

## Laboratorio 3: Lógica Combinacional y Aritmética I

### Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Procedimiento . . . . .	1
<b>2. Investigación</b>	<b>2</b>
<b>3. Ejercicios Prácticos</b>	<b>2</b>
3.1. Problema 1 . . . . .	2
3.2. Problema 2 . . . . .	3

### 1. Introducción

Un circuito combinacional es aquel en que las salidas solamente dependen combinaciones de las entradas y no del tiempo. Esta característica hace que los sistemas combinacionales sean en general rápidos, ya que no requieren ningún tipo de sincronización con señales de reloj. En sistemas digitales complejos, como los microprocesadores, los circuitos de lógica combinacional juegan un papel fundamental. Es por esto que en este laboratorio se realizará el diseño de una Unidad Lógico-Aritmética (ALU), componente esencial en la etapas de ejecución de cualquier microprocesador moderno.

La ALU es la unidad encargada de la ejecución de las instrucciones aritméticas y lógicas en un procesador. Dichas instrucciones requieren tiempos de ejecución cortos, para tener máximo desempeño en el procesador y es por esta razón por la que en general se requiere de una lógica combinacional en su implementación.

En este laboratorio el estudiante aplicará los conceptos de lógica combinacional en el diseño de circuitos digitales lógicos y aritméticos con un lenguaje de descripción de hardware. Además, se abordarán asuntos de tiempo relacionados con lógica combinacional, como lo son los tiempos de propagación y contaminación así como la ruta crítica.

#### 1.1. Procedimiento

A lo largo del laboratorio se desarrollarán dos ejercicios que guiarán a los estudiantes por el proceso de diseño, implementación y verificación de una ALU parametrizable de N bits, cuya especificación se detallará más adelante. Adicionalmente, se investigará sobre problemas y consideraciones de tiempo en diseños con lógica combinacional. Para el desarrollo de este laboratorio se deberán acatar las siguientes indicaciones.

1. Lea y trate de comprender todo el trabajo solicitado antes de iniciarlo.

2. Para cada ejercicio, utilice hojas de bitácora separadas. Muestre en la bitácora todas las tablas de verdad, circuitos, figuras, diagramas, etc. usados en las preguntas planteadas. Incluya una sección para la etapa de investigación.
3. Para la presentación funcional, se le pedirá que muestre algunos de los circuitos propuestos, ya sea en implementación o simulación. Dicha selección se hará el día de la presentación.

## 2. Investigación

Para el desarrollo de este laboratorio se deben responder las siguientes preguntas.

1. Investigue sobre el funcionamiento general de una ALU. Muestra tablas de verdad y diagramas de circuitos lógicos y aritméticos simples (sumas, restas, operaciones lógicas, etc.). Incluya una descripción de las banderas de estado de una ALU, por ejemplo las de la arquitectura ARMv4.
2. Explique los conceptos de tiempos de propagación y tiempos de contaminación, en circuitos combinacionales.
3. Investigue sobre la ruta crítica y cómo esta afecta en el diseño de sistemas digitales más complejos, por ejemplo un procesador con pipeline. Investigue su relación con la frecuencia máxima de operación de un circuito.

## 3. Ejercicios Prácticos

A continuación se presentan 2 ejercicios prácticos, los cuales debe resolver de manera completa. En cada ejercicio muestre en la bitácora todos los pasos de diseño que llevan a la solución del problema.

### 3.1. Problema 1

Para el problema inicial se deberá diseñar una ALU parametrizable de N bits, como se muestra en la figura 1. Dicha unidad deberá tomar dos números de N bits como entrada, así como un bus de control, denominado ALUControl, que debe permitir seleccionar cuál de las operaciones aplicar. Las diferentes operaciones que debe realizar la ALU son: suma, resta, and, or, not (sobre un solo operando), xor, corrimiento a la izquierda y corrimiento a la derecha (tanto lógicos, como aritméticos). Adicionalmente la ALU deberá generar 4 banderas de estado: Negativo (N), Cero (Z), Acarreo (C) y Desbordamiento (V).

Con base en la descripción anterior:

1. Diseñe la ALU con un modelo de estructura, utilizando SystemVerilog. Parta de los circuitos básicos (sumadores, restadores, de corrimiento, etc.) que considere necesarios. Muestre los diagramas de bloques, tablas de verdad y circuitos de cada módulo en el diseño.
2. Realice un *testbench* de auto-chequeo, en SystemVerilog, en el que se muestre de manera simple el funcionamiento de la ALU en 4 bits. Muestre el resultado de la prueba para al menos 4 valores diferentes para cada operación.
3. Implemente la ALU con parámetro de 3 bits, en la FPGA. Utilice los switches y botones como entradas y un display para 7 segmentos para visualizar el funcionamiento de la misma.

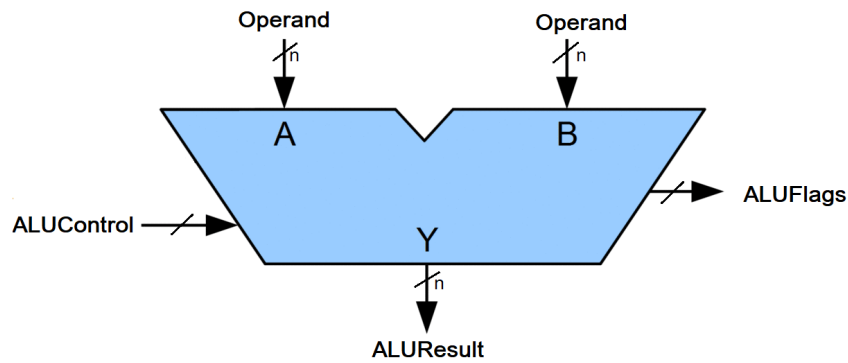


Figura 1: Diagrama de bloques de la ALU a diseñar

### 3.2. Problema 2

Un asunto importante, en sistemas combinacionales complejos, es la ruta crítica en tiempos de propagación, dato que finalmente determina la frecuencia de operación de un sistema digital. Un circuito simple para la medición de esta frecuencia se muestra en la figura 2, en la que se utilizan dos registros de carga paralela entre la lógica combinacional, para determinar la frecuencia máxima a la que puede operar un reloj. Para el circuito anterior y utilizando el

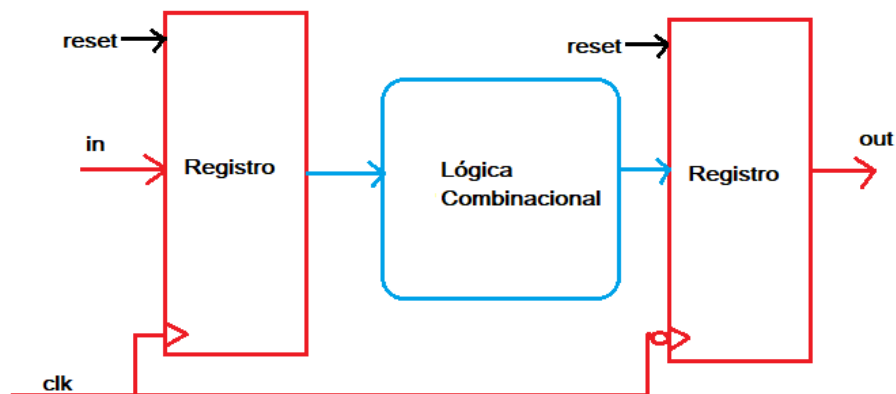


Figura 2: Circuito para la medición de frecuencia máxima de operación

diseño de la ALU anterior:

1. Diseñe el diagrama de la figura 2 con la ALU para 32, 64, 128 y 256 bits. Determine en cada caso el uso de los recursos de la FPGA en términos de celdas básicas. Utilice la herramienta TimeQuest de Altera para determinar la frecuencia máxima de operación para cada uno. Realice una tabla con estos datos. Discuta sobre el efecto de aumentar el tamaño de la lógica combinacional en la frecuencia de operación del sistema