

Fecha de asignación: 21 de abril de 2025

Fecha de entrega: 23:59 horas del 5 de mayo de 2025

1. Objetivo

- Emplear los conocimientos teóricos adquiridos en el curso en el proceso de diseño de sistemas digitales.
- Emplear herramientas de software para el diseño y la simulación de sistemas digitales.

2. Descripción

En la computación paralela el algoritmo de reducción es una técnica de paralelización ampliamente usada para la reducción de la complejidad computacional de algunos procesos con vectores de datos. Ejemplos de estos algoritmos son hallar la suma de todos los elementos del vector, hallar el valor menor, el valor mayor, etc. La complejidad computacional se reduce de $O(n)$ a $O(\log_2(n))$.

El proceso de la reducción se realiza por fases reutilizando los mismos recursos computacionales entre ellas. A lo largo del desarrollo de la solución de esta práctica se debe emplear un estilo de diseño estructural y jerárquico, en el que se diseñan bloques básicos que luego son instanciados para crear otros más complejos y de mayor nivel en la jerarquía de diseño.

El laboratorio debe ser desarrollado empleando la herramienta [Logisim Evolution v.3.9.0](#). Su biblioteca de componentes podrá ser usada con total libertad, excepto para aquellos componentes para los que se fije algún tipo de restricción sobre los bloques funcionales que puede emplear.

En esta práctica, cada equipo de trabajo debe diseñar y simular un sistema digital que corresponda a la implementación de un algoritmo de reducción. La Figura 1 muestra un ejemplo de cómo opera este algoritmo.

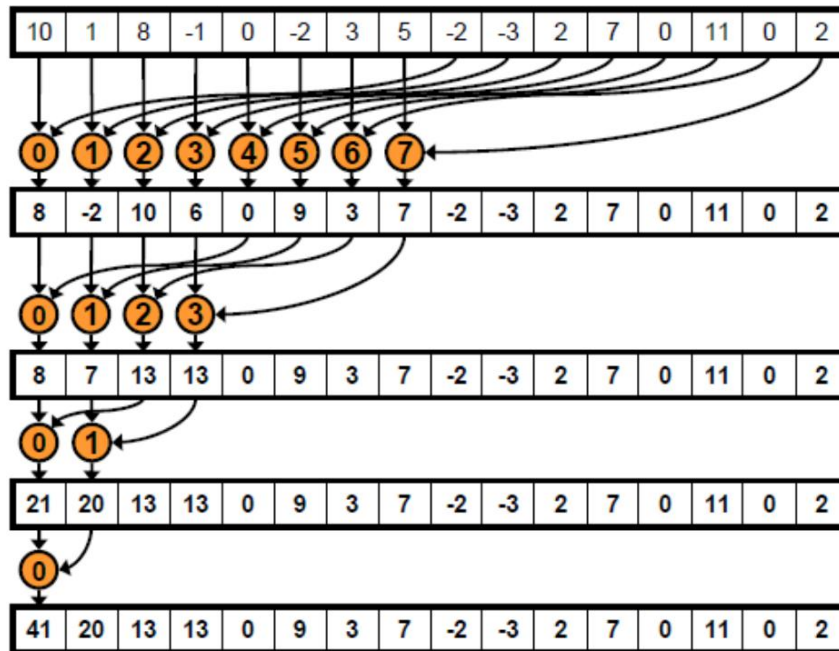


Figura 1. Reducción para hallar la sumatoria de un vector.

Para un arreglo de n valores, se necesitan $n/2$ unidades computacionales (**Circuitos**). Cada unidad toma 2 valores y los opera, el resultado es almacenado en la posición correspondiente. El ejemplo de la figura halla la sumatoria de los 16 valores del arreglo en 4 fases, usando a lo sumo 8 sumadores en paralelo. El sumador 0 toma las posiciones 0 y 8 y almacena el resultado en la posición 0. El sumador 1 opera las posiciones 1 y 9 y almacena el resultado en la posición 1 y así con los demás. Este ejemplo tiene un patrón de ordenamiento $0 \& n/2$, $1 \& (n/2+1)$.

Cada equipo de trabajo debe implementar un sistema de reducción para un arreglo de 32 datos de 8 bits. La operación a realizar es asignada a cada equipo, tal como se establece en la sección 3 de esta guía. La arquitectura del sistema debe ser definida por el equipo de trabajo. La construcción del bloque básico de operación deberá seguir un diseño jerárquico comenzando con un operador de números de 1 bit. Este último tendrá que ser diseñado empleando solamente los operadores lógicos básicos: AND, OR y NOT.

El sistema bajo diseño debe ofrecer la posibilidad de visualizar el contenido de los elementos del arreglo de datos mediante *displays* de siete segmentos, se debe visualizar además el número de la fase que se está ejecutando. El recorrido por las diferentes fases estará gobernado por una Máquina de Estados Finito (MEF). Esta MEF tendrá que ser diseñada empleando el tipo de *flip flop* que corresponda de acuerdo con el identificador del equipo de trabajo, tal como se establece en la sección 3 de esta guía. Si en algún momento el sistema detecta un resultado anómalo tal como un overflow en operaciones de suma o resta, la reducción se aborta y se avisa al usuario. El sistema digital debe disponer de una señal de *reset* asíncrona global.

El componente de visualización debe ser diseñado como un sistema combinacional, en el que los decodificadores que puedan ser necesarios tendrán que ser diseñados a partir de una tabla de verdad, las expresiones mínimas obtenidas con mapas de Karnaugh, y serán implementados sólo con compuertas AND, OR y NOT. La convención para ilustrar los dígitos decimales en un *display* de siete segmentos se muestra en la Figura 3.



Figura 3. Convención para ilustrar los dígitos decimales en un display de siete segmentos

Cualquier componente **secuencial** que el equipo quiera agregar a su sistema deberá ser diseñado a partir de una máquina de estados finitos, empleando el tipo de *flip flop* que le corresponda al equipo de trabajo.

Durante el proceso de diseño del sistema de reducción pueden presentarse diversas situaciones ante las cuales cada equipo debe tomar decisiones. En todos los casos, las decisiones de diseño estarán orientadas por el ingenio y la creatividad del equipo de trabajo, y deben ser ampliamente explicadas y justificadas.

El sistema digital tiene que ser simulado exhaustivamente para verificar su correcto funcionamiento.

3. Asignaciones para los equipos de trabajo

De acuerdo con el identificador asignado a cada equipo de trabajo, en la Tabla 1 se fija la operación a realizar, el patrón de ordenamiento y el tipo de *flip flop* a emplear para construir el contador que permitirá segmentar el proceso de reducción (y cualquier otro componente secuencial que precisen).

El número de asignación para cada equipo se calcula como (*Número del equipo* % 16).

Tabla 1. Asignaciones para los equipos de trabajo

# de asignación	Operación	Flip flop	Patrón de ordenamiento
0	Suma	SR	$0 \& n/2, 1 \& (n/2+1), \dots$
1	Menor	SR	$0 \& n/2, 1 \& (n/2+1), \dots$
2	Mayor	SR	$0 \& n/2, 1 \& (n/2+1), \dots$
3	Menor	SR	$0 \& n/2, 1 \& (n/2+1), \dots$
4	Suma	JK	$0 \& n/2, 1 \& (n/2+1), \dots$
5	Menor	JK	$0 \& n/2, 1 \& (n/2+1), \dots$
6	Mayor	JK	$0 \& n/2, 1 \& (n/2+1), \dots$
7	Menor	JK	$0 \& n/2, 1 \& (n/2+1), \dots$
8	Suma	SR	$0 \& 1, 2 \& 3, \dots$
9	Menor	SR	$0 \& 1, 2 \& 3, \dots$
10	Mayor	SR	$0 \& 1, 2 \& 3, \dots$
11	Menor	SR	$0 \& 1, 2 \& 3, \dots$
12	Suma	JK	$0 \& 1, 2 \& 3, \dots$
13	Menor	JK	$0 \& 1, 2 \& 3, \dots$
14	Mayor	JK	$0 \& 1, 2 \& 3, \dots$
15	Menor	JK	$0 \& 1, 2 \& 3, \dots$

4. Entrega

El equipo de trabajo deberá escribir un reporte que contenga los siguientes elementos:

- Una descripción completa del proceso de diseño: diagramas arquitectónicos, decisiones de diseño, diagramas de estados, tabla de estados y salidas, tablas de verdad, mapas de simplificación, funciones lógicas minimizadas, esquemáticos de los circuitos. Se aceptan imágenes escaneadas o fotografías siempre que sean claras.
- Resultados de la simulación exhaustiva del sistema, mostrando que su diseño realiza correctamente la funcionalidad solicitada.
- Observaciones, análisis de resultados y conclusiones.

Este documento debe estar en formato **PDF** y ser subido a la plataforma, acompañado de los archivos de diseño de **Logisim Evolution**, antes del cierre del plazo de entrega. El informe tiene un peso del 30% en la calificación global de la práctica.

Cree un archivo comprimido que incluya el reporte y los archivos de diseño de **Logisim Evolution** como se describe a continuación:

- Reporte:** archivo con extensión .pdf
- Archivos fuente:** .circ.

El nombre del archivo comprimido debe tener el siguiente formato:
eqXX_primerNombreApellidointegrante1_primerNombreApellidointegrante2.zip.

Ejemplo: eqXX_AnaPerez_LuisSuarez.zip.

5. Evaluación y Sustentación

Ambos miembros del equipo de trabajo deben demostrar un dominio completo del desarrollo de la práctica, mediante un video explicativo con una duración máxima de diez minutos, en el que intervengan de manera equitativa.

La entrega del video se debe realizar como un enlace de acceso al mismo incluido en el informe, y no como un archivo adjunto. El profesor podrá citar a algunos equipos de trabajo para que respondan preguntas que permitan aclarar o ampliar la información suministrada en el video con la sustentación. La sustentación tiene un peso del **70%** en la calificación global de la práctica.

NOTA: No cumplir con alguno de los requerimiento solicitados (Informe formato PDF, adjuntar archivos del Logisim, adjuntar enlace del video dentro del informe) puede generar la no calificación de la práctica.

6. Bibliografía

- Notas de Clase
- Computer organization and design. The hardware/software interface, 5th ed.