

INSTITUTO TECNOLÓGICO DE COSTA RICA

ÁREA ACADÉMICA DE INGENIERÍA MECATRÓNICA

MT 4002
Laboratorio de Electrónica Digital

Proyecto 2:
Diseño e implementación de una ALU

Jose Fabio Navarro Naranjo – 2019049626

Adrián Dittel Retana – 2019007945

Gabriel González Rodríguez – 2019057548

Emmanuel Naranjo Blanco – 2019053605

David Rodríguez Camacho – 2019024546

Profesora: Ana María Murillo Morgan

Semestre II - 2021

Contenido

I. Descripción del proyecto.....	3
II. Diseño propuesto	4
III. Banderas.....	5
1. Bandera ZF	5
2. Bandera PF	5
3. Bandera OF.....	5
IV. Proceso de diseño de las operaciones.....	6
4. Operación XOR	6
5. Operación AND.....	6
6. Operación OR.....	6
7. Desplazamiento lógico a la derecha	6
8. Suma con signo y Resta con signo	7
9. Multiplicación con signo	8
10. Quintuplicado de un número.....	10
V. Anexos	13

I. Descripción del proyecto

En el presente proyecto se diseñará una unidad aritmética lógica que realiza todas las operaciones mostradas en la Tabla 1. Para esto se cuenta con tres entradas y cuatro salidas, las cuales a través de una implementación en Verilog permitirá integrar las operaciones correspondientes.

En la Figura 1 se indica el diagrama de primer nivel del problema por resolver, donde las entradas A y B son los operandos de 6 bits, SEL es el seleccionador de la ALU de 3 bits. En cuanto a las salidas se tiene Y de 6 bits como resultado de la operación, y tres banderas, ZF como bandera de cero, PF como bandera de paridad y OF como bandera de desbordamiento.

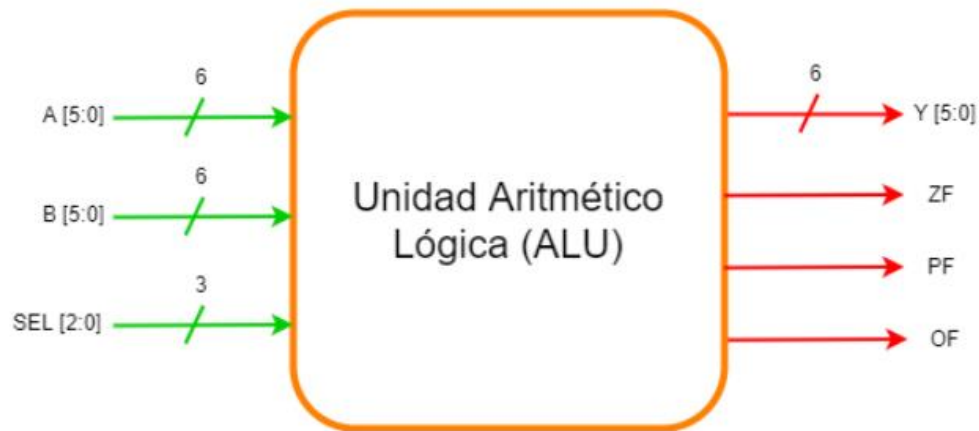


Figura 1. Diagrama de entradas y salidas del sistema a diseñar.

Tabla 1. Descripción de las operaciones de la ALU.

SEL [2:0]	Tipo de Operación	Y
000	XOR	$Y = A \wedge B$
001	AND	$Y = A \& B$
010	OR	$Y = A B$
011	Desplazamiento lógico a la derecha	$Y = A \gg 1$
100	Suma con signo	$Y = A + B$
101	Multiplicación con signo	$Y = A * B$
110	Resta con signo	$Y = A - B$
111	Quintuplicado de un número	$Y = 5 * A$

II. Diseño propuesto

Con el fin de llegar a una solución concreta, se particionó el problema en módulos, cada uno con su función correspondiente. En la sección de Anexos se presenta el diagrama de primer y segundo nivel para la solución propuesta del proyecto y en las siguientes secciones se explicará a fondo cada módulo planteado. En este se observa que según el SEL, la ALU escoge el resultado de una operación, cuyas entradas A y B, y la salida Y son de 6 bits. Además, se presentan los módulos de las tres banderas ZF, PF y OF, cada uno con su lógica combinacional correspondiente.

Cabe destacar que en los MUX empleador, todos se encuentran habilitados, por lo que el Enable es 1 y se decidió omitir de los diagramas.

III. Banderas

1. Bandera ZF

Para la bandera de cero, o la “Zero Flag”, el diseño se basó en colocar una compuerta del tipo NOR que capte todos los bits de salida del resultado de las operaciones, de modo que (al ser una compuerta NOR) arroje un 1 únicamente cuando todos los bits de Y sean 0.

2. Bandera PF

Para la bandera de paridad, el diseño se basó en el hecho de que dicha bandera debe activarse cuando la cantidad de 1's en el resultado es par. Por este motivo, se utilizó una compuerta XNOR, la cual recibe como entrada todos los bits del resultado Y, debido a su funcionamiento (arroja un 1 siempre que la cantidad de 1's en la entrada sea par) provocará que la bandera de paridad se active cuando la cantidad de bits iguales a 1 en la salida sea par.

3. Bandera OF

Esta bandera presenta un grado de complejidad mayor a las anteriores. Para diseñar esta bandera lo que se hizo fue implementar en el circuito de cada operación, una salida que representa si existe Over Flow en la operación realizada (dichos diseños se presentan más adelante en el diseño del circuito para cada operación). Entonces, se toman todas estas salidas y se dirigen hacia un multiplexor 8 a 1, el cual usa el mismo selector del multiplexor que saca el bus de datos de las operaciones, de esta manera, se conecta la bandera de Over Flow a la salida de este multiplexor, y se refleja el Over Flow de la operación que se está realizando. Es importante mencionar que para las operaciones lógicas (AND, OR, XOR, Desplazamiento), el Over Flow siempre es 0, por este motivo, las primeras 4 entradas del multiplexor se conectan a tierra. Por último, también se hace énfasis, en que se diseñó un solo circuito para la operación de suma y resta, y por ende, este circuito saca el Over Flow de la operación que se está realizando, por este motivo, las entradas 4 y 6 del multiplexor se encuentran en corto circuito con la salida del circuito Sumador/Restador.

El diagrama de diseño de las 3 banderas se presenta en la sección de Anexos junto con los demás diagramas.

IV. Proceso de diseño de las operaciones

En la presente sección se muestra la partición del problema, con el fin de desarrollar subsistemas y posteriormente integrarlo en un producto final. Cada operación se realiza según el seleccionador de la ALU, SEL [2:0].

4. Operación XOR

Para esta operación, el diseño implementó el uso de 6 compuertas XOR (representadas como una sola, que tiene como ingreso y salida un bus de datos), las cuales comparan los bits de A contra los de B en el orden de significancia, es decir, la primera compuerta XOR compara el bit A0 con el bit B0, la segunda compuerta compara el bit A1 con el bit B1, y así sucesivamente hasta que se llega a la última compuerta que compara el bit más significativo de cada entrada. Por último, el bus de salida tiene 6 bits, los cuales son tomados de cada compuerta utilizada.

5. Operación AND

El diseño para el circuito que ejecuta esta operación es bastante similar al implementado en la operación anterior, solo que con otro tipo de compuertas lógicas. Utiliza 6 compuertas del tipo AND, las cuales comparan bit por bit, las entradas A y B, desde el bit menos significativo, hasta el bit más significativo, de modo que este circuito arroja como salida un bus de 6 bits, que incluye todas las salidas individuales de cada compuerta.

6. Operación OR

De manera bastante similar a las anteriores, esta operación se realiza con 6 compuertas OR que comparaban uno a uno los bits de las entradas A y B, conforme a su significancia, de modo que al circuito le entran los 6 bits de A, y los 6 bits de B, y tiene como salida un bus de datos que se dirige al multiplexor, y que representa las salidas de cada compuerta OR individual.

7. Desplazamiento lógico a la derecha

Para el diseño de este circuito lógico, no se utilizó ninguna compuerta lógica. Lo que se realizó fue que se alambraron los bits de A, de modo que calzaran con el bit de Y esperado. Para esto, se utilizó de manera ilustrativa la Tabla 2, con el fin de tener una idea más clara de las conexiones a realizar.

Tabla 2. Tabla para las conexiones del circuito de desplazamiento a la derecha.

Bus de datos	Bits					
A	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
Y	0	A ₅	A ₄	A ₃	A ₂	A ₁

Con ayuda de la tabla anterior, se observó que lo que se debía realizar era tomar como 0 el bit más significativo de la salida, y luego, únicamente alambrar las entradas de A con las salidas, de manera que se cumpliera lo expuesto en la Tabla 2.

8. Suma con signo y Resta con signo

Se determinó apropiado la integración de la suma y resta con signo en un único módulo, el cual realizará una de las dos operaciones según el bit 1 del seleccionador de la ALU, SEL[1]. Como se observa en la Tabla 1, este corresponde a 0 cuando se escoge la suma de A y B, y equivale a 1 cuando se realiza la resta.

A partir de esto, se procedió a utilizar un sumador donde A se conecta directamente a una de sus entradas y B procede de la salida de un multiplexor gobernado por SEL[1], en una entrada del MUX B se mantiene igual y en la otra se invierte con compuertas NOT. Igualmente, el acarreo de entrada corresponde a SEL[1]. De este modo, al escoger la suma, A y B se mantienen iguales y Cin=0, pero al escoger la resta, A se mantiene y se aplica el complemento a 2 de B gracias a que este se niega a causa de la salida del MUX y se suma un 1 con el Cin=1. Para comprender mejor esto, se invita a observar la sección de Anexos, donde se muestra el diagrama de tercer nivel para la operación conjunta de suma y resta en complemento a 2.

En cuanto al desbordamiento, por un lado, en la suma esta condición puede ocurrir solo cuando los operandos son del mismo signo y siempre produce un resultado incorrecto. Por otro lado, en la resta el desbordamiento sucede cuando los números tienen signos diferentes (Tocci, 2007). Además, al trabajar bajo el sistema de complemento a 2, se sabe que tanto para resta como suma, si el número es negativo, se realizará el complemento a 2 de este y se hará la operación de suma. Por esta razón, el desbordamiento ocurre únicamente cuando el

bit más significativo de los operandos es igual y el bit más significativo del resultado es distinto al MSB de los operandos.

Para obtener el desbordamiento se hizo la tabla de verdad (Tabla 3) y mediante la suma de productos se obtuvo la ecuación correspondiente, la cual fue implementada en compuertas lógicas.

Tabla 3. Tabla de verdad para detectar el desbordamiento de la suma y resta con signo.

#	Entradas			Salidas
	A [5]	B [5]	Y [5]	OF
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

La función de desbordamiento corresponde a:

$$OF = \overline{A[5]} \cdot \overline{B[5]} \cdot Y[5] + A[5] \cdot B[5] \cdot \overline{Y[5]}$$

9. Multiplicación con signo

Para el diseño del multiplicador se utilizó el mismo proceso que se utilizaría para resolver una operación binaria en complemento a dos a mano. Este proceso consiste de tres pasos. Primero, se hace la conversión de complemento a 2 a binario natural para los operandos que tengan signo negativo. Segundo, se realiza la multiplicación binaria habitual con ambos operandos en binario natural. Tercero, se hace una comprobación, por “ley de signos”, para determinar si el resultado debe de tener signo negativo o positivo. Esto significa que, si ambos operandos tienen el mismo signo, el resultado va a ser positivo, y si tienen signos opuestos, el resultado va a ser negativo.

Entrando un poco más a fondo en el diseño del primer paso, explicado en el párrafo anterior, se utiliza un multiplexor (Mux) 2 a 1 para elegir entre dos opciones, o se deja pasar el operando a como entró al módulo originalmente, o se deja pasar este, pero habiéndole hecho

la conversión en complemento a 2. Como selector del Mux se utiliza el bit más significativo del dato de entrada, ya que este corresponde al signo, si es 0 (positivo) se deja pasar el operando tal y como venía, si es 1 (negativo) se realiza la conversión. La conversión en complemento a 2, ya propiamente, consiste en un inversor, para cada bit del operando, y de un sumador entre el operando y el número 1 (procedimiento para la conversión en complemento a 2). A la salida del mux entonces saldría únicamente el dato en su representación en binario natural. La anterior explicación se puede visualizar en la página 5 de los anexos, que corresponde al Diagrama de Tercer Nivel: Multiplicador con Signo.

Una vez teniendo ambos operandos en su representación en binario natural, se procede con el multiplicador, el cual está formado a partir de sumadores, a como se observa en la página 6 de los anexos que corresponde al Diagrama de Tercer Nivel: Detalle Multiplicador. En este se anidan los sumadores en la configuración mostrada, y con las compuertas AND; para realizar la multiplicación entre ambos operandos. Para el caso de este diseño, como se está trabajando con un tamaño de palabra de 6 bits, únicamente se dejan como salida los bits del 0 al 5, y al bit 5 se le asigna un 0 lógico ya que este correspondería al signo en complemento a dos. De esta forma se obtienen solo los datos de los bits se necesitan representar. Para detectar si existe desbordamiento, u “overflow”, se toman como entrada para una compuerta OR los bits no utilizados anterior mente (bits del 6 al 11) y de esta forma si alguno posee un 1 lógico (lo que significa que hubo desbordamiento) la salida de la compuerta (la bandera de desbordamiento, o bandera OF) tendría un 1 lógico a la salida, en caso contrario tendría un 0 lógico.

Por último, para el caso del convertir el resultado de la multiplicación, que se encuentra en binario natural, a complemento a 2, se utiliza el mismo conversor, descrito anteriormente, para la conversión del operando negativo en complemento a 2 a binario natural, ya que el procedimiento es el mismo sin importar la dirección de la conversión. La única diferencia con esta configuración anterior es que al selector le entra el resultado de una compuerta XOR que tiene como entradas el bit más significativo de cada operando. De esta forma, si los signos de los operandos son diferentes (que corresponde a un resultado negativo), se realiza la conversión, en caso contrario se deja pasar el resultado a como se obtuvo de la multiplicación en binario natural.

10. Quintuplicado de un número

En cuanto a la operación del quintuplicador, se hizo la multiplicación por 5 a partir del principio de desplazamiento a la izquierda. Dado que esto solo sirve para multiplicar por números resultantes de 2^n se separó el cinco en una suma de potencias de 2, 2^2+2^0 , entonces se corre el número entrante dos veces a la izquierda y se suma a la entrada original para obtener la función $x * (2^2 + 2^0)$. Para esto se hizo el desplazador mostrado en la Figura 2 para correr el complemento a dos del número.

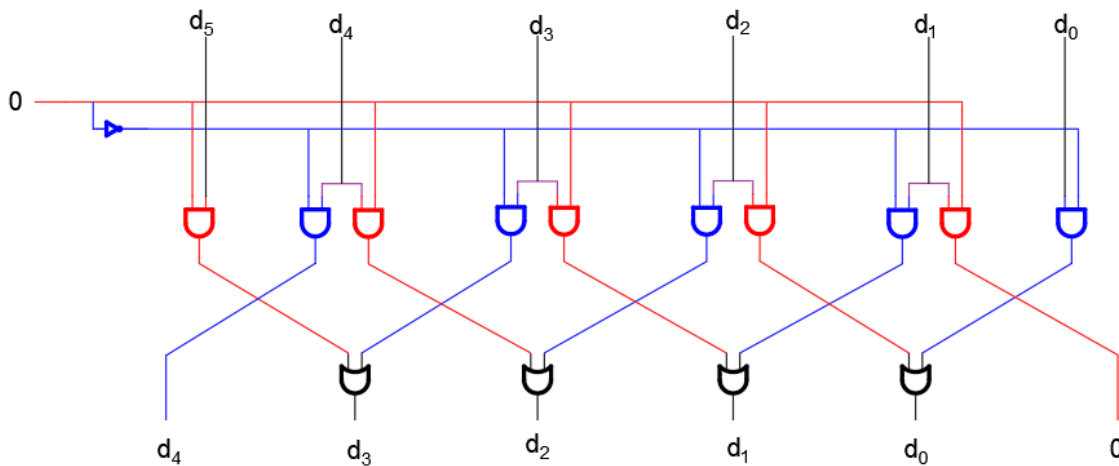


Figura 2. Circuito desplazador del quintuplicador.

En este circuito, cuando se le ingresa un 0 en las entradas de las AND rojas, esto va a hacer que las AND rojas corten el paso a su dígito de entrada y las azules, gracias a la NOR van a tener un 1 y van a dejar pasar su entrada (si entra un 1 va a salir un 1 y si entra un 0 va a pasar un 0). Así, como las salidas están organizadas se va a correr el dígito de entrada hacia la izquierda una vez. Para conseguir que se corriera dos veces, las salidas de este circuito se conectan a otro desplazador igual y luego este resultado se ingresa a un sumador de 6 bits junto al número original para obtener la multiplicación por cinco.

Para calcular el desbordamiento se tomó en cuenta que “*overflow occurs if and only if $p_n \oplus p_{n-1} = 1$* ” (Gok *et al.*, 2006, p.1065); es decir, si el XOR de los bits n y $n-1$ del producto (en este caso 6 y 5 porque n es la cantidad de bits de las entradas) va a haber desbordamiento. Entonces para cumplir esto en la salida del quintuplicador se conectaron los bits 6 y 5 a una compuerta XOR y si esta devuelve un 1 el desbordamiento ocurre. Para observar el diseño

completo es indispensable visitar la sección de Anexos, donde está el diagrama de tercer nivel de esta función quintuplicador.

Referencias

- Gok, M. Schulte, M. Arnold, M. (2006). *Integer Multiplier with Overflow Detection*.
https://www.researchgate.net/publication/3045091_Integer_multipliers_with_overflow_detection
- Tocci, R. (2007). *Sistemas digitales: Principios y aplicaciones*. Décima Edición. Pearson Education.

V. Anexos

En la presente sección se adjuntan los diagramas de primer, segundo y tercer nivel propuestos para la integración del diseño final del proyecto. Estos diagramas se dividen en páginas, cada una indicando los diseños particulares explicados anteriormente.

Índice de los diagramas para implementar la ALU:

Página 1. Diagrama de primer nivel.

Página 2. Diagrama de segundo nivel.

Página 3. Diagrama de tercer nivel: XOR, AND, OR y desplazamiento a la derecha.

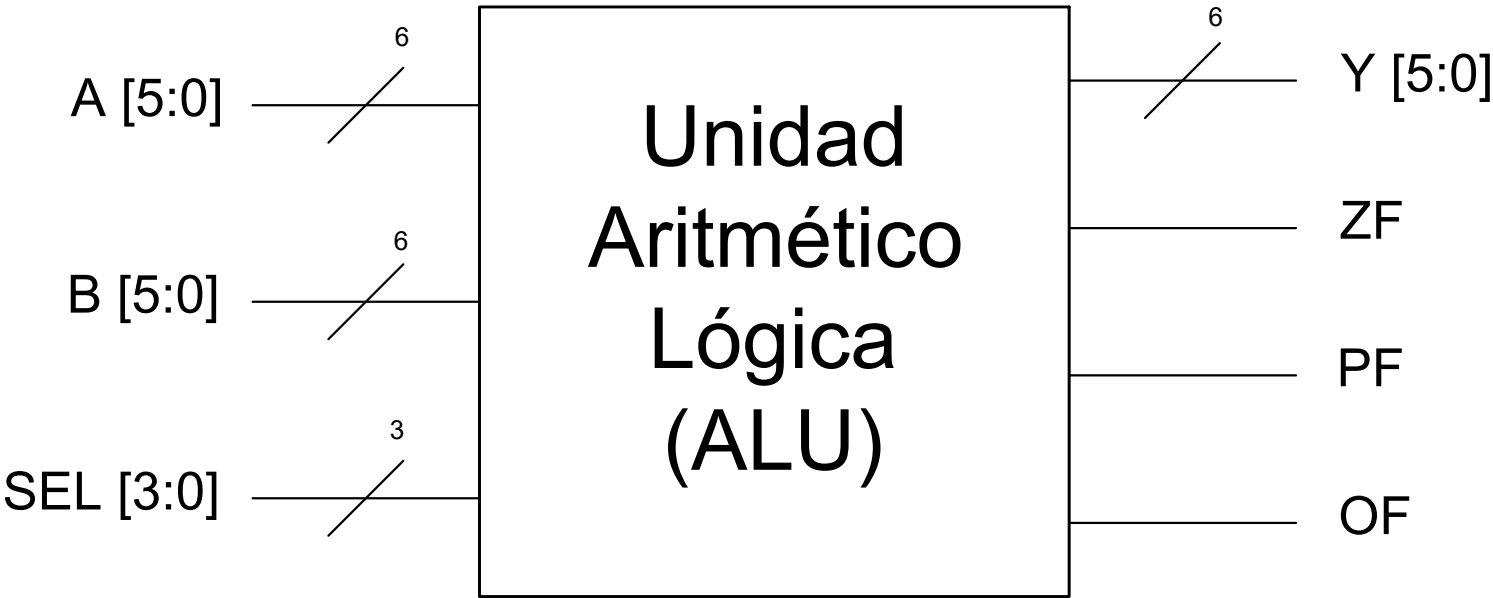
Página 4. Diagrama de tercer nivel: Banderas de Zero, Paridad, y de Desbordamiento.

Página 5. Diagrama de tercer nivel: multiplicador con signo.

Página 6. Diagrama de tercer nivel: detalle multiplicador.

Página 7. Diagrama de tercer nivel circuito sumador y restador.

Página 8. Diagrama de tercer nivel del circuito quintuplicador.



TEC | Tecnológico
de Costa Rica

PROYECTO 2: DISEÑO DE UNA ALU

DIAGRAMA DE PRIMER NIVEL

ÁREA ACADÉMICA DE INGENIERÍA MECATRÓNICA

CURSO: MT4002 - LAB. ELECTRÓNICA DIGITAL

INTEGRANTES:

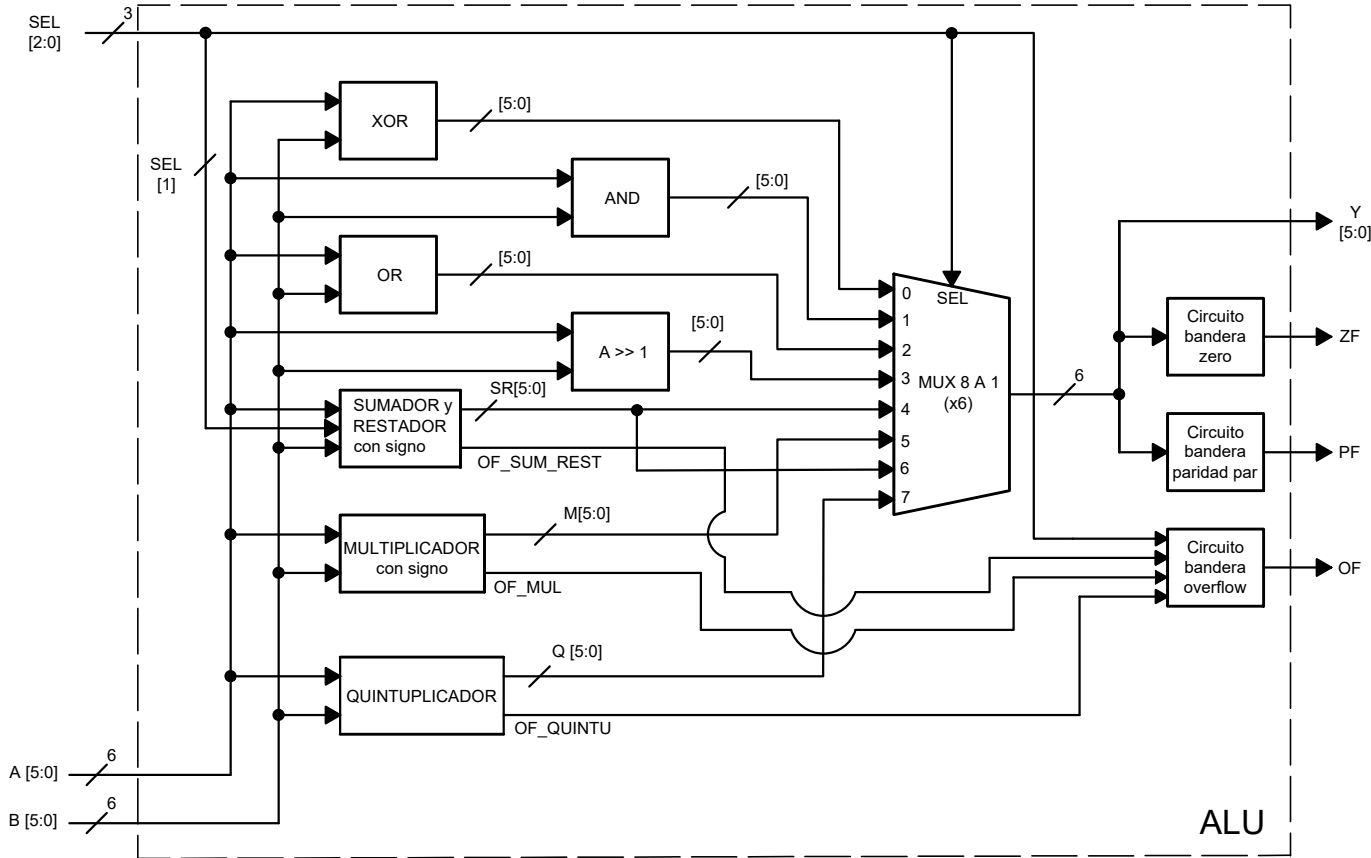
ADRIÁN DITTEL RETANA
GABRIEL GONZÁLEZ RODRÍGUEZ
EMMANUEL NARANJO BLANCO
JOSÉ FABIO NAVARRO NARANJO
DAVID RODRÍGUEZ CAMACHO

LÁMINA

1

8

CREADO CON UNA VERSION PARA ESTUDIANTES DE AUTODESK



CREADO CON UNA VERSION PARA ESTUDIANTES DE AUTODESK

TEC | Tecnológico de Costa Rica

PROYECTO 2: DISEÑO DE UNA ALU

DIAGRAMA DE SEGUNDO NIVEL

ÁREA ACADÉMICA DE INGENIERÍA MECATRÓNICA
CURSO: MT4002 - LAB. ELECTRÓNICA DIGITAL
INTEGRANTES:

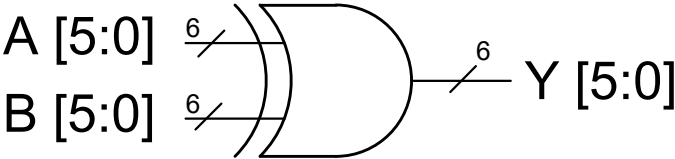
ADRIÁN DITTEL RETANA
GABRIEL GONZÁLEZ RODRÍGUEZ
EMMANUEL NARANJO BLANCO
JOSÉ FABIO NAVARRO NARANJO
DAVID RODRÍGUEZ CAMACHO

LÁMINA

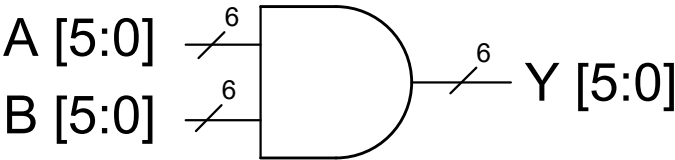
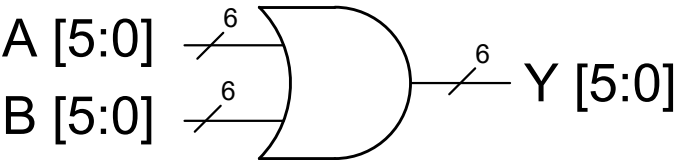
2

8

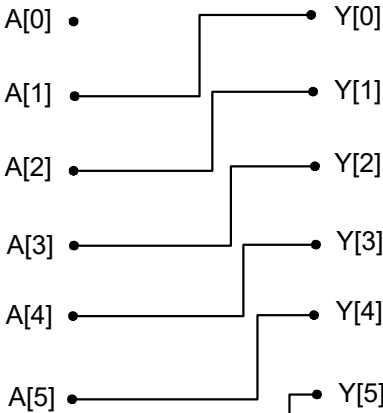
XOR



OR



AND



Desplazamiento Derecha

TEC | Tecnológico
de Costa Rica

PROYECTO 2: DISEÑO DE UNA ALU

DIAGRAMA DE TERCER NIVEL
XOR, AND, OR, DESPLAZAMIENTO A
LA DERECHA.

ÁREA ACADÉMICA DE INGENIERÍA MECATRÓNICA

CURSO: MT4002 - LAB. ELECTRÓNICA DIGITAL

INTEGRANTES:

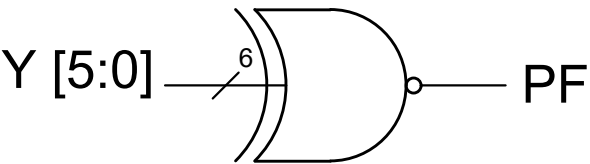
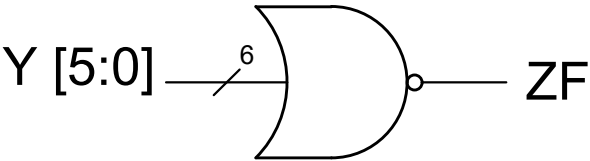
ADRIÁN DITTEL RETANA
GABRIEL GONZÁLEZ RODRÍGUEZ
EMMANUEL NARANJO BLANCO
JOSÉ FABIO NAVARRO NARANJO
DAVID RODRÍGUEZ CAMACHO

LÁMINA

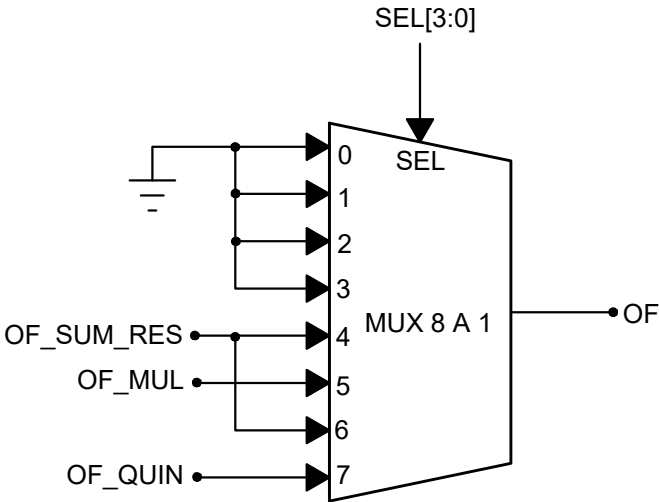
3

8

Bandera de Cero



Bandera de Paridad



Bandera de desbordamiento

TEC | Tecnológico
de Costa Rica

PROYECTO 2: DISEÑO DE UNA ALU

DIAGRAMA DE TERCER NIVEL
BANDERAS DE ZERO, PARIDAD, Y DE
DESBORDAMIENTO.

ÁREA ACADÉMICA DE INGENIERÍA MECATRÓNICA

CURSO: MT4002 - LAB. ELECTRÓNICA DIGITAL

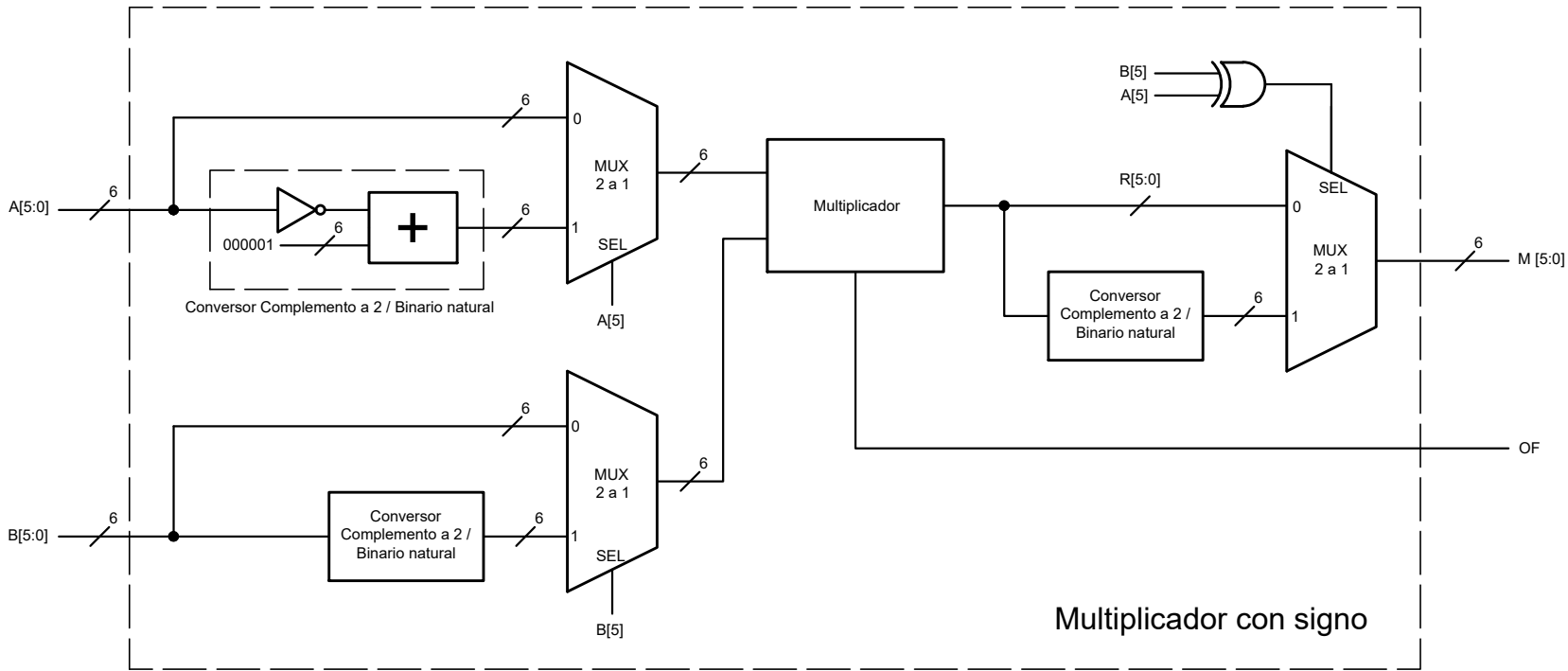
INTEGRANTES:

ADRIÁN DITTEL RETANA
GABRIEL GONZÁLEZ RODRÍGUEZ
EMMANUEL NARANJO BLANCO
JOSÉ FABIO NAVARRO NARANJO
DAVID RODRÍGUEZ CAMACHO

LÁMINA

4

8



TEC | Tecnológico
de Costa Rica

PROYECTO 2: DISEÑO DE UNA ALU

DIAGRAMA DE TERCER NIVEL:
MULTIPLICADOR CON SIGNO

ÁREA ACADÉMICA DE INGENIERÍA MECATRÓNICA

CURSO: MT4002 - LAB. ELECTRÓNICA DIGITAL

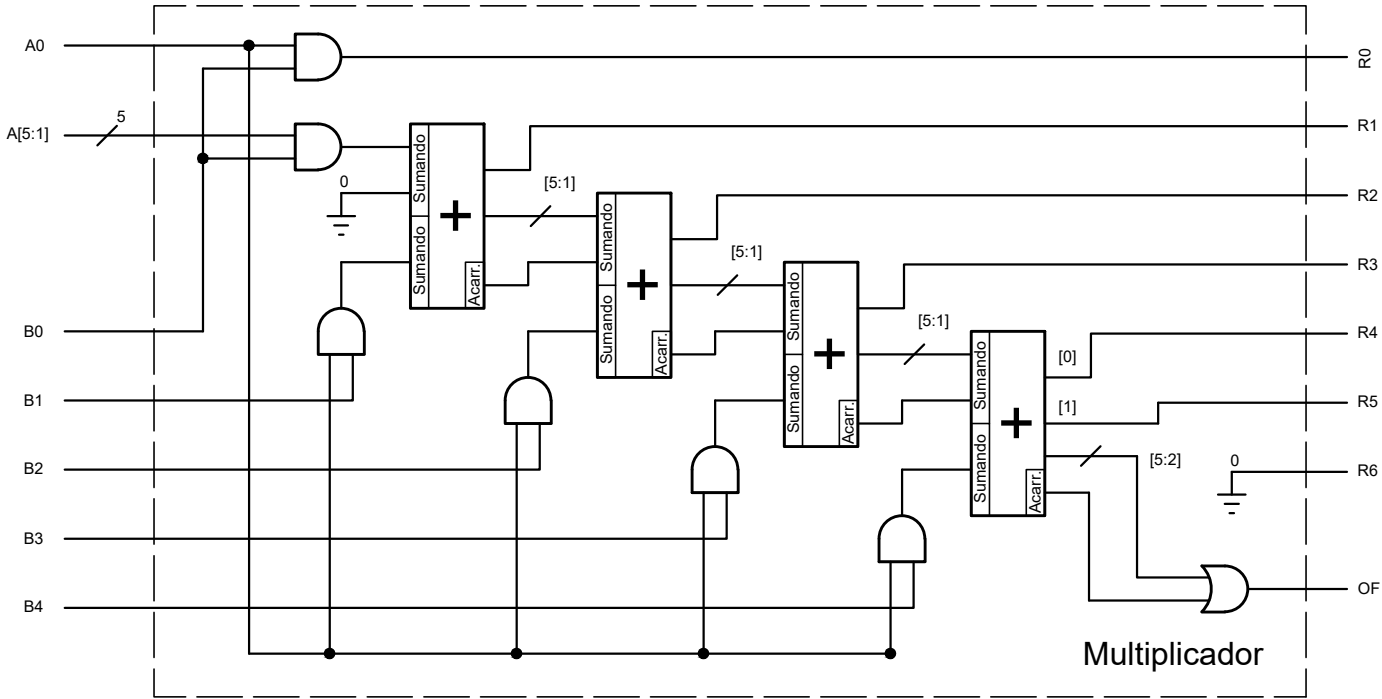
INTEGRANTES:

ADRIÁN DITTEL RETANA
GABRIEL GONZÁLEZ RODRÍGUEZ
EMMANUEL NARANJO BLANCO
JOSÉ FABIO NAVARRO NARANJO
DAVID RODRÍGUEZ CAMACHO

LÁMINA

5

8



TEC | Tecnológico
de Costa Rica

PROYECTO 2: DISEÑO DE UNA ALU

DIAGRAMA DE TERCER NIVEL:
DETALLE MULTIPLICADOR

ÁREA ACADÉMICA DE INGENIERÍA MECATRÓNICA

CURSO: MT4002 - LAB. ELECTRÓNICA DIGITAL

INTEGRANTES:

ADRIÁN DITTEL RETANA

GABRIEL GONZÁLEZ RODRÍGUEZ

EMMANUEL NARANJO BLANCO

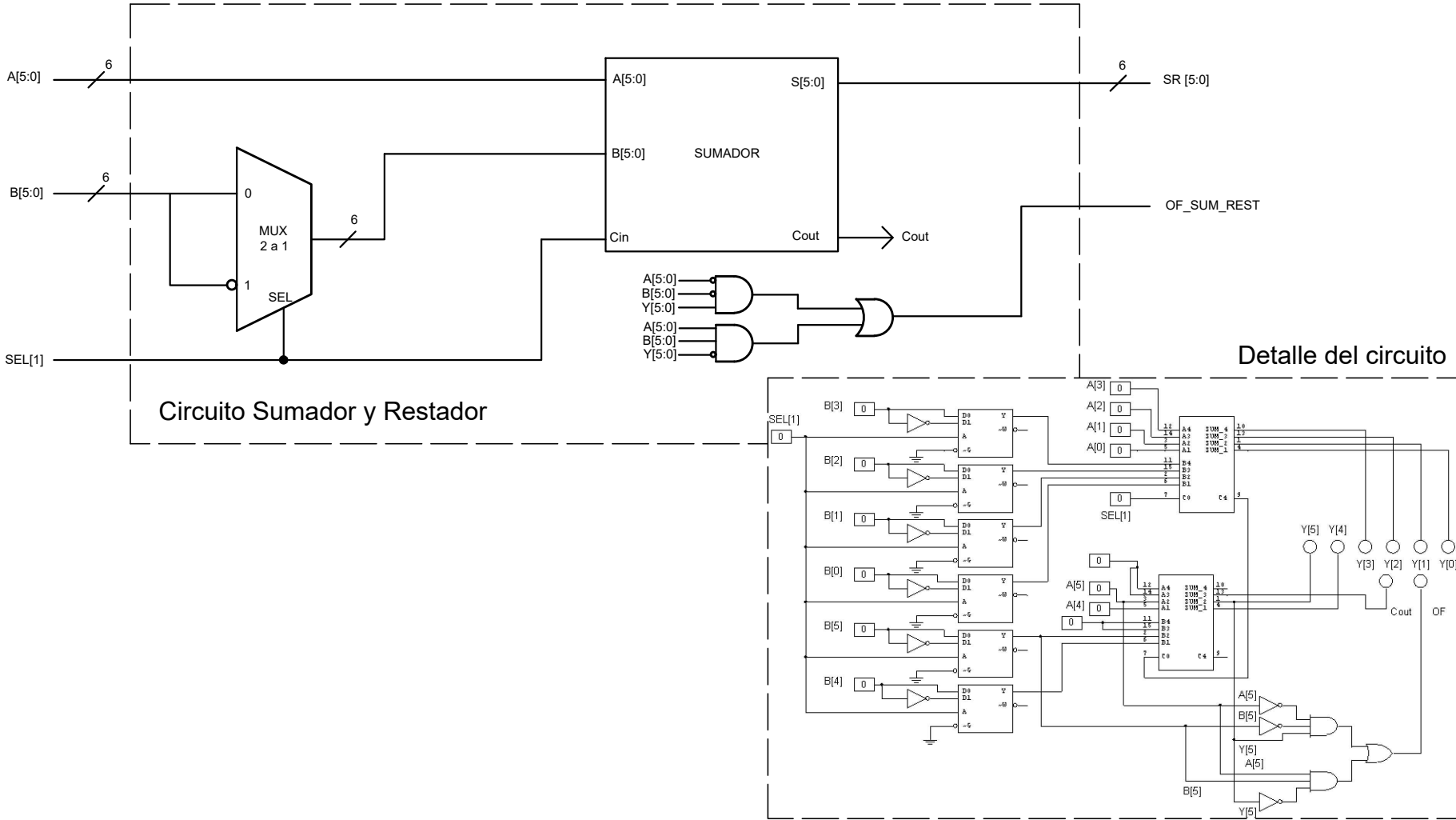
JOSÉ FABIO NAVARRO NARANJO

DAVID RODRÍGUEZ CAMACHO

LÁMINA

6

8



TEC | Tecnológico de Costa Rica

PROYECTO 2: DISEÑO DE UNA ALU

DIAGRAMA DE TERCER NIVEL
CIRCUITO SUMADOR Y RESTADOR

ÁREA ACADÉMICA DE INGENIERÍA MECATRÓNICA
CURSO: MT4002 - LAB. ELECTRÓNICA DIGITAL
INTEGRANTES:

ADRIÁN DITTEL RETANA
GABRIEL GONZÁLEZ RODRÍGUEZ
EMMANUEL NARANJO BLANCO
JOSÉ FABIO NAVARRO NARANJO
DAVID RODRÍGUEZ CAMACHO

LÁMINA

7
8

