

**CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS
DEL INSTITUTO POLITECNICO
NACIONAL.**



DEPARTAMENTO DE INGENIERIA ELECTRICA.

SECCION DE ELECTRONICA DEL ESTADO SOLIDO.

**DISEÑO DE UNA ALU
EMPLEANDO LOGICA DE CONFIGURACION EXTERNA.**

Tesis que presenta:

Ing. Eleazar Adrián Cortés Barrón

Para obtener el grado de Maestro en Ciencias
Especialidad en Ingeniería Eléctrica

Director de la Tesis:

Dr. Mario Alfredo Reyes Barranca

México D.F.

Noviembre 2012

Diseño de una ALU empleando Lógica de Configuración Externa

Capítulo I. Introducción	1
1.1 Antecedentes.....	1
1.2 Estructura Básica de los Transistores MOS de compuerta flotante Multi-entrada (MIFG).....	2
1.3 Diagrama de Potencial de Compuerta Flotante.....	4
1.4 Ventajas y Desventajas del uso de Transistores MIFG MOS	8
1.5 Conclusiones.....	8
1.6 Referencias	8
Capítulo II. Metodología del Diseño	10
2.1 Introducción.....	10
2.2 Celda Básica SHL de 4 bits.....	10
2.3 Coeficientes de Acoplamiento de la etapa de pre-carga	13
2.4 Coeficientes de Acoplamiento de la etapa de Neurona.....	15
2.5 Coeficientes de Acoplamiento de la etapa de Inversores Programables	17
2.6 Voltajes de Conmutación de los inversores programables	20
2.7 Conclusiones	26
2.8 Referencias	28
Capítulo III. Diseño de la Unidad Aritmética Lógica de 4 bits	29
3.1 Introducción	29
3.2 Metodología de Diseño de la ALU de 4 bits	29
3.3 Diseño del sub-bloque de Generación de Acarreo (f_{carry}).....	33
3.4 Diseño del bloque de Función F1	35
3.5 Diseño del bloque de Función F2	38
3.6 Conclusiones	43
3.7 Referencias	44
Capítulo IV. Simulaciones	45
4.1 Introducción.....	45
4.2 Determinación del punto de Inversión (V_{SP}) y de W/L de los transistores NMOS y PMOS empleados en la simulación de la celda básica SHL	45

4.3 Simulaciones de las compuertas AND, NAND, OR, NOR, XOR, XNOR y bloques que componen la ALU	53
4.4 Conclusiones	65
4.5 Referencias	65
Capítulo V. Comparaciones de las ALU de 4 bits: MC14581B / FPGA / SHL	66
5.1 Introducción	66
5.2 ALU SHL de 4 bits	66
5.3 Chip Motorola MC14581B	70
5.4 Implementación de una ALU en un FPGA	76
5.5 Comparación de resultados entre las ALUs: SHL, CHIP y FPGA	81
5.6 Conclusiones	84
Capítulo VI. Layout de la ALU SHL de 4 bits.....	85
6.1 Introducción.....	85
6.2 Diseño geométrico de los inversores CMOS de la celda básica SHL	85
6.3 Diseño del circuito de pre-carga	86
6.4 Diseño del circuito de Neurona para los bloques Suma-Acarreo y F1-F2	87
6.5 Diseño del circuito de Inversores Programables.....	88
6.6 Diseño de los bloques de Sumador Completo y selección-Complemento de la ALU.....	88
6.7 Diseño final del chip de la ALU SHL de 4 bits	88
6.8 Conclusiones	88
6.9 Referencias	91
Capítulo VII. Conclusiones	92
7.1 Conclusiones	92
7.2 Trabajos Futuros	93
7.3 Trabajos generados a partir de esta tesis	93

Capítulo I. Introducción.

1.1 Antecedentes.

La tecnología CMOS (Metal-Oxido-Semiconductor Complementario) ha sido una de las más revolucionarias y transformadoras, desde su aparición en 1985, especialmente en el desarrollo de circuitos integrados, ya que en la actualidad abarca más del 85% del mercado mundial de semiconductores, el cual aún está dominado por el diseño de sistemas digitales, especialmente por memorias y microprocesadores [1], donde la accesibilidad y el desarrollo de la capacidad tecnológica, en conjunto con nuevas metodologías y herramientas de diseño, han permitido incorporar secciones analógicas y digitales en un mismo chip, ofreciendo una solución hasta para las operaciones más complejas.

A pesar de que la tendencia en el desarrollo tecnológico actual es hacia sistemas de Ultra Larga Escala de Integración (ULSI), cuyo enfoque es el incremento de la densidad de integración de transistores y la versatilidad de funciones que pueda realizar un chip, la complejidad en el diseño (lo cual implica mayor número de interconexiones), el desempeño mismo del chip (estabilidad) y el incremento en el consumo de potencia, han sido factores que han limitado su crecimiento.

Por lo que la investigación de métodos alternativos para dar mayor funcionalidad al transistor MOS se ha convertido en un tema de suma importancia actualmente. El elemento, objeto base de esta tesis y prometedor concepto, introducido por Shibata y Ohmi [2], es el transistor MOS neurona (llamado también neuMOS o ν MOS), el cual posee una compuerta flotante, cuyo potencial (Φ_{FG}) es controlado por la suma ponderada de los potenciales aplicados a cada una de sus compuertas de control (entradas), mediante sus respectivos coeficientes de acoplamiento capacitivos (K_{CG}) [3].

La versatilidad del transistor neuMOS aplicado en circuitos de lógica binaria, así como circuitos análogos, ha quedado demostrada ampliamente y el rango de aplicaciones para el dispositivo neuMOS, van desde su uso en filtros análogos, convertidores de datos, memorias analógicas y digitales, hasta el desarrollo de redes neuronales [4-7].

El objetivo del presente trabajo, es demostrar que con la metodología propuesta es posible diseñar e implementar una unidad aritmética lógica de 4 bits (ALU) basada en Lógica SHL (Soft-Hardware Logic). La parte fundamental de la implementación, un sumador completo, está basado en el bien conocido transistor neu-MOS, el cual es un transistor MOS de compuerta flotante con más de una compuerta de control (MIFG), y en este caso es empleado para el procesamiento digital de la señal.

En los subsecuentes capítulos, se demostrará que mediante el empleo de una celda básica SHL, las funciones booleanas básicas AND, NAND, OR, NOR, Exclusive-OR (X-OR) y Exclusive-NOR (X-NOR), pueden ser implementadas empleando Transistores de compuerta flotante multi-entrada (MIFG) ν -MOS, para dar paso al diseño de una ALU de 4 bits, empleando bloques con esta misma configuración.

Basados en un método gráfico llamado “Diagrama de Potencial de Compuerta Flotante (FPD)”, una ALU de 4 bits fue diseñada y simulada para un par de funciones aritméticas y lógicas, tomando ventaja de la suma ponderada que se lleva a cabo en la compuerta flotante del transistor neu-MOS. Este método gráfico se abordará con más detalle en la sección 1.3. Las ponderaciones de la entrada pueden obtenerse directamente del FPD y entonces convertirse en capacitancias efectivas, escogiendo una tecnología CMOS dada, OnSemi de 0.5 micras para nuestro caso. Los resultados obtenidos de la simulación del diseño propuesto son comparados con resultados experimentales de una ALU implementada en un kit de evaluación FPGA Spartan 3E de Xilinx y una ALU de 4 bits comercial fabricada por Motorola: el chip MC14581B.

1.2 Estructura Básica de los Transistores MOS de compuerta flotante Multi-entrada (MIFG).

El MOSFET de compuerta flotante multi-entrada (MIFG), ilustrado en la figura 1.1, como es el caso del transistor neuMOS [8], se puede implementar sobre un proceso CMOS convencional de doble poli-silicio. El neuMOS está formado por una compuerta Flotante (Poly1), aislada eléctricamente y acoplada capacitivamente (poly1 – poly2) a varias compuertas de control (Poly2).

La compuerta flotante se extiende sobre el canal y el óxido de campo del MOSFET. Las compuertas de control o entradas del transistor, se forman sobre la compuerta flotante empleando una segunda capa de poli-silicio (poly2).

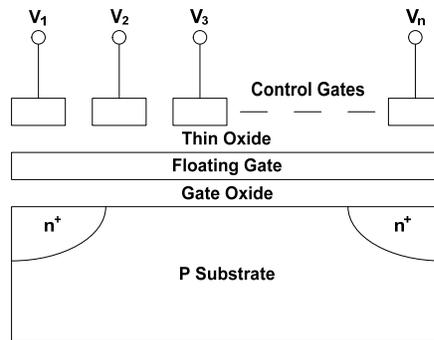


Figura 1.1. Estructura de un MOSFET de compuerta flotante con múltiples compuertas de control.

La figura 1.2 representa un circuito equivalente de la fig. 1.1, en el cual se muestran los acoplamientos capacitivos entre las compuertas de control $V_1, V_2, V_3, \dots, V_n$ y la compuerta flotante, denotados como $C_1, C_2, C_3, \dots, C_n$, donde n , corresponde al número de entradas al transistor. El término C_0 corresponde al capacitor formado entre la compuerta flotante y el sustrato. Mientras que V_{SS} corresponde a la polarización del sustrato y ϕ_{FG} el potencial de la compuerta Flotante.

Aplicando la ley de conservación de cargas para el nodo de la compuerta Flotante, tendremos que en cualquier instante, la carga de la compuerta Flotante, Q_{FG} , estará dada por:

$$Q_{FG}(t) = \sum_{i=1}^n C_i(V_i(t) - \phi_{FG}) - C_0(\phi_{FG} - V_{SS}) \quad \dots(1.1)$$

Desarrollando y agrupando términos, tendremos:

$$Q_{FG}(t) = \sum_{i=1}^n C_i V_i(t) + C_0 V_{SS} - \sum_{i=1}^n C_i \phi_{FG} - C_0 \phi_{FG} \quad \dots(1.2)$$

Por lo que la Ec. (1.2) se puede reescribir como:

$$Q_{FG}(t) = \sum_{i=1}^n C_i V_i(t) - \sum_{i=0}^n C_i \phi_{FG} + C_0 V_{SS} \quad \dots(1.3)$$

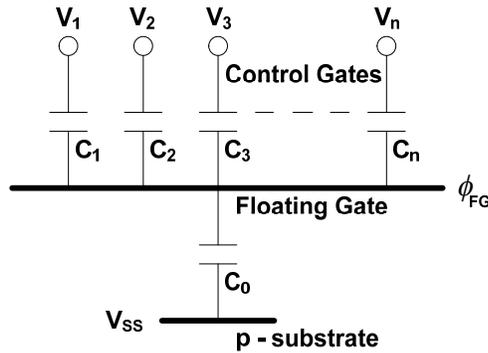


Figura 1.2. Capacitancias de Acoplamiento y terminales de voltaje de un MOSFET multi-entrada.

Asumiendo que la carga inicial en la compuerta Flotante y la polarización \$V_{ss}\$ son Cero, la Ec. (1.3) puede reescribirse de la siguiente manera:

$$\phi_{FG} = \frac{\sum_{i=1}^n C_i V_i(t)}{\sum_{i=0}^n C_i} = \frac{\sum_{i=1}^n C_i V_i(t)}{C_{TOT}} \quad \dots (1.4)$$

Donde la capacitancia total del circuito equivalente (\$C_{TOT}\$), estará definida por:

$$C_{TOT} = \sum_{i=0}^n C_i \quad \dots (1.5)$$

Sin embargo, uno de los parámetros a considerar en el desempeño de los transistores MOS de compuerta Flotante, es el coeficiente de acoplamiento, \$K_i\$, el cual representa una fracción del voltaje presente en la compuerta Flotante debido a un divisor de voltaje capacitivo equivalente en la estructura, el cual está dado por:

$$K_i = \frac{C_i}{\sum_{i=0}^n C_i} = \frac{C_i}{C_{TOT}} \quad \dots (1.6)$$

Donde, n , es el número de Entradas a la compuerta Flotante.

De la Ec. (1.4) puede observarse que la conmutación entre el encendido y apagado del transistor neuMOS dependerá respectivamente, de si el potencial de la compuerta flotante Φ_{FG} es mayor o menor que el voltaje de umbral del transistor (V_{th}).

Sin embargo, el valor del potencial de la compuerta flotante determinado por la Ec. (1.4) será válido, siempre y cuando, todos los coeficientes capacitivos del transistor (K_i), permanezcan sin cambios durante el funcionamiento del dispositivo. La capacitancia de óxido, C_o , se asume constante.

El desempeño del transistor MIFG, es similar al de una neurona biológica, [9], ya que éste realiza la suma ponderada de sus entradas, lo cual determinará el momento de encendido y apagado del transistor MOS, haciendo el uso del dispositivo más apropiado en el desarrollo de redes neuronales [10].

No obstante, aprovechando este principio, el dispositivo es empleado para configurar un celda básica neuMOS, que realice las seis compuertas lógicas, de una manera diferente a como se realizan las compuertas lógicas CMOS convencionalmente, es decir, con una sola configuración básica y sin alterar sus interconexiones, por lo que cualquier compuerta podrá obtenerse simplemente modificando los voltajes externos aplicados a la etapa de inversores programables, lo cual es llamado Lógica de Configuración Externa o Soft-Hardware Logic (SHL).

Una ventaja en la implementación de funciones lógicas empleando esta metodología con respecto a la forma convencional de implementación CMOS, es el número reducido de dispositivos empleados. Por ejemplo, un sumador completo SHL contendrá 8 transistores neuMOS, mientras que un diseño convencional con compuertas CMOS emplea 50 transistores [11]. Sin embargo un inconveniente que puede tener la implementación SHL, es la gran capacitancia creada por las entradas, dado que su área puede ser grande, afectando así la velocidad de la lógica.

1.3 Diagrama de Potencial de compuerta Flotante.

El desarrollo de la unidad aritmética lógica (ALU) de 4 Bits a partir de una celda básica aplicable en su configuración, se realizó mediante el apoyo del método grafico llamado "Diagrama de Potencial de Compuerta Flotante (FPD)" [12].

El FPD es una representación del comportamiento del circuito SHL, cuyo objetivo no solo es mostrar el potencial presente en la compuerta flotante del inversor MIFG CMOS, que determinará la configuración de salida (V_{OUT}) del circuito SHL a medida que las posibles combinaciones digitales se presenten en las compuertas de entrada de la neurona MOS, sino que también permite la estimación de los coeficientes de acoplamiento capacitivo, el numero de inversores programables requeridos y más importante aún, los voltajes de inversión necesarios para la implementación de dicha función.

La figura 1.3 muestra el FPD para implementar la función lógica X-NOR de 4 bits, sin embargo, este circuito también puede implementar una compuerta AND, NAND, OR, NOR o X-OR, cuando los voltajes externos de los inversores programables V_A, V_B, \dots, V_F de la celda básica SHL son debidamente seleccionados, lo cual se mostrará a detalle en el capítulo 2.

Podemos determinar la magnitud de las capacitancias de acoplamiento en términos de la ganancia del dispositivo de compuerta flotante γ , la cual corresponde a la razón entre la suma de las capacitancias de acoplamiento y la capacitancia total.

$$\gamma = \frac{C_1 + C_A + C_B + C_C + C_D + C_E + C_F}{C_{TOT}} \quad \dots(1.7)$$

Donde:

$$C_{TOT} = C_0 + C_1 + \dots + C_N \quad \dots(1.8)$$

$$C_0 = C_{0n} + C_{0p} \quad \dots(1.9)$$

$$C_1 = C_{X1} + C_{X2} + C_{X3} + C_{X4} \quad \dots(1.10)$$

$$\phi_{FG} = \frac{1}{C_{TOT}} * (V_P C_1 + V_A C_A + V_B C_B + V_C C_C + V_D C_D + V_E C_E + V_F C_F) \quad \dots(1.11)$$

Donde C_{0n} y C_{0p} , son las capacitancias de óxido de los transistores NMOS y PMOS del inversor ν MOS complementario respectivamente, por lo que C_0 será considerada como la capacitancia unitaria mínima; C_1 representa la suma de las capacitancias de acoplamiento de los 4 bits de la señal de entrada, como lo indica la ecuación (1.10). V_P representa los posibles valores de voltaje que se puede tener a la entrada de la configuración, el cual se considera continuo para facilidad del análisis presentado. V_A, \dots, V_F son los voltajes externos que se aplican a los inversores programables y con los cuales se configuran las diferentes funciones lógicas.

Las subdivisiones que debe presentar el eje de las abscisas del FPD, están directamente relacionadas al número de bits de entrada, n , del circuito SHL, y este número se determina como sigue:

$$\# \text{ subdiv. (eje-x)} = 2^N \quad \dots(1.12)$$

Mientras que el número de subdivisiones del eje de las ordenadas, se obtiene de la siguiente manera:

$$\# \text{ subdiv. (eje-y)} = 2^{N+1} \quad \dots(1.13)$$

Debido a que nuestra celda básica será de 4 bits, las subdivisiones del eje de las abscisas del FPD deben ser 16, mientras que para el eje de las ordenadas deben ser 32. Por otro lado, las ordenadas tendrán un valor máximo de V_{DD} , mientras que las abscisas tendrán un valor máximo de γV_{DD} cuando se lea el voltaje de ϕ_{FG} o γC_{TOT} cuando se lean capacitancias.

Diseño de una ALU empleando Lógica de Configuración Externa.

Para el uso de este tipo de representación de funciones, es necesario una referencia sobre la cual determinar cuándo el voltaje de salida es '1' o '0' lógico; debido a que se emplean inversores MIFG MOS simétricos, dicha referencia estará indicada a la mitad del FPD ($\gamma \frac{V_{DD}}{2}$) y se llamará línea de umbral.

Para el caso propuesto, cada una de las subdivisiones de la escala del eje de las abscisas, establecen las 16 combinaciones de entrada posibles, es decir desde "0000" hasta "1111" a medida que V_p aumenta, como puede observarse en la figura 1.3.2. De tal forma que:

Si $\phi_{FG} > \gamma \frac{V_{DD}}{2}$, entonces V_{OUT} será un '1' lógico.

Y cuando $\phi_{FG} < \gamma \frac{V_{DD}}{2}$, el V_{OUT} será un '0' lógico.

La línea con mayor grosor de la figura 1.3.2, representa el voltaje presente en la compuerta flotante (ϕ_{FG}) de la etapa de salida del v -MOS a medida que V_p aumenta desde 0V hasta V_{DD} . Sin embargo, para poderla graficar, ésta debe seguir la pendiente de una línea base, que analíticamente puede obtenerse a partir de la Ec. (1.11), cuando el voltaje aplicado a las entradas $V_A=V_B=V_C=V_D=V_E=V_F=0V$, resultando en la siguiente ecuación:

$$\phi_{FG} = \frac{V_p C_1}{C_{TOT}} \quad \dots(1.14)$$

Por lo que a medida que V_p aumenta, la línea base tendrá las coordenadas $(0, 0)$ hasta $(\frac{2^{N+1}}{2}, \frac{2^{N+1}}{2})$. Lo anterior se puede observar en la figura 1.3.2. La tabla de verdad en que se basó el FPD representado, es la tabla de la función X-NOR mostrada en la tabla 1.1

Tabla 1.1 Tabla de Verdad de la compuerta X-NOR de 4 bits.

Entradas.				Salida.
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

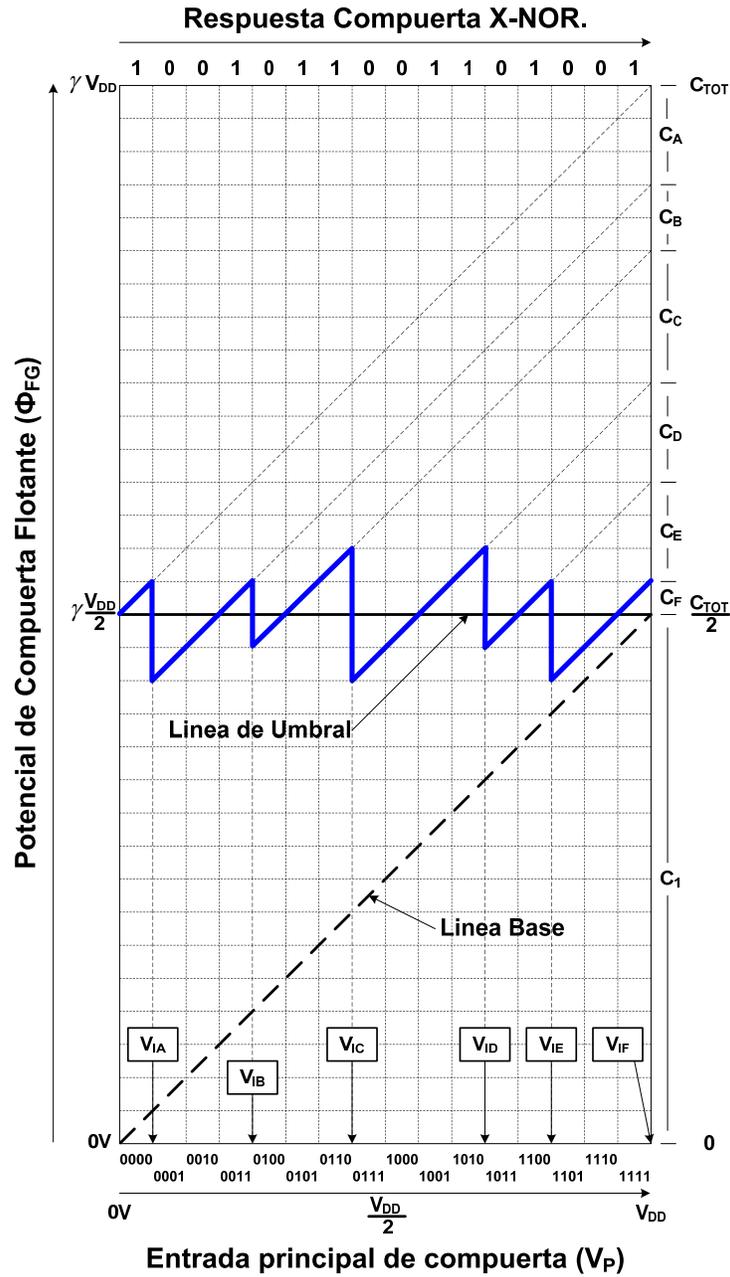


Fig. 1.3. Diagrama de Potencial de Compuerta Flotante para la implementación de una compuerta con función X-NOR de 4 Bits.

1.4 Ventajas y Desventajas del uso de Transistores MIFG MOS.

Los dispositivos de compuerta flotante tienen diversas características que los hacen diferentes de los dispositivos CMOS convencionales, entre los que se encuentran:

Para transistores MOS convencionales:

- Los inversores CMOS convencionales, poseen como máximo una compuerta de control.
- Para la implementación de una función, requerirá de un área fija y específica que dependerá del número de componentes. Por lo que si se requiere alguna otra implementación adicional, ocupará un área mayor.
- Los tiempos de retardo en arreglos con transistores convencionales son de valor pequeño.
- Los límites de frecuencia, para arreglos con transistores convencionales (ALU) son en su mayoría de valores elevados (por arriba de 10MHz).

Para la lógica vMOS:

- Pueden tener más de una compuerta de control.
- Para la implementación de funciones, presentan un ahorro de transistores, ya que con una sola configuración y la misma área pueden implementar diferentes funciones.
- Los tiempos de retardo son mayores en referencia a los de circuitos MOS convencionales.
- Por lo tanto, sus límites de frecuencia son menores a los de la lógica convencional.

1.5. Conclusiones.

En este capítulo se presentaron los antecedentes y la estructura básica del transistor MIFG MOS, el cual se empleará en el desarrollo de una celda básica SHL para el diseño de una ALU de 4 bits.

Debido a que la celda básica SHL emplea un método gráfico denominado (FPD) para la implementación de diversas funciones, se ha incluido en este capítulo una breve descripción de cómo se construye y qué información puede ser extraída de dicha representación con base a su tabla de verdad.

También se presentaron algunas ventajas y desventajas de emplear circuitos SHL con respecto a los circuitos fabricados con tecnología CMOS convencional.

1.6 Referencias.

- [1] . "Overview of Floating-Gate Devices, Circuits, and Systems". IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing. vol. 48. No. 1, January 2001.
- [2] T. Shibata, T Ohmi. "An intelligent MOS transistor featuring gate-level weighted sum and threshold operations". *Electron Devices Meeting, 1991. IEDM '91. Technical Digest., International* , vol., no., pp.919-922, 8-11 Dec 1991
- [3] T. Shibata. "A Functional MOS Transistor Featuring Gate-Level weighed Sum and Threshold Operations". IEEE Transactions on electron devices, vol. 39. No. 6, Jne 1992.
- [4] D. Kahng, S.M Sze, "A floating gate and its application to memory devices," Bell Syst. Tech. J. vol. 46, p. 1283, 1967.
- [5] T. C. Ong, P. K. Ko, C. Hu. "The EEPROM as an analog memory device," IEEE Transactions on Electron Devices. vol. 36, pp. 1840-1841, 1989.
- [6] T. Kaminski, P. Holajn. "Multiple-input floating-gate MOS transistor in analogue electronics circuit". Bulletin of the Polish academy of Sciences. Vol. 52. No. 3, 2004.
- [7] M. Holler, S. Tam, H Castro and R. Benson, "An electrical trainable artificial neural network (ETANN) with 10240 floating gate synapses," in Proc. Of int. Joint Conf. Neural Networks, vol. II, pp. 191-196, 1989.
- [8] B. Minch, P. Hasler. "A floating-gate technology for digital CMOS processes". Proceedings of the IEEE International Symposium on Circuits and Systems. Vol. 2. p. 400 – 403. 1999.
- [9] H. Ishii, T. Shibata, H. Kosaka, T. Ohmi. "Hardware-learning neural network LSI using a highly-functional transistor simulating neuron actions". Proceedings of International Joint Conference on Neural Networks. Vol. 1. p. 907 – 910. 1993-Nagoya.
- [10] T. Shibata, T. Ohmi. "A Self-Learning Neural-Network LSI using Neuron MOSFET's". Symposium on VLSI Technology. Digest of Technical Papers. p. 84 – 85. 1992.
- [11] K. Kotani, T. Shibata, T. Ohmi. "Neuron-MOS Binary-Logic Circuits Featuring Dramatic Reduction in Transistor Count and Interconnections". International Meeting on Electron Devices, Technical Digest. 1992.
- [12] A. Medina-Santiago, M. A. Reyes-Barranca. "Circuit for Logical-binary Functions using MOS Floating-gate Devices". 2nd International Conference on Electrical and Electronics Engineering. p. 211-214. 2005.

Capítulo II. Metodología del Diseño.

2.1 Introducción.

Este capítulo comprende la metodología de diseño de un Circuito Lógico de Configuración Externa o SHL, basado en el Diagrama de Potencial de Compuerta Flotante (FPD), el cual se empleó en el desarrollo de una Unidad Aritmética Lógica (ALU) de 4 bits, con tecnología CMOS de doble poli-silicio de 0.5 micras (μm) de OnSemi.

El circuito lógico SHL tratado en este capítulo, implementará la Función lógica X-OR, como base para la explicación de la metodología, y que forma parte del bloque principal de la ALU diseñada: Un sumador completo con propagación de acarreo.

2.2 Celda Básica SHL de 4 bits.

La Figura 2.1 muestra la configuración propuesta de la celda básica SHL de este trabajo, la cual es capaz de representar cualquiera de las 6 funciones lógicas: AND, NAND, OR, NOR, X-OR y X-NOR de 4 bits de entrada, lo cual se logra modificando únicamente el valor del voltaje aplicado externamente a la etapa de inversores programables de la celda.

Sin embargo la capacidad de representación de Funciones de esta celda, puede extenderse a un mayor número, al emplear el FPD para su implementación, como se verá más adelante en el Capítulo III. Esto hace de este concepto algo innovador, debido a que una sola configuración puede representar múltiples funciones lógicas, mientras que las configuraciones de compuertas lógicas convencionalmente emplean una configuración diferente para cada implementación.

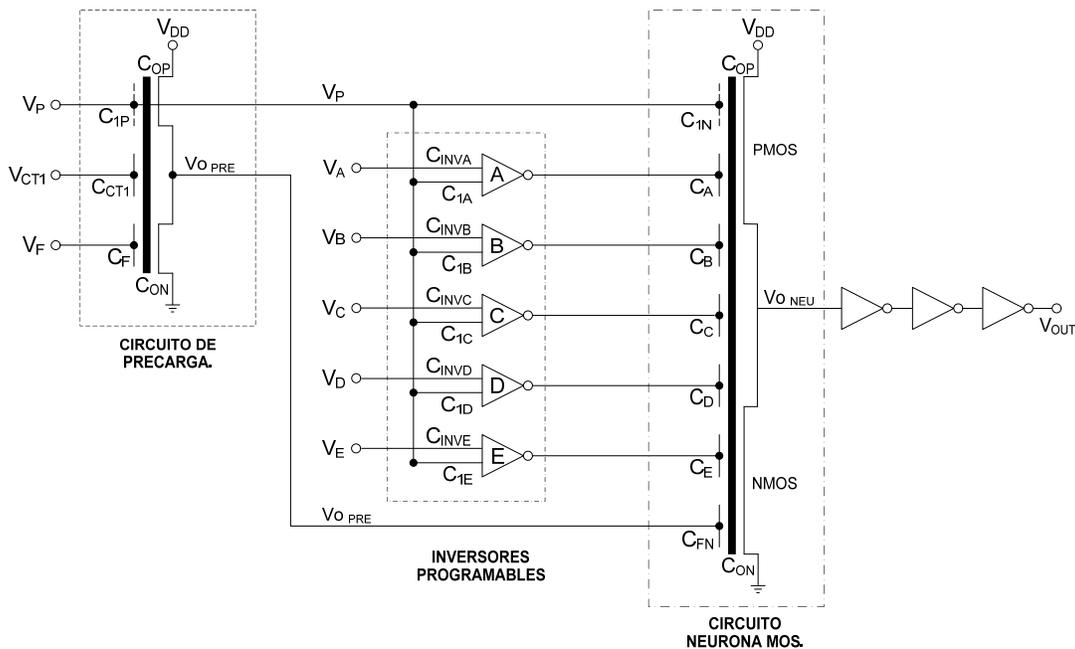


Figura 2.1. Configuración de la Celda Básica SHL neuMOS de 4 bits.

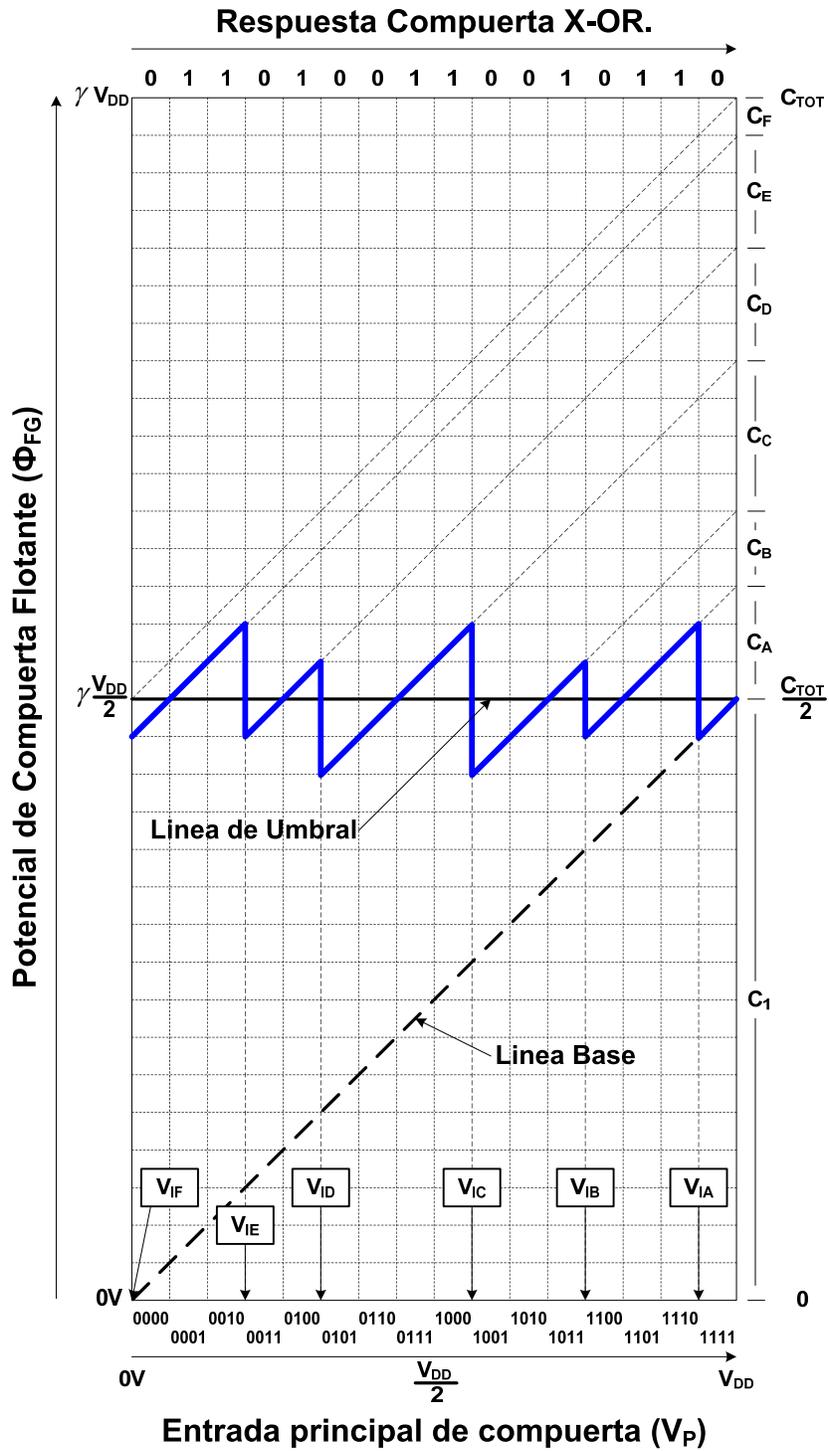


Figura 2.2. Diagrama de Potencial de compuerta Flotante correspondiente a las funciones X-OR.

La celda básica ilustrada en la figura 2.1, consta de tres etapas [1]:

La etapa inicial de pre-carga, está formada por un inversor MIFG CMOS con cuatro compuertas de entrada de señal de 4 bits, X_1 , X_2 , X_3 y X_4 , las cuales se han agrupado en una señal multivaluada, V_p , simplemente para propósitos de simulación en DC, y dos compuertas de entradas de control, V_{CTL} y V_F , empleadas para ajustar la salida de la etapa de pre-carga, $V_{0\ PRE}$, la cual también es una señal multivaluada, que depende directamente de la señal digital de 4 bits presente a la entrada.

La etapa intermedia de inversores programables, está formada por seis inversores MIFG CMOS, debido a que los FPDs de las compuertas X-OR y X-NOR presentan más transiciones que los FPDs de las compuertas AND, NAND, OR y NOR. Cabe mencionar que uno de los inversores programables, el inversor F, se encuentra asociado a la etapa de precarga.

Cada inversor programable MIFG CMOS se compone de cuatro compuertas de entrada de señal, X_1 , X_2 , X_3 y X_4 , agrupadas en V_p , y una compuerta de entrada de inversión, V_A , V_B , V_C , V_D o V_F , dependiendo del inversor que se trate. Los inversores programables son necesarios para formar las transiciones que se encuentran a lo largo de la línea de umbral del FPD que se desee implementar, como se ilustra en la figura 2.2.

La etapa final, llamada neurona MOS, está formada por un inversor MIFG CMOS con cuatro compuertas de entrada de señal de 4 bits, X_1 , X_2 , X_3 y X_4 , que también se han agrupado en una señal multivaluada, V_p , cinco compuertas de entrada provenientes de la etapa de salida de los inversores programables, V_{INVA} , V_{INVB} , V_{INVC} , V_{INVD} y V_{INVE} , y una compuerta de entrada proveniente de la etapa de salida del circuito de precarga, $V_{OUT\ PRE}$.

En la Compuerta Flotante del inversor de la etapa de neurona, se realiza la suma de las entradas ponderadas provenientes de la señal de entrada de 4 bits, de la salida de la etapa de inversores programables y de la salida de la etapa de precarga [2]. Habrá que recordar que cuando el voltaje de la compuerta flotante está por encima de la línea de umbral, se considera que se tiene un '1' lógico y un '0' lógico cuando está por debajo, como se mencionó en la sección 1.3 del Capítulo 1.

La Figura 2.2 ilustra el FPD de la Función Lógica X-OR de 4 bits, el cual es uno de los FPD que presenta mayores transiciones, empleando por consiguiente diferentes valores de voltaje de conmutación en los inversores programables de la celda, a diferencia del resto de las compuertas lógicas AND, NAND, OR y NOR.

Como se había mencionado con anterioridad, el número de inversores requeridos para la implementación de una Función, se realiza contando simplemente el número de transiciones presentes en el correspondiente FPD, para nuestro caso, serán seis.

Los coeficientes de acoplamiento capacitivo (K_i) pueden estimarse al leer sobre el eje de ordenadas (eje y) la extrapolación de las pendientes de cada una de las transiciones observadas en el FPD, mientras que los voltajes de conmutación de los inversores programables (V_{INV}), estarán definidos al extrapolar sobre el eje de las abscisas (eje x), los cruces de dichas transiciones.

A continuación se presenta el cálculo de los coeficientes de acoplamiento capacitivo de cada una de las etapas de la celda básica SHL.

2.3 Coeficientes de Acoplamiento de la etapa de pre-carga.

En la Figura 2.3(a), se muestra la configuración y en (b) la representación equivalente de la etapa del circuito de pre-carga de la celda básica SHL neuMOS de 4 bits ilustrada en la Fig. 2.1. Las señales de Entrada de 4 Bits, X_1 , X_2 , X_3 , y X_4 , están acopladas capacitivamente mediante C_{X1P} , C_{X2P} , C_{X3P} y C_{X4P} a la compuerta flotante de un inversor vMOS complementario, cuya salida pasa a través de dos inversores convencionales, con el fin de hacer mas abruptas las pendientes de la señal $V_{O\ PRE}$.

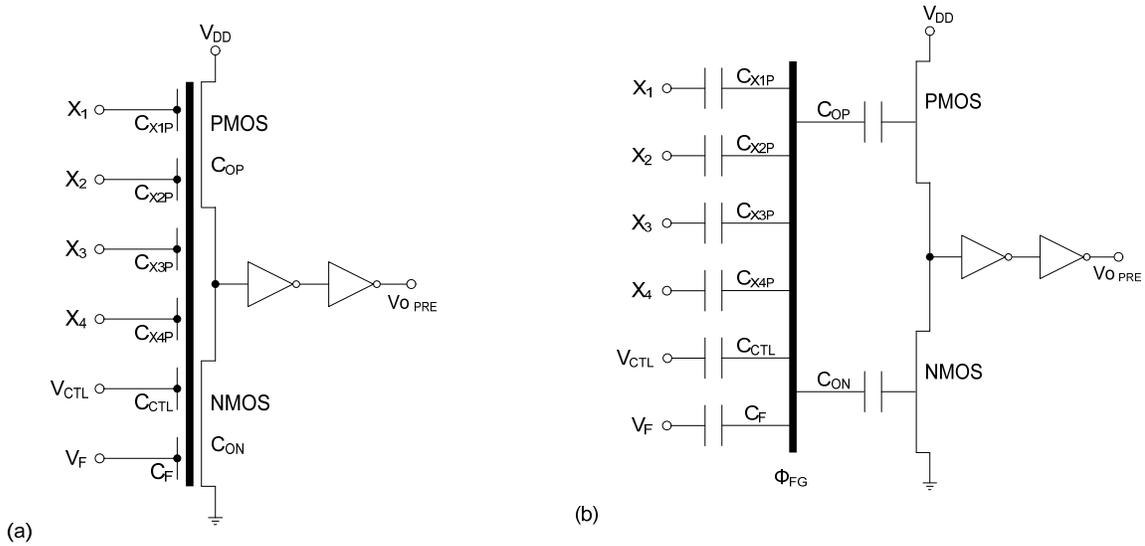


Figura 2.3. Configuración (a) y representación equivalente (b) del Circuito de Precarga de la celda básica SHL neuMOS de 4 bits mostrada en la Fig. 2.1.

De la figura 2.3 (b) y tomando como referencia la Ec. (1.4), el potencial de la compuerta Flotante del circuito de pre-carga ($\Phi_{FG\ PRE}$), está dado por:

$$\Phi_{FG\ PRE} = \frac{X_1 C_{X1P} + X_2 C_{X2P} + X_3 C_{X3P} + X_4 C_{X4P} + V_{CTL} C_{CTL} + V_F C_F}{C_{TOT\ PRE}} \quad \dots(2.2)$$

Donde:

$$C_{TOT\ PRE} = C_0 + C_{X1P} + C_{X2P} + C_{X3P} + C_{X4P} + C_{CTL} + C_F \quad \dots(2.3)$$

Donde la capacitancia unitaria mínima C_0 , que aparece en el término anterior, se obtiene de la suma de las capacitancias de compuerta de los transistores que forman el inversor neuMOS complementario y haciendo que la capacitancia C_F sea igual a la capacitancia C_0 , tendremos que:

$$C_0 = C_{ON} + C_{OP} = C_F \quad \dots(2.4)$$

El criterio de diseño empleado para cuantificar las capacitancias de acoplamiento de entrada de señal, C_{X1P} , C_{X2P} , C_{X3P} y C_{X4P} a la compuerta flotante del circuito de precarga, fue respetar el peso binario de cada una de las señales de Entrada de 4 Bits, tomando como referencia la capacitancia unitaria C_0 , es decir:

$$C_{X1P} = 2^0 * C_0 = C_0 \quad \dots(2.5)$$

$$C_{X2P} = 2^1 * C_0 = 2C_0 = 2C_{X1P} \quad \dots(2.6)$$

$$C_{X3P} = 2^2 * C_0 = 4C_0 = 4C_{X1P} \quad \dots(2.7)$$

$$C_{X4P} = 2^3 * C_0 = 8C_0 = 8C_{X1P} \quad \dots(2.8)$$

Haciendo que la capacitancia de control (C_{CTL}), sea de igual magnitud que la de Entrada (C_{1P}), tendremos que:

$$C_{CTL} = C_{1P} = C_{X1P} + C_{X2P} + C_{X3P} + C_{X4P} \quad \dots(2.9)$$

Por lo que sustituyendo las Ecs. (2.4 - 2.9) en la Ec. (2.3), tendremos que la capacitancia total del circuito de carga está dada por:

$$C_{TOT PRE} = 32C_{X1P} \quad \dots(2.10)$$

Despejando C_{X1P} en términos de $C_{TOT PRE}$ de la Ec. (2.10) y sustituyendo las Ecs. (2.5 - 2.9), los coeficientes de acoplamiento capacitivo de la etapa de precarga, serán los mostrados en la tabla 2.1.

Tabla 2.1. Coeficientes de acoplamiento de la etapa de precarga del circuito básico SHL propuesto.

$\frac{C_{X1P}}{C_{TOT PRE}} = \frac{1}{32}$	$\frac{C_{X2P}}{C_{TOT PRE}} = \frac{2}{32}$	$\frac{C_{X3P}}{C_{TOT PRE}} = \frac{4}{32}$
$\frac{C_{X4P}}{C_{TOT PRE}} = \frac{8}{32}$	$\frac{C_{CTL}}{C_{TOT PRE}} = \frac{15}{32}$	$\frac{C_F}{C_{TOT PRE}} = \frac{1}{32}$

Cabe mencionar que los coeficientes de acoplamiento de esta etapa de precarga son independientes de la función que se quiere implementar mediante la celda básica SHL.

2.4 Coeficientes de Acoplamiento de la etapa de Neurona.

Ahora se presenta en la Figura 2.4(a) la configuración y en (b) la representación equivalente de la etapa neurona de la celda básica SHL neuMOS de 4 bits ilustrada en la Fig. 2.1. Aquí se puede observar también que las cuatro señales de Entrada X_1, X_2, X_3, X_4 , están acopladas mediante $C_{X1N}, C_{X2N}, C_{X3N}$ y C_{X4N} a la compuerta flotante de un inversor vMOS complementario, cuya salida alimenta 3 inversores convencionales, los cuales son los encargados de hacer mas abruptas las conmutaciones de la señal $V_{O_{NEU}}$, y de generar el FPD del Circuito Lógico deseado.

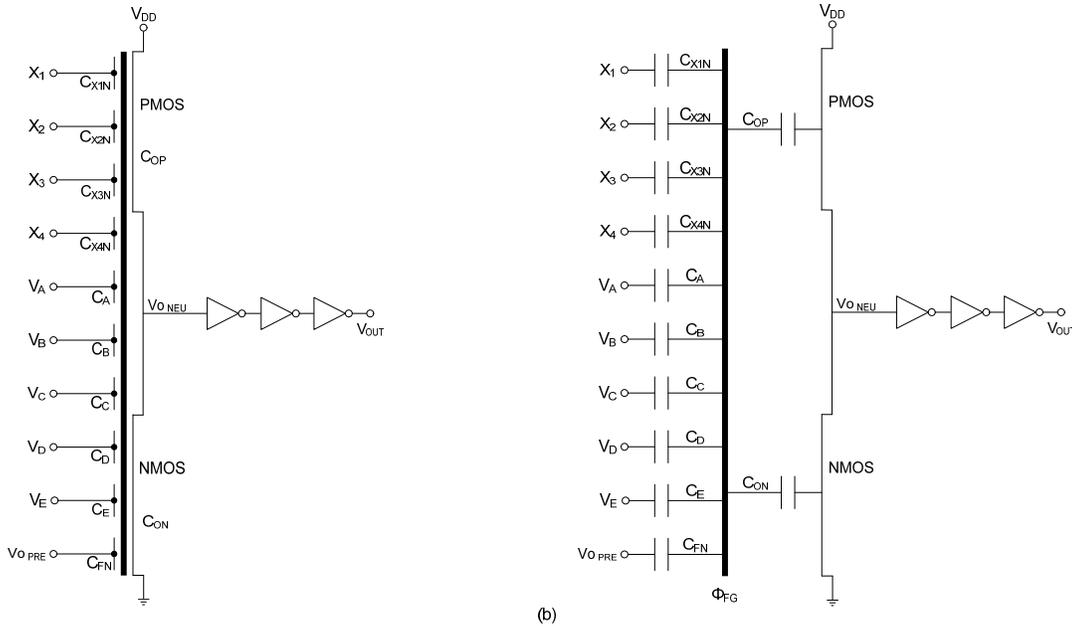


Figura 2.4. Configuración (a) y representación equivalente (b) de la etapa de neurona (neuMOS) de la celda básica SHL neuMOS de 4 bits mostrada en la Fig. 2.1.

Procediendo de forma similar como se derivaron las capacitancias de la etapa de precarga, procedemos ahora con la etapa neurona, como sigue:

El potencial de la compuerta flotante de la neurona ($\Phi_{FG_{NEU}}$), de acuerdo a la figura 2.4(b), esta dado por:

$$\Phi_{FG_{NEU}} = \frac{X_1 C_{X1N} + X_2 C_{X2N} + X_3 C_{X3N} + X_4 C_{X4N} + V_A C_A + V_B C_B + V_C C_C + V_D C_D + V_E C_E + V_{O_{PRE}} C_{FN}}{C_{TOT_{NEU}}} \quad \dots(2.11)$$

Donde, la capacitancia total, estará dada por:

$$C_{TOT_{NEU}} = C_0 + C_{X1N} + C_{X2N} + C_{X3N} + C_{X4N} + C_A + C_B + C_C + C_D + C_E + C_{FN} \quad \dots(2.12)$$

Nuevamente, haciendo que C_{FN} sea igual a la capacitancia unitaria (mínima) C_0 , tendremos:

$$C_0 = C_{ON} + C_{OP} = C_{FN} \quad \dots(2.13)$$

Siguiendo el criterio de diseño sugerido anteriormente ahora para cuantificar las capacitancias de acoplamiento de entrada de señal C_{X1N} , C_{X2N} , C_{X3N} y C_{X4N} a la compuerta flotante del circuito neurona, tendremos:

$$C_{X1N} = 2^0 * C_0 = C_0 \quad \dots(2.14)$$

$$C_{X2N} = 2^1 * C_0 = 2C_0 = 2C_{X1N} \quad \dots(2.15)$$

$$C_{X3N} = 2^2 * C_0 = 4C_0 = 4C_{X1N} \quad \dots(2.16)$$

$$C_{X4N} = 2^3 * C_0 = 8C_0 = 8C_{X1N} \quad \dots(2.17)$$

Debido a que el comportamiento del inversor NeuMOS de esta etapa, determinará la función que implementará el circuito SHL, X-OR en nuestro caso, los coeficientes de acoplamiento capacitivo K_A , K_B , K_C , K_D , K_E y K_{FN} , se obtienen mediante la lectura del correspondiente FPD (Fig. 2.2). Tendremos que:

$$\begin{array}{ccc} \frac{C_A}{C_{TOT NEU}} = \frac{3}{32} & \frac{C_B}{C_{TOT NEU}} = \frac{2}{32} & \frac{C_C}{C_{TOT NEU}} = \frac{4}{32} \\ \frac{C_D}{C_{TOT NEU}} = \frac{3}{32} & \frac{C_E}{C_{TOT NEU}} = \frac{3}{32} & \frac{C_{FN}}{C_{TOT NEU}} = \frac{1}{32} \end{array}$$

Sustituyendo las Ecs. (2.13 - 2.17) en la Ec. (2.12), tendremos que la capacitancia total del inversor de la etapa de neurona, estará dada por:

$$C_{TOT NEU} = 1 = C_0 + C_{X1N} + C_{X2N} + C_{X3N} + C_{X4N} + \frac{3}{32} + \frac{2}{32} + \frac{4}{32} + \frac{3}{32} + \frac{3}{32} + \frac{1}{32} \quad \dots(2.18)$$

O bien:

$$C_{TOT NEU} = 1 = C_0 + C_{X1N} + C_{X2N} + C_{X3N} + C_{X4N} + \frac{16}{32} \quad \dots(2.19)$$

Sustituyendo (2.14 – 2.17) en la Ec. (2.19), tendremos:

$$C_{TOT NEU} = \frac{16}{32} = 16 C_{X1N} \quad \dots(2.20)$$

Por lo que es claro observar, que al despejar C_{X1N} de la Ec. (2.20) los coeficientes de acoplamiento de la señal de entrada a la neurona neuMOS serán los enlistados en la tabla 2.2.

$$\frac{C_{X1N}}{C_{TOT NEU}} = \frac{1}{32}$$

$$\frac{C_{X2N}}{C_{TOT NEU}} = \frac{2}{32}$$

$$\frac{C_{X3N}}{C_{TOT NEU}} = \frac{4}{32}$$

$$\frac{C_{X4N}}{C_{TOT NEU}} = \frac{8}{32}$$

De esta manera, los coeficientes de acoplamiento correspondientes a la etapa Neuronal para la implementación de la función X-OR, serán los enlistados en la tabla 2.2.

Tabla 2.2. Coeficientes de acoplamiento de la etapa Neuronal del circuito básico SHL propuesto, para la implementación de la función X-OR.

$\frac{C_{X1N}}{C_{TOT NEU}} = \frac{1}{32}$	$\frac{C_{X2N}}{C_{TOT NEU}} = \frac{2}{32}$	$\frac{C_{X3N}}{C_{TOT NEU}} = \frac{4}{32}$	$\frac{C_{X4N}}{C_{TOT NEU}} = \frac{8}{32}$	$\frac{C_A}{C_{TOT NEU}} = \frac{3}{32}$
$\frac{C_B}{C_{TOT NEU}} = \frac{2}{32}$	$\frac{C_C}{C_{TOT NEU}} = \frac{4}{32}$	$\frac{C_D}{C_{TOT NEU}} = \frac{3}{32}$	$\frac{C_E}{C_{TOT NEU}} = \frac{3}{32}$	$\frac{C_{FN}}{C_{TOT NEU}} = \frac{1}{32}$

Es importante hacer notar en este momento que las capacitancias de acoplamiento mostradas en la tabla 2.2, las cuales fueron obtenidas analíticamente con base a criterios de diseño, coinciden con las derivadas del FPD ilustrado en la figura 2.2, por lo que se puede afirmar que una forma simple para la derivación de estas capacitancias, es mediante el método gráfico del FPD referida a la compuerta que se desee diseñar.

Finalmente a continuación presentamos el cálculo de los coeficientes de la etapa de inversores programables.

2.5 Coeficientes de Acoplamiento de la etapa de inversores Programables.

En la Figura 2.5(a) se muestra la configuración y en 2.5(b) la representación equivalente del inversor A, de la etapa de inversores programables de la celda básica SHL neuMOS de 4 bits ilustrada en la Fig. 2.1.

Como se observó en las anteriores etapas, las cuatro entradas de señal X_1, X_2, X_3, X_4 , están directamente acopladas mediante $C_{X1A}, C_{X2A}, C_{X3A}$ y C_{X4A} a la compuerta flotante de un inversor vMOS complementario, cuya salida alimenta 2 inversores convencionales, los cuales son los encargados de hacer mas abruptas las transiciones de la señal V_A .

Procediendo nuevamente de forma similar al cálculo realizado para las capacitancias de las etapas anteriores, procedemos ahora con la etapa de inversores programables como se describe a continuación:

A partir de la Fig. 2.5 (b), el potencial de la compuerta Flotante de la etapa de inversores programables ($\Phi_{FG INV}$) estará dado por:

$$\Phi_{FG INV} = \frac{X_1 C_{X1A} + X_2 C_{X2A} + X_3 C_{X3A} + X_4 C_{X4A} + V_{INVA} C_{INVA}}{C_{TOT INV}} \quad \dots(2.21)$$

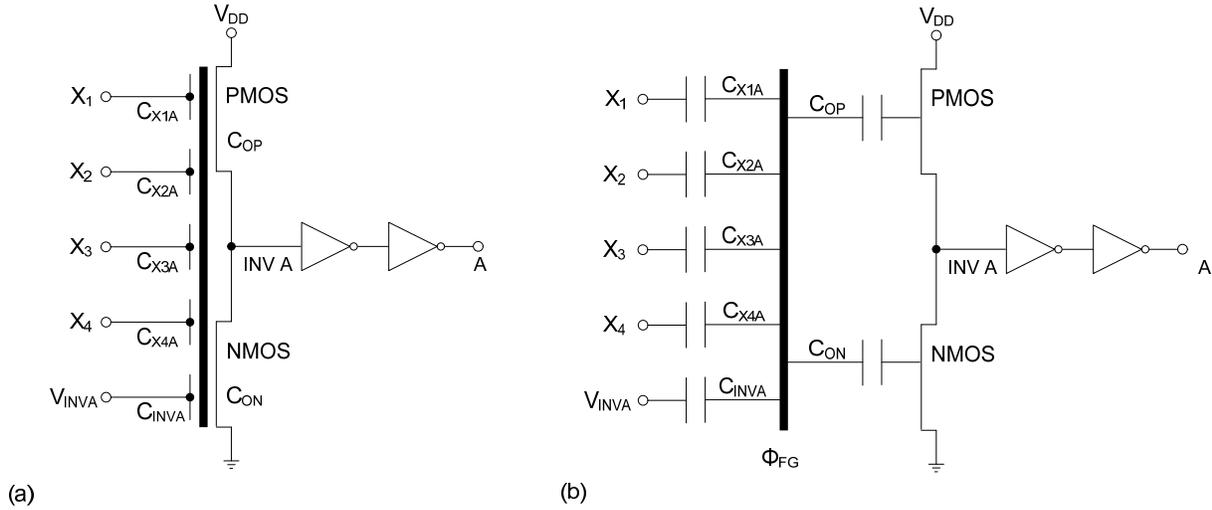


Figura 2.5. Configuración (a) y representación equivalente (b) de la etapa de inversores programables de la celda básica SHL neuMOS de 4 bits mostrada en la Fig. 2.1.

La capacitancia total de la etapa de inversores programables, está definida por:

$$C_{TOT\ INV} = C_0 + C_{X1A} + C_{X2A} + C_{X3A} + C_{X4A} + C_{INVA} \quad \dots(2.22)$$

Definiendo nuevamente la capacitancia unitaria (mínima) C_0 , como la suma de las capacitancias formadas por los substratos del inversor, tendremos:

$$C_0 = C_{ON} + C_{OP} \quad \dots(2.23)$$

Siguiendo el criterio empleado para la ponderación de los coeficientes de acoplamiento de señal de entrada, como en el caso de la etapa de pre-carga y la etapa de Neurona, tendremos que:

$$C_{X1A} = 2^0 * C_0 = C_0 \quad \dots(2.24)$$

$$C_{X2A} = 2^1 * C_0 = 2C_0 = 2C_{X1A} \quad \dots(2.25)$$

$$C_{X3A} = 2^2 * C_0 = 4C_0 = 4C_{X1A} \quad \dots(2.26)$$

$$C_{X4A} = 2^3 * C_0 = 8C_0 = 8C_{X1A} \quad \dots(2.27)$$

Diseño de una ALU empleando Lógica de Configuración Externa.

Por lo que igualando la capacitancia de inversión C_{INVA} del inversor A con la capacitancia de entrada a la compuerta Flotante C_{1A} , la cual reúne las cuatro capacitancias de entrada de señal C_{X1A} , C_{X2A} , C_{X3A} , C_{X4A} , tendremos que:

$$C_{INVA} = C_{1A} = C_{X1A} + C_{X2A} + C_{X3A} + C_{X4A} \quad \dots(2.28)$$

Proponiendo que las capacitancias involucradas sean de igual tamaño, tendremos que:

$$C_0 = C_{1A} = C_{INVA} \quad \dots(2.29)$$

Entonces, sustituyendo la Ec. (2.29) en la Ec. (2.22), observamos que:

$$C_{TOT\ INV} = C_0 + C_{1A} + C_{INVA} = 3C_{1A} \quad \dots(2.30)$$

Sustituyendo la Ec. (2.28) y las Ecs. (2.24-27), en la Ec. (2.30), tendremos:

$$C_{TOT\ INV} = 3C_{1A} = 3(C_{X1A} + 2C_{X1A} + 4C_{X1A} + 8C_{X1A}) \quad \dots(2.31)$$

Resultando en:

$$C_{TOT\ INV} = 45C_{X1A} \quad \dots(2.32)$$

Por lo que despejando C_{X1A} en función de $C_{TOT\ INV}$, de la Ec. (2.32) y reemplazando las Ecs. (2.24 - 2.28), los coeficientes de Acoplamiento de la etapa de inversores programables, quedaran definidos como se muestra en la tabla 2.3.

Tabla 2.3. Coeficientes de acoplamiento de la etapa de inversores programables del circuito básico SHL propuesto, cuando el criterio de diseño es $C_0 = C_{X1A} = C_{INVA}$.

$\frac{C_{X1A}}{C_{TOT\ INV}} = \frac{1}{45}$	$\frac{C_{X2A}}{C_{TOT\ INV}} = \frac{2}{45}$	$\frac{C_{X3A}}{C_{TOT\ INV}} = \frac{4}{45}$
$\frac{C_{X4A}}{C_{TOT\ INV}} = \frac{8}{45}$	$\frac{C_{INVA}}{C_{TOT\ INV}} = \frac{15}{45}$	

Cabe mencionar que los coeficientes de acoplamiento de la etapa de inversores programables, tampoco dependen de la función que el circuito SHL implementará.

2.6. Voltajes de conmutación de los inversores programables.

Una vez que se obtuvieron los coeficientes de acoplamiento capacitivos de cada una de las etapas del circuito básico SHL, procedemos a estimar los voltajes de inversión requeridos por la etapa de inversores programables para implementar la función lógica X-OR.

Partiendo de que los inversores programables deber ser simétricos, es decir, que la conmutación entre un '0' y un '1' lógico debe realizarse cuando en su compuerta flotante se encuentre presente la mitad del voltaje de polarización, es decir:

$$\Phi_{FG\ INV} = \frac{1}{2}V_{DD} \quad \dots(2.33)$$

Y sustituyendo los coeficientes de acoplamiento de la tabla 2.3 en la Ec. (2.21), harán que $\Phi_{FG\ INV}$ quede expresado como:

$$\Phi_{FG\ INV} = \frac{1}{45}X_1 + \frac{2}{45}X_2 + \frac{4}{45}X_3 + \frac{8}{45}X_4 + \frac{15}{45}V_{INVA} \quad \dots(2.34)$$

Considerando la Ec. (2.33), la Ec. (2.34) puede reescribirse como:

$$\frac{1}{2}V_{DD} = \frac{1}{3}X_{1A} + \frac{1}{3}V_{INVA} = \frac{1}{3}(X_{1A} + V_{INVA}) \quad \dots(2.35)$$

Nuevamente, a partir del FPD de la Función X-OR ilustrado en la Figura 2.2, los voltajes de inversión para cada uno de los inversores programables se deducen como se explicó anteriormente, los cuales se observan en la tabla 2.4.

Tabla 2.4. Voltajes de Inversión de los inversores programables de la celda básica SHL para la implementación de la función X-OR.

$V_{IA} = \frac{15}{16}V_{DD}$	$V_{IB} = \frac{12}{16}V_{DD}$	$V_{IC} = \frac{9}{16}V_{DD}$
$V_{ID} = \frac{5}{16}V_{DD}$	$V_{IE} = \frac{3}{16}V_{DD}$	$V_{IF} = 0V_{DD}$

Despejando V_{INVA} de la Ec. (2.35), tendremos que:

$$V_{INVA} = \frac{3}{2}V_{DD} + X_{1A} \quad \dots(2.36)$$

Si reemplazamos el factor X_{1A} de la Ec. (2.36) por los voltajes enlistados en la tabla 2.4 y resolviendo para V_{INVA} , obtendremos los Voltajes que aplicados externamente producirán la conmutación de los

inversores programables o en otras palabras, la Función que se desea implementar, en este caso la función X-OR.

La tabla 2.5, muestra los voltajes externos aplicados a la etapa de inversores programables, los cuales producirán la implementación de la función X-OR en la celda básica SHL considerando que $V_{DD}=5V$.

Tabla 2.5. Voltajes Externos aplicados a los inversores programables de la celda básica SHL para producir la Función X-OR, cuando el criterio de diseño es $C_0 = C_{X1A} = C_{INVA}$ con $V_{DD}=5V$.

$V_A = 2.81 V$	$V_B = 3.75 V$	$V_C = 4.68 V$
$V_D = 5.93 V$	$V_E = 6.56 V$	$V_F = 7.5 V$

Sin embargo y debido al criterio de igualdad entre capacitancias de la Ec. (2.29) para la etapa de inversores programables, surge un inconveniente, los valores externos de voltaje encontrados para los inversores programables D, E y F de esta etapa están por encima de V_{DD} .

Lo anterior supone el uso de una fuente de voltaje más alta para poder derivar de ella, el voltaje de alimentación V_{DD} y los demás voltajes correspondientes, por lo que sería deseable que estos voltajes externos estuvieran dentro del rango de 0 a V_{DD} .

Para cumplir con el lineamiento anterior, se realizó una modificación al criterio de igualdad, se mantuvo la relación entre la capacitancia unitaria C_0 y la capacitancia de entrada a la compuerta Flotante C_{X1A} , y se propuso que la capacitancia de inversión C_{INVA} del inversor A, fuera del doble del tamaño de la capacitancia de entrada a la Compuerta Flotante C_{X1A} , es decir:

$$C_0 = C_{1A} \quad \dots(2.37)$$

$$C_{INVA} = 2C_{1A} \quad \dots(2.38)$$

Por lo que sustituyendo las Ecs. (2.28), (2.37) y (2.38) en la Ec. (2.22), tendremos que:

$$C_{TOT INV} = C_0 + C_{1A} + C_{INVA} = 4C_{1A} \quad \dots(2.39)$$

Sustituyendo la Ecs. (2.24 - 2.27) en la Ec. (2.39), tendremos:

$$C_{TOT INV} = 4C_{1A} = 4(C_{X1A} + 2C_{X1A} + 4C_{X1A} + 8C_{X1A}) \quad \dots(2.40)$$

Obteniendo finalmente que la capacitancia total de la etapa de inversores programables, estará dada por:

$$C_{TOT INV} = 60C_{X1A} \quad \dots(2.41)$$

Por lo que despejando C_{X1A} en función de $C_{TOT INV}$, de la Ec. (2.41) y reemplazando las Ecs. (2.24 - 2.27) y la Ec. (2.38), los coeficientes de acoplamiento de la etapa de inversores programables, quedarán definidos como se muestran en la tabla 2.6.

Tabla 2.6. Coeficientes de acoplamiento de la etapa de inversores programables del circuito básico SHL propuesto, cuando el criterio de diseño es $C_0 = C_{X1A}$ y $C_{INVA} = 2 C_{X1A}$.

$\frac{C_{X1A}}{C_{TOT INV}} = \frac{1}{60}$	$\frac{C_{X2A}}{C_{TOT INV}} = \frac{2}{60}$	$\frac{C_{X3A}}{C_{TOT INV}} = \frac{4}{60}$
$\frac{C_{X4A}}{C_{TOT INV}} = \frac{8}{60}$		$\frac{C_{INVA}}{C_{TOT INV}} = \frac{30}{60}$

Por lo que retomando, la Ec. (2.21) y sustituyendo los coeficientes de acoplamiento enlistados en la tabla 2.6, tendremos que el potencial de la compuerta Flotante de la etapa de inversores programables, estará dado por:

$$\Phi_{FG INV} = \frac{1}{60}X_1 + \frac{2}{60}X_2 + \frac{4}{60}X_3 + \frac{8}{60}X_4 + \frac{15}{60}V_{INVA} \quad \dots(2.42)$$

Considerando la Ec. (2.33), la Ec. (2.42) puede reescribirse como:

$$\frac{1}{2}V_{DD} = \frac{15}{60}X_{1A} + \frac{30}{60}V_{INVA} = \frac{1}{4}X_{1A} + \frac{1}{2}V_{INVA} \quad \dots(2.43)$$

Despejando V_{INVA} de la Ec. (2.43), tendremos que:

$$V_{INVA} = V_{DD} - \frac{1}{2}X_{1A} \quad \dots(2.44)$$

Por lo que reemplazando, como se hizo anteriormente, el factor X_{1A} de la Ec. (2.44) por los voltajes de la tabla 2.6 y resolviendo para V_{INVA} , obtendremos nuevamente los voltajes externos que aplicados a los inversores programables producirán la Función X-OR, los cuales son presentados en la tabla 2.7.

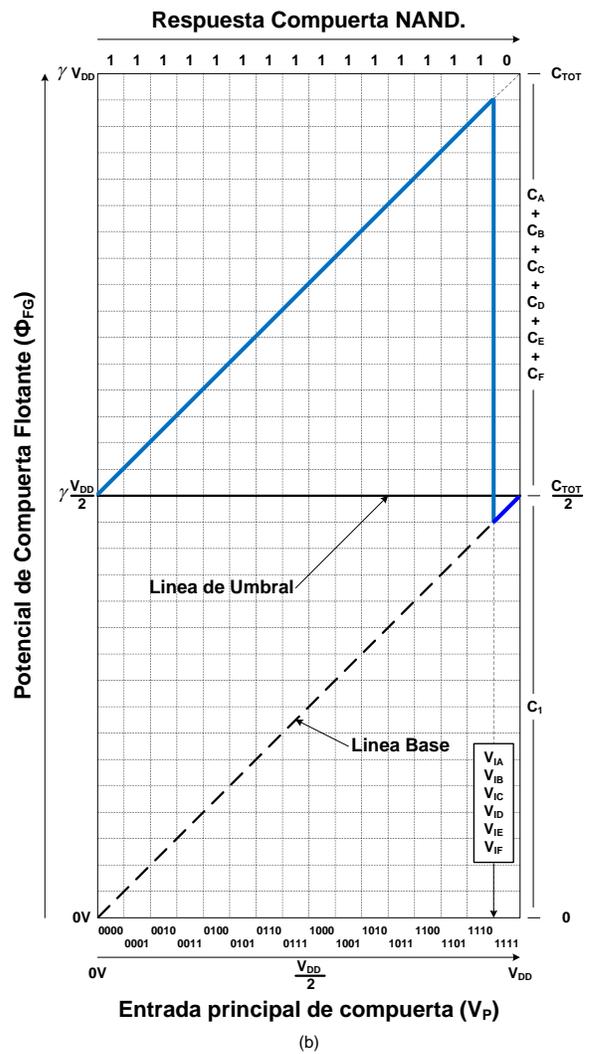
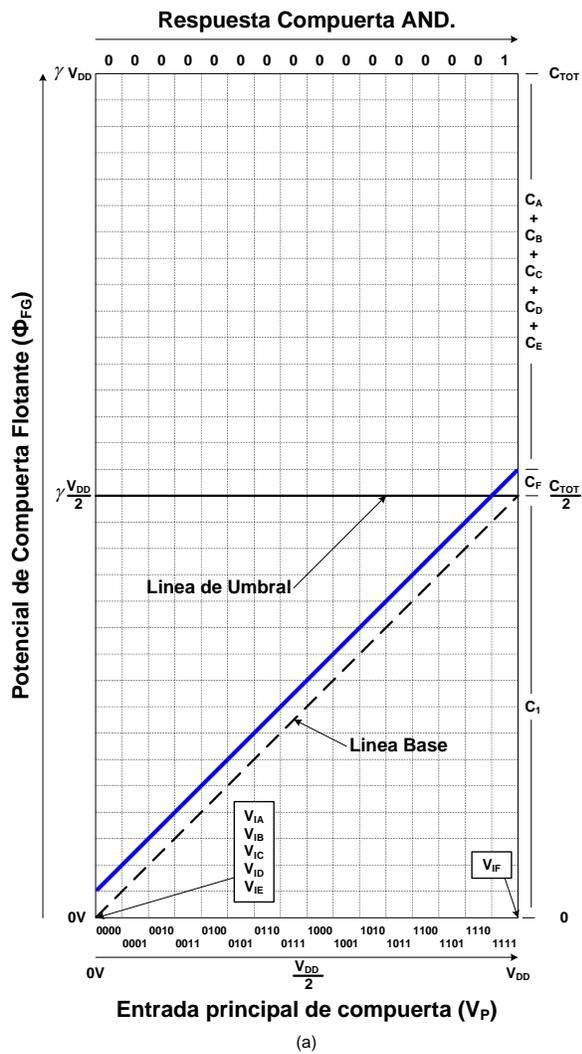
Consecuentemente, se puede observar una significativa reducción en los voltajes de los inversores programables D, E y F, los cuales ya se encuentran dentro del rango de 0 a V_{DD} .

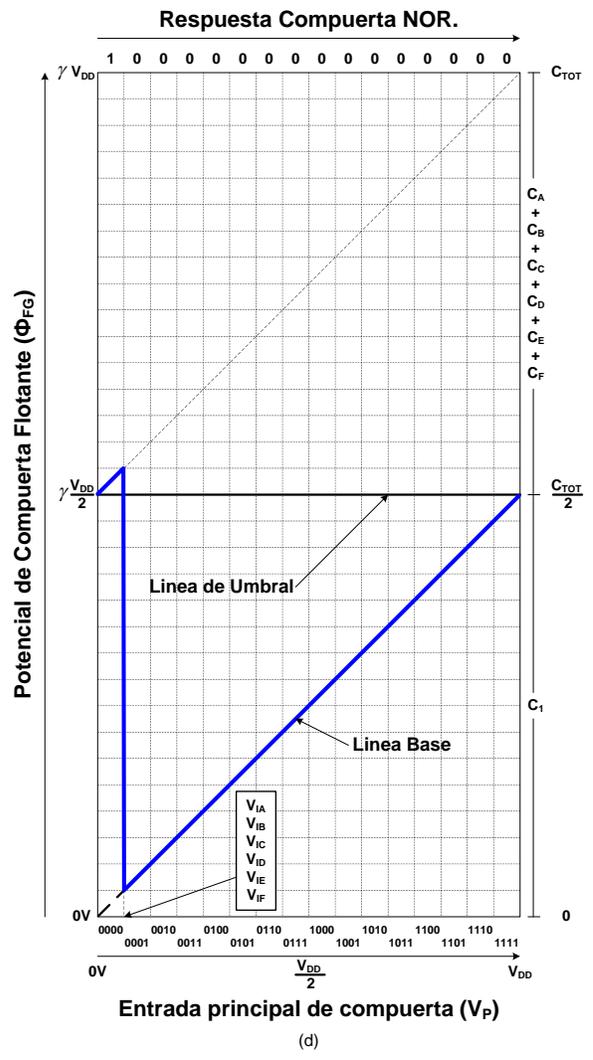
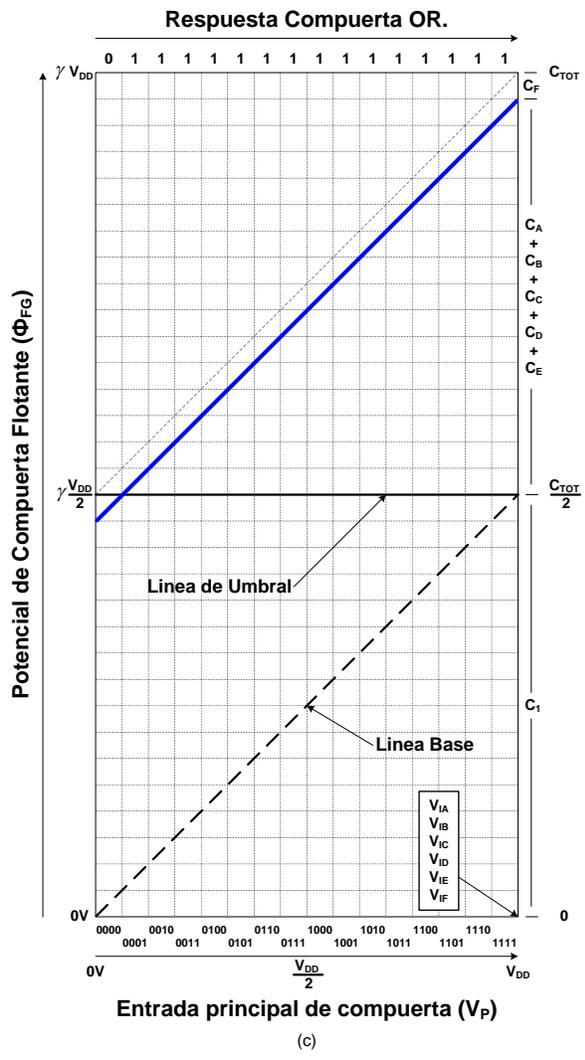
Diseño de una ALU empleando Lógica de Configuración Externa.

Tabla 2.7. Voltajes Externos aplicados de los inversores programables de la celda básica SHL para producir la Función X-OR, cuando el criterio de diseño es $C_0 = C_{X1A}$ y $C_{INVA} = 2 C_{X1A}$.

$V_A = 2.65 \text{ V}$	$V_B = 3.12 \text{ V}$	$V_C = 3.59 \text{ V}$
$V_D = 4.21 \text{ V}$	$V_E = 4.53 \text{ V}$	$V_F = 5 \text{ V}$

A continuación, las figuras 2.6(a), (b), (c), (d), (e) y (f), muestran los FPDs de las seis funciones Lógicas Básicas que puede realizar la celda básica SHL respectivamente.





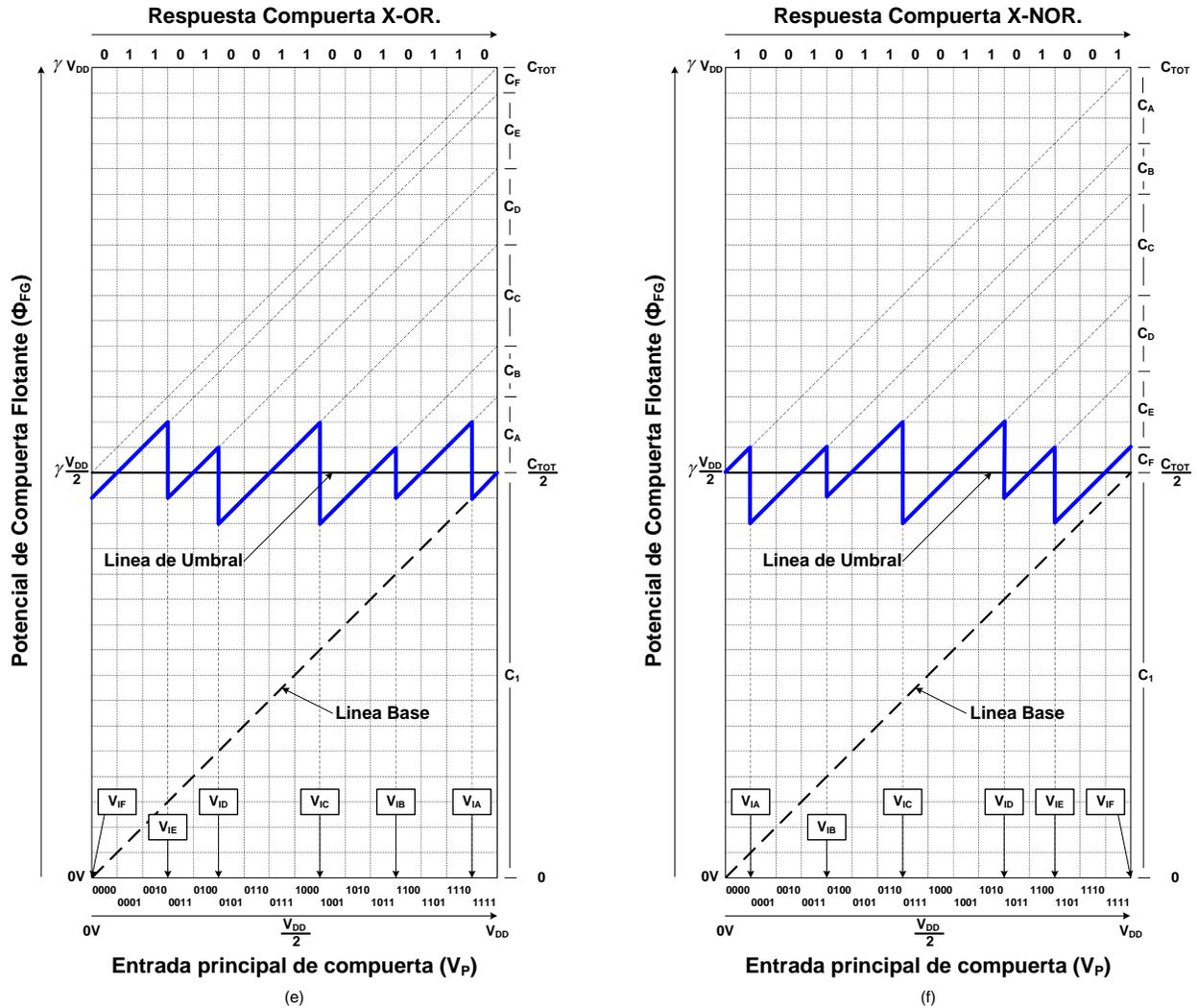


Figura 2.6. FPDs de las seis Compuertas Lógicas Básicas: (a) AND, (b) NAND, (c) OR, (d) NOR, (e) X-OR y (f) X-NOR, que pueden realizarse mediante la celda básica SHL de 4 bits.

Aplicando a cada una de las compuertas lógicas restantes, el mismo procedimiento que se empleó para la implementación de la función X-OR y basados en su respectivo FPD, se obtienen los coeficientes de acoplamiento para cada una de las etapas de la celda básica SHL de 4 bits, así como los voltajes de inversión y los voltajes externos aplicados a la etapa de inversores programables para la implementación de la Función Lógica AND, NAND, OR, NOR y XNOR respectivamente.

Por lo anteriormente expuesto y con base a la ventaja que representa la implementación de funciones lógicas mediante configuración externa, una aplicación natural que se puede dar a este tipo de circuitos es para sistemas reconfigurables en tiempo real, ya que dependiendo de la función que se quiera

implementar, se pueden usar las celdas básicas que se explicaron en este capítulo para diseñar sistemas más complejos. Es así que por esta propiedad, se presenta en este trabajo de tesis el diseño de una Unidad Aritmética Lógica (ALU) empleando la metodología que se acaba de explicar.

2.7. Conclusiones.

En este capítulo se ha demostrado que mediante el circuito de configuración externa de 4 bits propuesto e ilustrado en la figura 2.1, se pueden implementar las seis Funciones Lógicas Básicas AND, NAND, OR, NOR, X-OR y X-NOR, al modificar simplemente los valores de voltaje externos aplicados a la etapa de inversores programables de la celda.

No obstante, se pueden implementar mas funciones mediante este circuito SHL, con la técnica aquí expuesta, como se verá en el siguiente capítulo.

La ventaja que ofrece esta configuración como ya se mencionó anteriormente, es que mediante un mismo circuito se pueden implementar diversas funciones a diferencia de la Lógica convencional que emplea una configuración diferente para cada compuerta.

En resumen, siguiendo la metodología expuesta en este capítulo, a partir del correspondiente FPD se determinaron los coeficientes de acoplamiento de la etapa de precarga, inversores programables y neurona de la celda SHL, los cuales se resumen en la tabla 2.8.

Tabla 2.8. Coeficientes de acoplamiento de cada una de las etapas de la celda básica SHL para la implementación de las seis compuertas básicas AND, NAND, OR, NOR, X-OR y X-NOR.

Etapa de la Celda SHL.	Coeficientes de Acoplamiento.					
Precarga.	$K_{X1P} = \frac{1}{32}$	$K_{X2P} = \frac{2}{32}$	$K_{X3P} = \frac{4}{32}$	$K_{X4P} = \frac{8}{32}$	$K_{CTL} = \frac{15}{32}$	$K_F = \frac{1}{32}$
Inversores Programables.	$K_{X1A} = \frac{1}{60}$	$K_{X2A} = \frac{2}{60}$	$K_{X3A} = \frac{4}{60}$	$K_{X4A} = \frac{8}{60}$	$K_{INVA} = \frac{30}{60}$	
Neurona.	$K_{X1N} = \frac{1}{32}$		$K_{X2N} = \frac{2}{32}$		$K_{X3N} = \frac{4}{32}$	
	$K_A = \frac{3}{32}$	$K_B = \frac{2}{32}$	$K_C = \frac{4}{32}$	$K_D = \frac{3}{32}$	$K_E = \frac{3}{32}$	

Así mismo se determinaron los voltajes de inversión de la etapa de inversores programables de la celda básica SHL, para la implementación de cada una de las seis funciones lógicas básicas, los cuales se muestran en la tabla 2.9.

Diseño de una ALU empleando Lógica de Configuración Externa.

Tabla 2.9. Voltajes de Inversión de la etapa de inversores programables de la celda básica SHL para la implementación de las seis compuertas básicas AND, NAND, OR, NOR, X-OR y X-NOR.

Compuerta Lógica.	Voltajes de Inversión de los inversores programables.		
AND	$V_{IA} = V_{IB} = V_{IC} = V_{ID} = V_{IE} = 0 V_{DD}$		$V_{IF} = V_{DD}$
NAND	$V_{IA} = V_{IB} = V_{IC} = V_{ID} = V_{IE} = V_{IF} = \frac{15}{16} V_{DD}$		
OR	$V_{IA} = V_{IB} = V_{IC} = V_{ID} = V_{IE} = V_{IF} = V_{DD}$		
NOR	$V_{IA} = V_{IB} = V_{IC} = V_{ID} = V_{IE} = V_{IF} = \frac{1}{16} V_{DD}$		
X-OR	$V_{IA} = \frac{15}{16} V_{DD}$	$V_{IB} = \frac{12}{16} V_{DD}$	$V_{IC} = \frac{9}{16} V_{DD}$
	$V_{ID} = \frac{5}{16} V_{DD}$	$V_{IE} = \frac{3}{16} V_{DD}$	$V_{IF} = 0 V_{DD}$
X-NOR	$V_{IA} = \frac{1}{16} V_{DD}$	$V_{IB} = \frac{4}{16} V_{DD}$	$V_{IC} = \frac{7}{16} V_{DD}$
	$V_{ID} = \frac{11}{16} V_{DD}$	$V_{IE} = \frac{13}{16} V_{DD}$	$V_{IF} = V_{DD}$

Por último, se determinaron los voltajes externos aplicados a la etapa de inversores programables para implementar las Función Lógica deseada, a partir del potencial de compuerta flotante de la etapa de inversores ($\Phi_{FG INV}$) programables, Ec. (2.44), los cuales se muestran en la tabla 2.10.

Tabla 2.10. Voltajes de Inversión de la etapa de inversores programables de la celda básica SHL para la implementación de las seis compuertas básicas AND, NAND, OR, NOR, X-OR y X-NOR.

Voltaje Externo.	Compuerta Lógica Básica.					
	AND	NAND	OR	NOR	X-OR	X-NOR
V_A	5 V	2.65 V	2.5 V	4.84 V	2.65 V	4.84 V
V_B					3.12 V	4.37 V
V_C					3.59 V	3.90 V
V_D					4.21 V	3.28 V
V_E					4.53 V	2.96 V
V_F	2.5 V				5 V	2.5 V
Vctl.						

2.8. Referencias.

- [1] T. Shibata, T. Ohmi. "Neuron MOS Binary-Logic Integrated Circuits. Part I: Design Fundamentals and Soft-Hardware-Logic Circuit Implementation". IEEE Transactions on Electron Devices. Vol. 40. p. 570 – 576. 1993
- [2]. T. Shibata, T. Ohmi. "Neuron MOS Binary-Logic Integrated Circuits. Part II: Simplifying techniques of circuits configuration and their practical applications". IEEE Transactions on Electron Devices. Vol. 40. p. 570 – 576. 1993.
- [3]. M. A. Reyes-Barranca, A. Medina-Santiago " Methodology for the design of a 4-bit soft-hardware-logic circuit based on neuron MOS transistors". International Journal of Electronics. 95:6, p. 517-530. 2008.
- [4]. M. A. Reyes-Barranca, J. A. Moreno-Cardenas. "Simple method for the determination of the coupling coefficient of floating-gate MOSFETs programmed with Fowler-Nordheim tunneling". IEE Proceedings on Circuits Devices and Systems. Vol. 146. No. 4. p. 215-217. 1999.
- [5] J. J. Ramirez-Angulo, G. Gonzalez-Altamirano, S. C. Choi. "Modeling Multiple-input floating-gate transistors for analog signal processing". IEEE International Symposium on Circuits and Systems. p. 2020 – 2023. 1997.
- [6]

Capítulo III. Diseño de la unidad Aritmética Lógica de 4 bits.

3.1 Introducción.

Este capítulo describe la metodología empleada para el diseño de una ALU de 4 Bits capaz de realizar dos funciones aritméticas y dos funciones Lógicas, las cuales se pueden seleccionar mediante la palabra de control S de 4 bits, S[0-3], entregando como respuesta una función de salida F de 4 bits, F[0-3] y una salida de acarreo generado de 1 bit, G, la cual solo tiene sentido cuando se realiza la operación aritmética de Suma.

Las funciones que puede desarrollar la ALU propuesta fueron seleccionadas aleatoriamente de la tabla de verdad de la ALU comercial de 4 bits CMOS de Motorola: MC14581B (Anexo A).

La ALU propuesta está formada por un bloque principal que implementa la función de Sumador Completo, el cual está dividido en dos sub-bloques, cuyo objetivo es el de realizar la operación aritmética XOR y la Generación de acarreo (f_{CARRY}) respectivamente y además, por dos bloques secundarios, denotados como F1 y F2, los cuales realizan las operaciones lógicas y la selección/complemento de las señales de entrada A o B respectivamente, los cuales como ya se mencionó son controlados por una palabra de control de 4 bits S[0-3].

Como se verá a continuación, la metodología de diseño de la celda básica SHL empleada en la implementación de las compuertas lógicas básicas, también es aplicable al diseño de los bloques de generación de acarreo y los bloques secundarios F1 y F2, dándonos así un mayor grado de libertad en la implementación de cualquier función de 4 bits mediante el uso del FPD y la celda básica SHL.

3.2 Metodología de Diseño de la ALU de 4 bits.

El diagrama a bloques del circuito propuesto para la implementación de una ALU de 4 bits, se muestra en la figura 3.1, cuyos bloques están basados en la celda básica SHL presentada en la figura 2.1 del capítulo anterior.

El bloque principal de la ALU, está compuesto por un sumador completo (FA) con propagación de arrastre, el cual se muestra en la figura 3.2, constituido básicamente de 2 sub-bloques, uno que realiza la función de Sumador Completo (Compuerta XOR) y otro que realiza la función de Generación de acarreo (f_{CARRY}) para tres señales entradas de 4 bits, correspondientes a las palabras A y B y el acarreo de entrada (C_{in} , C_0 , C_1 o C_2), dependiendo del bloque de entrada de señal de 4 bits que se esté evaluando.

El bloque secundario F1, se emplea para implementar las operaciones lógicas que puede desempeñar la ALU, mientras que el bloque secundario F2, se emplea para implementar un multiplexor/complemento de las señales de entrada al bloque principal, las cuales son gobernadas mediante la combinación de la señal de control de 4 bits, S[0-3], como se mostrará más adelante.

Diseño de una ALU empleando Lógica de Configuración Externa.

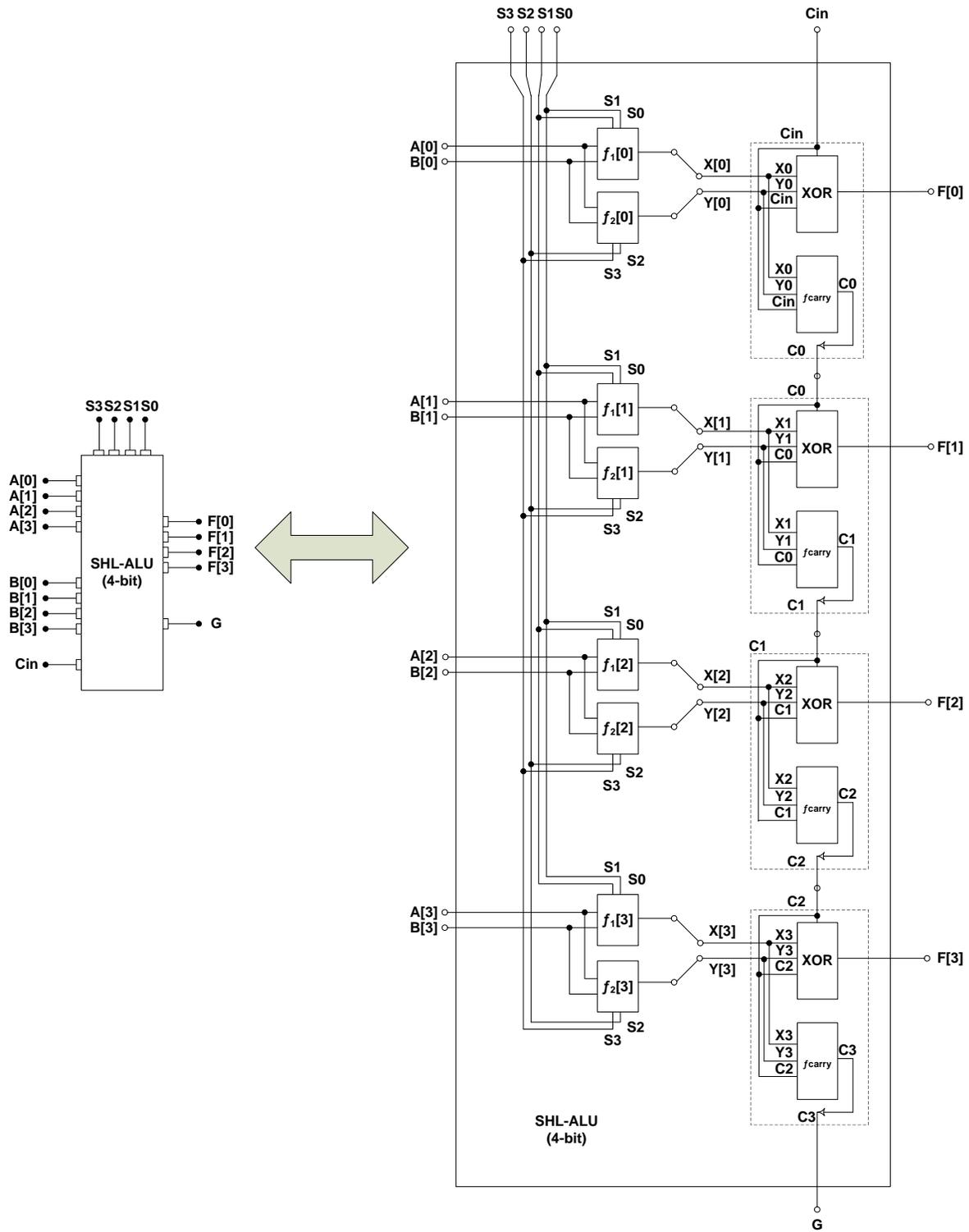


Figura 3.1. Circuito propuesto para la implementación de una ALU de 4 bits mediante la configuración de una celda básica SHL.

Diseño de una ALU empleando Lógica de Configuración Externa.

Como ya se mencionó anteriormente, la ALU está configurada para realizar dos operaciones lógicas y dos operaciones aritméticas de acuerdo con la tabla 3.1, la cual presenta las cuatro funciones seleccionadas arbitrariamente de la tabla de verdad de la ALU CMOS comercial de 4 bits MC14581B de Motorola (ver anexo A).

Tabla 3.1. Operaciones Aritméticas y Lógicas que puede realizar la ALU propuesta para demostrar la factibilidad del diseño presentado.

Lógicas:	$A \text{ and } B$ $A \text{ xnor } B$
Aritméticas:	$A + B$ $A \bullet B - 1$

Aunque los bloques se configuraron para cumplir con una función en específico, esto no está limitado ya que estas operaciones se pueden ampliar, haciendo uso de la metodología que a continuación se presenta.

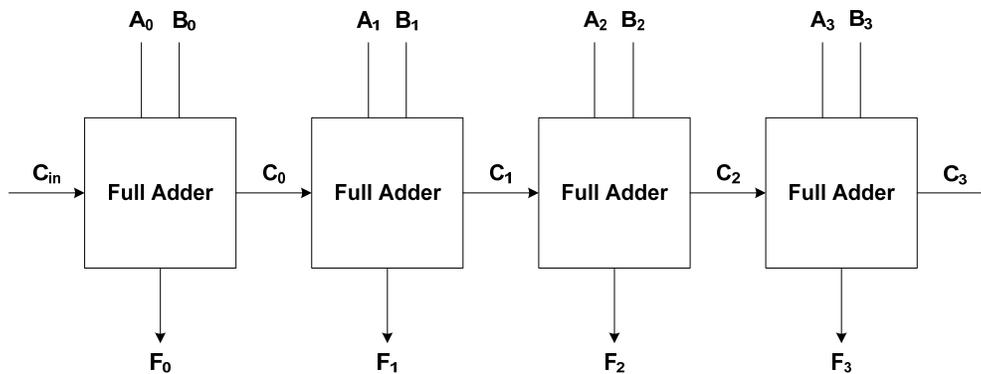


Figura 3.2 Diagrama a bloques de un sumador completo con propagación de arrastre para 4 bits, empleado en el diseño de la ALU.

Las señales de salida de la ALU propuesta, corresponden a la salida de los sub-bloques del bloque principal: La señal del sumador Completo (XOR) de 4 bits y la señal de acarreo (f_{CARRY}) de 1 bit, las cuales se rigen por las siguientes ecuaciones:

$$F_i (\text{sum}) = A_i \oplus B_i \oplus C_{i-1} \quad \dots(3.1)$$

$$C_i (\text{carry}) = A_i \bullet B_i + (A_i \oplus B_i) \bullet C_{i-1} \quad \dots(3.2)$$

Donde i representa cada una de las etapas de arrastre.

Diseño de una ALU empleando Lógica de Configuración Externa.

A partir de las Ecs. (3.1) y (3.2), se generó la tabla de verdad del Sumador Completo con propagación de arrastre, la cual se muestra en la tabla 3.2.

En este capítulo, se omitirá el diseño del sub-bloque de Suma del bloque principal de la ALU, debido a que la implementación de la función XOR (columna 'F' de la tabla 3.2) ya se trató en el capítulo anterior.

Por lo tanto los coeficientes de acoplamiento, los voltajes de inversión y voltajes externos aplicados a los inversores programables para la implementación de esta función (X-OR) pueden consultarse en las tablas 2.8, 2.9 y 2.10 respectivamente.

Tabla 3.2. Tabla de Verdad del Sumador Completo de 3 bits.

Entradas.			Salidas.	
X	Y	C _{in}	Suma (F)	Acarreo (C)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Así mismo, los coeficientes de acoplamiento de las etapas de precarga e inversores programables de los sub-bloques (X-OR y f_{CARRY}) así como de los bloques secundarios F1 y F2 del bloque principal de la ALU, serán los mismos que se calcularon en el capítulo 2, debido a que dichas etapas no dependen de la función a implementar. Por lo que los coeficientes de estas dos etapas se muestran en la tabla 3.3.

Tabla 3.3. Coeficientes de acoplamiento para las etapas de precarga e inversores programables de los sub-bloques del bloque principal y los dos bloques secundarios de la ALU.

Etapa de la Celda SHL.	Coeficientes de Acoplamiento para el bloque principal y los bloques secundarios de la ALU.					
Precarga.	$K_{X1P} = \frac{1}{32}$	$K_{X2P} = \frac{2}{32}$	$K_{X3P} = \frac{4}{32}$	$K_{X4P} = \frac{8}{32}$	$K_{CTL} = \frac{15}{32}$	$K_F = \frac{1}{32}$
Inversores Programables.	$K_{X1A} = \frac{1}{60}$	$K_{X2A} = \frac{2}{60}$	$K_{X3A} = \frac{4}{60}$	$K_{X4A} = \frac{8}{60}$	$K_{INVA} = \frac{30}{60}$	

A continuación se muestra el diseño de la etapa de inversores programables del sub-bloque que implementa la función de generación de acarreo (f_{CARRY}) y de los bloques secundarios F1 y F2 del bloque principal de la ALU, respectivamente.

3.3. Diseño del sub Bloque de Generación de Acarreo (f_{CARRY}).

El FPD que describe el comportamiento del bloque de la función de generación de acarreo (f_{CARRY}) se muestra en la figura 3.3, el cual corresponde a la columna de salida de acarreo (C) de tabla de verdad del sumador completo de la tabla 3.2.

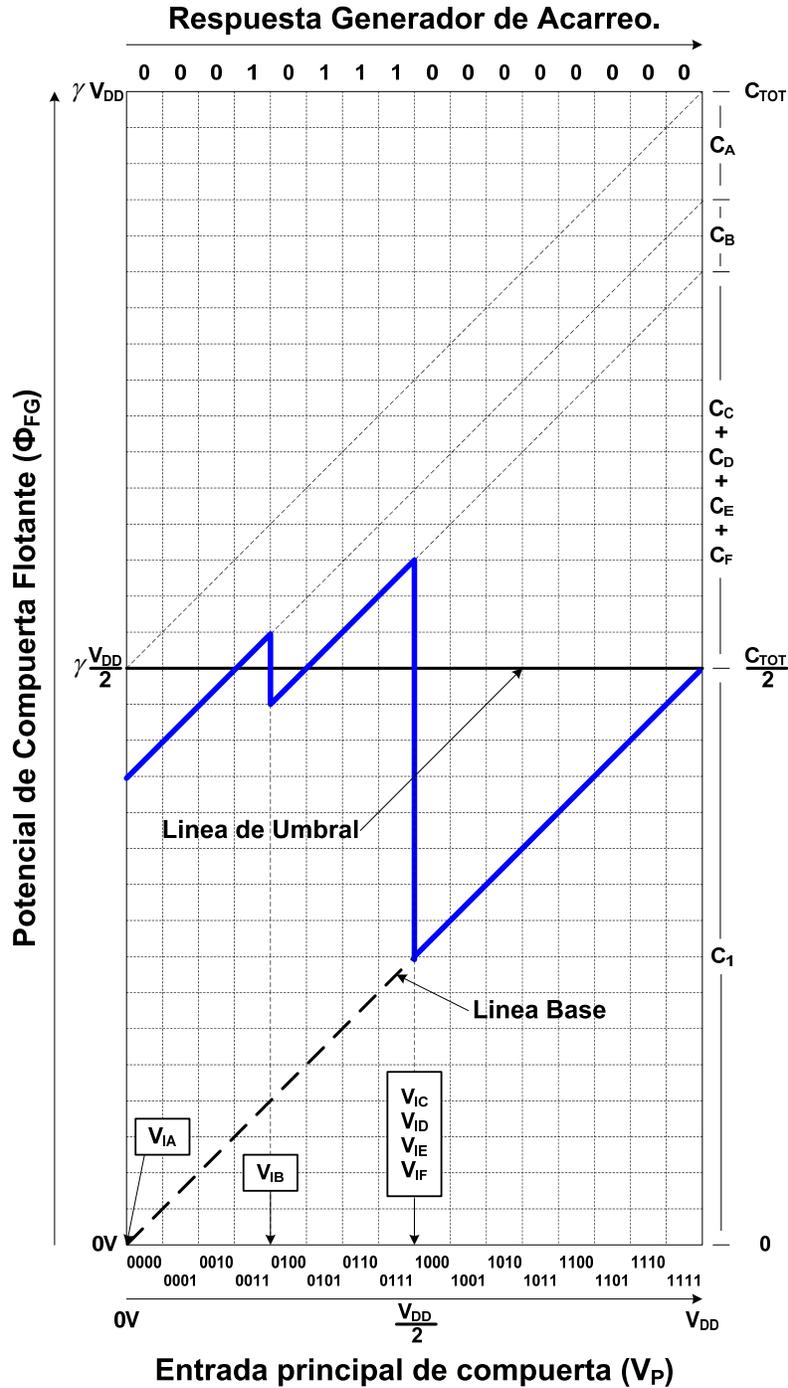


Figura 3.3. FPD de la Función de Generación de Acarreo (f_{CARRY}).

Diseño de una ALU empleando Lógica de Configuración Externa.

A partir del FPD, se tiene que los coeficientes de acoplamiento de la salida de los inversores programables hacia la etapa neurona MOS, deben cumplir con las siguientes proporciones:

$$\frac{C_A}{C_{TOT NEU}} = \frac{3}{32} \qquad \frac{C_B}{C_{TOT NEU}} = \frac{2}{32} \qquad \dots(3.3)$$

Donde:

$$\frac{C_C}{C_{TOT NEU}} + \frac{C_D}{C_{TOT NEU}} + \frac{C_E}{C_{TOT NEU}} + \frac{C_F}{C_{TOT NEU}} = \frac{11}{32} \qquad \dots(3.4)$$

Lo cual se satisface si:

$$\frac{C_C}{C_{TOT NEU}} = \frac{4}{32} \qquad \frac{C_D}{C_{TOT NEU}} = \frac{3}{32} \qquad \frac{C_E}{C_{TOT NEU}} = \frac{3}{32} \qquad \frac{C_F}{C_{TOT NEU}} = \frac{1}{32}$$

Lo cual significa que los coeficientes de acoplamiento presentados para la implementación de las seis funciones básicas en el capítulo 2, incluyendo los coeficientes de acoplamiento de entrada C_{X1N} , C_{X2N} , C_{X3N} y C_{X4N} a la Compuerta Flotante del circuito neurona, también son válidos para la implementación de la función f_{CARRY} del segundo sub-bloque del bloque principal FA.

La tabla 3.4, muestra los coeficientes de acoplamiento de la etapa de neurona de la celda básica SHL de 4 bits para la implementación de la función f_{CARRY} del bloque principal de la ALU.

Tabla 3.4. Coeficientes de acoplamiento de la etapa de neurona de la celda básica SHL de 4 bits para implementar la función f_{CARRY} del bloque principal de la ALU.

Etapa de la Celda SHL.	Coeficientes de Acoplamiento para el sub-bloque de la función f_{CARRY} del bloque principal de la ALU.			
Neurona.	$K_{X1N} = \frac{1}{32}$	$K_{X2N} = \frac{2}{32}$	$K_{X3N} = \frac{4}{32}$	$K_{X4N} = \frac{8}{32}$
	$K_A = \frac{3}{32}$	$K_B = \frac{2}{32}$	$K_C = \frac{4}{32}$	$K_D = \frac{3}{32}$

Los voltajes de inversión de la etapa de inversores programables para la implementación de la función f_{CARRY} del bloque principal de la ALU, extraídos del FPD correspondiente se presentan en la tabla 3.5.

Tabla 3.5. Voltajes de inversión de la etapa de neurona de la celda básica SHL de 4 bits para implementar la función f_{CARRY} del bloque principal de la ALU.

Función:	Voltajes de Inversión para la etapa de inversores programables.		
f_{CARRY}	$V_{IA} = 0 V_{DD}$	$V_{IB} = \frac{4}{16} V_{DD}$	$V_{IC} = V_{ID} = V_{IE} = V_{IF} = \frac{8}{16} V_{DD}$

Haciendo uso de la Ec. (2.36) y reemplazando el factor X_{1A} por los voltajes de inversión de la tabla 3.5 y resolviendo para V_{INVA} , encontraremos los voltajes externos que aplicados a los inversores programables producirán la función del bloque principal, f_{CARRY} , los cuales se muestran en la tabla 3.6.

Tabla 3.6. Voltajes Externos aplicados a los inversores programables de la celda básica SHL para producir la función f_{CARRY} del bloque principal.

Voltajes Externos para la implementación de la Función f_{CARRY} del bloque principal de la ALU.		
$V_A = 5 \text{ V}$	$V_B = 4.37 \text{ V}$	$V_C = 3.75 \text{ V}$
$V_D = 3.75 \text{ V}$	$V_E = 3.75 \text{ V}$	$V_F = 3.75 \text{ V}$

3.4. Diseño del Bloque de Función F1.

El bloque de Función F1 nos permite decidir, mediante los bits S[0] y S[1] de la señal de control 'S', entre implementar las operaciones Lógicas que puede realizar la ALU ($A \bullet B$ o $\overline{A \oplus B}$) o seleccionar entre las señales A o B para realizar la suma del bloque principal, de acuerdo con la tabla 3.7.

Nuevamente considerando que los coeficientes de acoplamiento de las etapas de precarga e inversores programables, no son dependientes de la función a implementar, estos tendrán el mismo valor que ya se había establecido en la tabla 3.3, y debido a que las consideraciones para el cálculo de los coeficientes de acoplamiento de las señales de entrada con la compuerta flotante del inversor CMOS de la etapa de neurona C_{X1N} , C_{X2N} , C_{X3N} , C_{X4N} del bloque F1 y la celda básica SHL no cambian, estos serán los mismos que se calcularon en el capítulo 2. Por lo tanto, a partir del FPD correspondiente al bloque de función F1, mostrado en la figura 3.4, se tiene que los coeficientes de acoplamiento de la salida de los inversores programables hacia la etapa neurona MOS, K_A , K_B , ..., K_F serán:

$$\begin{array}{ccc} \frac{C_A}{C_{TOT\ NEU}} = \frac{3}{32} & \frac{C_B}{C_{TOT\ NEU}} = \frac{2}{32} & \frac{C_C}{C_{TOT\ NEU}} = \frac{2}{32} \\ \frac{C_D}{C_{TOT\ NEU}} = \frac{3}{32} & \frac{C_E}{C_{TOT\ NEU}} = \frac{5}{32} & \frac{C_F}{C_{TOT\ NEU}} = \frac{1}{32} \end{array}$$

Al comparar los coeficientes de acoplamiento K_A , K_B , K_C , K_D , K_E y K_F de la etapa de neurona de la celda básica SHL y del bloque de función F1, observamos que K_C y K_E son diferentes, por lo que habrá la necesidad de ajustarlos para poder implementar la función del bloque F1 mediante la celda básica SHL. Los coeficientes de acoplamiento para la implementación de la función del bloque F1, se muestran en la tabla 3.8.

Tabla 3.7. Tabla de Verdad del bloque secundario F1.

Bits de Selección		Entradas		Salida.	Función implementada
S[1]	S[0]	B	A	X	
0	0	0	0	0	A • B
0	0	0	1	0	
0	0	1	0	0	
0	0	1	1	1	
0	1	0	0	0	A
0	1	0	1	1	
0	1	1	0	0	
0	1	1	1	1	
1	0	0	0	0	B
1	0	0	1	0	
1	0	1	0	1	
1	0	1	1	1	
1	1	0	0	1	$\overline{A \oplus B}$
1	1	0	1	0	
1	1	1	0	0	
1	1	1	1	1	

Tabla 3.8. Coeficientes de acoplamiento de la etapa de neurona de la celda básica SHL de 4 bits para implementar la función del bloque secundario F1 de la ALU.

Etapa de la Celda SHL.	Coeficientes de Acoplamiento para el sub bloque de la función f_{CARRY} del bloque principal de la ALU.				
Neurona.	$K_{X1N} = \frac{1}{32}$	$K_{X2N} = \frac{2}{32}$	$K_{X3N} = \frac{4}{32}$	$K_{X4N} = \frac{8}{32}$	
	$K_A = \frac{3}{32}$	$K_B = \frac{2}{32}$	$K_C = \frac{2}{32}$	$K_D = \frac{3}{32}$	$K_E = \frac{5}{32}$

Los voltajes de inversión para la etapa de inversores programables para la implementación de la función del bloque secundario F1 de la ALU, extraídos del FPD correspondiente se presentan en la tabla 3.9.

Tabla 3.9. Voltajes de inversión de la etapa de neurona de la celda básica SHL de 4 bits para implementar la función del bloque secundario F1 de la ALU.

Voltajes de Inversión del Bloque F1.		
$V_{IA} = 0 V_{DD}$	$V_{IB} = \frac{4}{16} V_{DD}$	$V_{IC} = \frac{6}{16} V_{DD}$
$V_{ID} = \frac{8}{16} V_{DD}$	$V_{IE} = \frac{13}{16} V_{DD}$	$V_{IF} = V_{DD}$

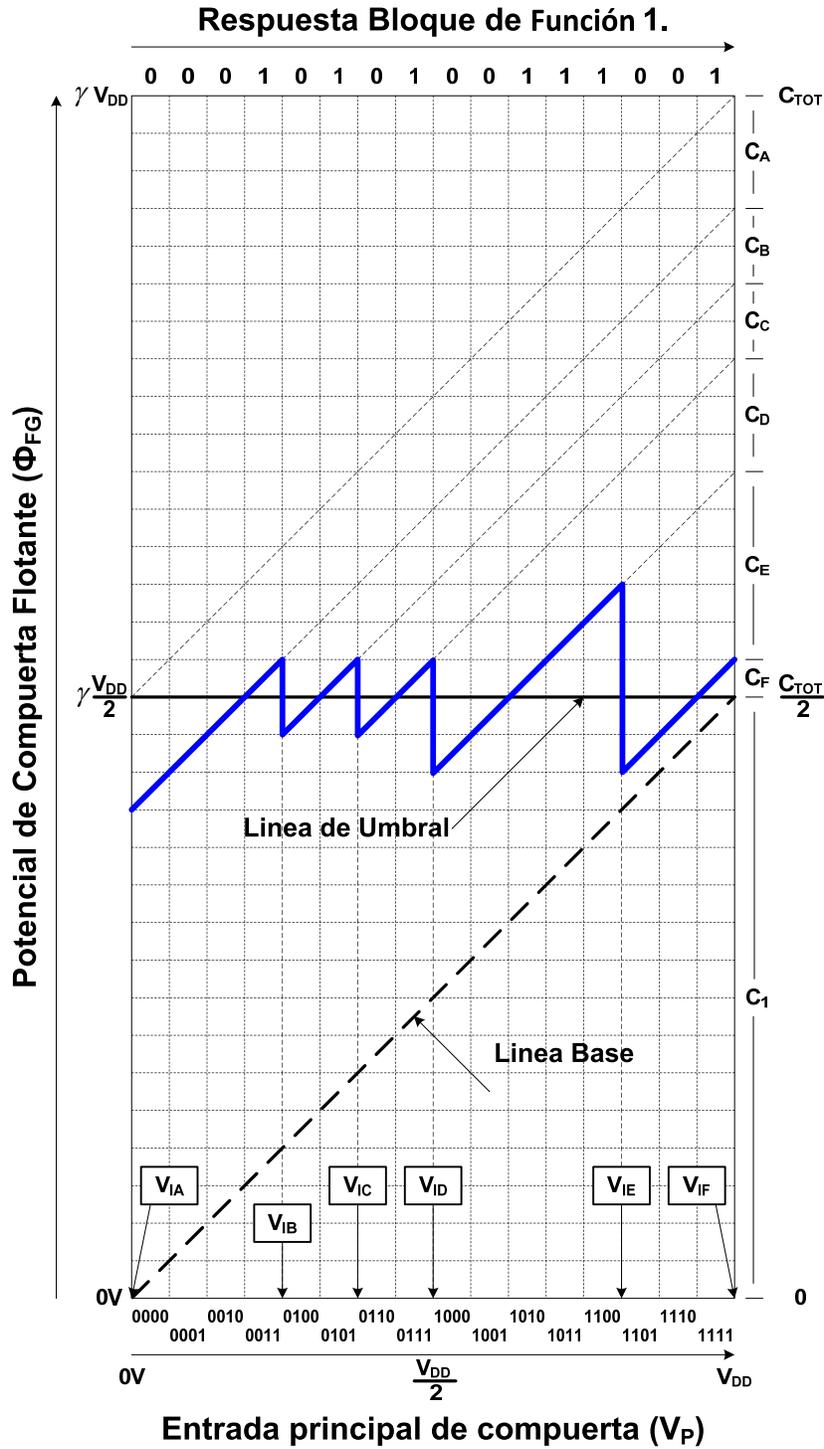


Figura 3.4. FPD del bloque secundario F1.

Por lo que realizando las sustituciones adecuadas, se encontrarán los voltajes externos aplicados a los inversores programables para implementar la función del Bloque F1, los cuales se muestran en la tabla 3.10.

Tabla 3.10. Voltajes Externos aplicados a los inversores programables de la celda básica SHL para implementar la función del bloque secundario F1.

Voltajes Externos para la implementación de la Función del bloque secundario F1 de la ALU.		
$V_A = 5\text{ V}$	$V_B = 4.37\text{ V}$	$V_C = 4.06\text{ V}$
$V_D = 3.75\text{ V}$	$V_E = 2.96\text{ V}$	$V_F = 2.5\text{ V}$

3.5. Diseño del Bloque de Función F2.

El bloque de función F2 de la ALU, nos permite decidir, mediante los bits S[2] y S[3] de la señal de control 'S', entre implementar el complemento de la operación aritmética $A \bullet B - 1$ o seleccionar entre las señales A o B para realizar la suma del bloque principal, de acuerdo a la tabla 3.11.

Los coeficientes de acoplamiento de las etapas de precarga e inversores programables de la celda SHL de 4 bits que implementará la función del bloque F2, serán los que se mostraron en la tabla 3.3 y debido a que las consideraciones para el cálculo de C_{X1N} , C_{X2N} , C_{X3N} , C_{X4N} de de la etapa de neurona del bloque F2 y la celda básica SHL no cambian, estos serán los mismos que se calcularon en el capítulo 2.

Tabla 3.11. Tabla de Verdad del bloque secundario F2.

Bits de Selección		Entradas		Salida.	Función implementada
S[3]	S[2]	B	A	Y	
0	0	0	0	0	0
0	0	0	1	0	
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	A
0	1	0	1	1	
0	1	1	0	0	
0	1	1	1	1	
1	0	0	0	0	B
1	0	0	1	0	
1	0	1	0	1	
1	0	1	1	1	
1	1	0	0	1	1
1	1	0	1	1	
1	1	1	0	1	
1	1	1	1	1	

Diseño de una ALU empleando Lógica de Configuración Externa.

A partir del FPD correspondiente al bloque de función F2, mostrado en la figura 3.5, tendremos que los coeficientes de acoplamiento de la salida de los inversores programables hacia la etapa neurona MOS, K_A, K_B, \dots, K_F tienen que cumplir las siguientes proporciones:

$$\begin{aligned} \frac{C_A}{C_{TOT NEU}} + \frac{C_B}{C_{TOT NEU}} &= \frac{5}{32} & \frac{C_D}{C_{TOT NEU}} &= \frac{3}{32} \\ \frac{C_C}{C_{TOT NEU}} &= \frac{2}{32} & \frac{C_E}{C_{TOT NEU}} + \frac{C_F}{C_{TOT NEU}} &= \frac{6}{32} \end{aligned}$$

Lo cual se satisface si:

$$\frac{C_A}{C_{TOT NEU}} = \frac{3}{32} \quad \frac{C_B}{C_{TOT NEU}} = \frac{2}{32} \quad \frac{C_E}{C_{TOT NEU}} = \frac{5}{32} \quad \frac{C_F}{C_{TOT NEU}} = \frac{1}{32}$$

Esto implica que los coeficientes que se calcularon para la implementación del bloque F1, también son válidos para el bloque F2. Por lo que los coeficientes de acoplamiento de la etapa de neurona del bloque F2 se muestran en la tabla 3.12.

Tabla 3.12. Coeficientes de acoplamiento de la etapa de neurona de la celda básica SHL de 4 bits para implementar la función del bloque secundario F2 de la ALU.

Etapa de la Celda SHL.	Coeficientes de Acoplamiento del F2 de la ALU.				
Neurona.	$K_{X1N} = \frac{1}{32}$	$K_{X2N} = \frac{2}{32}$	$K_{X3N} = \frac{4}{32}$	$K_{X4N} = \frac{8}{32}$	
	$K_A = \frac{3}{32}$	$K_B = \frac{2}{32}$	$K_C = \frac{2}{32}$	$K_D = \frac{3}{32}$	$K_E = \frac{5}{32}$

Los voltajes de inversión para la etapa de inversores programables para la implementación de la función del bloque secundario F2 de la ALU, extraídos del FPD correspondiente se presentan en la tabla 3.13.

Tabla 3.13. Voltajes de inversión de la etapa de neurona de la celda básica SHL de 4 bits para implementar la función del bloque secundario F2 de la ALU.

Voltajes de Inversión del Bloque F2 de la ALU.	
$V_{IA} = V_{IB} = 0 V_{DD}$	$V_{IC} = \frac{6}{16} V_{DD}$
$V_{ID} = \frac{8}{16} V_{DD}$	$V_{IE} = V_{IF} = V_{DD}$

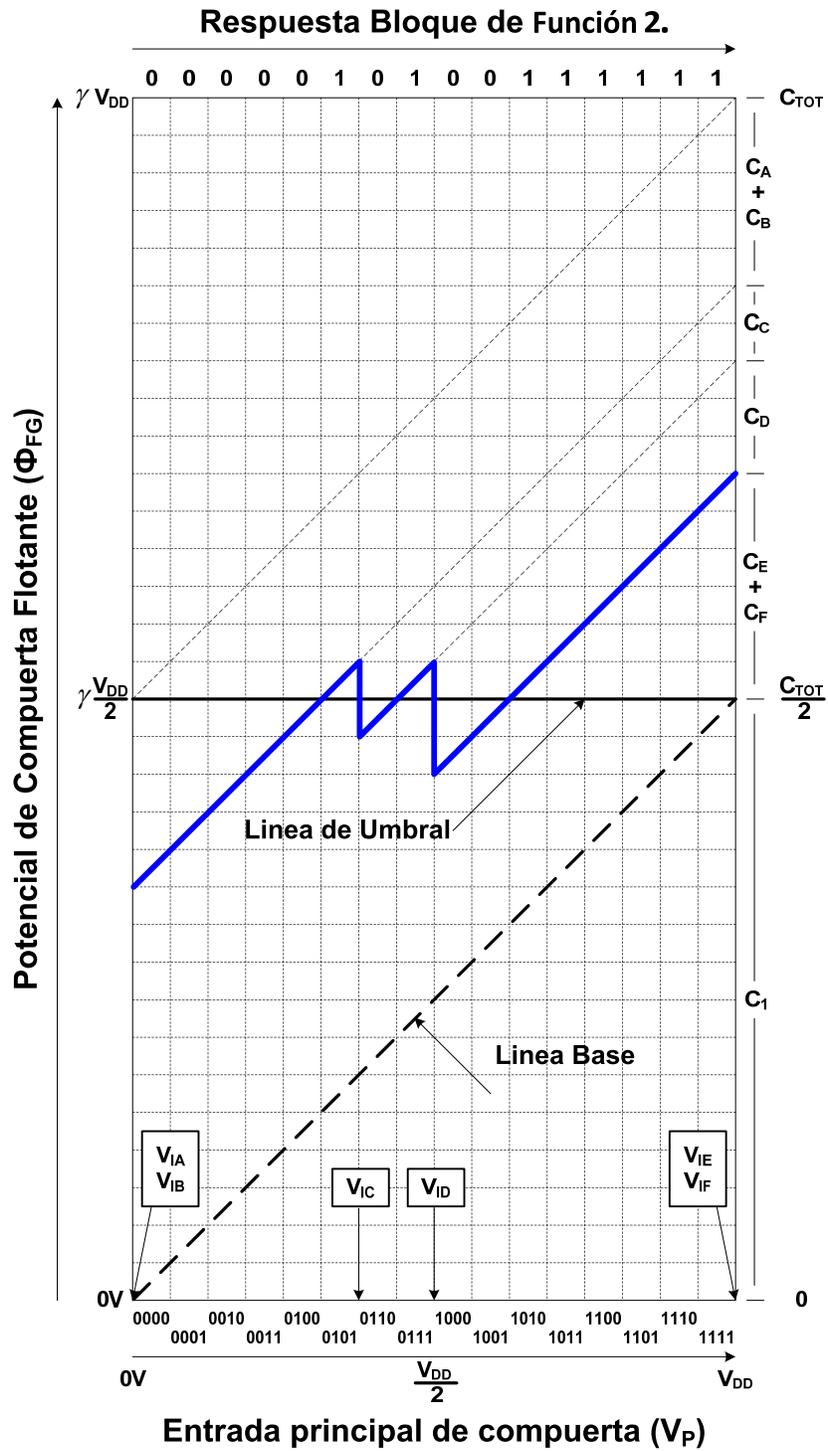


Figura 3.5. PDF del bloque secundario F2.

Diseño de una ALU empleando Lógica de Configuración Externa.

Por lo que una vez más, realizando las sustituciones pertinentes en la Ec. (2.36), se pueden obtener los voltajes externos que aplicados a los inversores programables, implementarán la función del bloque F2, los cuales se muestran en la tabla 3.14.

Tabla 3.14. Voltajes Externos aplicados a los inversores programables de la celda básica SHL para producir la función del bloque F2.

Voltajes Externos para la implementación de la Función del bloque F2 de la ALU.		
$V_A = 5\text{ V}$	$V_B = 5\text{ V}$	$V_C = 4.06\text{ V}$
$V_D = 3.75\text{ V}$	$V_E = 2.5\text{ V}$	$V_F = 2.5\text{ V}$

Por último, al principio del capítulo se mencionó que la ALU podía realizar dos funciones aritméticas y dos funciones lógicas de acuerdo a la tabla 3.1, elegidas mediante la señal de control S de 4 bits, sin embargo no se explicó el cómo se realiza dicha selección.

Para poder explicar el procedimiento de selección de operaciones, es necesario que las tablas 3.7 y 3.11, que muestran las tablas de verdad de los bloques secundarios F1 y F2, se reescriban como se muestra en las tablas 3.15 y 3.16, respectivamente.

Tabla 3.15. Tabla de Verdad simplificada del bloque de Función F1.

Entradas de Control.		Entradas de Señal.		Función implementada.
S[1]	S[0]	B	A	X
0	0	B	A	$A \bullet B$
0	1			A
1	0			B
1	1			$\overline{A \oplus B}$

Tabla 3.16. Tabla de Verdad simplificada del bloque de Función F2.

Entradas de Control.		Entradas de Señal.		Función implementada.
S[3]	S[2]	B	A	Y
0	0	B	A	0
0	1			A
1	0			B
1	1			1

Teniendo en cuenta que uno de los sub- bloques del bloque principal de la ALU, realiza la operación XOR sobre la función implementada por los bloques secundarios F1 y F2, es decir $X \oplus Y$, y de acuerdo a la tabla 3.1 y a las tablas simplificadas, la selección de cada una de las operaciones se realiza de la siguiente forma:

1. Para que la ALU pueda realizar la operación lógica $A \bullet B$, el sub-bloque F1 debe implementar la función $A \bullet B$, mientras que el sub-bloque F2 debe implementar la función '0', de tal forma que $(A \bullet B) \oplus 0 = A \bullet B$.
2. Para que la ALU pueda realizar la operación lógica $\overline{A \oplus B}$, el sub-bloque F1 debe implementar la función $\overline{A \oplus B}$, mientras que el sub-bloque F2 debe implementar la función '0', de tal forma que $(\overline{A \oplus B}) \oplus 0 = \overline{A \oplus B}$.
3. Para que la ALU pueda realizar la operación aritmética $A + B$, el sub-bloque F1 debe implementar la señal A, mientras que el sub-bloque F2 debe implementar la señal B, de tal forma que $A \oplus B = A + B$.
4. Para que la ALU pueda realizar la operación aritmética $A \bullet B - 1$, el sub-bloque F1 debe implementar la función $A \bullet B$, mientras que el sub-bloque F2 debe implementar la función '1', de tal forma que $A \bullet B \oplus 1 = A \bullet B - 1$ complementada. Lo anterior es muy útil, debido a que en las hojas de especificación del chip que se analizó, se indica que el resultado de esta operación es con complemento a 2.

La implementación del sub-bloque de acarreo (f_{CARRY}) solo tiene sentido cuando la ALU realiza la función $A + B$.

En consecuencia. La selección de operaciones de la ALU, puede resumirse en la tabla 3.17.

Tabla 3.17. Selección de las Operaciones de la ALU mediante la señal de control S de 4 bits.

Entradas de Control.				Entradas de Señal.		Función implementada por la ALU
S[3]	S[2]	S[1]	S[0]			
0	0	0	0	B	A	$A \bullet B$
0	0	1	1			$\overline{A \oplus B}$
1	0	0	1			$A + B$
1	1	0	0			$A \bullet B - 1$

Con lo anterior se concluye el diseño de todos los bloques que forman la ALU SHL de 4 bits propuesta en esta tesis.

3.6. Conclusiones.

En este capítulo, se presentó el diseño del sub-bloque de Suma (XOR), del sub-bloque de generación de acarreo (f_{CARRY}), del bloque secundario F1 y F2, los cuales constituyen los bloques para el desarrollo de la ALU SHL de 4 bits.

Además, se demostró que el concepto SHL no solo puede aplicarse a compuertas lógicas básicas, sino que tiene una viabilidad enorme en la implementación de funciones, a diferencias de los circuitos fabricados con tecnología CMOS convencional.

Una ALU SHL de 4 bits, fue diseñada empleando varias celdas básicas SHL, las cuales están basadas en el método grafico FPD.

También se demostró que empleando los criterios y consideraciones de diseño de la celda básica SHL del capítulo 2, se pudieron definir bloques para la implementación de una ALU SHL de 4 bits, la cual es capaz de realizar dos funciones aritméticas y dos funciones lógicas, mediante una señal de control 'S' de 4 bits, como se mostró en la tabla 3.17.

A continuación en la tabla 3.18, se muestra un resumen de los coeficientes de acoplamiento de las etapas de precarga, inversores programables y neurona para la implementación de cada uno de los bloques que integran la ALU de 4 bits propuesta.

Tabla 3.18. Coeficientes de Acoplamiento de las etapas de precarga, inversores programables y neurona para la implementación de cada bloque de la ALU.

Bloque de la ALU.	Coeficientes de Acoplamiento de la Etapa:			
	Precarga.	Inversores Programables.	Neurona.	
Suma (XOR).	$K_{X1P} = \frac{1}{32}$	$K_{X1A} = \frac{1}{60}$	$K_{X1N} = \frac{1}{32}$	$K_C = \frac{4}{32}$
f_{CARRY} .	$K_{X2P} = \frac{2}{32}$		$K_{X2N} = \frac{2}{32}$	
F1.	$K_{X3P} = \frac{4}{32}$	$K_{X2A} = \frac{2}{60}$	$K_{X3N} = \frac{4}{32}$	$K_C = \frac{2}{32}$
F2.	$K_{X4P} = \frac{8}{32}$	$K_{X3A} = \frac{4}{60}$	$K_{X4N} = \frac{8}{32}$	
	$K_{CTL} = \frac{15}{32}$	$K_{X4A} = \frac{8}{60}$	$K_A = \frac{3}{32}$	
	$K_F = \frac{1}{32}$	$K_{INVA} = \frac{30}{60}$	$K_B = \frac{2}{32}$	$K_E = \frac{5}{32}$
			$K_D = \frac{3}{32}$	

Los voltajes de inversión y voltajes externos aplicados a la etapa de inversores programables para la implementación de cada operación que puede desarrollar la ALU, se muestran en la tabla 3.19 y 3.20, respectivamente.

Diseño de una ALU empleando Lógica de Configuración Externa.

Tabla 3.17. Voltajes de inversión requeridos para la implementación de la función de cada uno de los bloques de la ALU.

Voltajes de Inversión para la etapa de inversores programables.							
Función del Bloque:	V_{IA}	V_{IB}	V_{IC}	V_{ID}	V_{IE}	V_{IF}	Vctl
Suma (XOR).	$\frac{15}{16}V_{DD}$	$\frac{12}{16}V_{DD}$	$\frac{9}{16}V_{DD}$	$\frac{5}{16}V_{DD}$	$\frac{3}{16}V_{DD}$	$0V_{DD}$	
f_{CARRY} .	$0V_{DD}$	$\frac{4}{16}V_{DD}$	$\frac{8}{16}V_{DD}$				
F1.	$0V_{DD}$	$\frac{4}{16}V_{DD}$	$\frac{6}{16}V_{DD}$	$\frac{8}{16}V_{DD}$	$\frac{13}{16}V_{DD}$	V_{DD}	
F2.	$0V_{DD}$	$0V_{DD}$	$\frac{6}{16}V_{DD}$	$\frac{8}{16}V_{DD}$	V_{DD}	V_{DD}	

Tabla 3.20. Voltajes Externos aplicados a la etapa de inversores programables para implementar la función de cada uno de los bloques de la ALU.

Voltajes externos aplicados a la etapa de inversores programables para la implementación de las funciones de los bloques de la ALU.							
Función del Bloque:	V_A	V_B	V_C	V_D	V_E	V_F	Vctl
Suma (XOR).	2.65 V	3.12 V	3.59 V	4.21 V	4.53 V	5 V	
f_{CARRY} .	5 V	4.37 V	3.75 V				
F1.			4.06 V	3.75 V	2.96 V	2.5 V	
F2.		5 V			2.5 V		

3.7. Referencias.

Capítulo IV. Simulaciones de la celda básica.

4.1 Introducción.

Las simulaciones presentadas en este capítulo, se obtuvieron mediante la herramienta gráfica PSpice de OrCAD Capture, incluida en el paquete de Diseño de Cadence ver. 16, la cual nos proporciona la posibilidad de declarar componentes electrónicos empleando los símbolos disponibles en las librerías del paquete, lo cual hace más ágil la declaración esquemática de componentes, en comparación con la declaración de componentes mediante Netlist.

Las simulaciones en DC (FPD) y estado transitorio que se presentan corresponden a las seis compuertas lógicas básicas (AND, NAND, OR, NOR y XNOR) que se estudian en esta tesis, además las correspondientes a los bloques de Suma (XOR), Generación de Acarreo (f_{CARRY}) y los bloques secundarios de Función F1 y F2, los cuales forman parte de la ALU de 4 bits presentada en el capítulo anterior.

4.2. Determinación del punto de Inversión (V_{SP}) y de las geometrías (W/L) de los transistores NMOS y PMOS empleados en la simulación de la celda básica SHL.

El componente en el que reside el desempeño de la celda básica SHL empleada en los bloques de Precarga, Inversores programables y Neurona de la ALU de 4 bits propuesta, es el inversor MIFG CMOS.

Por lo tanto, a continuación se definirán W/L de los transistores NMOS y PMOS que forman parte de los inversores empleados.

Los parámetros de PSpice que se ocuparon tanto en el cálculo de W/L como del V_{SP} del inversor MIFG CMOS de cada etapa, se obtuvieron del archivo de corrida V11L efectuado para una tecnología de 0.5 μm , proporcionado por la compañía OnSemi (MOSIS) correspondientes a BSIM3 ver.3.1 o nivel 8 de PSpice.

El reporte de este archivo (V11L) contiene los resultados promedio del lote obtenidos a partir de mediciones de estructuras de prueba que MOSIS realizó sobre cada oblea de este lote de fabricación (ver anexo B).

La tabla 4.1, muestra algunos de los parámetros empleados en el modelo PSpice de los transistores NMOS y PMOS del inversor MIFG CMOS de la celda básica.

Así que considerando el Inversor CMOS y la característica de transferencia [1], que se presentan en la figura 4.1, se puede definir el voltaje del punto de conmutación del inversor CMOS (V_{SP}) como el punto en la curva donde el voltaje de entrada es igual al voltaje de salida (C).

Dado que en este punto (V_{SP}), los dos transistores del inversor trabajan en la región de saturación y que la corriente del Drenador en cada MOSFET debe ser igual, se tiene que:

$$\frac{\beta_p}{2} (V_{DD} - V_{thp} - V_{sp})^2 = \frac{\beta_n}{2} (V_{sp} - V_{thn})^2 \quad \dots(4.1)$$

Diseño de una ALU empleando Lógica de Configuración Externa.

Tabla 4.1. Parámetros SPICE para el modelo de los transistores NMOS y PMOS.

Parámetros de SPICE BSIM3 Ver. 3.1 (SPICE nivel 8, Star-HSPICE nivel 49, UTMOST nivel 8)		
Parámetro.	Transistor NMOS.	Transistor PMOS.
Voltaje de umbral (V).	$V_{thn} = 0.6411523$	$V_{thp} = -0.9152268$
Movilidad Intrínseca ($\text{cm}^2/\text{V}\cdot\text{s}$).	$U_n = 454.5474058$	$U_p = 201.3603195$

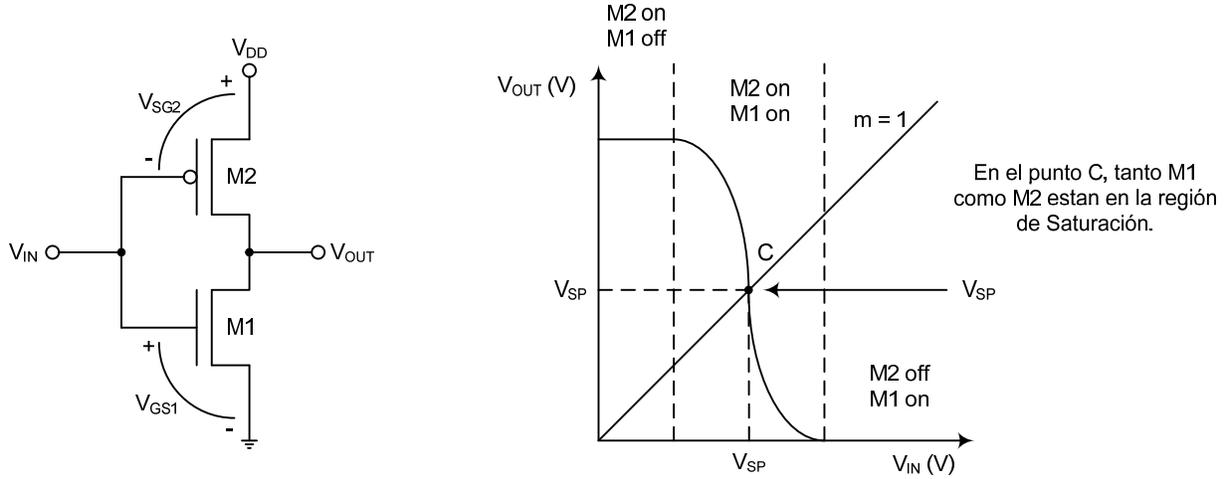


Figura 4.1. El Inversor CMOS y su característica de transferencia, mostrando el punto de conmutación..

Resolviendo para V_{SP} , se obtiene,

$$V_{SP} = \frac{\sqrt{\frac{\beta_n}{\beta_p} * V_{thn} + (V_{DD} - V_{thp})}}{1 + \sqrt{\frac{\beta_n}{\beta_p}}} \quad \dots(4.2)$$

Donde:

$$\sqrt{\frac{\beta_n}{\beta_p}} = \sqrt{\frac{\mu_n \frac{W_n}{L_n}}{\mu_p \frac{W_p}{L_p}}} \quad \dots(4.3)$$

Resolviendo la Ec. (4.1) en términos del aspecto geométrico de los transistores NMOS y PMOS y al sustituir el valor de β_p y β_n (implícitas en la Ec. 4.3), la Ec. (4.1) se puede reescribir como sigue:

$$W_n = W_p \left(\frac{\mu_p * L_n}{\mu_n * L_p} \right) * \frac{(V_{DD} - V_{thp} - V_{sp})^2}{(V_{sp} - V_{thn})^2} \quad \dots(4.4)$$

Idealmente el punto de voltaje de conmutación de un inversor CMOS se presenta cuando $V_{SP} = V_{DD}/2$, sin embargo, para el diseño de la celda básica, y recordando que la ganancia del inversor MIFG CMOS de la etapa de Neurona del dispositivo está acotada por las Ecs. (1.7-10), se tiene que:

$$\gamma_{neu} = \frac{31}{32} = 0.96875 V \quad \dots(4.5)$$

Por lo que el máximo voltaje del Φ_{FG} de la etapa de neurona, será:

$$\gamma_{neu} * V_{DD} = \frac{31}{32} * 5V = 4.84375 V \quad \dots(4.6)$$

Y en consecuencia para lograr una función de transferencia simétrica, se tiene que:

$$\frac{\gamma_{neu} V_{DD}}{2} = 2.421875 V \quad \dots(4.7)$$

Al sustituir V_{SP} en la Ec. (4.4), por el valor encontrado en la Ec. (4.7) y los valores de la tabla 4.1, se obtiene la relación entre los parámetros de los transistores NMOS y PMOS para su simulación en PSpice, como:

$$W_p = W_n \left(\frac{L_p}{L_n} \right) * 1.7048 \quad \dots(4.8)$$

Eligiendo como regla de diseño, asignar la mínima dimensión permitida por la tecnología de 0.5 micras al transistor NMOS ($W_n = 3\mu m$ y $L_n = 0.6\mu m$) y considerando que las longitudes de canal de ambos transistores son iguales, ($L_p = L_n$), se tiene que:

$$W_p = 5.11 \mu m \quad \dots(4.9)$$

Debido a que $5.11\mu m$ no es un múltiplo exacto de la lambda de la tecnológica ($\lambda = 0.3\mu m$), éste requiere un ajuste, el cual será:

$$W_p = 5.1 \mu m \quad \dots(4.10)$$

Las dimensiones físicas de los transistores NMOS y PMOS empleadas en la simulación, se muestran en la tabla 4.2, sin embargo cabe resaltar que la dimensión de los transistores PMOS de la etapa de inversores programables es diferente a la de las otras etapas, debido a que estos deben tener un V_{sp} de 2.5V.

Diseño de una ALU empleando Lógica de Configuración Externa.

Tabla 4.2. Dimensiones físicas de los transistores NMOS y PMOS empleados en las simulaciones de la celda básica SHL.

Etapa de la celda básica:	W_p / L_p ($\mu\text{m}/\mu\text{m}$).	W_n / L_n ($\mu\text{m}/\mu\text{m}$).
Precarga.	5.1 / 0.6	3 / 0.6
Neurona.		
Inversores Programables.		

La tabla 4.3 muestra los V_{sp} calculados, al sustituir las dimensiones de la tabla 4.2 en las Ecs. (4.2) y (4.3), los cuales nos servirán como referencia al presentar las simulaciones obtenidas en OrCAD Capture.

Tabla 4.3. V_{sp} para las dimensiones de los transistores empleados en la simulación.

Etapa de la Celda:	Dimensión Transistores ($\mu\text{m}/\mu\text{m}$).		V_{sp} (V).
	PMOS	NMOS	
Precarga y Neurona.	5.1 / 0.6	3 / 0.6	2.405
Inversores Programables.	5.7 / 0.6		2.4958

Debido al ajuste en las geometrías de los transistores NMOS y PMOS inherente a la tecnología empleada, es necesario re-calcular los voltajes externos empleados en la implementación de las seis compuertas lógicas básicas y los bloques que comprenden la ALU.

Siguiendo el procedimiento descrito en el capítulo II y partiendo de la Ec. (2.44), al utilizar los coeficientes de acoplamiento, así como los voltajes de inversión para la implementación de funciones ilustrados en las Tablas 2.8 y 2.9, se obtienen los voltajes externos empleados para la simulación, los cuales se presentan en la tabla 4.4.

A causa de la suma ponderada que se realiza en cada una de las compuertas flotantes del inversor CMOS en las etapas de la celda básica SHL, los coeficientes de acoplamiento deben ajustarse multiplicándose por un factor relacionado directamente al número de coeficientes de acoplamiento en las etapas de precarga, inversores programables y Neurona, los cuales son 6, 5 y 10 respectivamente.

Los coeficientes de acoplamiento ajustados de cada una de las etapas de la celda básica SHL empleados en simulación, son los presentados en la tabla 4.5.

Diseño de una ALU empleando Lógica de Configuración Externa.

Tabla 4.4. Voltajes de Inversión de la etapa de Inversores Programables empleados en la simulación de la celda básica SHL para la implementación de las seis funciones básicas y los bloques de la ALU.

Voltaje Externo (ajustado)	Función a implementar por la celda básica SHL.								
	AND	NAND	OR	NOR	X-OR	X-NOR	f_{CARRY}	F1	F2
V_A	4.99 V	2.64 V	2.49 V	4.83 V	2.64 V	4.83 V	4.99 V		4.99 V
V_B					3.11 V	4.36 V	4.36 V		
V_C					3.58 V	3.89 V	3.74 V	4.05 V	
V_D					4.21 V	3.27 V		3.74 V	
V_E					4.52 V	2.96 V		2.96 V	2.49 V
V_F					2.49 V	4.99 V		2.49 V	
Vctl.									

Tabla 4.5. Coeficientes de Acoplamiento ajustados de las etapas de precarga, Inversores Programables y neurona para la simulación de las funciones básicas lógicas y bloques de la ALU.

Bloque de la ALU.	Coeficientes de Acoplamiento ajustados para simulación de la Etapa:			
	Precarga.	Inversores Programables.	Neurona.	
AND, NAND, OR, NOR, X-OR, X-NOR	$K_{X1P} = \frac{6}{32} = 0.1875$	$K_{X1A} = \frac{5}{60} = 0.0833$	$K_{X1N} = \frac{10}{32} = 0.3125$	$K_C = \frac{40}{32} = 1.25$
$f_{CARRY.}$	$K_{X2P} = \frac{12}{32} = 0.375$		$K_{X2A} = \frac{10}{60} = 0.1666$	$K_{X2N} = \frac{20}{32} = 0.625$
F1.	$K_{X3P} = \frac{24}{32} = 0.75$	$K_{X3A} = \frac{20}{60} = 0.3333$	$K_{X3N} = \frac{40}{32} = 1.25$	$K_C = \frac{20}{32} = 0.625$
	$K_{X4P} = \frac{48}{32} = 1.5$	$K_{X4A} = \frac{40}{60} = 0.6666$	$K_{X4N} = \frac{80}{32} = 2.5$	
F2.	$K_{CTL} = \frac{90}{32} = 2.8125$	$K_{INVA} = \frac{150}{60} = 2.5$	$K_A = \frac{30}{32} = 0.9375$	
	$K_F = \frac{6}{32} = 0.1875$		$K_B = \frac{20}{32} = 0.625$	
			$K_D = \frac{30}{32} = 0.9375$	$K_E = \frac{50}{32} = 1.5625$

El diagrama a bloques que se presenta en la figura 4.2, corresponde al de la celda básica SHL de 4 bits que se ilustró en la figura 2.3 del capítulo 2, cuyo diagrama esquemático para simulación en OrCAD se presenta en la Figura 4.3.

Diseño de una ALU empleando Lógica de Configuración Externa.

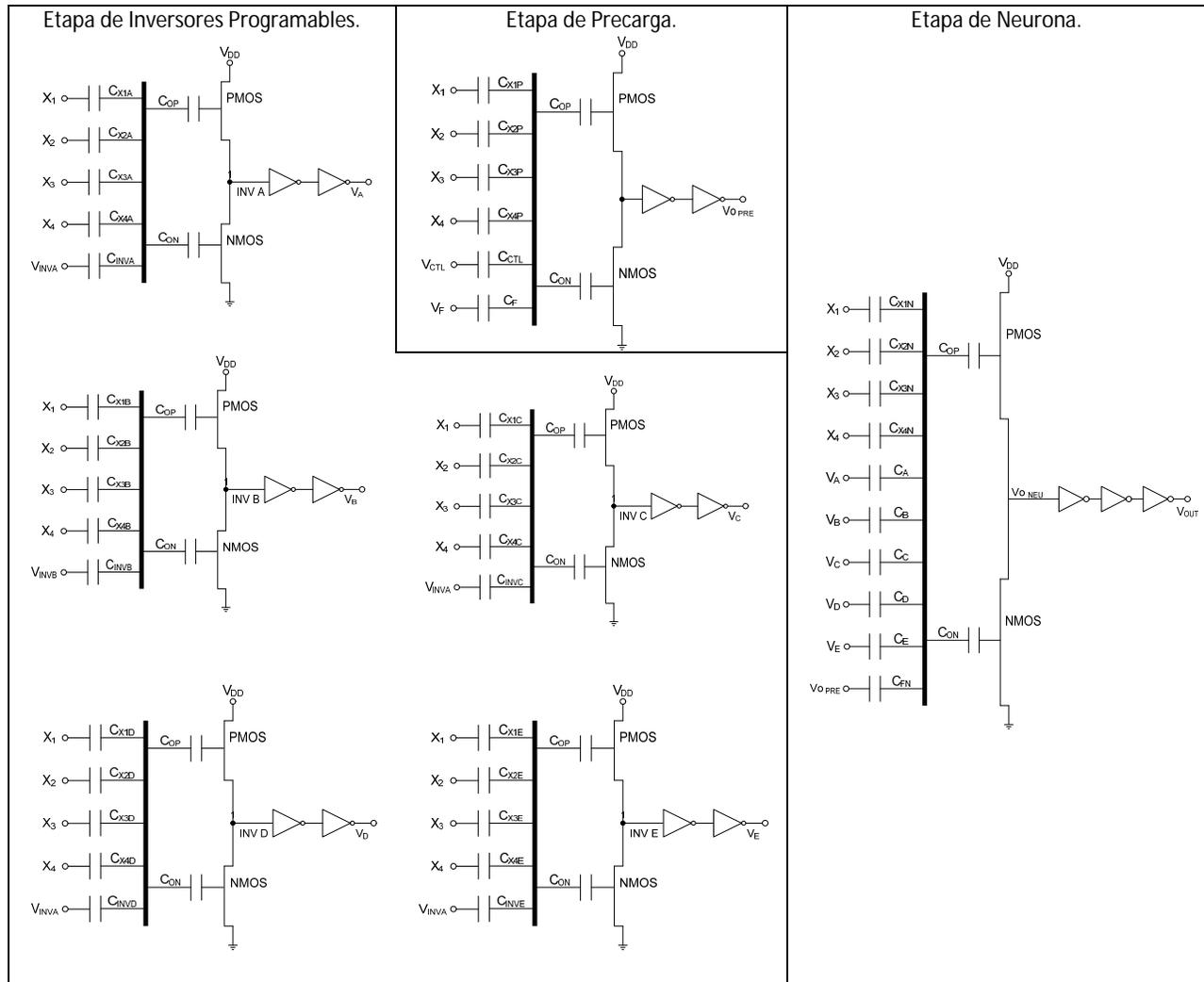


Figura 4.2. Diagrama a bloques de la figura 2.3 correspondiente a la celda básica SHL de 4 bits.

En la figura 4.3, se puede observar que los coeficientes capacitivos de cada una de las etapas de la celda SHL (ilustrados en la figura. 4.2), fueron declarados mediante el factor de ganancia que poseen las Fuentes de Voltaje dependientes de Voltaje.

Las señales de entrada V_A , V_B , V_C , V_D , V_E , V_F y V_{CTL} , correspondientes a las etapas de inversores programables y precarga respectivamente, se declararon mediante Fuentes Independientes de Voltaje, cuyos valores estarán determinados de acuerdo a la Función que se desea implementar, los cuales se mostrarán a medida que se presenten las simulaciones en DC y transitorio de cada implementación.

Para propósitos de simulación en DC, las entradas de señal de 4 bits (X_1 , X_2 , X_3 y X_4) de cada una de las etapas de la celda SHL, se declararon y agruparon en una Fuente de Voltaje Independiente, V_p , la cual se varió desde 0V hasta 5V con incrementos de 1mV.

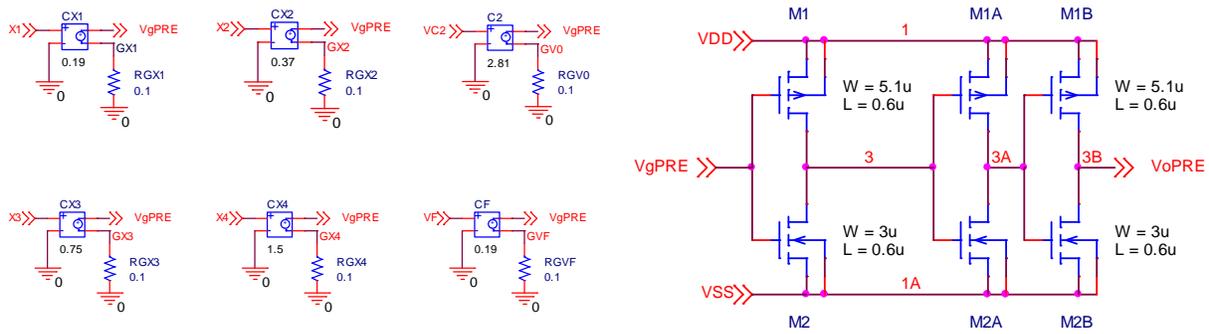
Para la simulación en estado transitorio, las entradas de señal de 4 bits (X_1 , X_2 , X_3 y X_4) de las etapas de precarga, inversores programables y neurona respectivamente, fueron declaradas mediante una Fuente

Diseño de una ALU empleando Lógica de Configuración Externa.

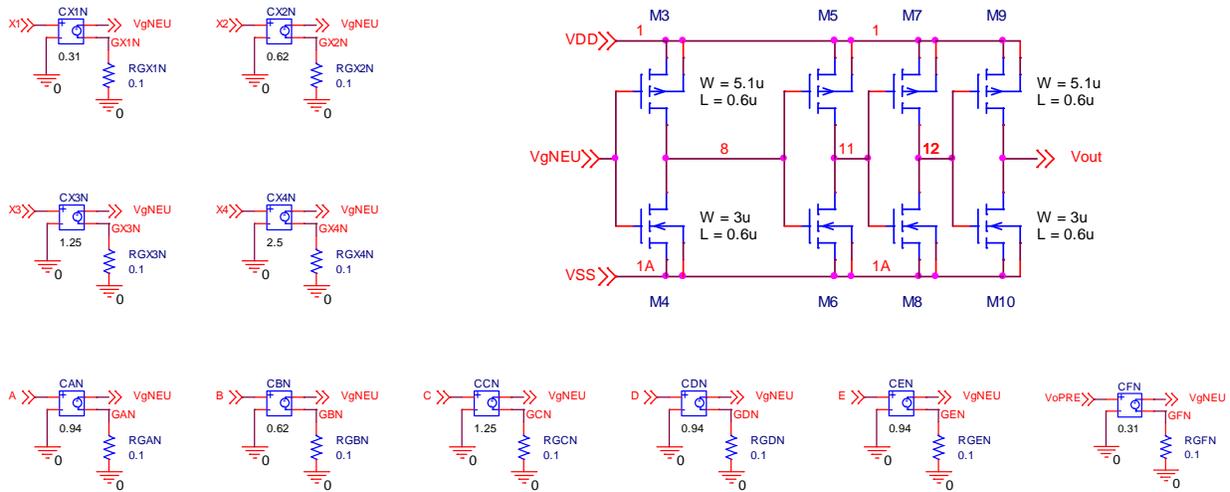
de Voltaje Pulsado (V_{PULSE}), cuyos parámetros se definieron de la siguiente forma: el voltaje de inicio igual a cero volts ($V_1 = 0V$), el voltaje pulsado igual a V_{DD} ($V_2 = 5V$).

Mientras que los tiempos de retraso (TD), subida (TR), bajada (TF), así como el ancho del pulso (PW) y periodo del mismo (PER), se determinaron a partir de las características de conmutación de las hojas de especificaciones del circuito MC14581B (Anexo A, pág. 3), con el cual se comparara en el capítulo V.

Las declaraciones de las fuentes y parámetros empleados en la simulación en DC y estado transitorio, se muestran en la figura 4.4 (a) y (b), las formas de onda generadas por tales fuentes, pueden observarse en la figura 4.5 (a) y (b) respectivamente.

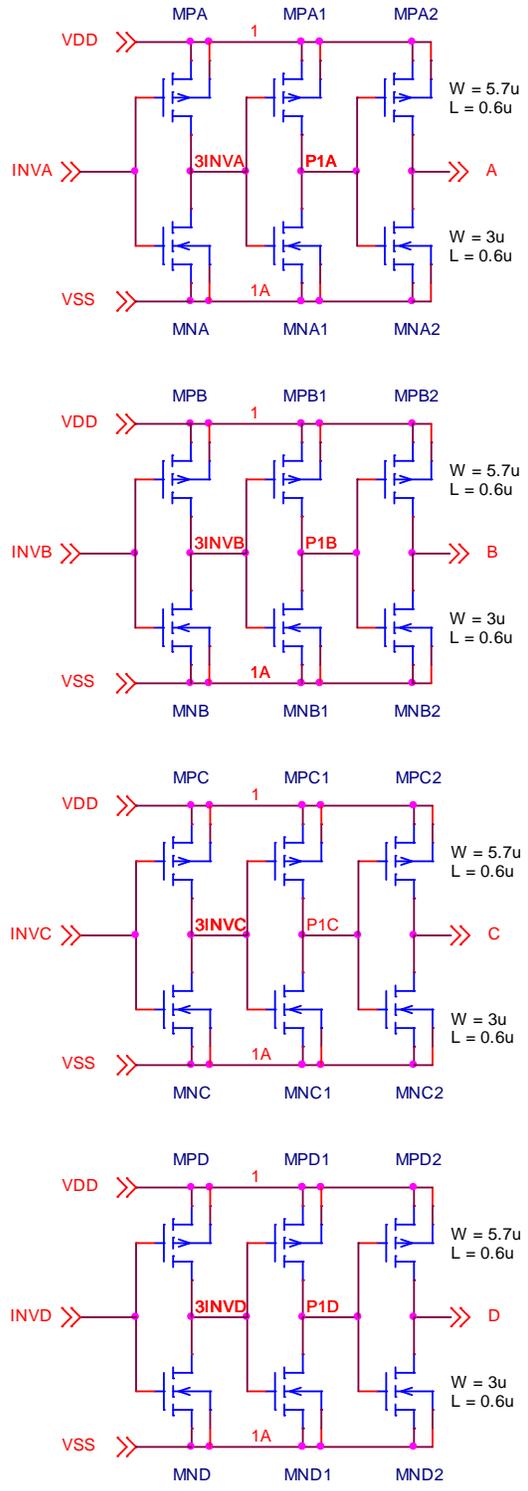
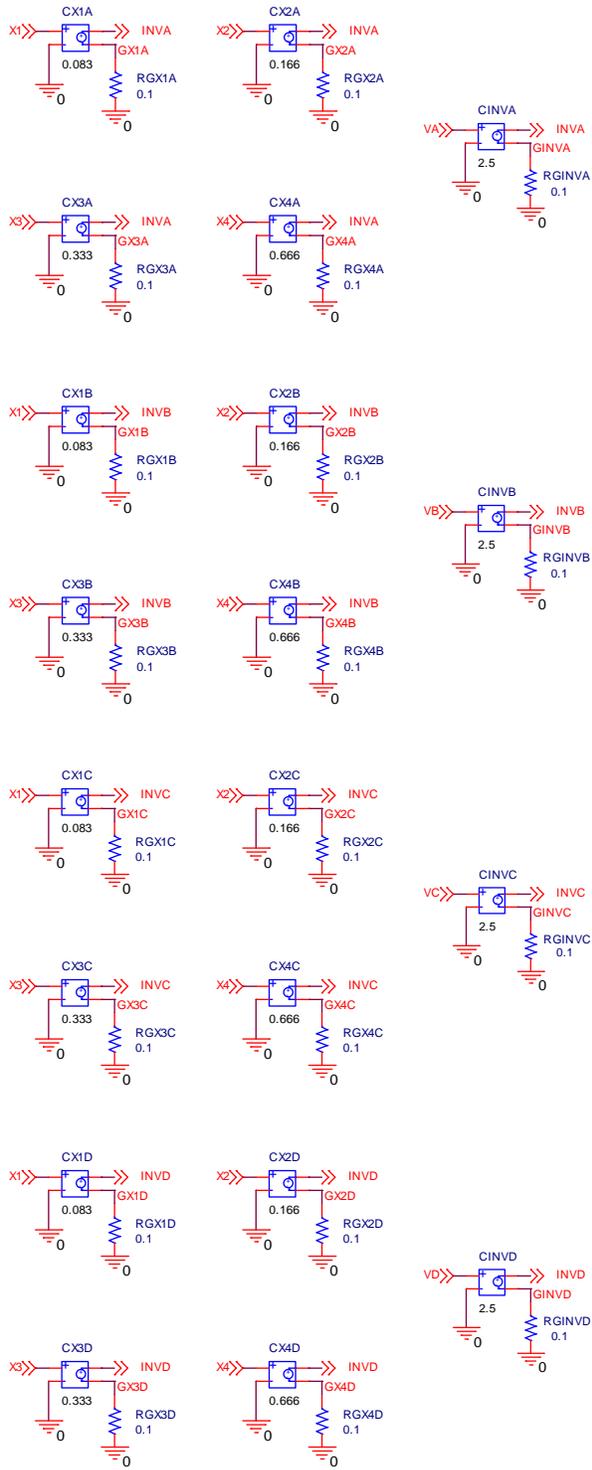


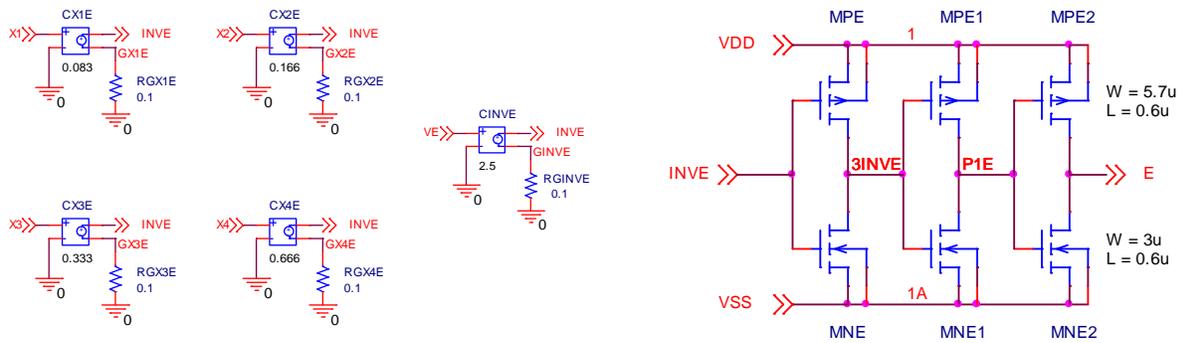
a) Coeficientes de acoplamiento e inversor CMOS de la etapa de Precarga.



b) Coeficientes de acoplamiento e inversor CMOS de la etapa de Neurona.

Diseño de una ALU empleando Lógica de Configuración Externa.





c) Coeficientes de acoplamiento e inversor CMOS de la etapa de Inversores Programables A, B, C, D y E respectivamente.

Figura 4.3. Diagrama Esquemático de las etapas de precarga (a), inversores programables (b) y neurona (c) para la simulación de la celda básica SHL de 4 bits

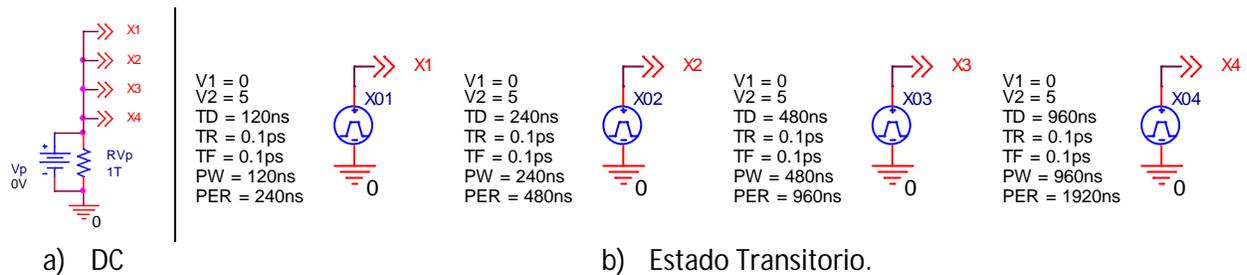


Figura 4.4. Declaración de las fuentes empleadas en la simulación en DC y transitorio para el esquemático de la figura 4.3.

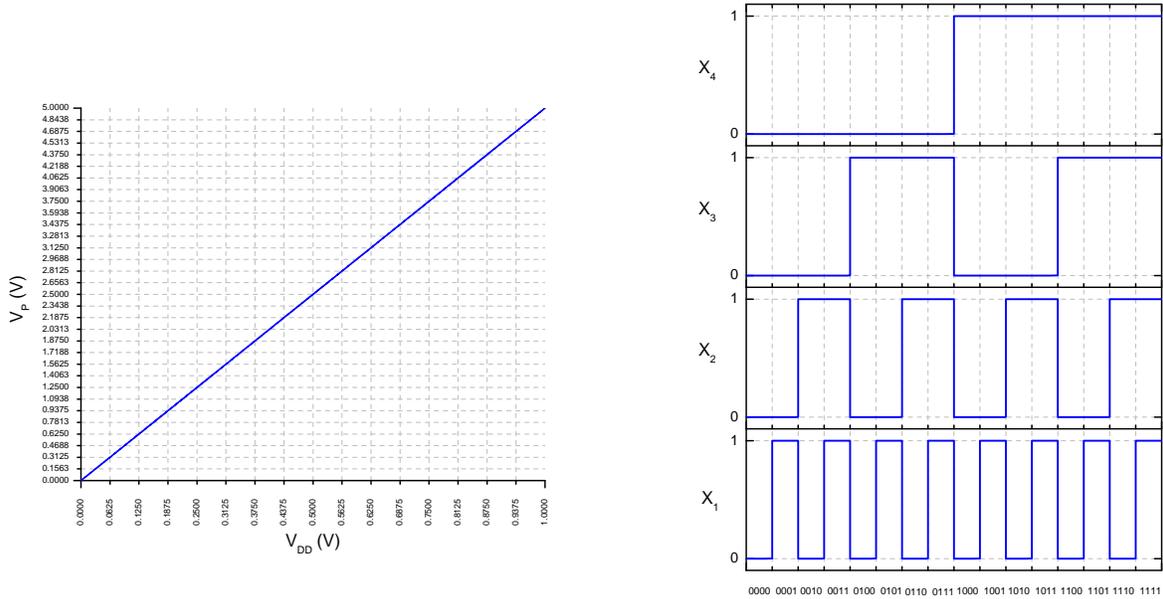
4.3. Simulación de las Compuertas AND, NAND, OR, NOR, XOR, XNOR y bloques que componen la ALU.

En las figuras 4.7 – 4.16, se muestran las simulaciones en DC y transitorio del diagrama esquemático presentado en la figura 4.3, correspondientes a las implementaciones de las funciones lógicas AND, NAND, OR, NOR, XOR y XNOR, así como a las funciones de SUMA y generación de acarreo (f_{CARRY}) del bloque principal y las funciones del sub-bloque F1 y F2 de la ALU, respectivamente.

Tales implementaciones se obtuvieron aplicando los voltajes externos apropiados a la etapa de inversores programables, los cuales se pueden observar en la parte superior de cada una de las figuras. Además para una rápida verificación de la salida de la celda básica ($V_{OUT\ NEU}$), se incluyó la tabla de verdad de cada función implementada.

Cabe recordar que los coeficientes de acoplamiento K_C y K_E de la etapa de neurona de la celda básica SHL empleada para implementación del bloque principal y los bloques secundarios de la ALU, son diferentes.

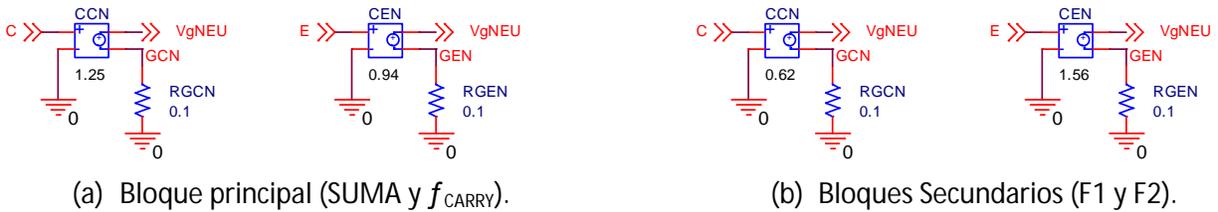
Por lo cual, la única consideración al realizar las simulaciones en DC y transitorio de los bloques principal y secundarios de la ALU, fue ajustar el factor de ganancia empleado para dichos coeficientes de acoplamiento, cuyos valores se pueden observar de la tabla 4.5 y cuya declaración se presenta en la figura 4.6.



(a) Salida generada por V_p .

(b) Salida generada por X_1, X_2, X_3 y X_4 .

Figura 4.5. Forma de Onda empleadas para la simulación en DC (a) y estado transitorio (b).



(a) Bloque principal (SUMA y f_{CARRY}).

(b) Bloques Secundarios (F1 y F2).

Figura 4.6. Declaración esquemática de los coeficientes K_C y K_E de la etapa de neurona para la implementación del bloque principal (a) y bloques secundarios (b) de la ALU.

Diseño de una ALU empleando Lógica de Configuración Externa.

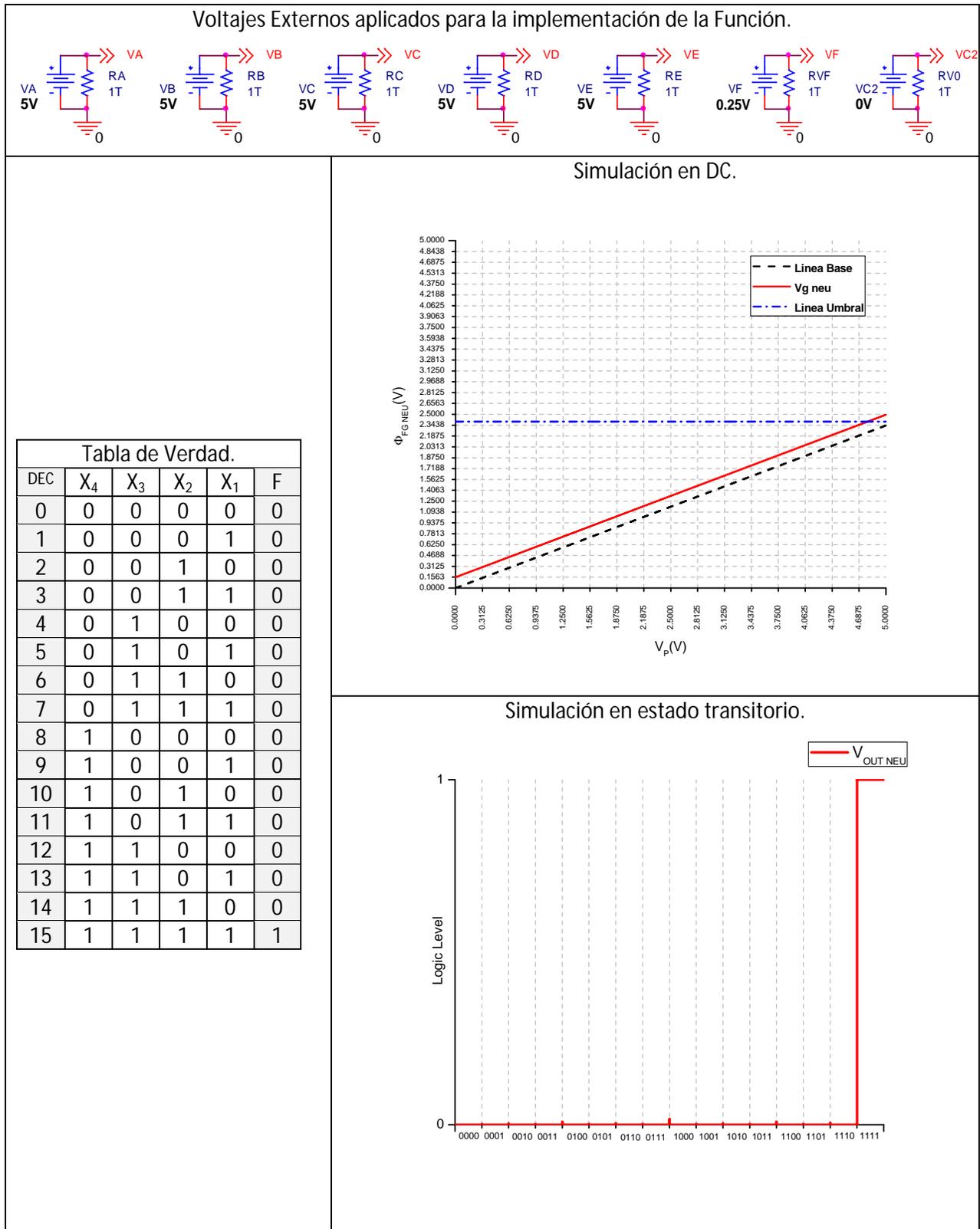


Figura 4.7. Simulaciones en DC y en estado transitorio de la Función AND.

Diseño de una ALU empleando Lógica de Configuración Externa.

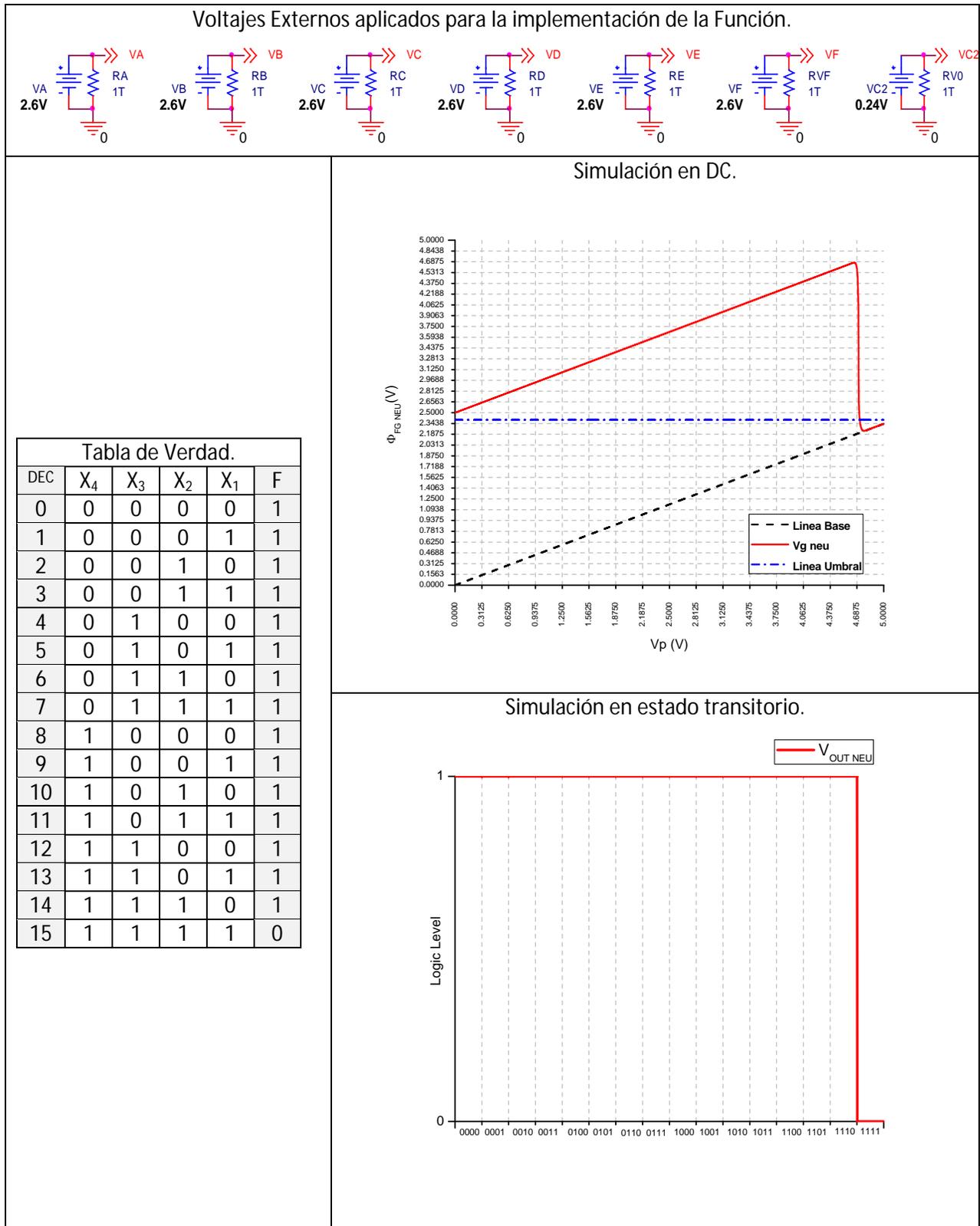


Figura 4.8. Simulaciones en DC y en estado transitorio de la Función NAND.

Diseño de una ALU empleando Lógica de Configuración Externa.

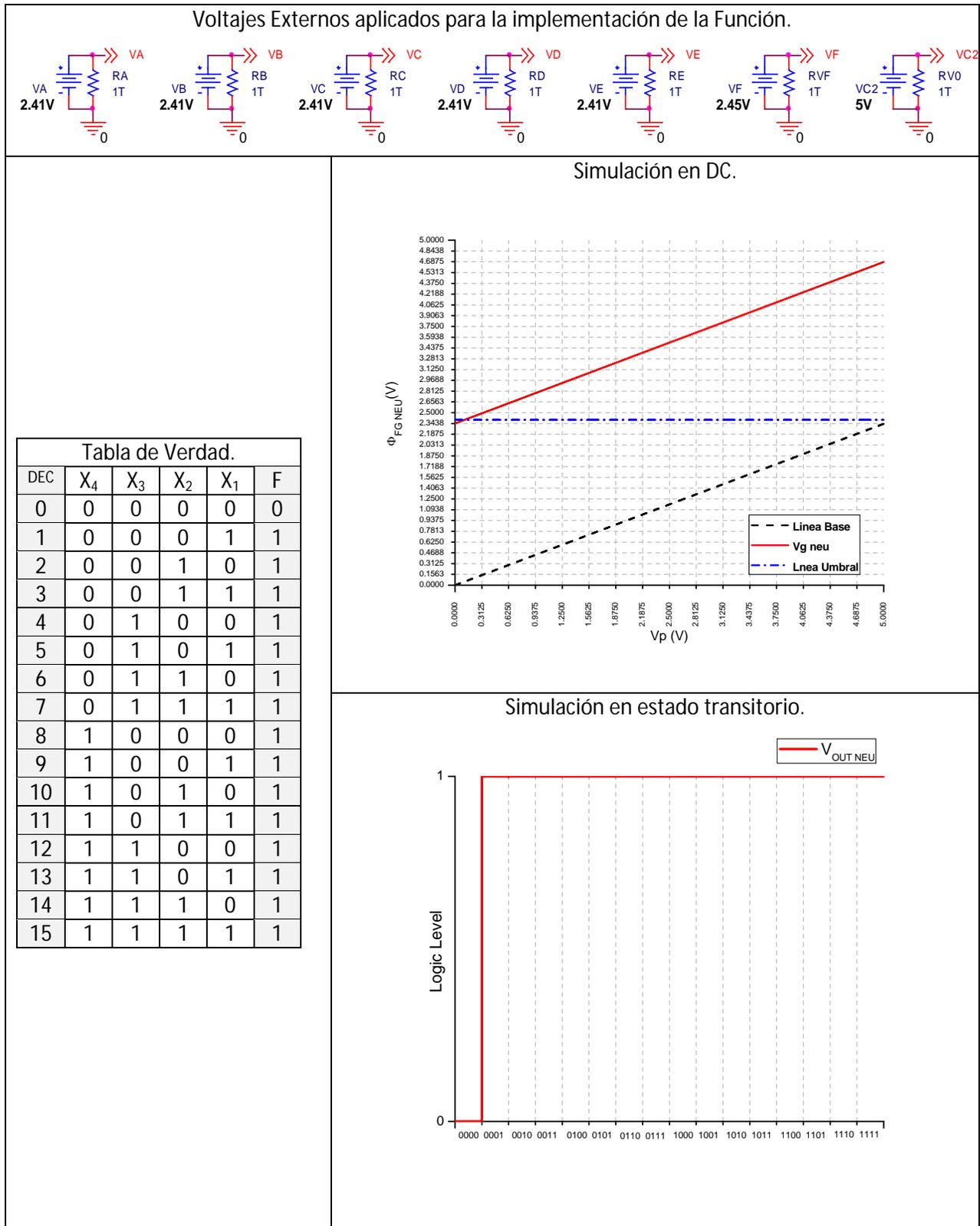


Figura 4.9. Simulaciones en DC y en estado transitorio de la Función OR.

Diseño de una ALU empleando Lógica de Configuración Externa.

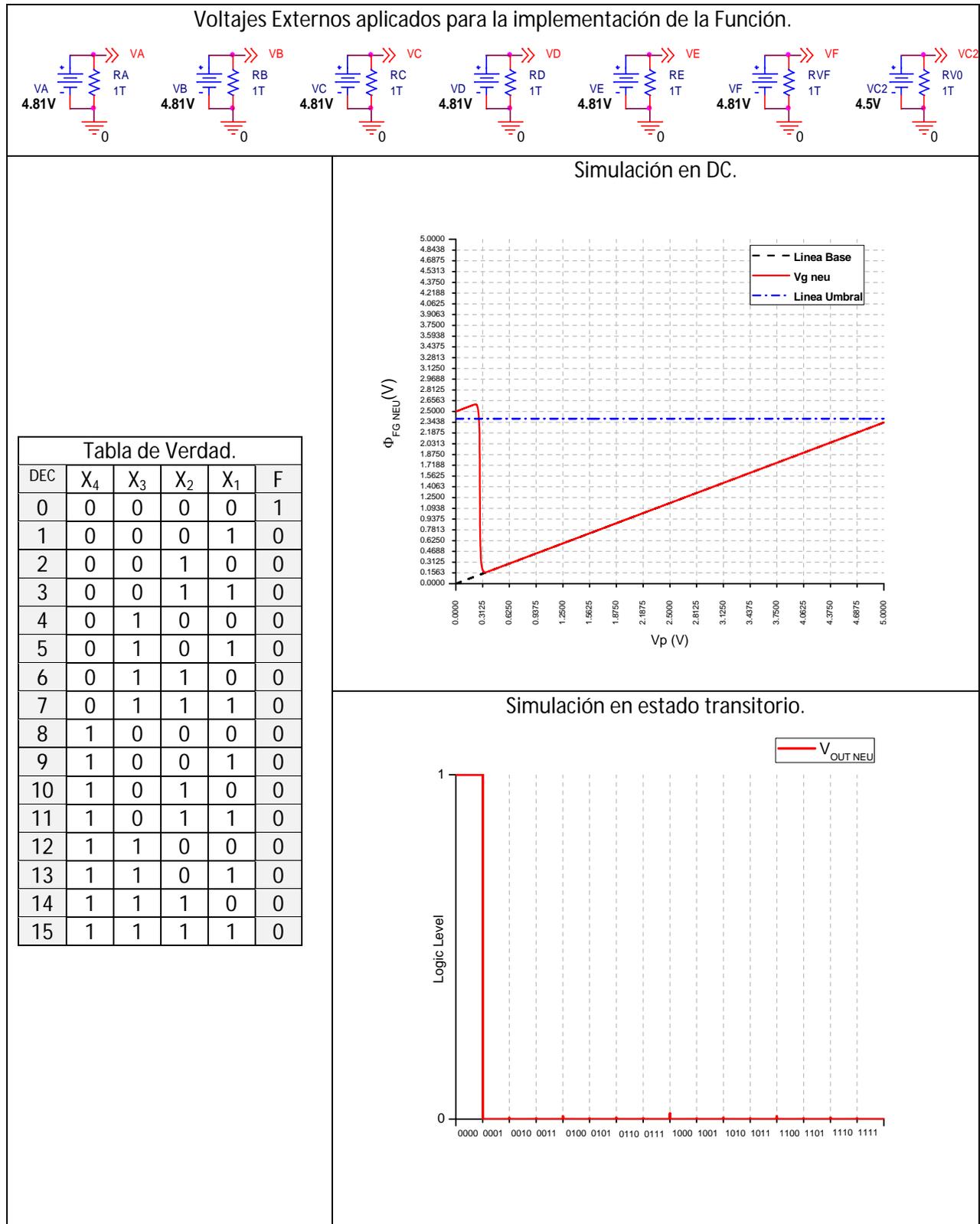


Figura 4.10. Simulaciones en DC y en estado transitorio de la Función NOR.

Diseño de una ALU empleando Lógica de Configuración Externa.

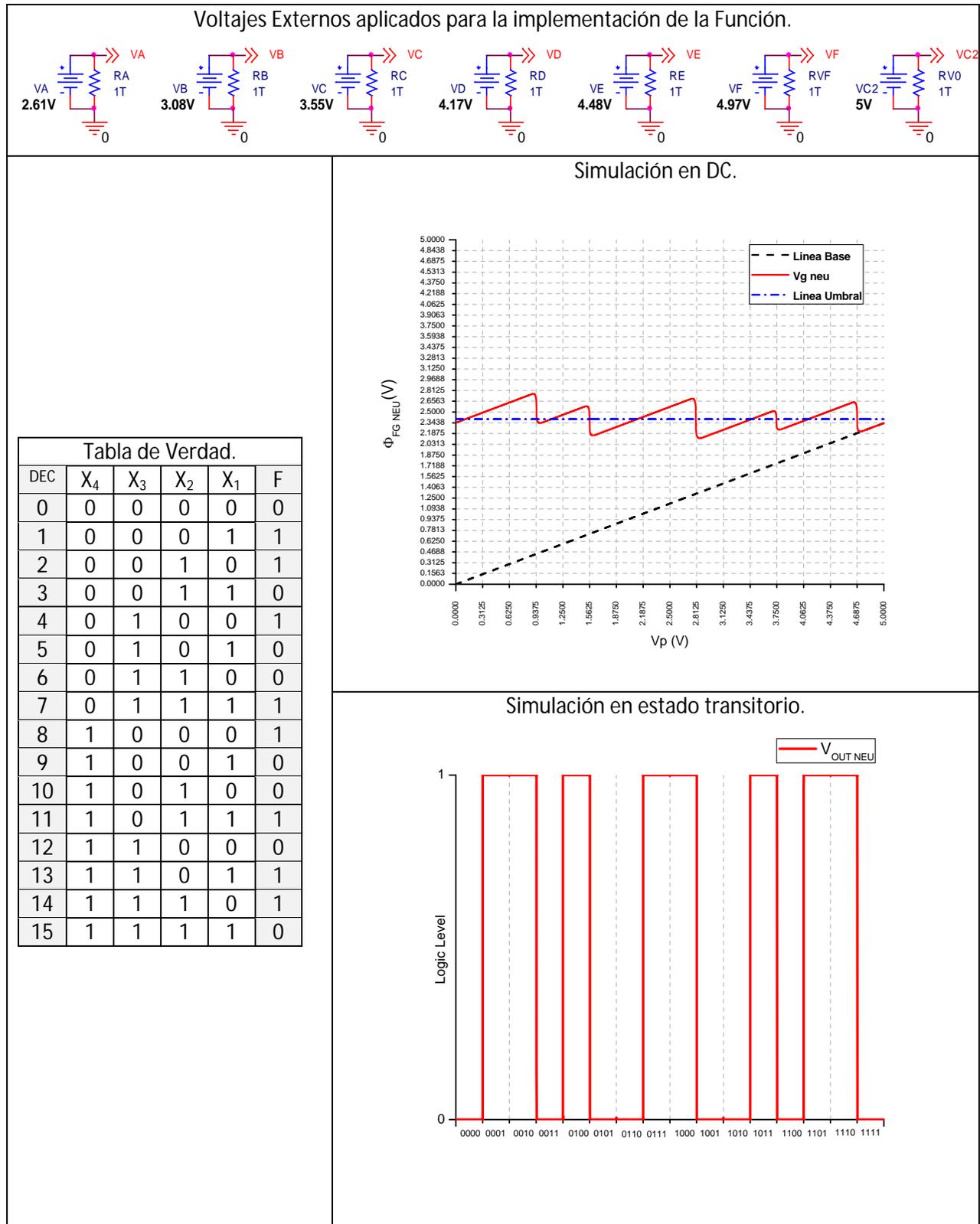


Figura 4.11. Simulaciones en DC y en estado transitorio de la Función XOR.

Diseño de una ALU empleando Lógica de Configuración Externa.

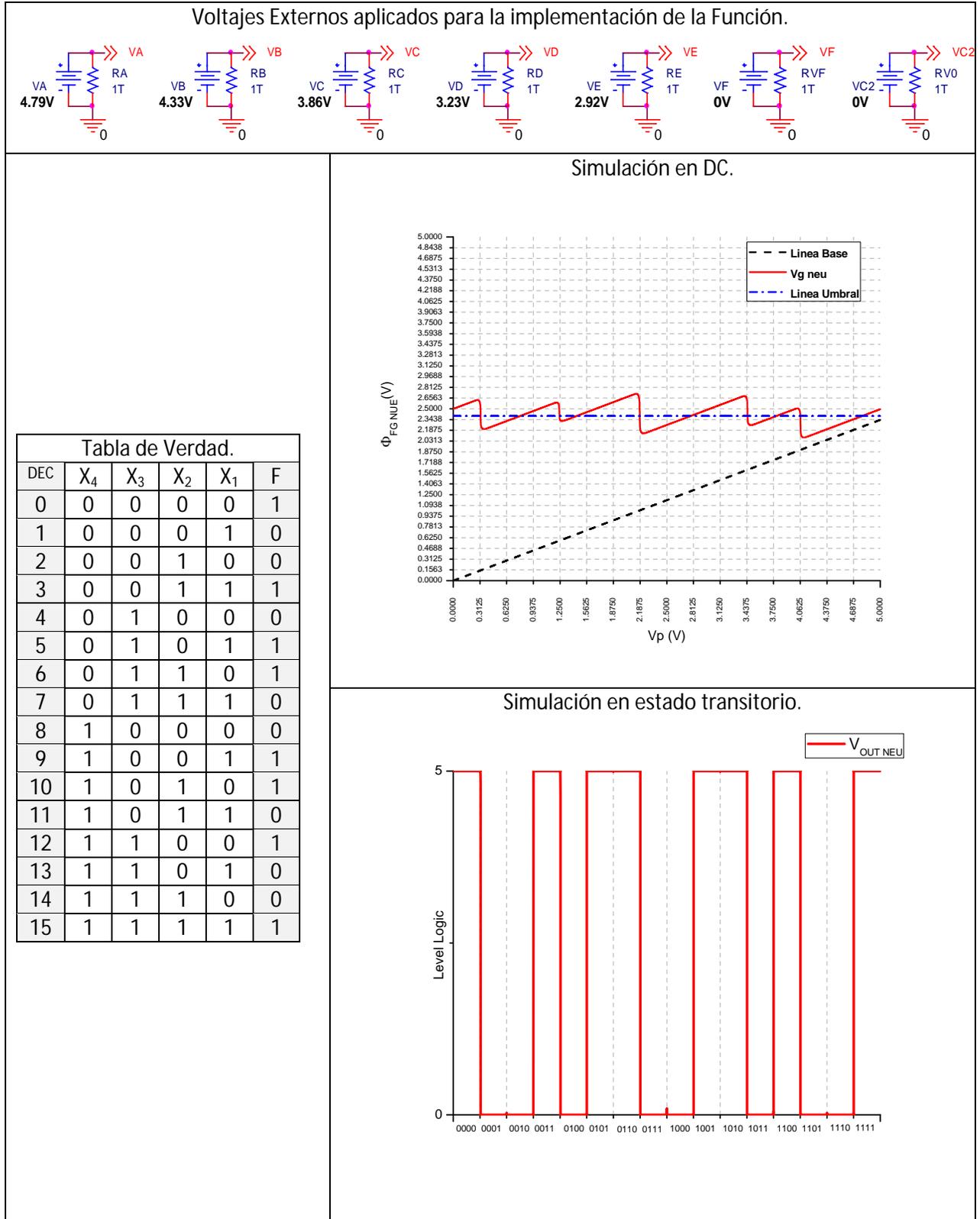


Figura 4.12. Simulaciones en DC y en estado transitorio de la Función XNOR.

Diseño de una ALU empleando Lógica de Configuración Externa.

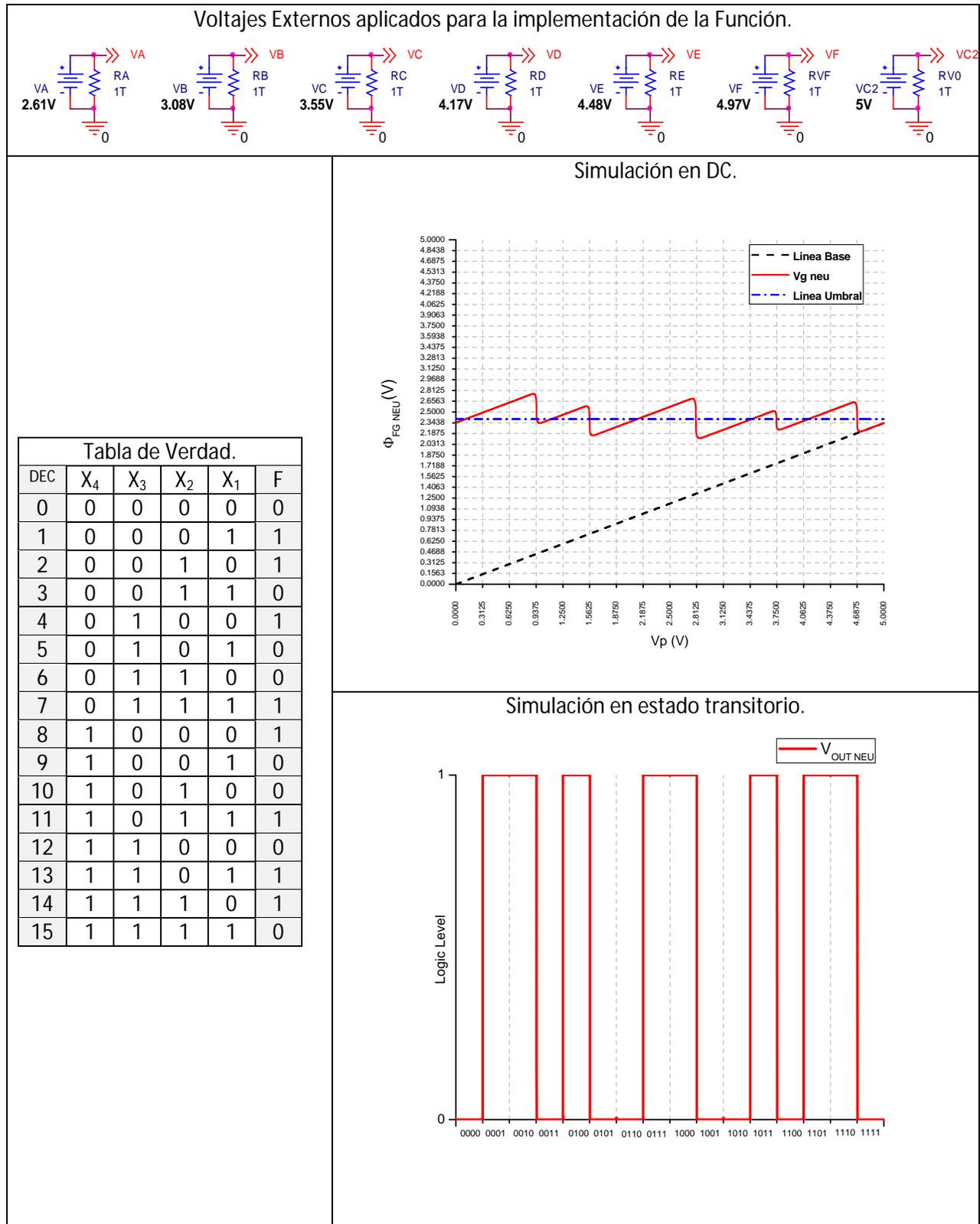


Figura 4.13. Simulaciones en DC y en estado transitorio del bloque de SUMA de la ALU.

Diseño de una ALU empleando Lógica de Configuración Externa.

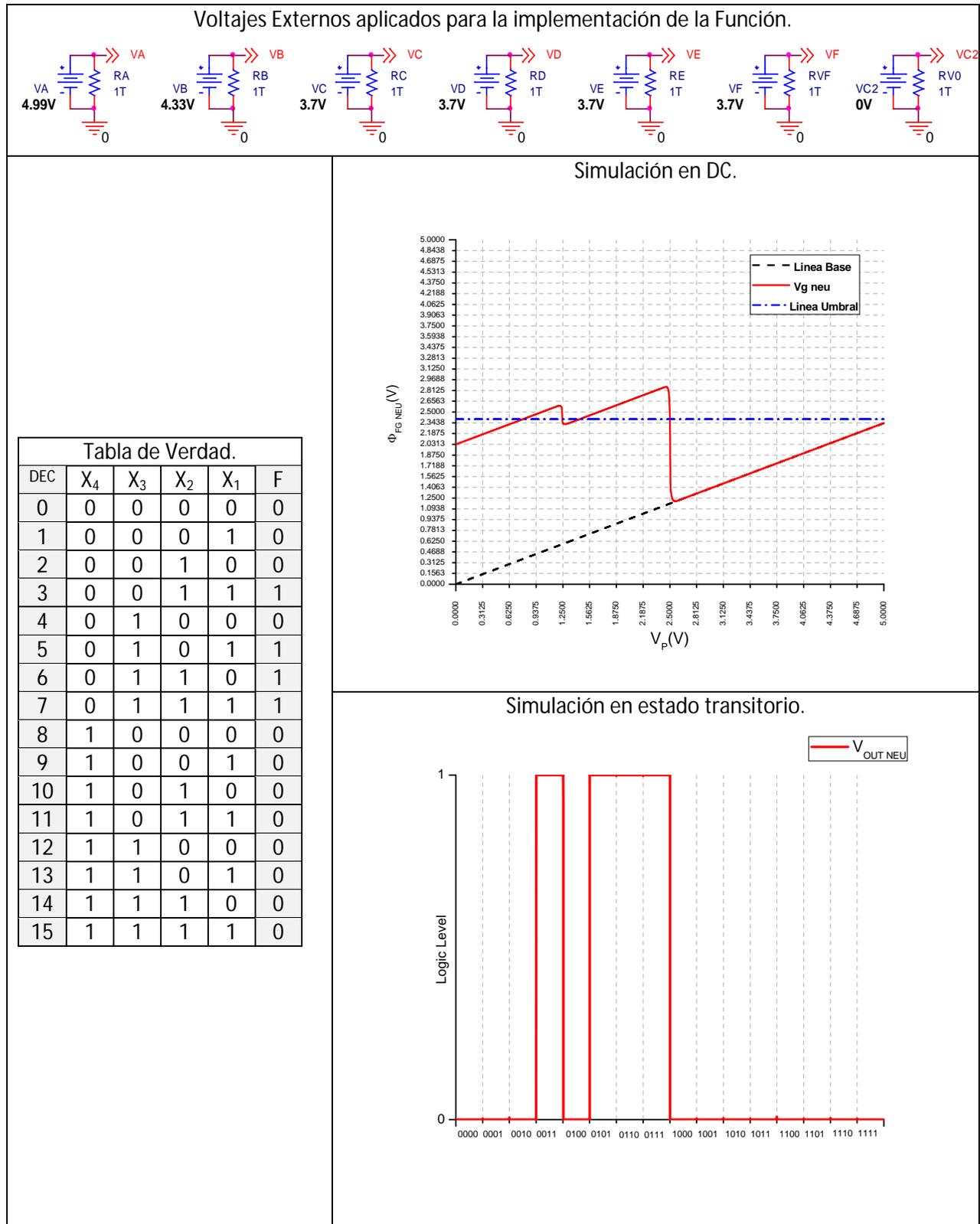


Figura 4.14. Simulaciones en DC y en estado transitorio del bloque de Acarreo de la ALU.

Diseño de una ALU empleando Lógica de Configuración Externa.

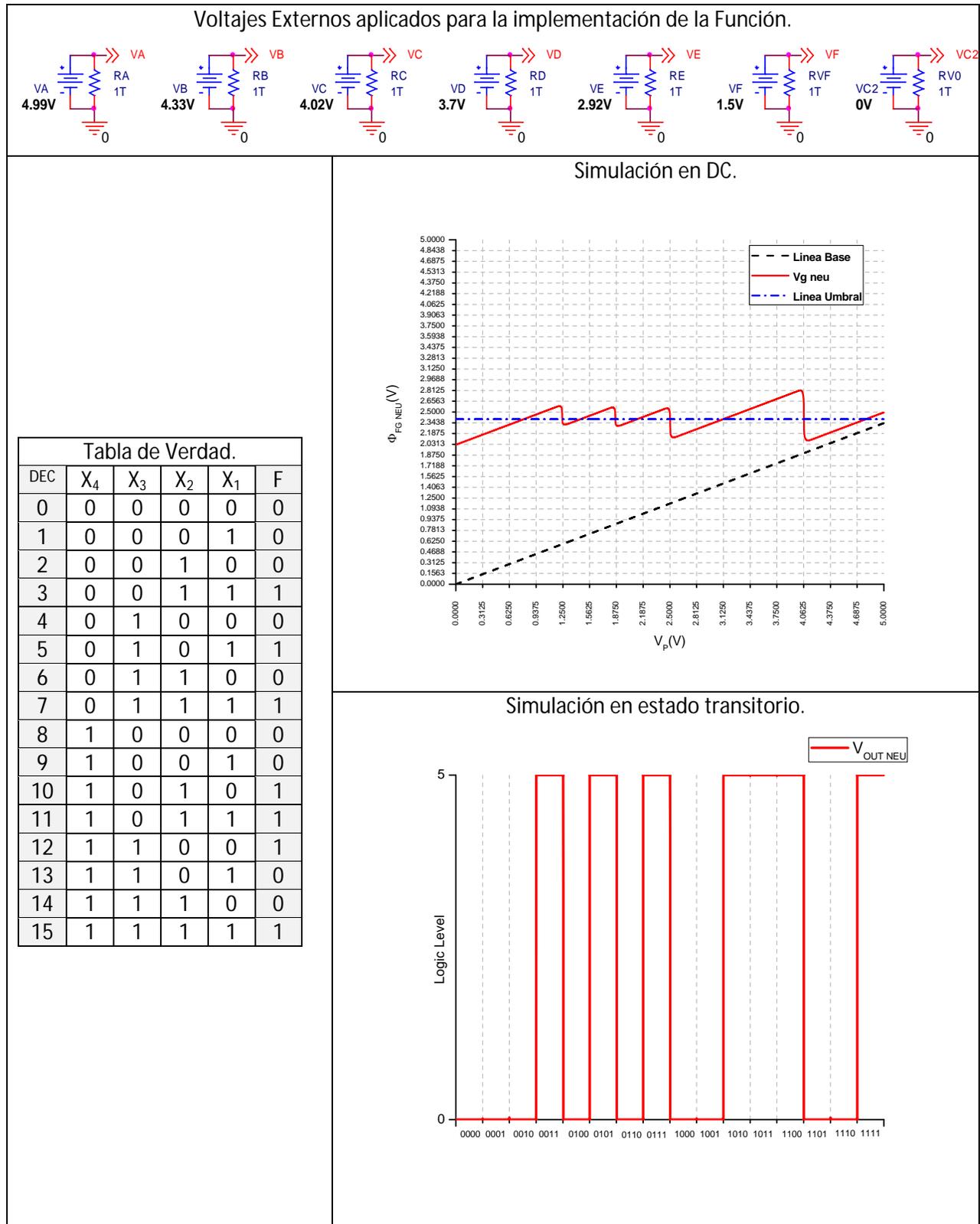


Figura 4.15. Simulaciones en DC y en estado transitorio del sub-bloque de Función F1 de la ALU.

Diseño de una ALU empleando Lógica de Configuración Externa.

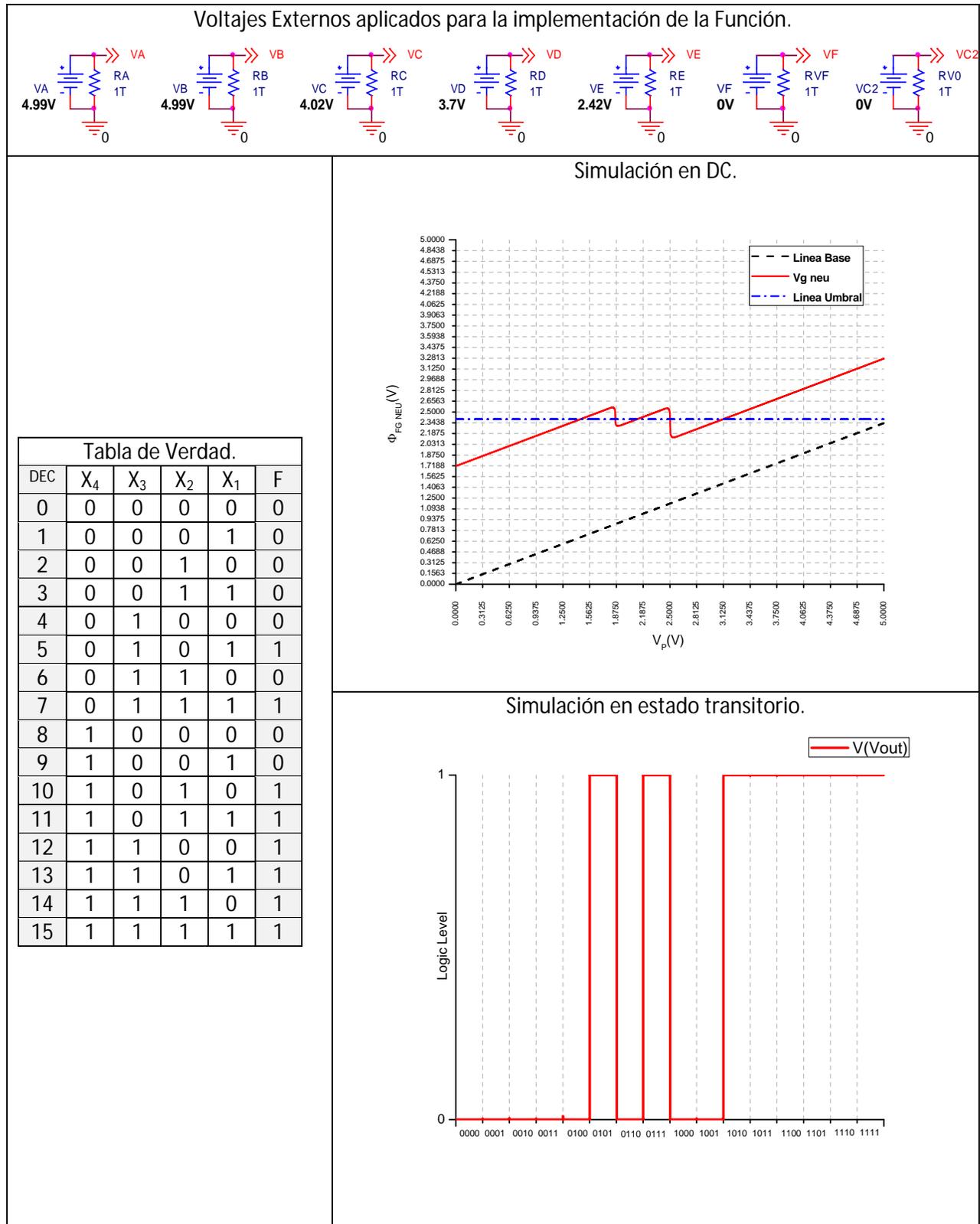


Figura 4.16. Simulaciones en DC y en estado transitorio del sub-bloque de Función F2 de la ALU.

4.4. Conclusiones.

En este capítulo, se demostró mediante las simulaciones en DC y estado transitorio, la factibilidad del uso de la celda básica SHL para la implementación de las seis funciones lógicas básicas, así como de las funciones del bloque principal (SUMA y f_{CARRY}) y bloques secundarios (F1 y F2) de la ALU de 4 bits SHL basada en transistores neuMOS.

La versatilidad de implementación de funciones de la celda básica SHL, puede extenderse a un número mayor siguiendo el método gráfico ilustrado FPD, como restadores completos por citar algún ejemplo, lo cual nos llevó al diseño de una celda básica universal capaz de implementar cualquier función de 4 bits que se desee, la cual se tratará como un trabajo futuro.

Sin embargo, aun hay que trabajar aun en la frecuencia de operación del dispositivo, la cual sigue siendo baja en comparación con los circuitos MOS actualmente en el mercado.

4.5. Referencias.

- [1] R. Jacob Baker, "CMOS Circuit Design, Layout and Simulation", John Wiley & Sons inc, 2nd. Ed. 2005, pg 331- 352.

[2]

Capítulo V. Comparaciones de las ALU de 4 bits: MC14581B/ FPGA/ SHL.

5.1 Introducción.

Una vez que se verificó que los bloques de SUMA, f_{CARRY} , F1 y F2 funcionan correctamente, se realizaron las simulaciones de las dos operaciones aritméticas y las dos operaciones lógicas que es capaz de realizar la ALU SHL, las cuales se presentan en el primer parte de este capítulo.

En la segunda parte del capítulo y con la finalidad de tener una referencia para poder realizar la comparación de los resultados de simulación obtenidos para la ALU diseñada mediante la celda básica SHL de 4 bits, se presentan las mediciones de las mismas operaciones en el circuito integrado MC14581B.

En la tercer parte del capítulo, se presentan las mediciones obtenidas de la implementación de una ALU de 4 bits en un FPGA Spartan III de Xilinx, el cual implementa las mismas operaciones que la ALU SHL y el chip.

Finalmente se presentan la comparaciones de cada una de las operaciones realizadas (A and B, A xnor B, A + B y $A \bullet B - 1$) por las 3 ALU's (SHL, FPGA y CHIP) respectivamente.

5.2 ALU SHL de 4 Bits.

De acuerdo al diagrama a bloques ilustrado en la fig. 3.1 del capítulo 3, se realizó la declaración de cada una de las celdas que integran los bloques de la ALU SHL empleando el programa PSpice esquemático de OrCAD, sin embargo y debido a lo extenso del archivo generado, la declaración mencionada tendrá que ser consultada en el anexo C de esta tesis.

Las señales de excitación de los bloques de palabra A y B, se realizaron mediante Fuentes pulsadas, cuya declaración se muestra en la figura 5.1, las cuales básicamente realizan un conteo binario de 0 a 15 para el bloque de palabra A y un sub-conteo binario de 0 a 15 por cada cuenta de A, para el bloque de palabra B, generando de esta forma las 256 posibles combinaciones automáticamente.

Debido a la complejidad de las formas de onda de salida de la ALU's, generadas a partir de las 256 posibles combinaciones de entrada (palabras de A y B), todos los resultados obtenidos tanto en las mediciones como en las simulaciones de este capítulo, se mostraran en el rango de conteo binario de 8 a 11 para A y para B de 0 a 15 para cada valor de A, como se mostrará más adelante en la figuras presentadas para cada simulación o medición respectivamente. Sin embargo todas las simulaciones o mediciones que aquí se exponen, pueden ser revisadas en el anexo D del presente trabajo, para el rango completo.

En la tabla 5.1, se reprodujo la tabla de selección de operación de la ALU SHL del capítulo 3, en la cual nos basamos para la obtención de las simulaciones de cada una de las funciones que es capaz de realizar la ALU SHL propuesta en este trabajo de tesis.

Cabe mencionar que para todas las simulaciones presentadas de la ALU SHL, el bit del acarreo de entrada del bloque de Sumador Completo tendrá asignado por default el valor de cero ($C_{\text{IN}} = 0$).

Diseño de una ALU empleando Lógica de Configuración Externa.

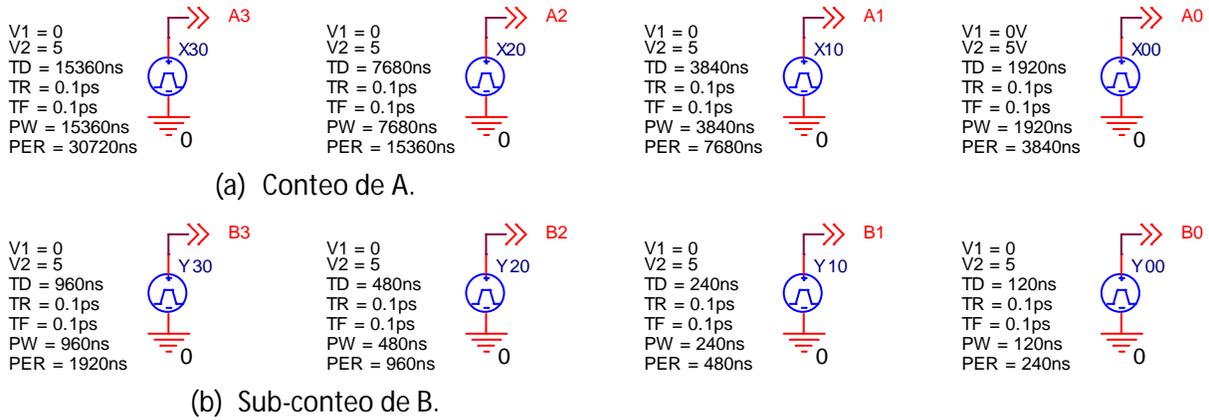


Figura 5.1. Declaración de las Fuentes pulsadas para la generación de las 256 combinaciones de los bloques de palabra A y B respectivamente.

Tabla 5.1. Selección de operaciones de la ALU mediante la señal de control 'S'.

Entradas de Control.				Entradas de Señal.		Función implementada por la ALU
S[3]	S[2]	S[1]	S[0]			
0	0	0	0	B	A	$A \bullet B$
0	0	1	1			$\overline{A \oplus B}$
1	0	0	1			$A+B$
1	1	0	0			$A \bullet B - 1$

En la figura 5.2, se muestran la declaración de los voltajes externos aplicados a las etapas de inversores programables de cada uno de los bloques de la ALU SHL, mediante fuentes de voltaje independientes.

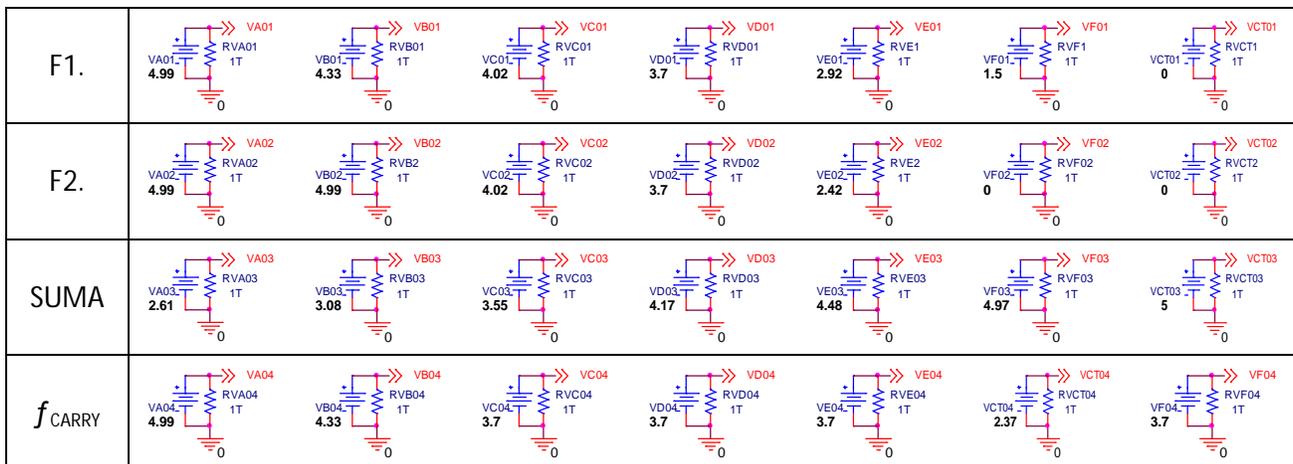


Figura 5.2. Voltajes externos aplicados a la etapa de inversores programables de cada uno de los bloques de la ALU SHL.

Diseño de una ALU empleando Lógica de Configuración Externa.

A continuación en las figuras 5.3, 5.4, 5.5 y 5.6 se presentan las formas de onda resultantes de las simulaciones de la ALU SHL para las funciones de $A \bullet B$, $A \text{xnor} B$, $A+B$ y $A \bullet B - 1$ respectivamente, conforme a los valores mostrados para la señal de control 'S' de la tabla 5.1.

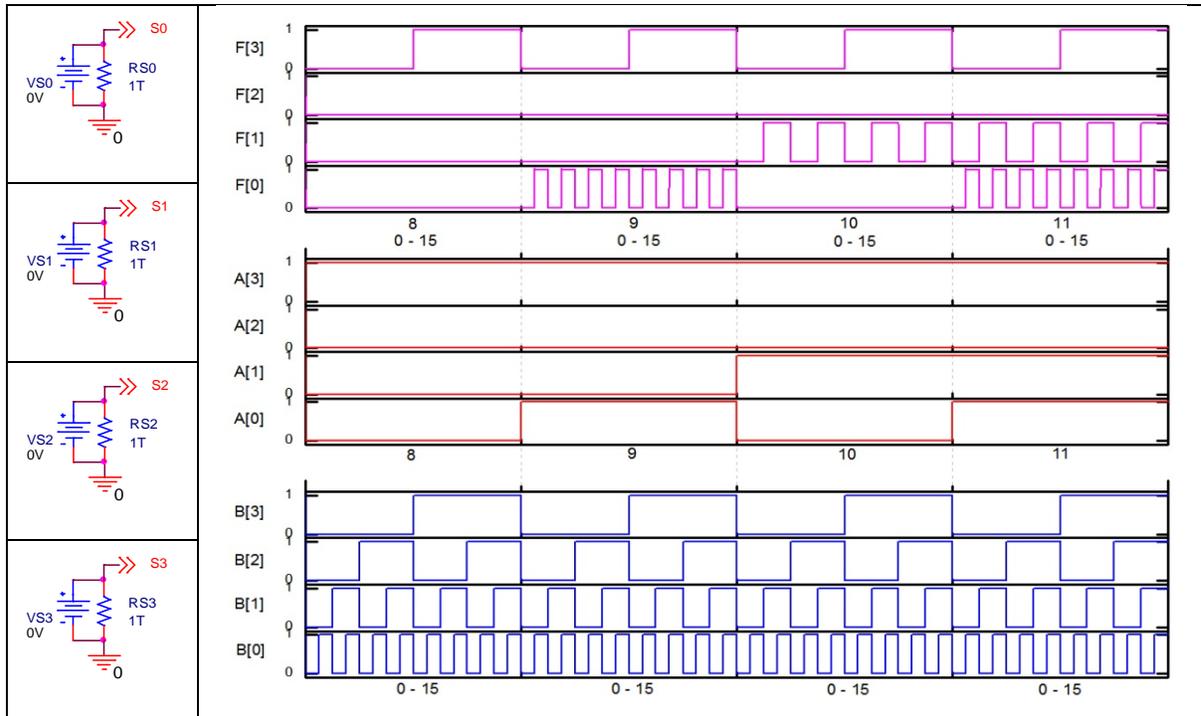


Figura 5.3. Resultado de la implementación $A \bullet B$ de la ALU SHL cuando 'S' = "0000".

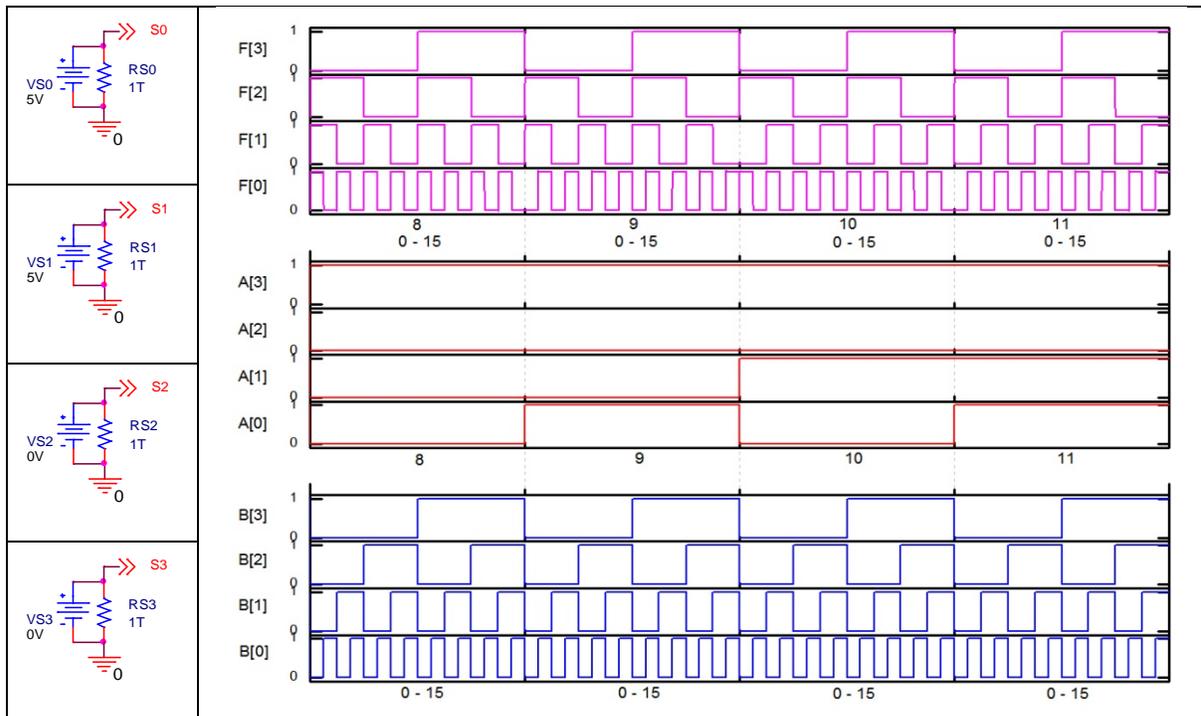


Figura 5.4. Resultado de la implementación $A \oplus B$ de la ALU SHL cuando 'S' = "0011".

Diseño de una ALU empleando Lógica de Configuración Externa.

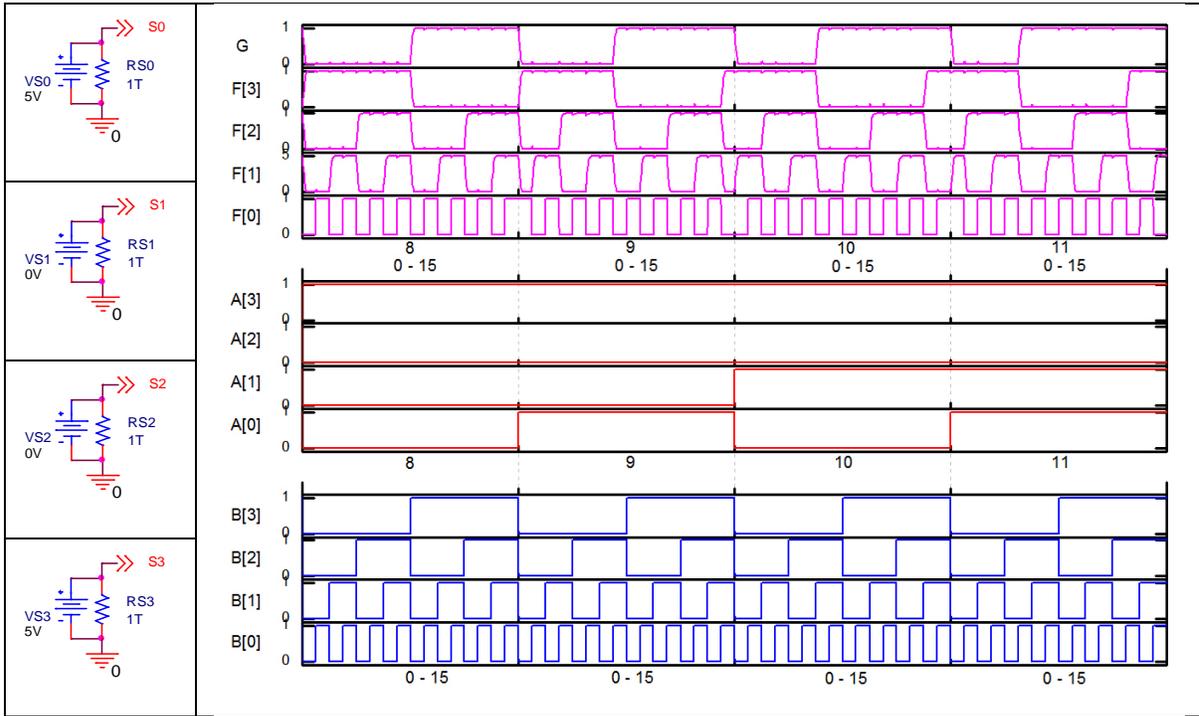


Figura 5.5. Resultado de la implementación $A + B$ de la ALU SHL cuando $'S' = "1001"$.

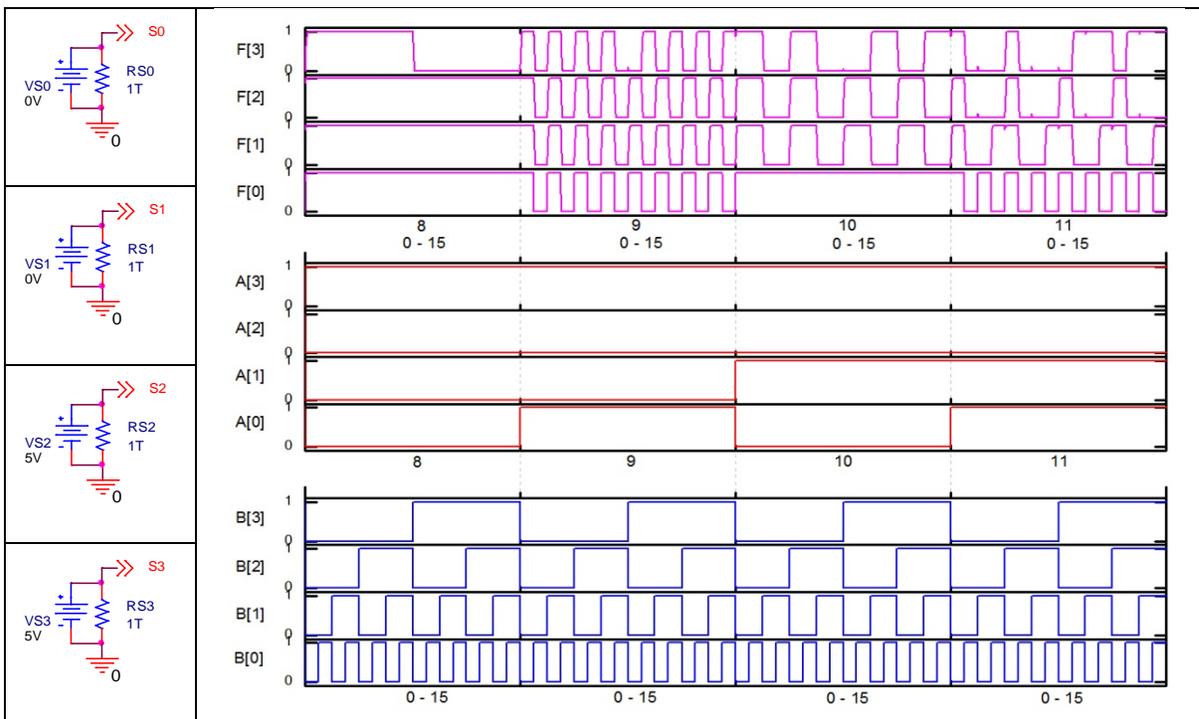


Figura 5.6. Resultado de la implementación $A \bullet B - 1$ de la ALU SHL cuando $'S' = "1100"$.

5.3 Chip Motorola MC14581B.

El circuito integrado o chip MC14581B es una ALU CMOS de 4 bits capaz de realizar 16 funciones de dos variables booleanas y 16 operaciones aritméticas binarias sobre palabras de 14 bits.

La forma en que el chip funciona es el siguiente:

El nivel de la entrada "Modo de Control" (MC) determina si la función de salida es lógica o aritmética.

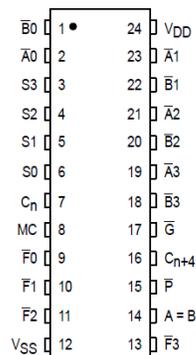
Una función lógica se selecciona al aplicar adecuadamente una palabra binaria a las entradas de selección de 4 bits 'S', cuando la entrada MC esta en nivel alto, mientras que una operación aritmética se selecciona al aplicar un voltaje bajo a la entrada MC, teniendo el nivel requerido para el acarreo de entrada (C_n) y la palabra correcta en las entradas de selección 'S', de acuerdo a la tabla 5.2, la cual es un extracto de la tabla de verdad completa del chip (anexo A) seleccionada aleatoriamente.

El chip incluye las salidas de acarreo propagado (\bar{P}) y acarreo generado (\bar{G}), las cuales proveen un esquema completo de acarreo anticipado (look-ahead carry) generando un rápido acarreo simultaneo sobre los 4 bits del paquete.

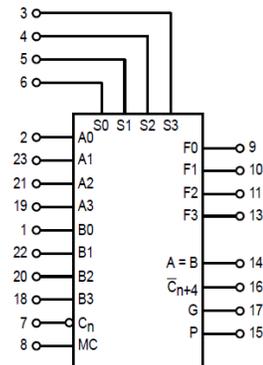
En la Figura 5.7, se muestra la asignación de pines del chip MC14581B (a) y la configuración activa en alto (b), que se empleo para definir el nivel de las palabras de entrada, así como el de las funciones de salida del chip para su tratamiento.

Tabla 5.2. Tabla de Verdad empleada en las mediciones del chip MC14581B.

Entradas de selección.				Entradas/ Salidas activas en alto	
S[3]	S[2]	S[1]	S[0]	Funciones Lógicas (MC = 1).	Funciones Aritméticas (MC = 0, $\bar{C}_n = 1$)
1	0	0	1	$\overline{A \oplus B}$	A + B
1	0	1	1	A • B	A • B - 1



(a)



(b)

Figura 5.7. (a) Asignación de Pines del IC y (b) Configuración de entrada/salidas activas en alto.

Las conexiones para poder realizar las mediciones en el chip, se muestran en la figura 5.8.

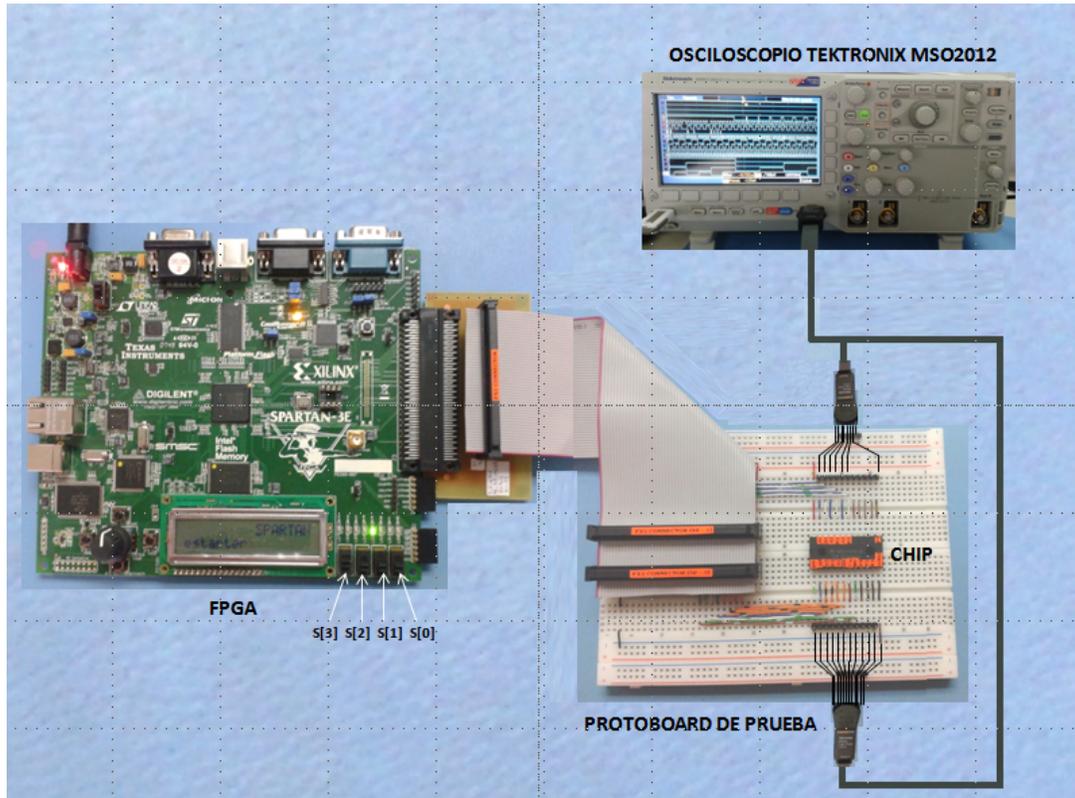


Figura 5.8. Conexiones para realizar la medición del chip.

A través del conector de Hirose FX2 de la tarjeta Spartan III, el FPGA alimenta las señales de excitación (palabras para A y B, así como MC) del chip de prueba, donde la operación a realizar se determina mediante los valores de la señal de control de 4 bits 'S', de acuerdo a la tabla 5.2, para finalmente en un osciloscopio mixto Tektronix realizar la captura de las formas de onda de salida del chip, F[0-3] y G, respectivamente

Cabe recordar que la salida G (acarreo generado), tiene sentido solamente cuando se realiza la operación de suma completa (A + B), por lo tanto los oscilogramas de las demás operaciones no contendrán dicha señal.

El FPGA genera las señales de excitación mediante el programa desarrollado en código VHDL que se muestra en la figura 5.9, el cual realiza principalmente las siguientes funciones.

1. Generar el nivel de entrada de MC automáticamente (MC=0 / MC=1), de acuerdo a la tabla 5.2.
2. Realizar un conteo binario de 0 a 15 para el bloque de palabra A y un sub-conteo de 0 a 15 por cada cuenta de A, para el bloque de palabra B, generando automáticamente las 256 posibles combinaciones de entrada al chip.

- Permitir la asignación manual de la palabra de selección de 4 bits, 'S', mediante los switches deslizables SW0 al SW3 de la tarjeta Spartan III, como se muestra en la figura 5.8.

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_UNSIGNED.ALL;
4  use IEEE.STD_LOGIC_ARITH.ALL;
5  use IEEE.NUMERIC_STD.ALL;
6  entity ALU is
7      port ( clk, reset: in std_logic;
8            A,B,S,sel: inout std_logic_vector (4 downto 0);
9            MC, Cn: inout std_logic);
10 end ALU;
11
12 architecture Arq_ALU of ALU is
13     signal clkdiv: std_logic_vector (12 downto 0);
14
15     begin
16
17         Cn <= '1';           -- Declaración de la configuración del IC: Activo en alto.
18         S <= sel;          -- Asignación de la señal de los switches SW0 - SW3 a la palabra de control 'S' de 4 bits.
19
20         process (reset,clk,clkdiv) -- Generador de las señales de sincronización del reloj F = 50MHz, T = 20ns.
21         begin
22             if reset = '1' then
23                 clkdiv <= (others => '0');
24             elsif clkdiv = X"1FFF" then
25                 clkdiv <= (others => '0');
26             elsif clk'event and clk = '1' then
27                 clkdiv <= clkdiv + 1;
28             end if;
29         end process;
30
31         process (reset,clkdiv(12),A) -- Contador principal para A [0-15]
32         begin
33             if reset = '1' then
34                 A <= (others => '0');
35             elsif clkdiv(12)'event and clkdiv(12) = '0' then
36                 A <= A + 1;
37             end if;
38         end process;
39
40         process (reset,clkdiv(8),B) -- Contador secundario para B [0-15] por cada cuanta de A.
41         begin
42             if reset = '1' then
43                 B <= (others => '0');
44             elsif clkdiv(8)'event and clkdiv(8) = '0' then
45                 B <= B + 1;
46             end if;
47         end process;
48
49         process (reset,MC,A) -- Generador de la señal de control MC
50         begin
51             if reset = '1' then
52                 MC <= '0'; -- Implementación de la Función Aritmética.
53             elsif A > "01111" then
54                 MC <= '1'; -- Implementación de la Función Lógica.
55             elsif A = "00000" then
56                 MC <= '0';
57             end if;
58         end process;
59
60     end Arq_ALU;

```

Figura 5.9 Código VHDL empleado para generar las señales de excitación del FPGA.

La asignación de las terminales de la figura 5.10 (b), que se emplearon para interconectar las señales entre el FPGA (conector Hirose) y el chip de prueba (MC14581B), fueron definidas mediante el archivo de restricciones de usuario, mostrado en la figura 5.10 (a).

Diseño de una ALU empleando Lógica de Configuración Externa.

```

1 #####
2 ### SPARTAN-3E STARTER KIT BOARD CONSTRAINTS FILE
3 #####
4
5 NET "clk" LOC = "C9" | IOSTANDARD = LVCMOS33 ;
6 NET "reset" LOC = "K17" | IOSTANDARD = LVTTTL | PULLDOWN ;
7 #NET "reset" CLOCK_DEDICATED_ROUTE = FALSE;
8
9 # === Slide Switches (SW) ===
10 NET "Sel<0>" LOC = "L13" | IOSTANDARD = LVTTTL | PULLUP ;
11 NET "Sel<1>" LOC = "L14" | IOSTANDARD = LVTTTL | PULLUP ;
12 NET "Sel<2>" LOC = "H18" | IOSTANDARD = LVTTTL | PULLUP ;
13 NET "Sel<3>" LOC = "N17" | IOSTANDARD = LVTTTL | PULLUP ;
14
15 NET "S<0>" LOC = "B4" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
16 NET "S<1>" LOC = "D5" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
17 NET "S<2>" LOC = "A6" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
18 NET "S<3>" LOC = "E7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
19
20 NET "A<0>" LOC = "D7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
21 NET "A<1>" LOC = "E8" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
22 NET "A<2>" LOC = "F7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
23 NET "A<3>" LOC = "C5" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
24
25 NET "B<0>" LOC = "F8" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
26 NET "B<1>" LOC = "C7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
27 NET "B<2>" LOC = "B6" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
28 NET "B<3>" LOC = "A4" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
29
30 NET "MC" LOC = "F9" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
31 NET "Cn" LOC = "D11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
32

```

(a)

Señal	Localización	FPGA Pin
5.0 V	-	1
GND	-	46
S[0]	B4	6
S[1]	D5	8
S[2]	A6	10
S[3]	E7	12
A[0]	D7	14
A[1]	E8	17
A[2]	F7	13
A[3]	C5	9
B[0]	F8	16
B[1]	C7	15
B[2]	B6	11
B[3]	A4	7
MC	F9	18
Cn	D11	20

(b)

Figura 5.10. Archivo de restricciones de usuario (a) y asignación de pines en el FPGA (b) para el chip.

Por lo que de acuerdo a la tabla 5.2,

La salida de la operación $A \bullet B$, que se muestra en la figura 5.11, se obtuvo cuando 'S' = "1011" y MC = '1',

La salida de la operación $\overline{A \oplus B}$, que se muestra en la figura 5.12, se obtuvo cuando 'S' = "1001" y MC = '1'.

La salida de la operación $A + B$, que se muestra en la figura 5.13, se obtuvo cuando 'S' = "1001" y MC = '0'.

La salida de la operación $A \bullet B - 1$, que se muestra en la figura 5.14, se obtuvo cuando 'S' = "1011" y MC = '0'.

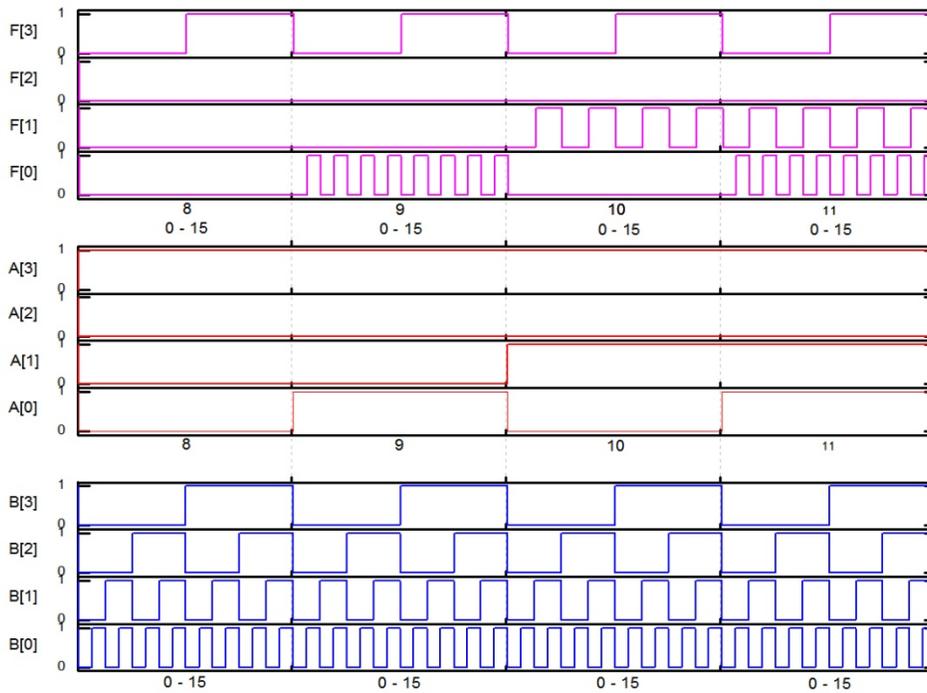


Figura 5.11. Resultado de la operación $A \bullet B$ del chip MC14581B.

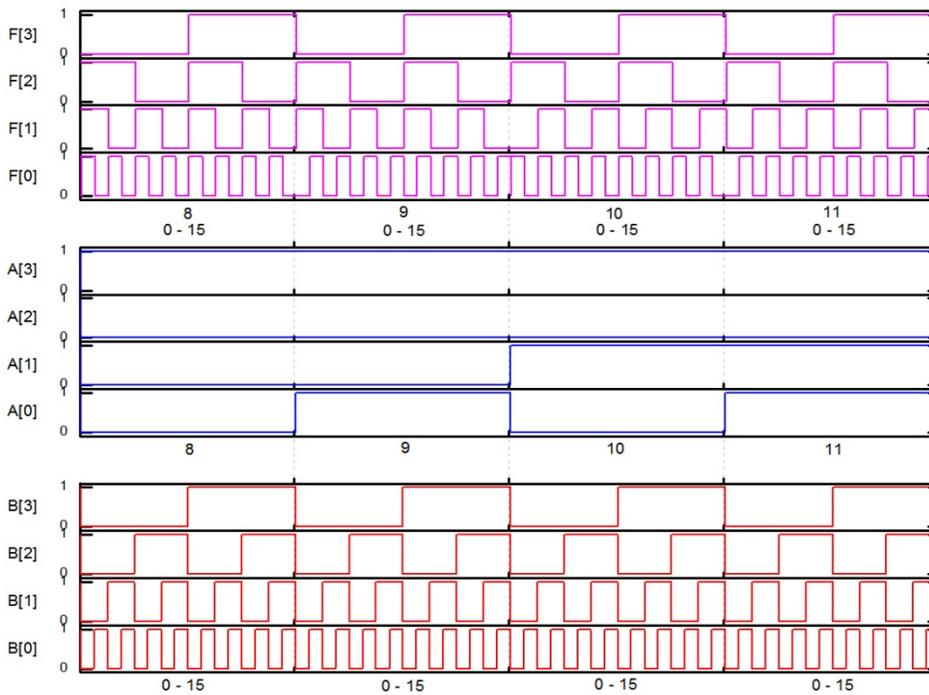


Figura 5.12. Resultado de la operación $A \oplus B$ del chip MC14581B.

Diseño de una ALU empleando Lógica de Configuración Externa.

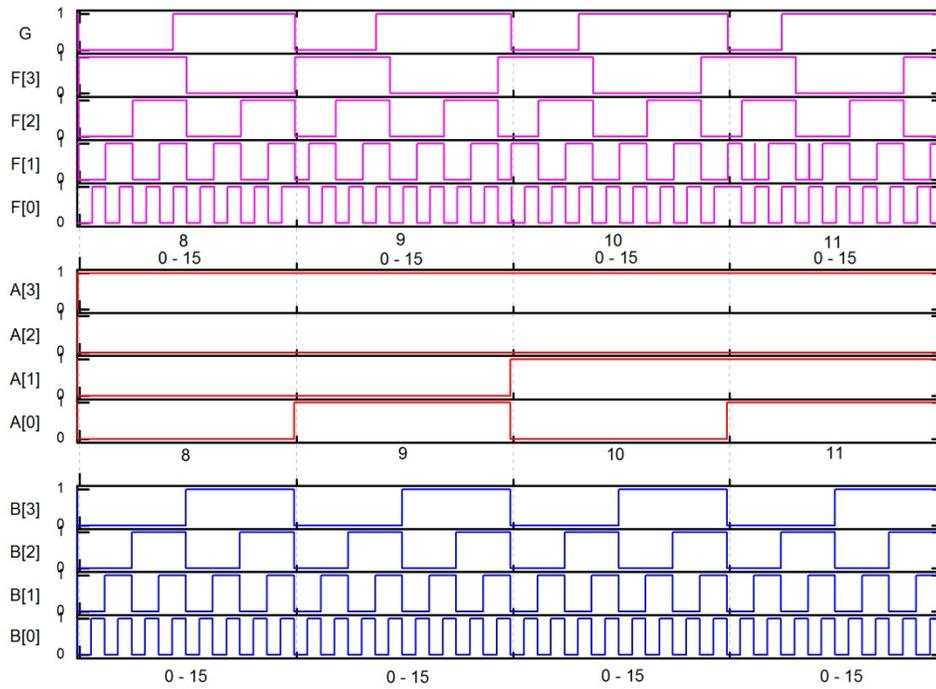


Figura 5.13. Resultado de la operación $A + B$ del chip MC14581B.

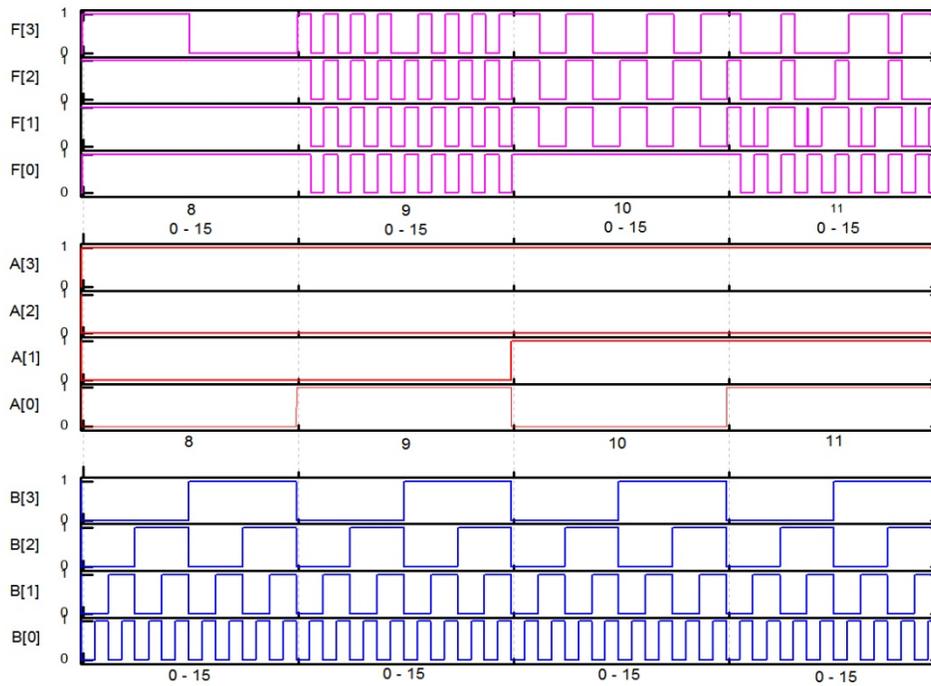


Figura 5.14. Resultado de la operación $A \cdot B - 1$ del chip MC14581B.

5.4 Implementación de una ALU en un FPGA.

Con la finalidad de obtener una referencia más de comparación para la ALU SHL, a demás de las mediciones del chip MC14581B, se realizo la implementación de las funciones de la tabla 5.2 en el FPGA de la tarjeta Spartan III mediante otro programa desarrollado en código VHDL, con la única diferencia de que la selección de la operación se determina a partir del valor de la señales de control 'S' y la señal 'MC', ambas de 1 bit, de acuerdo a la tabla 5.3.

Tabla 5.3. Tabla de Verdad empleada en la implementación de la ALU en un FPGA.

Entrada de selección.	Funciones Lógicas. (MC = 1)	Funciones Aritméticas. (MC = 0)
S = 0	$\overline{A \oplus B}$	A + B
S = 1	A • B	A • B - 1

Las conexiones para realizar las mediciones de la implementación del FPGA, se muestran en la figura 5.15, aquí el FPGA genera las señales de excitación internamente (palabras de 4 bits para A y B, así como MC), mientras que la función a realizar se determina mediante los valores de la señal de control 'S' de 1 bit de acuerdo a la tabla 5.3, para finalmente capturar las formas de onda de salida F[0-3] y G, respectivamente mediante un osciloscopio mixto Tektronix.

Cabe recordar que la salida G (acarreo generado), tiene sentido solamente cuando se realiza la operación de suma completa (A + B), por lo tanto las demás operaciones, no contendrán dicha señal.

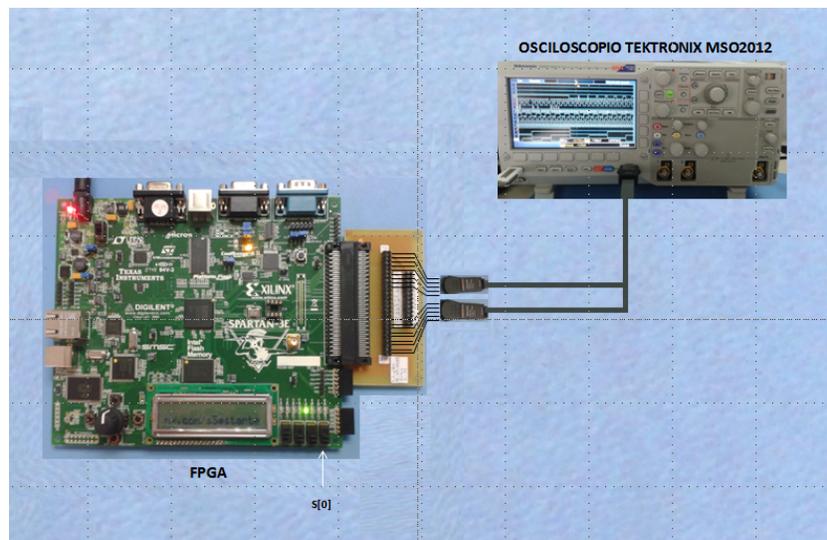


Figura 5.15. Conexiones empleadas para la medición del FPGA.

El programa que se muestra en la figura 5.16, básicamente realiza las siguientes funciones:

1. Generar el nivel de entrada de MC automáticamente (MC=0 / MC=1), de acuerdo a la tabla 5.3.

2. Realizar un conteo binario de 0 a 15 para el bloque de palabra A y un sub-conteo de 0 a 15 por cada cuenta de A, para el bloque de palabra B, generando automática e internamente las 256 posibles combinaciones de entrada.
3. Realiza las operaciones de la tabla 5.3, permitiendo la asignación manual del valor de la señal de selección 'S', mediante el switch deslizable SW0 de la tarjeta Spartan III, como se muestra en la figura 5.15.

```

1  library ieee;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_UNSIGNED.ALL;
4  use IEEE.STD_LOGIC_ARITH.ALL;
5  use IEEE.NUMERIC_STD.ALL;
6
7  entity automatic_MC14581 is
8      port (clk, reset: in std_logic;
9            A,B: inout std_logic_vector (4 downto 0);
10           sel : inout std_logic;
11           MC: inout std_logic;
12           F : inout std_logic_vector (4 downto 0));
13  end automatic_MC14581;
14
15
16  architecture arq of automatic_MC14581 is
17  signal clkdiv: std_logic_vector (12 downto 0);
18  signal C: std_logic_vector (4 downto 0);
19  begin
20  -- Cn <= '1'; -- Asignacion de Cn = '0', Circuito en activo alto.
21  process (reset,clk,clkdiv) -- Generador de las seales de sincronizacion del reloj f = 50MHz, T = 20ns.
22  begin
23      if reset = '1' then
24          clkdiv <= (others => '0');
25      elsif clkdiv = X"1FFF" then
26          clkdiv <= (others => '0');
27      elsif clk'event and clk = '1' then
28          clkdiv <= clkdiv + 1;
29      end if;
30  end process;
31
32  process (reset,clkdiv(12),A) -- Contador (principal) para A [0-15]
33  begin
34      if (reset = '1' or A > "1111") then
35          A <= (others => '0');
36      elsif clkdiv(12)'event and clkdiv(12) = '0' then
37          A <= A + 1;
38      end if;
39  end process;
40
41  process (reset,clkdiv(8),B) -- Contador (secundario) para B [0-15]
42  begin
43      if ( reset = '1' or B > "1111") then
44          B <= (others => '0');
45      elsif clkdiv(8)'event and clkdiv(8) = '0' then
46          B <= B + 1;
47      end if;
48  end process;
49
50  process (reset,clkdiv(12),C,MC) -- contador de apoyo de 0-31.
51  begin
52      if reset = '1' or C > "11111" then
53          C <= (others => '0');
54      elsif clkdiv(12)'event and clkdiv(12) = '0' then
55          C <= C + 1;
56      end if;
57  end process;

```

```

58
59 process (reset,MC,C) -- Generador de señal de control MC
60 begin
61     if reset = '1' or C = "0000" then
62         MC <= '0';
63     elsif C > "01111" then
64         MC <= '1';
65     end if;
66 end process;
67
68 process (A,B,F,SEL,MC,F)
69 begin
70     -- sel <= '0';
71     F <= "00000";
72     if (MC = '1') then --logic function--
73         case SEL is
74             when '0' => F <= A XNOR B;
75             when '1' => F <= A AND B;
76             when others => F <= "00000";
77         end case;
78     else if (MC = '0') then --arithmetic unit --
79         case SEL is
80             when '0' => F <= A + B;
81             when '1' => F <= (A AND B) - 1;
82             when others => F <= "00000";
83         end case;
84     end if;
85 end process;
86
87
88 end arq;
89
90

```

Figura 5.16. Programa en VHDL para las mediciones de la ALU implementada por el FPGA.

La asignación de las terminales del conector Hirose de la tarjeta Spartan III, figura 5.17(b), que se emplearon para realizar las mediciones de la ALU implementada mediante el FPGA, fueron definidas mediante el archivo de restricciones de usuario, mostrado en la figura 5.17 (a).

```

1 #####
2 ### SPARTAN-3E STARTER KIT BOARD CONSTRAINTS FILE
3 #####
4
5 NET "clk" LOC = "C9" | IOSTANDARD = LVCMOS33 ;
6 NET "reset" LOC = "K17" | IOSTANDARD = LVTTTL | PULLDOWN ;
7 #NET "reset" CLOCK_DEDICATED_ROUTE = FALSE;
8
9 # === Slide Switches (SW) ===
10 NET "sel" LOC = "L13" | IOSTANDARD = LVTTTL | PULLUP ;
11 #NET "SW<1>" LOC = "L14" | IOSTANDARD = LVTTTL | PULLUP ;
12 #NET "SW<2>" LOC = "H18" | IOSTANDARD = LVTTTL | PULLUP ;
13 #NET "SW<3>" LOC = "N17" | IOSTANDARD = LVTTTL | PULLUP ;
14
15 NET "A<0>" LOC = "B4" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
16 NET "A<1>" LOC = "A4" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
17 NET "A<2>" LOC = "D5" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
18 NET "A<3>" LOC = "C5" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
19 NET "A<4>" LOC = "A6" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
20
21 NET "B<0>" LOC = "B6" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
22 NET "B<1>" LOC = "E7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
23 NET "B<2>" LOC = "F7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
24 NET "B<3>" LOC = "D7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
25 NET "B<4>" LOC = "C7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
26
27 NET "F<0>" LOC = "F9" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
28 NET "F<1>" LOC = "E9" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
29 NET "F<2>" LOC = "D11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
30 NET "F<3>" LOC = "C11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
31 NET "F<4>" LOC = "F11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
32
33
34 NET "MC" LOC = "F8" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
35
36 #NET "Cn" LOC = "D11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
37

```

(a)

Señal	Localización	FPGA Pin
5.0 V	-	1
GND	-	46
A[0]	B4	6
A[1]	A4	7
A[2]	D5	8
A[3]	C5	9
B[0]	B6	11
B[1]	E7	12
B[2]	F7	13
B[3]	D7	14
F[0]	F9	18
F[1]	E9	19
F[2]	D11	20
F[3]	C11	21
G	F11	22
MC	F8	16

(b)

Figura 5.17. Archivo de restricciones de usuario (a) y asignación de pines en el FPGA (b).

Diseño de una ALU empleando Lógica de Configuración Externa.

Por lo que al programar una memoria flash del FPGA, se obtuvieron las siguientes mediciones:

La salida de la operación $A \bullet B$, que se muestra en la figura 5.18, se obtuvo colocando 'S' = '1' y MC = '1',

La salida de la operación $\overline{A \oplus B}$, que se muestra en la figura 5.19, se obtuvo cuando 'S' = '0' y MC = '1'.

La salida de la operación $A + B$, que se muestra en la figura 5.20, se obtuvo cuando 'S' = '0' y MC = '0'.

La salida de la operación $A \bullet B - 1$, que se muestra en la figura 5.21, se obtuvo cuando 'S' = '1' y MC = '0'.

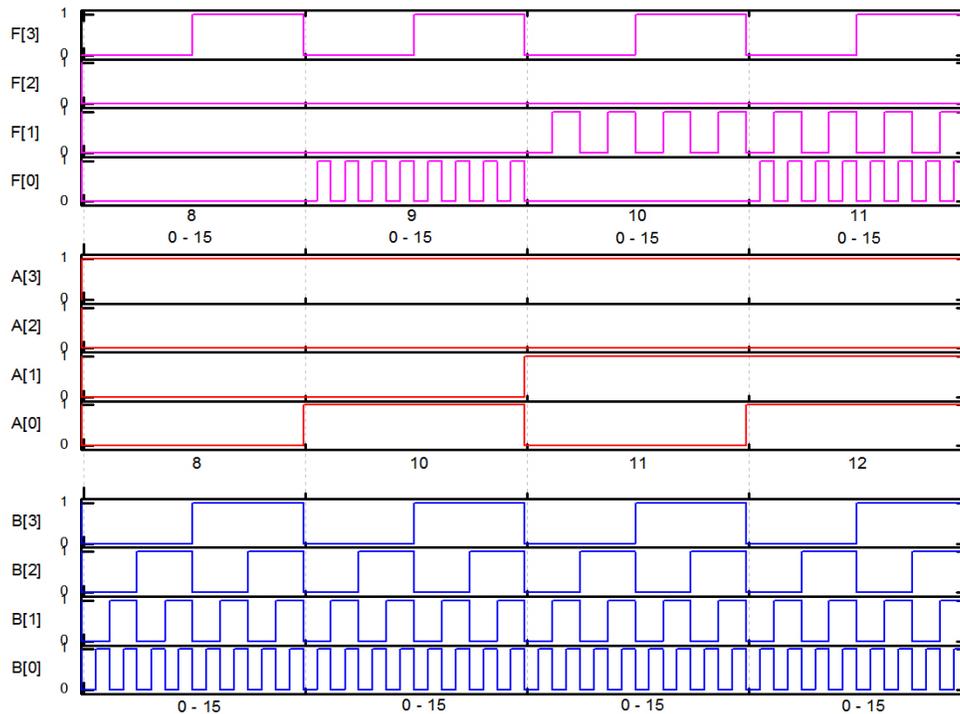


Figura 5.18. Resultado de la operación $A \bullet B$ de la ALU implementada en el FPGA.

Diseño de una ALU empleando Lógica de Configuración Externa.

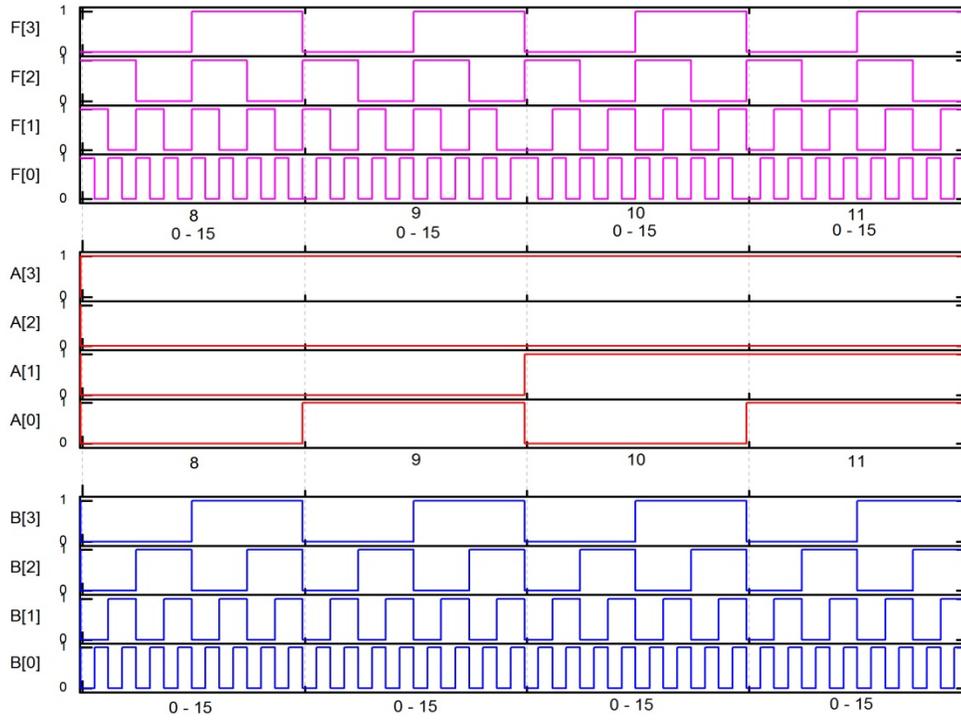


Figura 5.19. Resultado de la operación $\overline{A \oplus B}$, de la ALU implementada en el FPGA.

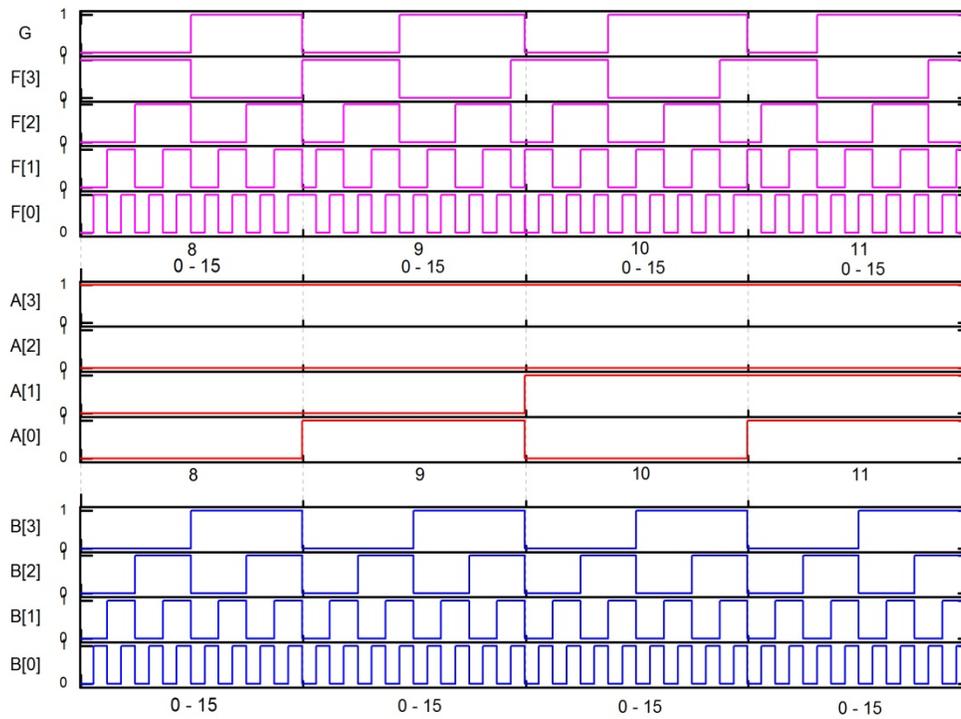


Figura 5.20. Resultado de la operación $A + B$ de la ALU implementada en el FPGA.

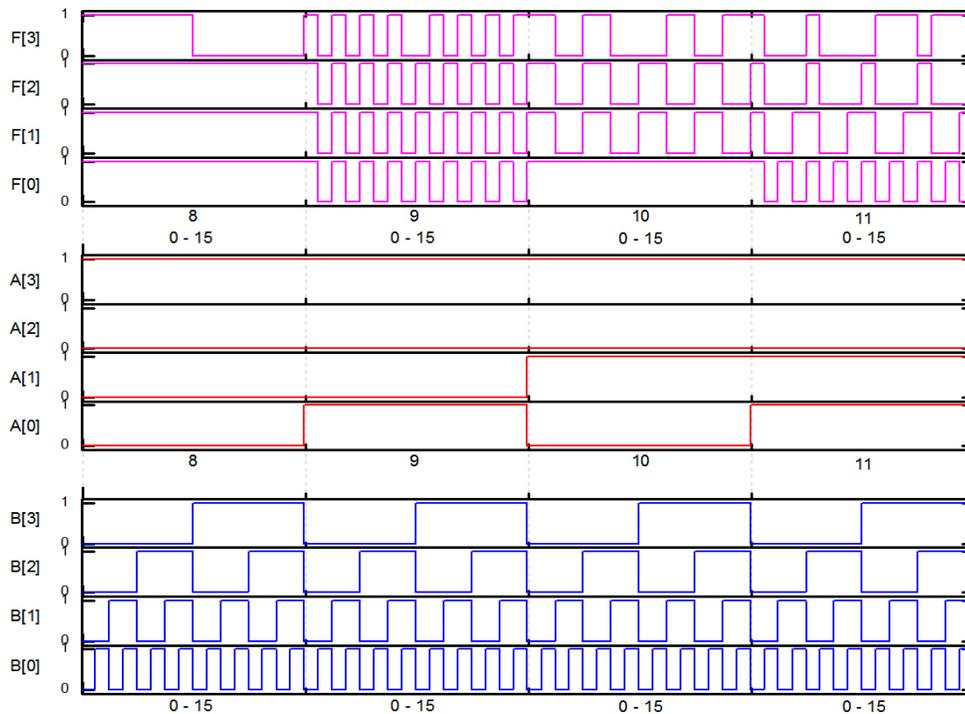


Figura 5.21. Resultado de la operación $A \bullet B - 1$ de la ALU implementada en el FPGA.

5.5 Comparación de resultados entre las ALU's: SHL, CHIP y FPGA.

A continuación, se presentan las comparaciones entre los resultados obtenidos para la simulación en PSpice de la ALU SHL (propuesta en este trabajo) y los resultados obtenidos de las mediciones realizadas al chip MC14581B (ALU 4-bits CMOS comercial) y la implementación de la ALU mediante un FPGA (Spartan III).

Las figuras 5.22, 5.23, 5.24 y 5.25, muestran las comparaciones de los resultados de las operaciones $A \bullet B$, $\overline{A \oplus B}$, $A + B$ y $A \bullet B - 1$, respectivamente para cada una de las ALUs evaluadas: SHL, Chip (MC14581B) y FPGA.

Cabe resaltar que todas las simulaciones presentadas en este capítulo, se realizaron a una frecuencia de 826.44 KHz, la cual se definió a partir los tiempos de retraso de propagación típicos (t_{PHL} y t_{PLH}) reportados en las hojas de especificación del IC MC14581B (anexo A), como se muestra a continuación.

$$f = \frac{1}{t_{PLH} + t_{PHL}} = \frac{1}{605 \text{ ns} + 605 \text{ ns}} = 826.446 \text{ KHz.}$$

Diseño de una ALU empleando Lógica de Configuración Externa.

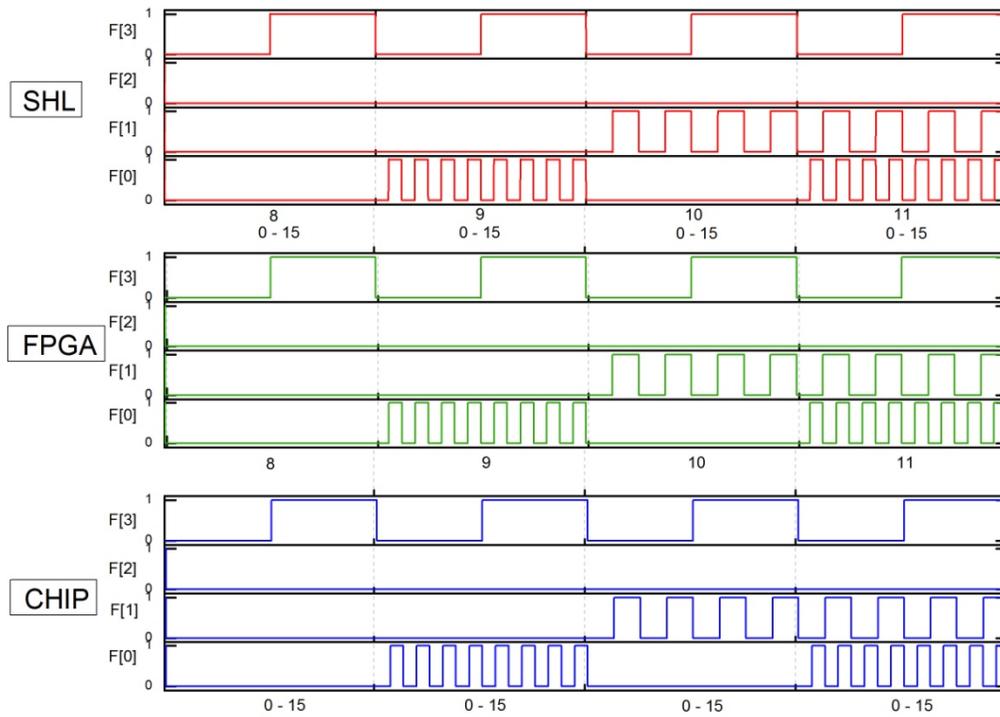


Figura 5.22. Comparativo de los resultados de la operación $A \cdot B$ entre las tres ALUs.

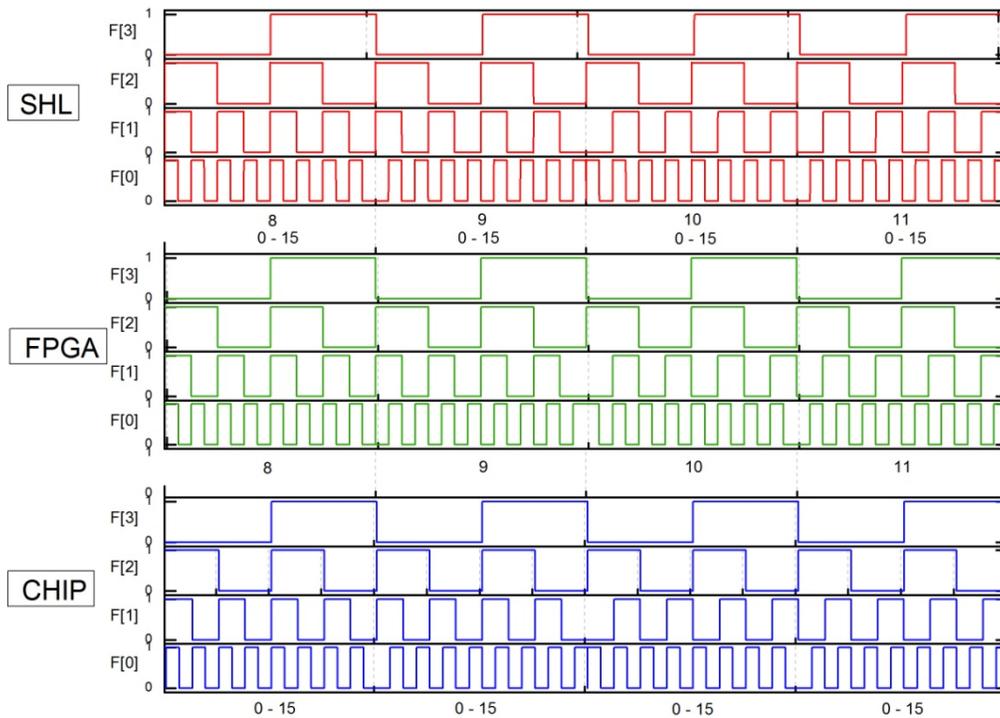


Figura 5.23. Comparativo de de los resultados de la operación $A \oplus B$ entre las tres ALUs.

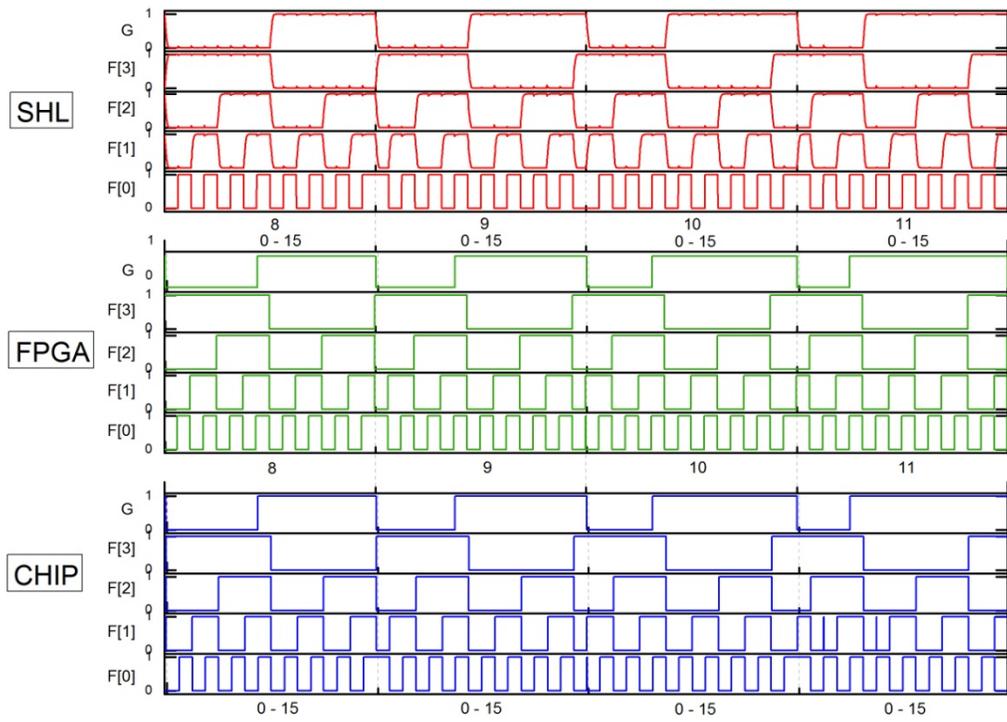


Figura 5.24. Comparativo de los resultados de la operación $A + B$ entre las tres ALUs.

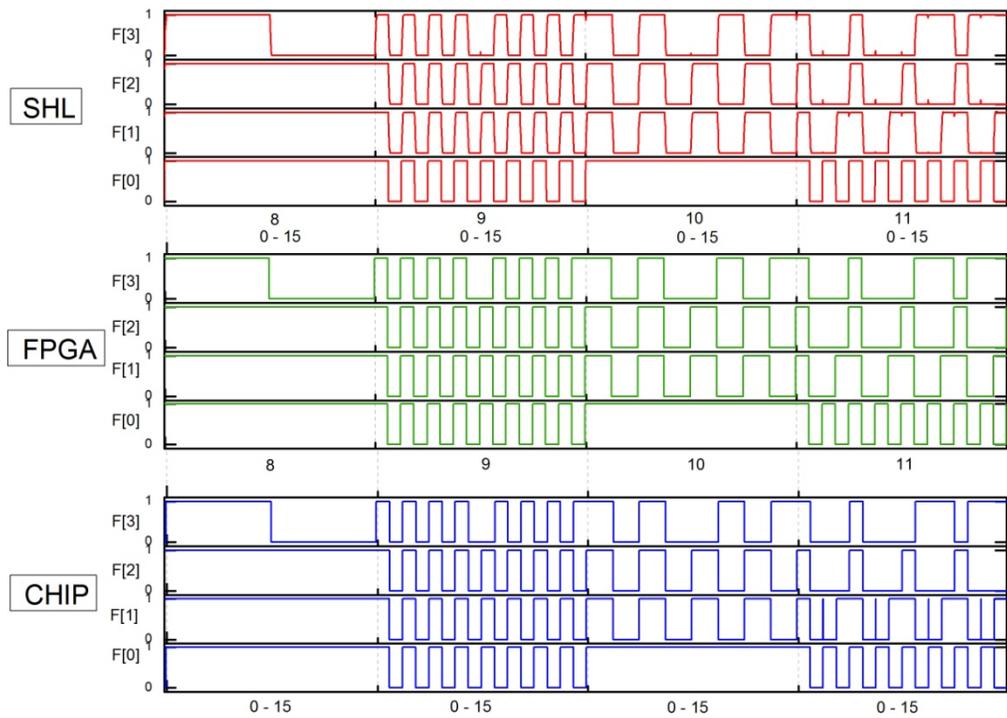


Figura 5.25. Comparativo de los resultados de la operación $A \cdot B - 1$ entre las tres ALUs.

Como se puede observar en las figuras 5.22, 5.23, 5.24 y 5.25, el comportamiento de las operaciones que puede realizar la ALU SHL, es similar al mostrado por el IC MC14581 y por la implementación realizada en el FPGA.

Con lo cual, se demuestra la alta viabilidad del uso de celdas básicas SHL para la implementación de funciones de n bits, más allá de la mostrada en este trabajo (una ALU básica de 4 bits).

Sin embargo, aun existe bastante trabajo por realizar, ya que como también se puede observar en las fig. 5.24 y 5.25, hay presencia de "glitches", causados principalmente por los retrasos de la señal debido a la frecuencia de operación empleada y en menor grado por al arreglo de la celda básica, disminuyendo la precisión en los resultados mostrados.

Por lo que, en un futuro, será necesario realizar un análisis en frecuencia para determinar el rango de operación de la ALU SHL propuesta en este trabajo.

5.6 Conclusiones.

En este capítulo se ha demostrado que la versatilidad del bloque básico SHL, cuyo desarrollo está basado en la celda SHL de los capítulos anteriores, se puede emplear para la generación de cualquier función básica de 4 bits y que con la correcta combinación de bloques, se puede ampliar el rango de posibilidades, como aquí se expuso.

Como se había mencionado, un análisis en frecuencia es necesario, para conocer los límites de operación del bloque SHL y en nuestro caso en particular de la ALU SHL. Sin embargo de los resultados que se presentaron en este capítulo, se puede inferir que es menor al de la tecnología comercial existente.

Aun así, en lo que se refiere a la reducción de transistores, nuestro diseño SHL (basado en el FPD) emplea menos que la tecnología comercial, dando como resultado una diferencia de 400 transistores aproximadamente.

Capítulo VI. Layout de la ALU SHL de 4 bits.

6.1 Introducción.

En este capítulo se presenta el diseño geométrico o layout de cada uno de los componentes de los bloques que integran la ALU de 4 bits presentada en esta tesis. El layout se basó en las reglas de diseño especificadas por la compañía MOSIS para un proceso tecnológico de 0.5 micras ($\lambda=0.30\mu\text{m}$), cuyas descripciones pueden consultarse directamente en el archivo SCN3M_SUBM proporcionado en la página web de Mosis: <http://www.mosis.org> o bien en el archivo Mamis05 proporcionado en la página web del fabricante de la herramienta de L-edit empleada en el diseño del layout: <http://tanner.com.ces>.

El proceso tecnológico estándar dispone de un pozo N CMOS escalable, tres niveles de metal y la capacidad de agregar una segunda capa de polisilicio (poly2) como el electrodo superior de un capacitor poly [1].

El paquete de herramientas de TannerTM que contiene al programa L-edit, permite la generación de mascararas de diseño y su edición, así como la verificación de las reglas de diseño (DRC) y la vista en corte transversal de las mascararas diseñadas (L-Edit/XST).

A continuación se muestra la información de las capas, relleno y contornos utilizados en los diseños de este capítulo [2].

	Overglass		Metal2		Poly Contact
	Icon/Outline		Poly2		Pad Comment
	N Well		N Select		Poly2 Contact
	Metal3		Via2		Via1
	Poly		P Select		Active Contact
	Error Layer		Active		Metal1

6.2 Diseño Geométrico de los Inversores CMOS de la celda básica SHL.

La figura 6.1(a) y 6.1(b) muestran los transistores PMOS y NMOS empleados en los inversores CMOS de las etapas de Precarga y Neurona (PMOS=5.1 μm /0.6 μm , NMOS=3 μm /0.6 μm), así como de la etapa de Inversores Programables (PMOS=5.7 μm /0.6 μm , NMOS=3 μm /0.6 μm) de la celda básica SHL de 4 bits de esta tesis.

Aquí cabe mencionar que un dispositivo MOS canal-N, está basado en una compuerta de polisilicio depositada sobre la superficie de un substrato tipo P, aislada por una capa ultra delgada de óxido (llamada óxido de compuerta), y una implantación N+ que forma dos difusiones eléctricamente separadas en ambos lados de la compuerta.

Mientras que un dispositivo MOS canal-P, se construye usando polisilicio como material de compuerta y una difusión P+ para construir la Fuente y el Drenador, el dispositivo PMOS requiere de la adición de un

pozo tipo N sobre el polisilicio y las capas de difusión. Sin embargo la difusión P+ debe estar completamente dentro de del pozo N [3].

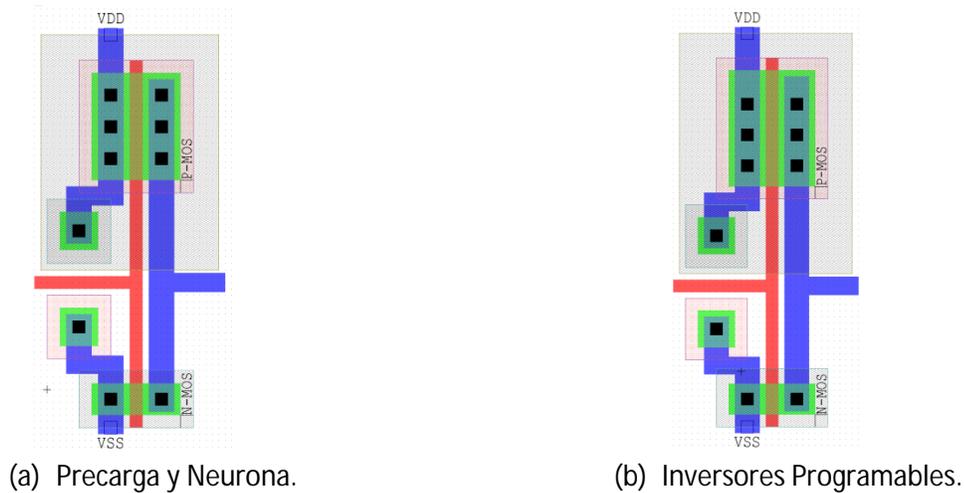


Figura 6.1. Diseño de los Inversores CMOS de las etapas de precarga y Neurona (a) e Inversores Programables (b) de la celda básica SHL.

6.3 Diseño del circuito de Precarga.

La figura 6.2 muestra el circuito de pre-carga que se emplea para convertir las señales binarias de entrada en la señal multivaluada, V_p , la cual alimentara a su vez a las etapas de inversores programables y Neurona, respectivamente.

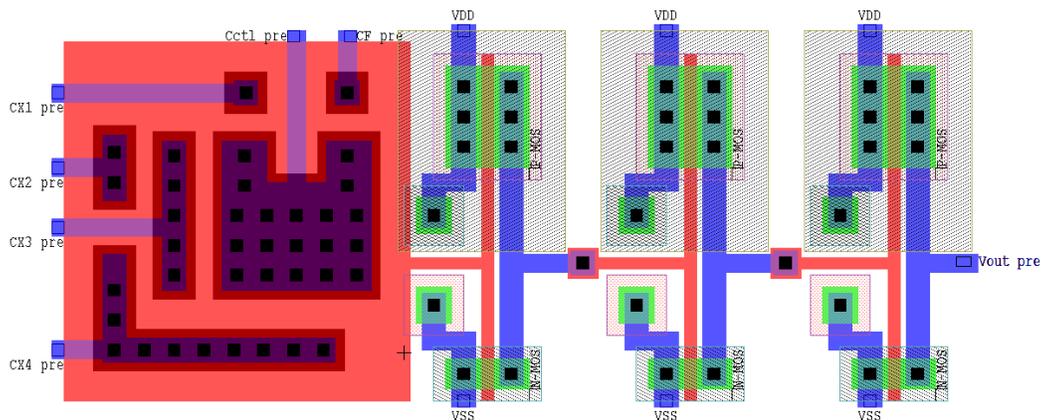


Figura 6.2. Diseño del circuito de Pre-carga.

Aquí se pueden apreciar los coeficientes de acoplamiento C_{X1} , C_{X2} , C_{X3} , C_{X4} , $C_{CTL PRE}$ y C_F que corresponden a las cuatro compuertas de señal de 4 bits de entrada, una compuerta de control, empleada en el ajuste de la salida de la etapa de precarga ($V_{OUT PRE}$) y a la entrada del inversor programable F, respectivamente. Así mismo se observan los dos inversores convencionales encargados de hacer más abrupta la salida de la etapa, como se explica en la sección 2.3 de esta tesis.

Los coeficientes de acoplamiento mostrados, se construyeron empleando diferentes polisilicios, poly2 sobre poly1, cuya área de traslape define el valor del coeficiente capacitivo.

6.4 Diseño del circuito de Neurona para los bloques Suma-Acarreo y F1-F2.

La figuras 6.3 y 6.4 muestran los diseños correspondientes a la etapa de Neurona (NeuMOS) para las celdas de los sub-bloques de suma-acarreo y de los sub-bloques F1-F2 respectivamente.

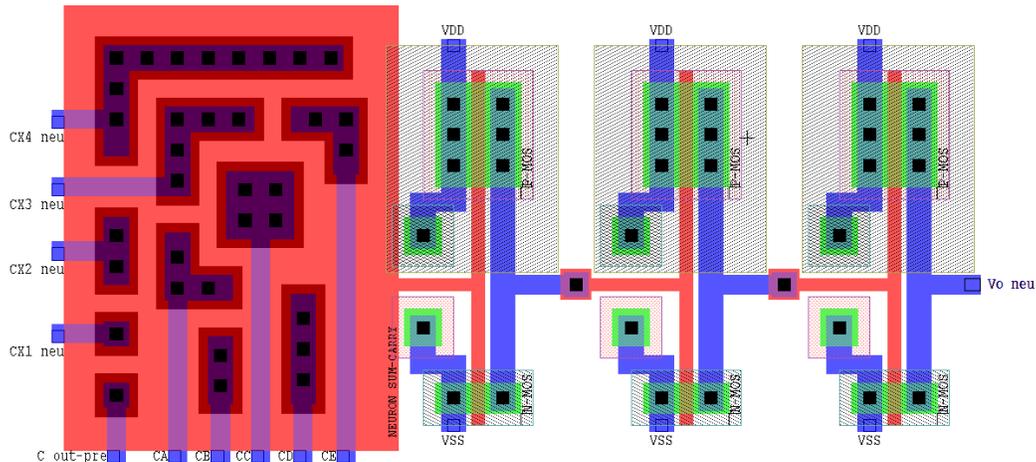


Figura 6.3. Diseño del circuito de Neurona de los sub-bloques de SUMA y f_{CARRY} .

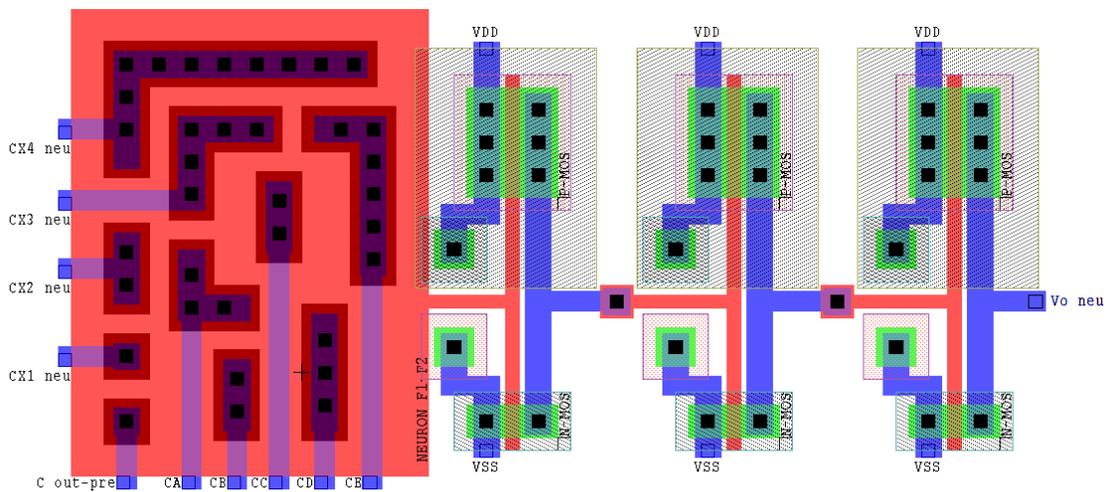


Figura 6.4. Diseño del circuito de Neurona de los sub-bloques F1 y F2.

Se puede observar que la señal de Entrada de 4 bits, también está acoplada mediante C_{X1} , C_{X2} , C_{X3} y C_{X4} a la compuerta flotante de un inversor vMOS complementario, cuya salida alimenta 3 inversores convencionales, los cuales son los encargados de hacer mas abruptas las conmutaciones de la señal V_O NEU, y de generar el FPD del Circuito Lógico deseado. Los nodos etiquetados con C_A , C_B , C_C , C_D , C_E y $C_{OUT PRE}$, corresponden a la salidas de los inversores A, B, C, D y E de la etapa de inversores programables y la salida de la etapa de precarga, respectivamente, como se vio en las secciones 2.4, 3.3 y 3.4 de esta tesis.

6.5 Diseño del circuito de Inversores Programables.

La figura 6.5 muestra el diseño de la etapa de inversores programables de la celda básica SHL, descrita en la sección 2.5 de esta tesis.

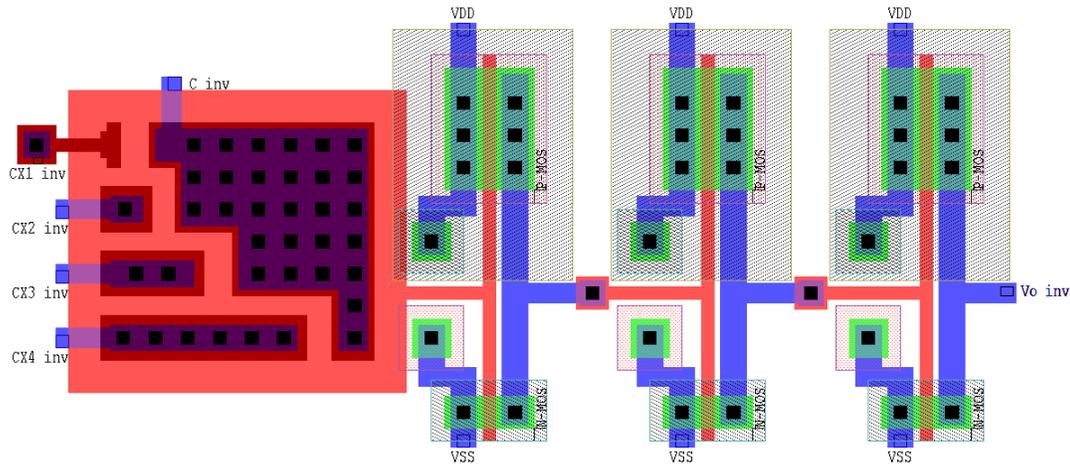


Figura 6.5. Diseño del circuito de Inversores Programables.

Como se observó en las etapas anteriores, las cuatro entradas de señal están directamente acopladas mediante C_{X1} , C_{X2} , C_{X3} y C_{X4} a la compuerta flotante de un inversor vMOS complementario, cuya salida alimenta 2 inversores convencionales, los cuales son los encargados de hacer más abruptas las transiciones de la señal de salida del inversor programable. El nodo etiquetado como C inv, corresponde a la entrada de voltaje externo necesario para lograr que el inversor conmute de acuerdo a lo establecido en el FPD, dependiendo de qué implementación se trate.

6.6 Diseño de los bloques de Sumador Completo y Selección-Complemento de la ALU.

A continuación las figuras 6.6 y 6.7 muestran los diseños correspondientes a los sub-bloques de Sumador Completo con propagación de acarreo (SUMA-Acarreo) y los sub-bloques de Selección-Complemento (F1-F2) que forman la ALU SHL de 4 bits, conforme a lo ilustrado en el capítulo 3.

6.7 Diseño final del chip de la ALU SHL de 4 bits.

La figura 6.8, presenta una propuesta del chip de la ALU SHL de 4 bits de acuerdo al diagrama presentado en la figura 3.1 del capítulo 3.

6.8 Conclusiones.

En este capítulo se presentaron los diseños geométricos correspondientes a cada una de las etapas que conforman la ALU SHL de 4 bits derivados del capítulo anterior.

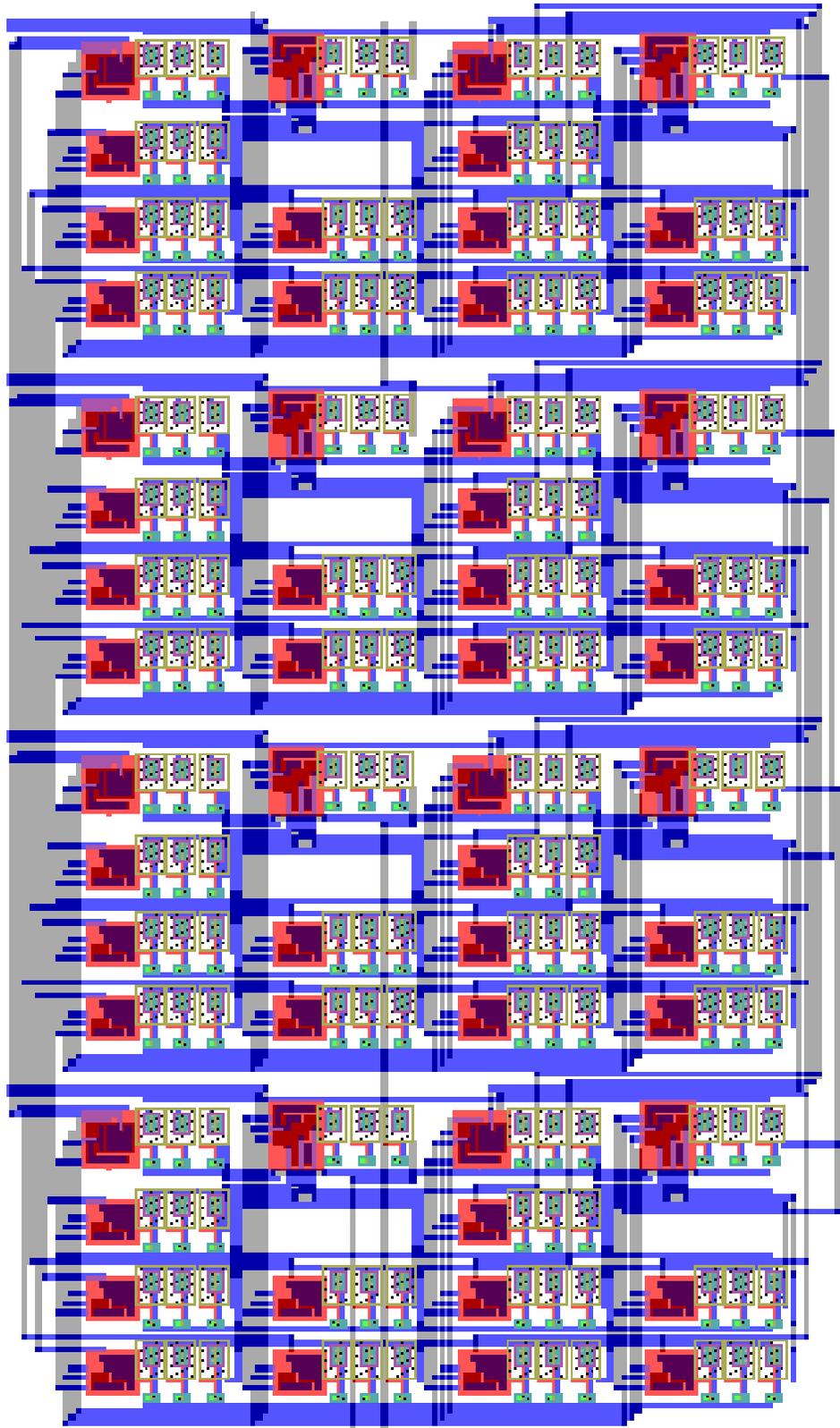


Figura 6.6. Diseño del Bloque de Sumador Completo (SUMA-Acarreo) de la ALU.

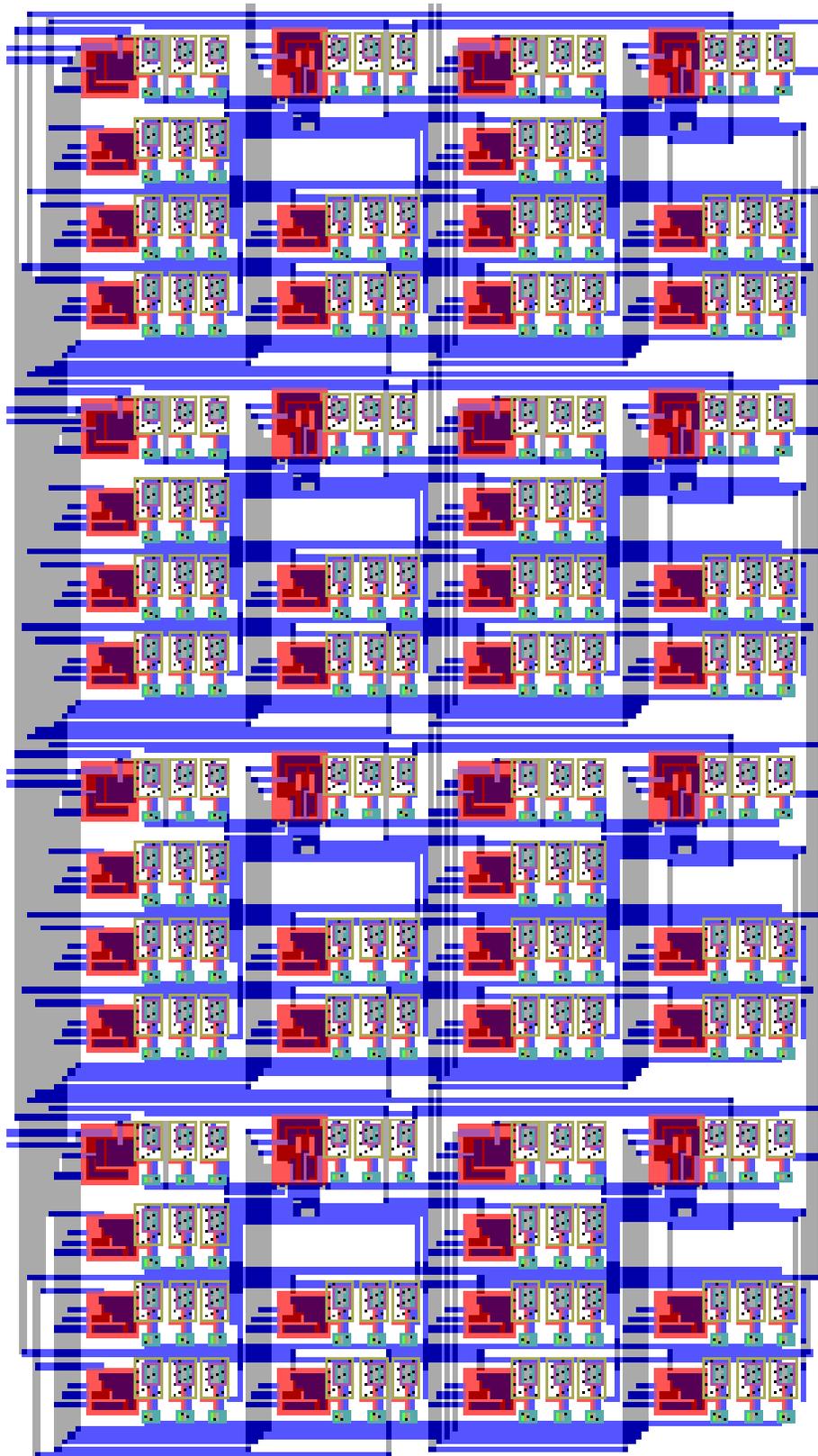


Figura 6.7. Diseño del Bloque Selección-Complemento (F1-F2) de la ALU.

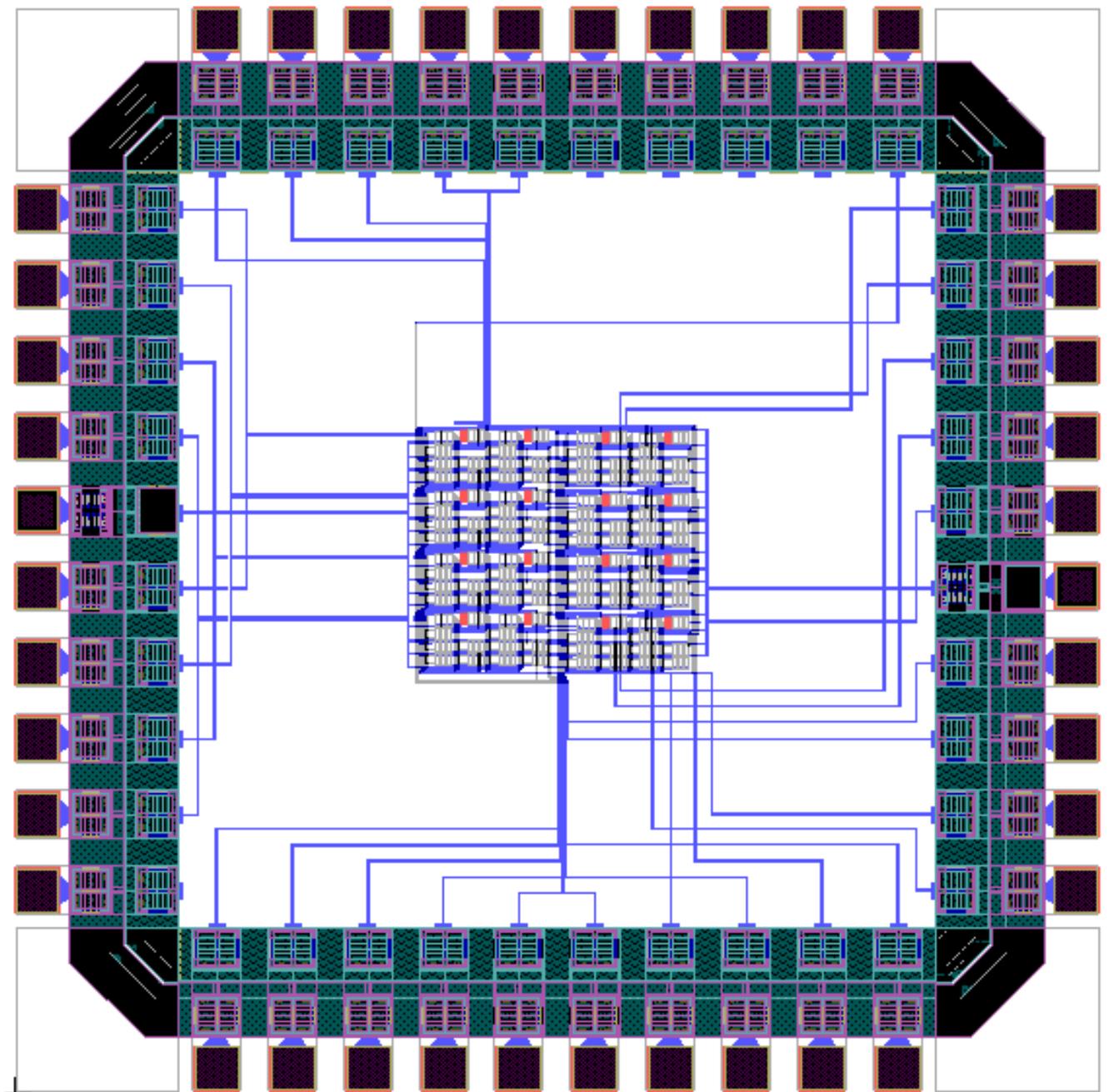


Figura 6.8. Diseño del chip de la ALU SHL de 4 bits.

6.9 Referencias.

- [1].
- [2]
- [3].

Capítulo VII. Conclusiones.

7.1. Conclusiones.

Es claro observar que el empleo de métodos alternativos en la concepción y diseño de circuitos integrados, como el que se presento y estudio en esta tesis, es decir el Diagrama de Potencial de Compuerta Flotante, es de suma importancia debido a la tendencia actual de reducción de área y por lo tanto a una mayor integración de elementos en un solo chip.

La metodología del FPD nos ofrece una herramienta sencilla para la obtención de cualquier función que se desee y muestra de ello, es la Unidad Aritmética Lógica de 4 bits que se desarrollo en el presente trabajo, la cual fue una aplicación específica de la celda básica SHL reconfigurable.

Sin embargo el inconveniente de emplear esta metodología, es la baja frecuencia de operación debida implícitamente a los valores tan amplios de los coeficientes capacitivos que se utilizaron, lo cual reduce la respuesta en frecuencia del dispositivo. En resumen, los resultados de la tesis se pueden destacar a través de los siguientes puntos:

- Se presentaron las características que conforman al diagrama de Potencial de Compuerta Flotante, a partir de la cual, es posible configurar cualquier función booleana, basada en un dispositivo muy versátil como el vMOS. El caso estudiado en la presente tesis correspondió a una ALU de cuatro bits de entrada, pero se puede extender a un número mayor.
- Se estableció una metodología a partir de la cual se pudo diseñar una Unidad Aritmética Lógica (ALU) de cuatro bits de entrada, pero además, se estableció una estrategia simple para decidir sobre la salida Lógica o Aritmética del bloque ALU, basado en bloques que se configuraban mediante niveles lógicos. Empleando esta misma metodología, se pueden diseñar más circuitos reconfigurables con los que se puedan realizar diversas funciones lógicas y para diferente número de bits de entrada, por lo tanto, lo desarrollado en la presente tesis sirve como base para diseños reconfigurables.
- Para determinar el desempeño de la ALU reconfigurable, se realizaron mediciones sobre un chip comercial de una ALU, así como sobre una ALU sintetizada mediante un FPGA Spartan III. La comparación hecha mostró que el diseño mediante SHL cumple con los requisitos, aunque con la limitante de trabajar a menor frecuencia debido a las capacitancias de acoplamiento que se emplean tanto en el bloque de precarga como en la etapa de la neurona.
- Por otro lado, una ventaja que se tiene con los circuitos basados en SHL, es que se puede simplificar su configuración, gracias a la reducción del número de transistores requeridos para implementar una función dada y su inherente reconfiguración a partir de la misma configuración, la cual se puede establecer únicamente a partir de la aplicación de voltajes externos.

7.2. Trabajos Futuros.

Del análisis realizado y los resultados derivados del desarrollo de esta tesis, se observo que se puede seguir investigando en los siguientes aspectos del MIFG MOS:

1. Particularmente en lo que se refiere a la celda básica, se puede desarrollar una celda universal SHL para implementar cualquier función de 4 bits al realizar una redistribución de los valores de los coeficientes de acoplamiento y el número de inversores programables como se muestra en la figura 8.1.

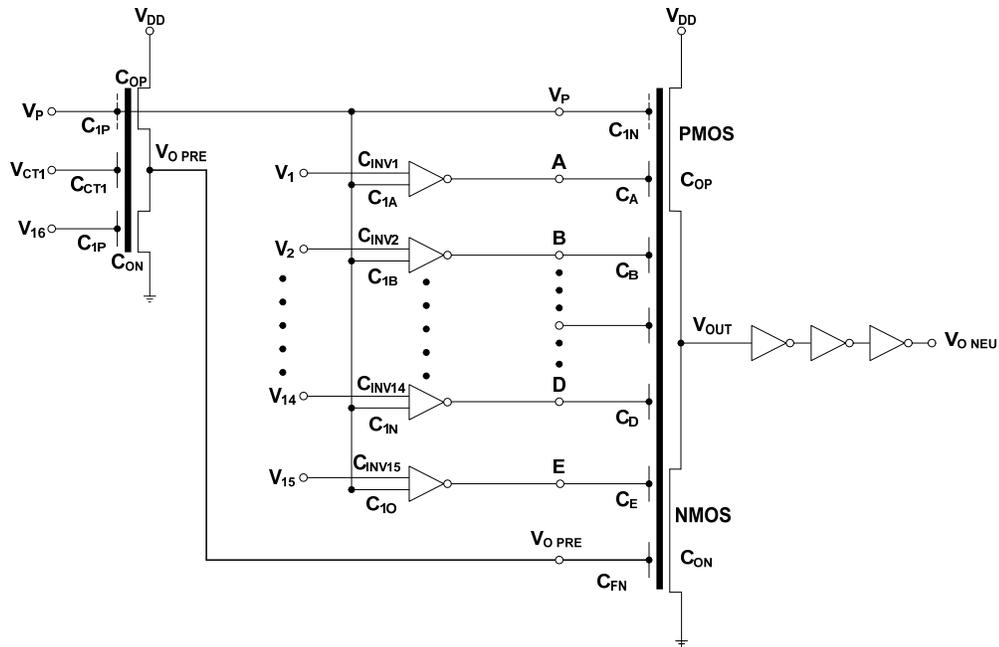


Figura 7.1. Diagrama a bloques de la celda básica universal SHL.

2. Análisis detallado en frecuencia.
3. Reducción o escalamiento de la alimentación del Circuito Integrado a 3.3 volts o inferiores, logrando así un menor consumo en potencia del dispositivo.
4. Integración de forma interna de los voltajes necesarios para que los inversores programables realicen su transición de acuerdo a un FPD, lo cual nos daría un mejor desempeño y la liberación de terminales del chip para poder colocar más de una celda universal por chip.

7.3. Trabajos generados a partir de esta tesis.

Los resultados presentados en esta tesis, dieron lugar a los siguientes trabajos:

"4-Bit Arithmetic Logic Unit (ALU) based on Neuron MOS Transistors". E. A. Cortés-Barrón, M. A. Reyes-Barranca, L. M. Flores-Nava, Medina-Santiago, , 9th International Conference on Electrical Engineering Computing Science and Automatic Control (CCE 2012), México City, 2012. pp.