

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Organización Computacional

Ing. Juan Carlos Maeda Juárez

Aux: Javier Gutierrez



PRÁCTICA #2

LogicCalc

INTRODUCCIÓN

En el ámbito de la electrónica digital, los bloques MSI (Medium Scale Integration, por sus siglas en inglés) desempeñan un papel crucial en la construcción de circuitos complejos. Dentro de esta categoría, los bloques MSI tipo aritmético son especialmente importantes para realizar operaciones aritméticas básicas, como sumas, restas y multiplicaciones, de manera eficiente y confiable.

Los bloques digitales combinacionales MSI tipo aritmético están diseñados para realizar operaciones matemáticas utilizando compuertas lógicas, registros y otros componentes digitales. Estos bloques están configurados para recibir dos o más operandos y generar una salida que representa el resultado de la operación.

OBJETIVOS

Objetivo General

1. Construir una Unidad Aritmética Lógica Básica (ALU).

Objetivos Específicos

1. Aprender el funcionamiento de Multiplexores, Demultiplexores, Comparadores y Decodificadores.
2. Construir un diseño óptimo, logrando utilizar la menor cantidad de dispositivos.
3. Aprender el funcionamiento de Operaciones Lógicas, Aritméticas y Comparativas con números binarios.

DESCRIPCIÓN

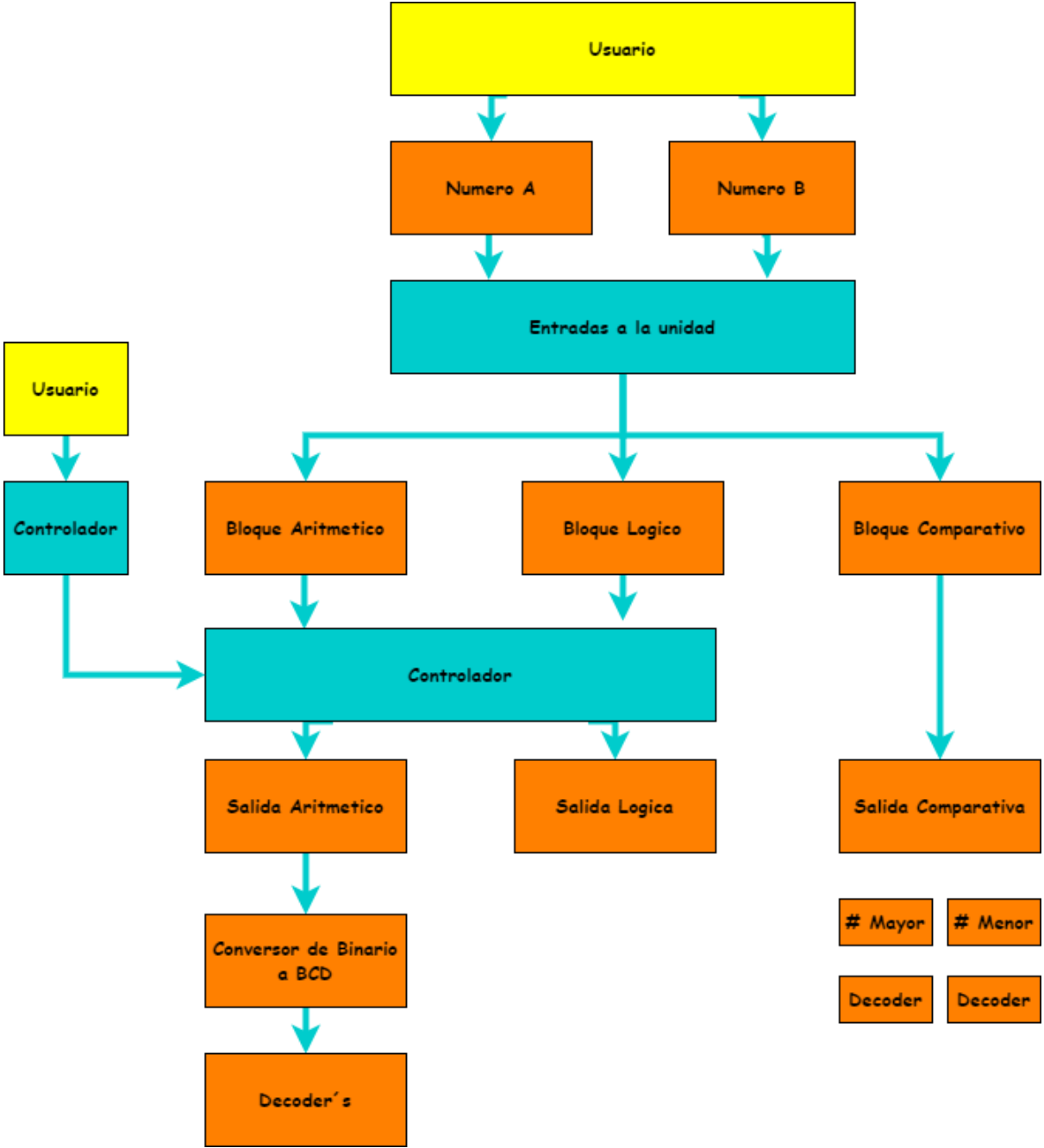
Como estudiantes del curso Organización Computacional, han sido contratados por Intel Corporation para desarrollar un prototipo de calculadora llamado "LogicCalc". Intel busca una solución óptima basada en lógica combinacional que sea capaz de realizar cálculos aritméticos y lógicos. Para cumplir con estos requisitos, Intel ha proporcionado las especificaciones únicas para una Unidad Aritmética Lógica Básica (ALU).

REQUERIMIENTOS

Se desea elaborar un circuito combinacional, el cual sea capaz de resolver operaciones Aritméticas, Lógicas y Comparativas entre 2 números binarios de 4 bits. Para lo cual se debe de implementar un controlador que será el encargado de seleccionar la operación deseada por el usuario. El controlador manejará las operaciones deseadas por medio de la definición de la siguiente tabla:

| C | B | A | Operación |
|---|---|---|----------------|
| 0 | 0 | 0 | Suma |
| 0 | 0 | 1 | Resta |
| 0 | 1 | 0 | Multiplicación |
| 0 | 1 | 1 | Número Elevado |
| 1 | 0 | 0 | XOR |
| 1 | 0 | 1 | OR |
| 1 | 1 | 0 | AND |
| 1 | 1 | 1 | NOT |

DIAGRAMA DE BLOQUES ALU REQUERIDO



UNIDAD ARITMÉTICA

Esta unidad estará encargada de realizar el cálculo aritmético de las entradas a la Unidad.

Por lo cual se deberá tener en cuenta la selección de la operación en el controlador:

- **Suma:** Procederá a realizar la Suma entre 2 Números binarios A y B (Entradas), teniendo en cuenta los valores de acarreo respectivamente. Para las entradas se manejarán solo números positivos. Esta operación se tendrá que implementar enteramente con compuertas lógicas, no se permite utilizar semi-sumadores o sumadores completos.
- **Resta:** Procederá a realizar la Resta entre 2 Números binarios A y B (Entradas), teniendo en cuenta los valores de préstamo respectivamente. Para las entradas se verificará si el minuendo es mayor que el sustraendo, si es así, realizar la resta y mostrar el resultado, de lo contrario, mostrar un mensaje de Error en el display. No se permite utilizar semi-sumadores o sumadores completos.

| Validación | Resultado |
|------------|-----------------------------------|
| $A \geq B$ | Realizar resta, mostrar Resultado |
| $A < B$ | Mostrar EE en display |

- **Multipliación:** Procederá a realizar la Multipliación entre 2 Números binarios A y B (Entradas), teniendo en cuenta los valores de acarreo respectivamente. Esta operación se tendrá que implementar enteramente con Sumadores.
- **Número Elevado:** Procederá a elevar al Cuadrado o al Cubo el Número binario A (Entrada), teniendo en cuenta los valores de acarreo respectivamente. Para definir a qué potencia se elevará el número, se tomará como base el valor del número B, si este es 2, el número A se elevará al cuadrado (Números del 0-9); si este el valor de B es 3, el número A se elevará al cubo (Números del 0-4).

Nota: Todos los resultados de esta Unidad se deberán mostrar en 2 displays de 7 Segmentos (Todos en los mismos). Teniendo en cuenta que el máximo resultado que se podrá visualizar es 99, las multiplicaciones que sobrepasan este valor no se evaluarán durante la calificación.

UNIDAD LÓGICA

Esta unidad estará encargada de realizar el cálculo lógico de las entradas a la Unidad. Por lo cual se deberá tener en cuenta la selección de la operación en el controlador:

- **AND:** Procederá a realizar la Multiplicación Lógica entre 2 Números binarios A y B (Entradas).
- **OR:** Procederá a realizar la Suma Lógica entre 2 Números binarios A y B (Entradas).
- **XOR:** Procederá a realizar la OR Exclusiva entre 2 Números binarios A y B (Entradas).
- **NOT:** Procederá a realizar la Negación Lógica del Número Binario A (Entrada).

Nota: Todos los resultados de esta Unidad se deberán mostrar en 4 LEDS que representarán los 4 bits resultantes como se muestra en el diagrama en bloques. Además de esto se deberá tener en cuenta que la unidad Aritmética y Lógica son mutuamente excluyentes, es decir que solo se deberá mostrar el resultado de la operación de la unidad en un tiempo dado.

Ej: Si se selecciona una operación Aritmética, sólo se deberá mostrar el resultado en el display de la operación aritmética, no se deberá mostrar nada en los 4 bits resultantes de la unidad lógica.

UNIDAD COMPARATIVA

Esta unidad estará encargada de realizar el cálculo Comparativo entre las entradas de la Unidad. Por lo cual se deberá mostrar en 2 display de 7 segmentos, el número mayor y menor de la operación dada. No importando que operación sea. Si en dado caso las entradas son iguales se deberá mostrar en los 2 display el mismo número.

Nota: Se deberá de mostrar mediante un Led de diferente color la unidad que está ejecutando la operación, por lo cual se definen los colores de cada unidad:

| Unidad | Color |
|-------------------|-------|
| Unidad Aritmética | Verde |
| Unidad Lógica | Rojo |

Para la calificación de la práctica se tomará en cuenta el orden del circuito, todo debe ser claro y legible de lo contrario habrá una penalización.

APLICACIÓN DE CONTROL

Se deberá crear un programa que se encargue de controlar el ingreso de los números y de la selección del cálculo a realizar, siendo enviado el dato por comunicación serial al Arduino y por medio de sus pines configurados como salidas deberán ingresar los datos a las entradas del ALU.

METODOLOGÍA DE TRABAJO

La práctica debe realizarse en los grupos asignados en la práctica 1.

- Se debe de simular toda la práctica en un solo archivo de Proteus.
- La documentación por entregar deberá ser en forma digital y deberá contener de manera obligatoria el siguiente orden y estar bien identificada cada una de las secciones:

1. Carátula.

- Nombre de cada integrante y carne.
- Indicar el porcentaje de participación de cada integrante

2. Introducción.

3. Descripción del problema.

4. Funciones Booleanas y Mapas de Karnaugh. (o bien proceso teórico realizado para llegar su diagrama solución).

5. Diagramas de los diseños desarrollados.

6. Equipo Utilizado.

7. Presupuesto.

- Gastos.
- Aporte de cada integrante.

8. Conclusiones.

NOTA:

Para tener derecho a calificación de documentación, deberán entregar simulación completa del circuito en Proteus.

También será necesario que se pruebe el circuito dos veces, una con el Arduino y la otra con dip-switch para verificar que la práctica no está hecha en su totalidad con Arduino.

CONSIDERACIONES

- La práctica será calificada sobre el 100% y se estará preguntando a los integrantes sobre el desarrollo de esta, de no responder correctamente se restará un cierto porcentaje a la nota obtenida del integrante cuestionado, para asegurar que todos hayan participado en dicha práctica.
- Todos los integrantes deben estar presentes durante la calificación de lo contrario se asumirá que no trabajaron y tendrán nota 0.
- Se penalizará con un porcentaje de la nota obtenida, por no intentar realizar un circuito en físico.
- Se penalizará con un porcentaje de la nota obtenida si la simulación en proteus se encuentra en distintos archivos.
- Se revisará detalladamente cada circuito si se llegaron a encontrar integrados no permitidos no se podrá seguir con la calificación y tendrán nota 0, todos los integrantes sin excusa alguna.
- Queda prohibido el uso de Logic State.
- Si se detectan copias tendrán nota de 0 y serán reportadas a la Escuela de sistemas.
- A continuación, se describen algunos circuitos integrados y compuertas lógicas permitidas. No se permite el uso de otro integrado que no esté en esta lista, si se desea usar un integrado que no se encuentre en esta lista abocarse al auxiliar.

| DESCRIPCIÓN | CÓDIGO |
|---------------|--------------|
| Compuerta AND | 7408 |
| Compuerta NOT | 7404 |
| Compuerta XOR | 7486 |
| Compuerta OR | 7432 |
| Comparador | 7485 / 74285 |
| Decoder | 7447 / 7448 |
| Sumador | 7483 / 74283 |
| Multiplexores | 74157 |

| | |
|------------------|--------|
| Demultiplexores | 74138 |
| Transistores NPN | 2N2222 |

- Usuario de GitHub al cual se debe de añadir como colaborador al auxiliar:
 - PikaGuty

ENTREGABLES

Link de drive con los siguientes documentos:

- Un archivo con extensión. pdsprj que contenga el circuito combinacional.
- Documentación.
- Se requiere entregar el enlace al repositorio grupal de GitHub con la siguiente nomenclatura: P2_ORGA_G# el cual contendrá todo lo mencionado anteriormente.

FORMA DE ENTREGA

Mediante UEDI, subiendo el enlace del repositorio. Solamente una persona del grupo debe realizar la entrega.

Nota: La fecha límite de entrega es el 16 de junio de 2024, antes de las 23:59.