlo dentro del proyecto como un subcircuito. Se pide a los alumnos que también configuren de forma adecuada el aspecto del subcircuito de tal manera que se presente algo similar a:



Subcircuito 3: Unidad lógica de 1 bit (UL1bit):

Como su nombre indica es un circuito en el cual vamos a implementar las siguientes operaciones lógicas con una serie de restricciones:

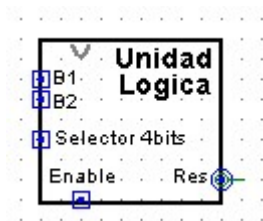
Función lógica	Restricciones	Bits que representan la operacion
NOT	Realizarlo exclusivamente con puertas NAND (nota: esta	000

	funcionalidad solo afecta al al bit del primer número al ser una operación sobre u único número)	
AND	Realizarlo exclusivamente con puertas NOR	001
OR	Realizarlo exclusivamente con puertas AND y NOT	010
NAND	Realizarlo exclusivamente con puertas OR y NOT	011
NOR	Realizarlo directamente con la puerta NOR de logisim	100
XOR	Realizarlo exclusivamente con puertas AND, OR y NOT	101
XNOR	Libertad para realizarlo con las puertas que considere, pero sin poder utilizar ni la puerta XNOR ni a XOR de logisim	110
Expresión Propia	Inventar una combinación cualquier de operaciones sobre dos variables. Indicar su fórmula algebraica simplificada e implementarla. Mínimo 3 puertas distintas, mínimo 3 sumandos o 3 productos ya que puede ser una suma de productos o un producto de sumas.	111

Entradas	<ul style="list-style-type: none"> • BN0: Bit del primer número • BN1: Bit del segundo número • Selector: es una entrada de tres bits que se emplea para indicar qué operación es la que va a realizar la unidad lógica sobre los bits de entrada • Enable: es una entrada de un bit que se utilizará para “activar” circuitos que se emplean dentro de éste. Recordad que hay circuitos combinacionales que necesitan un 1 en una entrada para activarlos.
Salidas	<ul style="list-style-type: none"> • R: Muestra el resultado de aplicar la operación seleccionada a los bits de entrada.

Pista: Habrá que pensar en algún tipo de circuito combinacional famoso que permita a partir de varias entradas y otras de selección , sacar una única salida. Se permite en este caso usar el componente que tiene logisim ya hecho, que implementa esta funcionalidad.

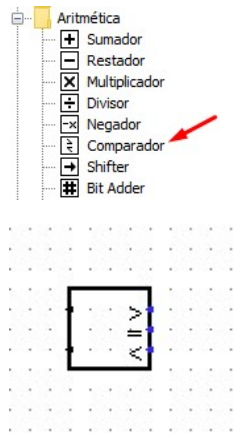
Cuando el circuito esté construido habrá que salvarlo dentro del proyecto como un subcircuito. Se pide a los alumnos que también configuren de forma adecuada el aspecto del subcircuito de tal manera que se presente algo similar a:



Subcircuito 4: Unidad aritmética de 4 bits (UA4bit):

Como su nombre indica es un circuito en el cual vamos a implementar las siguientes operaciones aritméticas con una serie de restricciones:

Función	Restricciones	Bits que
---------	---------------	----------

aritmética		representan la operacion
Complemento a 1	Realizarlo con puertas básicas exclusivamente. Aunque la unidad aritmética reciba dos números de 4 bits de entrada, solo se calcula el complemento a 1 de la primera entrada. Se permite el uso de puertas con varias entradas o incluso con una sola entrada de 4 bits.	000
Complemento a 2	Empleando el circuito anterior complemento a 1 y para sumarle uno usar el sumador completo de 4 bits. Aunque la unidad aritmética reciba dos números de 4 bits de entrada, solo se calcula el complemento a 2 de la primera entrada.	001
Desplazamiento de un bit la izquierda.	Equivale multiplicar por 2. Solo aplica a la primera entrada. Por ejemplo si en la primera entrada tenemos 0101 que equivale a 5 decimal, el desplazamiento da 1010 que equivale a 10 decimal	010
Desplazamiento de un bit la derecha.	Equivale dividir por 2. Solo aplica a la primera entrada. Por ejemplo si en la primera entrada tenemos 1010 que equivale a 10 decimal, el desplazamiento da 0101 que equivale a 5 decimal	011
Sumador	Simplemente llama con las dos entradas al sumador de 4 bits	100
Restador	Calcula primero el complemento a 2 de la segunda cifra de 4 bits. Seguidamente llama al sumador de 4 bits que hemos implementado y suma el primer número con el complemento a 2 del segundo que hemos calculado. El resultado estará en complemento a 2.	101
Bit de paridad de 1	Solo aplica a la primer cifra. Si el número de 1's de la primera cifra es par o la cifra es 0000 entonces esta operación dará como resultado 0001. En caso contrario dará 0000. Pista: esta función equivale a hacer la operación de XNOR entre todos los bits del primer número.	110
Comparador	<p>Compara el primer número con el segundo.</p> <p>Si el primero es mayor que el segundo el resultado será 0100</p> <p>Si son iguales entonces el resultado será 0010</p> <p>Si el primero es menor que el segundo el resultado será 0001</p> <p>Para implementar esta funcionalidad los alumnos pueden optar por cualquiera de las siguientes opciones:</p> <p>1. Usar el comparador que trae logisim implementado:</p>  <p>Pero debéis de configurarlo de esta forma:</p>	111

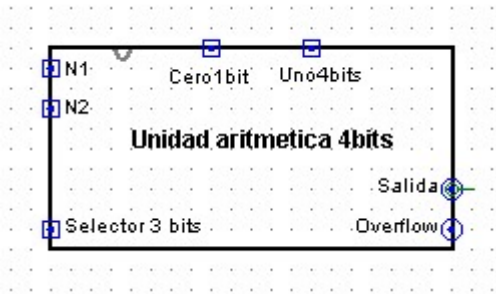
Selection: Comparador	
Bits De Datos	4
Tipo Numérico	Sin Signo

2. Crear vuestro propio comparador.

En cualquier caso se recomienda a los alumnos intentar hacer su propio comparador.

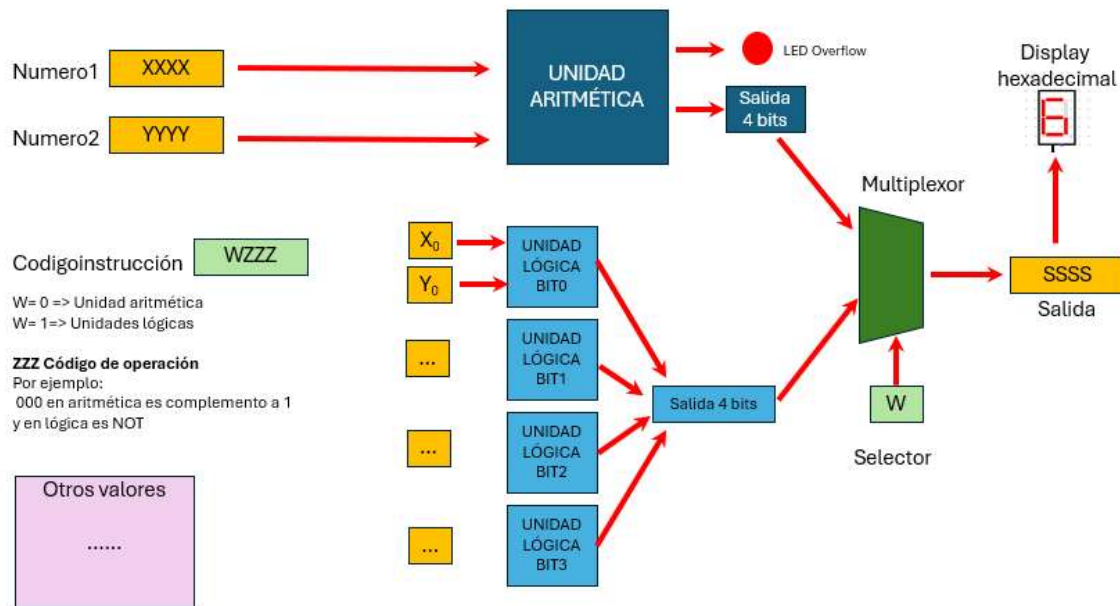
Entradas	<ul style="list-style-type: none"> N1: Primer número de 4 bits N2: Segundo número de 4 bits Selector: es una entrada de tres bits que se emplea para indicar qué operación es la que va a realizar la unidad aritmética sobre los números de entrada Cero1bit: esta entrada recibe un 0 de un bit que se puede usar dentro del circuito por ejemplo para poner el cero cuando desplazamos una cifra o para pasárselo a una puerta NOT y que nos de un 1 que actúe como Enable de cierto circuitos que podemos usar. Uno4bits: Corresponde a una entrada que siempre pasaremos como 0001 y puede servir por ejemplo para sumarle uno al complemento a 1 y obtener por tanto el complemento a 2.
Salidas	<ul style="list-style-type: none"> Salida: Muestra con 4 bits el resultado de aplicar la operación seleccionada a los números de entrada de entrada. Overflow: En este circuito tendremos que utilizar varios sumadores e 4 bits. Si cualquiera de ellos devuelve carryout = 1 esta salida devolverá un 1 indicando que una operación interna ha dado overflow

Cuando el circuito esté construido habrá que salvarlo dentro del proyecto como un subcircuito. Se pide a los alumnos que también configuren de forma adecuada el aspecto del subcircuito de tal manera que se presente algo similar a:



Circuito principal: Nuestra ALU

Es el circuito en el cual importaremos la librería de subcircuitos y los usaremos para implementar la funcionalidad de la ALU. El siguiente esquema presente ser una ayuda a los alumnos a la hora de colocar los subcircuitos en el circuito principal. No es obligatorio seguirlo pero se valorará positivamente tener un circuito limpio, claro y si fuese necesario emplear “túneles” que eviten una maraña de cables:



Pistas:

- El código de instrucción es un número de 4 bits. Si el bit más significativo vale cero queremos hacer una operación aritmética. Si vale 1 queremos hacer una operación lógica. Los tres bits restantes son los bits de selección de cada una de las unidades e indican qué operación queremos realizar.
- Hay que mostrar un led rojo que se encenderá si hay overflow.
- Se recomienda a los alumnos que vayan probando cada circuito por separado y luego lo vayan montando paso a paso. Montar circuitos se asemeja a programar en el sentido de que el famoso “divide y vencerás” es muy útil y modularizar funciones/subcircuitos ayuda a compartimentar la funcionalidad y revisar errores.
- El visualizador hexadecimal es parecido al de 7 segmentos pero más sencillo. Sólo necesita como entrada un número de 4 bits:



- Se recomienda que haya una sección donde establecer aquellos valores que vayan a necesitarse por diversos circuitos como el primer número, o el 1 de 4 bits, etc. Se recomienda el empleo de túneles que harán que el código sea más legible. Es importante que el reloj sea único e el circuito principal.

Ejemplo:



NORMAS GENERALES

- Si hay que inicializar valores por ejemplo a 0001 para poder sumar a un número en complemento a 1 y obtener así complemento a 2 también se incluirán en dicha zona del circuito principal.
- Se agradecen textos a modo de comentarios y/o explicaciones. Se puntuarán dentro del apartado de claridad, presentación y limpieza.
- La elección de puertas lógicas a usar, su número de entradas y tipo de biestables queda a criterio de los alumnos. No hay restricción al respecto, salvo que se indique expresamente, valorándose positivamente la claridad y sencillez.
- Cualquier duda de si se puede emplear algún elemento de logisim o librería adicional debe ser consultada con el profesor.
- Si uno de los integrantes del grupo no participa en el video explicativo, se considerará que dicho alumno no ha colaborado siendo calificado con un 0 en dicha práctica.
- El profesor se reserva el derecho de llamar a cualquiera de los grupos y preguntar a cualquier integrante sobre cualquier funcionalidad. En el caso de no saber responder esto afectará negativamente a la nota del trabajo en grupo.
- En caso de detectarse copia entre dos prácticas ambas (tanto el que copia como el copiado) serán calificados con un 0.

ENTREGABLES Y RÚBRICA DE EVALUACIÓN

La entrega de esta práctica se realizará en Blackboard exclusivamente por un solo miembro integrante del grupo y dentro de la fecha tope señala por el profesor. La entrega fuera de plazo conllevará una penalización en la nota.

Elementos a entregar:

- 1 fichero .circ(ficheros de logisim) con la librería de subcircuitos implementada
- Otro fichero .circ con el circuito principal implementado e importando la librería anterior
- Un pdf que debe contener:
 - o nombre y apellidos de los integrantes del grupo
 - o la siguiente tabla rellena indicando si se ha implementado o no cada apartado. Si se ha implementado y ha habido problemas y los habéis solucionados se incluye una breve explicación de cómo se ha hecho. En caso de que no se hay implementado hay que justificar el problema que os ha impedido hacerlo.
 - o Link a un video corto (no más de 5 minutos subido a onedrive, youtube o similar. NO SUBIR A BLACKBOARD) en el cual tiene que aparecer:

- La voz de ambos alumnos integrantes del equipo
 - La pantalla de un ordenador donde se vaya viendo la ejecución del circuito. Si hubiera que parametrizar inicialmente los valores de algunas entradas habría que explicarlo para que el profesor sea capaz de reproducir la ejecución.
 - Explicación y prueba de cada subcircuito y circuito principal. Hay que demostrar que se domina la práctica.
 - Mostrar en pantalla la tabla inferior que se incluye en el pdf.
- Finalizar con conclusiones finales explicando lo que os ha llamado más la atención, lo que os ha resultado más difícil etc.

La siguiente tabla muestra cómo se va a valorar esta práctica:

Elemento	Puntuación	Hecho/No Hecho	Problemas encontrados y cómo se solventaron si se arreglaron
Subcircuito 1: Sumador completo de 1 bit (SumCom1bit):	0,5 ptos		
Subcircuito 2: Sumador completo de 4 bits (Sum4bits):	0,5 ptos		
Subcircuito 3: Unidad lógica de 1 bit (UL1bit):	2,5 ptos		
Subcircuito 4: Unidad aritmética de 4 bits (UA4bit):	2,5 ptos		
Circuito Principal ALU	2 ptos		
Video	1 pto		
Presentación, limpieza y claridad	1 pto		