



FACULTAD DE CIENCIAS
POSGRADO EN CIENCIAS APLICADAS

TESIS PROFESIONAL:

CELDA LÓGICA RECONFIGURABLE

Tesis presentada por Ricardo Eliu Lozoya Ponce para obtener el grado de Doctor en Ciencias Aplicadas en la Universidad Autónoma de San Luis Potosí San Luis Potosí, S.L.P., México, febrero 2017.

Asesor:
Dr. Isaac Campos Cantón

Índice general

1. Introducción	1
2. Conceptos teóricos fundamentales	7
2.1. Definiciones algebraicas	7
Espacio vectorial real	7
Sub Espacio Vectorial	8
Función Lineal	8
Composición de funciones	8
3. Definición de celda lógica reconfigurable	9
3.1. Celda lógica reconfigurable basada en un sistema bidimensional	10
3.2. Circuito de celda lógica reconfigurable basada en un sistema bidimensional	14
3.3. Celda lógica reconfigurable basada en sistema lineal por partes	18
4. Diseño teórico de compuertas lógicas	23
Diseño de compuerta AND "N" entradas	24
Diseño de compuerta OR "N" entradas	25
Respuesta con incremento lineal	27
Respuesta lineal a trozos	29
5. Circuitos reconfigurables	33
5.1. Compuerta lógica reconfigurable	34
5.2. Circuitos combinacionales	59
5.2.1. Medio sumador - Medio Restador	59
Resultado medio sumador	64
Resultado medio restador	65
5.2.2. Decodificador	66
Resultado decodificador	70
5.2.3. Codificador	74
Resultado codificador	78
5.2.4. Comparador / Multiplexor / Demultiplexor	81
Resultado comparador	88
Resultado demultiplexor	89
Resultado multiplexor	93

5.2.5. Elementos de almacenamiento: Latches	97
Resultado Latch SR	101
5.2.6. Unidad Aritmética - Lógica	102
Sección Lógica	102
Sección Aritmética	103
Circuito de selección aritmética	106
ALU 1 Bit	110
ALU 2 Bit	110
6. Aplicaciones	117
6.1. Diseño de una neurona artificial usando circuitos reconfigurables	117
6.1.1. Neurobiología	118
6.1.2. Modelo Perceptrón	119
6.1.3. El aprendizaje en las redes neuronales artificiales	121
6.1.4. Metodología de diseño	122
6.2. Diseño e implementación para un convertidor multinivel	123
Topologías de conversión multinivel.	124
Técnicas de Conmutación	126
Modulación PWM.	126
7. Conclusiones	141
Bibliografía	145

Índice de figuras

3.1. Nodo estable en: (a) (0,0), (b) $(b_{12}/\lambda, b_{22}/\mu)$, (c) $(b_{11}/\lambda, b_{21}/\mu)$ y (d) $((b_{11} + b_{12})/\lambda, (b_{21} + b_{22})/\mu)$	13
3.2. Celda Lógica.	13
3.3. Ubicación del nodo y región cerrada para función lógica OR.	14
3.4. <i>Diagrama a bloques de la celda lógica.</i>	14
3.5. <i>Circuito ventana.</i>	16
3.6. <i>Nodos estables.- a) Representación de los cuatro casos, b) Función OR.</i>	17
3.7. <i>Función OR. (a) Señales de entrada a las terminales U_1 y U_2, potencial en el nodo M y salida Y de la celda lógica. (b) Potenciales de umbral $\pm K$, en el nodo M y de salida Y.</i>	18
3.8. <i>Celda lógica propuesta</i>	19
3.9. <i>a) Posibles valores para las diferentes permutaciones de la ecuación (3.23); b) Representación gráfica de la ecuación (3.24) por estados</i>	20
3.10. <i>Compuertas AND comerciales.</i>	20
3.11. <i>Arreglos en cascada para formar compuertas de múltiples entradas.</i>	21
4.1. <i>Estados para una compuerta de dos entradas.</i>	24
4.2. <i>Salidas de la celda lógica.</i>	24
4.3. <i>Estados para una compuerta AND de n entradas.</i>	25
4.4. <i>Salidas de la celda lógica en modo AND de n entradas.</i>	25
4.5. <i>Estados para una compuerta OR de n entradas.</i>	26
4.6. <i>Salidas de la celda lógica en modo OR de n entradas.</i>	26
4.7. <i>Respuesta con incremento lineal de la ecuación (3.23) utilizando la asignación de la ecuación (4.1).</i>	27
4.8. <i>Funciones que son obtenidas utilizando un solo umbral de operación y una respuesta lineal creciente en la ecuación (3.23).</i>	29
4.9. <i>Respuesta lineal a trozos para la ecuación (3.23) utilizando la asignación de la ecuación (4.2).</i>	29
4.10. <i>Funciones que son obtenidas utilizando un solo umbral de operación y una respuesta lineal a trozos en la ecuación (3.23).</i>	30
5.1. <i>Diagrama de compuerta lógica reconfigurable.</i>	34
5.2. <i>Respuesta de compuerta lógica reconfigurable en modo OR.</i>	37
5.3. <i>Amplificador operacional en modo de sumador restador.</i>	38

5.4. Arreglo de amplificadores operacionales en modo de circuito ventana.	39
5.5. Diagra de bloques para diseño de celda lógica reconfigurable.	39
5.6. Circuito de montaje superficial de celda lógica reconfigurable.	40
5.7. Respuesta con incremento lineal.	41
5.8. Función F_0	42
5.9. Función F_1	43
5.10. Función F_2	44
5.11. Función F_3	45
5.12. Función F_4	46
5.13. Función F_6	47
5.14. Función F_7	48
5.15. Función F_8	49
5.16. Función F_{12}	50
5.17. Función F_{14}	51
5.18. Función F_{15}	52
5.19. Respuesta lineal a trozos.	53
5.20. Función F_5	54
5.21. Función F_9	55
5.22. Función F_{10}	56
5.23. Función F_{11}	57
5.24. Función F_{13}	58
5.25. Diagrama de un circuito combinacional medio sumador	60
5.26. Diagrama de un circuito combinacional medio sumador usando un esquema basado en estructuras reconfigurables.	60
5.27. Diagrama de un circuito combinacional medio restador.	62
5.28. Resultados de circuito reconfigurable en modo <i>medio sumador</i>	64
5.29. Resultados de circuito reconfigurable en modo <i>medio restador</i>	65
5.30. Diagrama de un circuito combinacional <i>decodificador</i>	66
5.31. Diagrama de un circuito combinacional <i>decodificador</i> usando un esquema basado en estructuras reconfigurables.	67
5.32. Resultados de circuito reconfigurable en modo <i>decodificador</i> (Salidas D_0 y D_1).	70
5.33. Resultados de circuito reconfigurable en modo <i>decodificador</i> (Salidas D_2 y D_3).	71
5.34. Resultados de circuito reconfigurable en modo <i>decodificador</i> (Salidas D_4 y D_5).	72
5.35. Resultados de circuito reconfigurable en modo <i>decodificador</i> (Salidas D_6 y D_7).	73
5.36. Diagrama de un circuito combinacional <i>codificador</i>	74
5.37. Diagrama de un circuito combinacional <i>codificador</i> usando un esquema basado en estructuras reconfigurables.	75
5.38. Resultado D_0 para circuito reconfigurable en modo <i>codificador</i>	78
5.39. Resultado D_1 para circuito reconfigurable en modo <i>codificador</i>	79
5.40. Resultado D_2 para circuito reconfigurable en modo <i>codificador</i>	80
5.41. Circuitos empleados para sistema reconfigurable comparador / multiplexor / demultiplexor.- a) Bloque M_1 , modelo para sumador restador, b) Bloque M_2 , modelo para circuito ventana, c) Bloque M_3 , modelo para sumador con ganancia unitaria.	82

5.42. Circuitos empleados para sistema reconfigurable comparador / multiplexor / demultiplexor.- a) Bloque M_1 , modelo para sumador restador, b) Bloque M_2 , modelo para circuito ventana, c) Bloque M_3 , modelo para sumador con ganancia unitaria.	82
5.43. Esquema de entradas y salidas presentes en un comparador de 2 números de 2 bits.	83
5.44. Diagrama a bloques para un comparador de 2 números de 2 bits.	84
5.45. Resultados de circuito reconfigurable en modo <i>comparador</i>	85
5.46. Resultados de circuito reconfigurable en modo <i>comparador</i>	86
5.47. Resultados de circuito reconfigurable en modo <i>comparador</i>	87
5.48. Esquema de entradas y salidas presentes en un demultiplexor 1×4	88
5.49. Diagrama a bloques para un demultiplexor 1×4	89
5.50. Resultados de circuito reconfigurable en modo <i>Demultiplexor</i> . a) Señales de entrada x_0, x_1, x_2 y función F	89
5.51. Resultados de circuito reconfigurable en modo <i>comparador</i>	90
5.52. Resultados de circuito reconfigurable en modo <i>comparador</i>	91
5.53. Bloque esquemático que muestra las entradas, salidas y entradas de control en un multiplexor 4×1	92
5.54. Diagrama a bloques para el multiplexor 4×1	93
5.55. Resultados de circuito reconfigurable en modo <i>Multiplexor</i> . a) Señales de selección z_0, z_1 y función F	93
5.56. Resultados de circuito reconfigurable en modo <i>multiplexor</i>	94
5.57. Resultados de circuito reconfigurable en modo <i>multiplexor</i>	95
5.58. Resultados de circuito reconfigurable en modo <i>multiplexor</i>	96
5.59. Diagramas combinatoriales para <i>Latch SR</i>	97
5.60. Cronograma (diagrama de tiempo) de un <i>Latch SR</i>	98
5.61. Diagrama a bloques para un <i>Latch SR</i>	98
5.62. Diagrama a bloques para un <i>Latch SR</i> utilizando un lazo de retroalimentación.	99
5.63. Diagramas combinatoriales para <i>Latch SR</i>	100
5.64. Resultado de simulación para un <i>Latch SR</i>	101
5.65. Diagrama de entradas y salidas para un <i>ALU</i> . Los puertos <i>A</i> y <i>B</i> corresponden a los datos de entrada que se van a operar; los bits <i>C</i> corresponden a los bits de acarreo de entrada y salida; en el puerto <i>S</i> están los bits que corresponden a las señales de control que permiten seleccionar que función se realizará; el puerto <i>F</i> contiene los bits de salida del sistema.	102
5.66. Diagrama de la sección lógica del <i>ALU</i> (1 bit).	103
5.67. Diagrama lógico para la selección de funciones aritméticas.	104
5.68. Diagrama lógico de un <i>ALU</i> de 1 Bit de operación.	105
5.69. Diagrama lógico de un <i>ALU</i> de 2 Bits de operación.	105
5.70. Diagrama de la primera sección del seleccionador aritmético.	106
5.71. Diagrama de la segunda sección del seleccionador aritmético.	107
5.72. Respuesta en tiempo del sistema de la figura 5.70 (color azul) y regiones de umbral de acción (color verde).	108
5.73. Diagrama de bloques de un circuito seleccionador aritmético.	108

5.74. Respuesta en tiempo para circuito seleccionador aritmetico presentado en figura 5.73.	109
5.75. Diagrama a bloques de un ALU 1 Bit	111
5.76. Respuesta de funciones aritméticas para un ALU 1 Bit	112
5.77. Diagrama a bloques de un ALU 2 Bit	113
5.78. Respuesta de funciones aritméticas para un ALU 2 Bit	114
5.79. Respuesta de funciones lógicas para un ALU 1 Bit	115
5.80. Respuesta de funciones lógicas para un ALU 2 Bit	115
6.1. Partes de una neurona.	118
6.2. Unión entre neuronas, Sinapsis.	118
6.3. Esquema de neurona.	119
6.4. Función de activación tipo escalón.	120
6.5. Plano para función de 2 entradas (<i>OR</i>).	120
6.6. Solución con red de perceptrones para el caso <i>XOR</i>	121
6.7. Aprendizaje por corrección de error.	121
6.8. Señales de switcheo y comparación entre señal escalonada y señal sinusoidal.	123
6.9. Topología de capacitor flotante.	124
6.10. Topología de diodo anclado.	125
6.11. Topología de puente H.	125
6.12. Técnicas de conmutación.	126
6.13. Señales portadoras (rosa, azul, verde, rojo) y señal moduladora (negro).	127
6.14. Señal de switcheo obtenida tras comparar la portadora de nivel más bajo (rojo).	127
6.15. Señal de switcheo obtenida tras comparar cada portadora.	128
6.16. Sumatoria de todas las señales de switcheo (señal multinivel azul) y señal moduladora (negro)	128
6.17. Diagrama del proceso de emulación para un convertidor multinivel.	129
6.18. a) Configuración sumador no inversor. b) Diagrama a bloque	129
6.19. a) Configuración comparador. b) Diagrama a bloque	130
6.20. a) Configuración sumador. b) Diagrama a bloque	131
6.21. Tipos de arreglo para obtener señales portadoras.- a) Paralelo, b) Serie.	132
6.22. Resultado experimental usando señal cuadrada $K_n = 1.2KHz$	133
6.23. Resultado experimental usando señal triangular $K_n = 1.2KHz$	134
6.24. Resultado experimental usando señal senoidal $K_n = 5.5KHz$	134
6.25. Resultado experimental usando señal constante.	135
6.26. Resultado experimental usando señal de ruido blanco.	135
6.27. Resultado experimental usando estado <i>Z</i> del <i>oscilador de Lorenz</i>	136
6.28. Resultado experimental usando señal cuadrada $K_n = 1.2KHz$	136
6.29. Resultado experimental usando señal triangular $K_n = 1.2KHz$	137
6.30. Resultado experimental usando señal senoidal $K_n = 1.2KHz$	137
6.31. Resultado experimental usando señal constante.	138
6.32. Resultado experimental usando señal de ruido blanco.	138
6.33. Resultado experimental usando señal del estado <i>Z</i> del <i>oscilador de Lorenz</i>	139

Índice de cuadros

3.1. Combinación de las entradas en el sistema lineal.	10
3.2. Tabla de verdad para una función lógica OR de dos entradas.	12
3.3. <i>Soluciones de X_1 y X_2 para los distintos casos.</i>	17
4.1. Combinación de las entradas en el sistema lineal.	24
4.2. Funciones booleanas que se pueden obtener con una compuerta de dos entradas.	26
4.3. Operaciones booleanas para compuertas de dos entradas.	27
5.1. Valores de resistencias en una compuerta reconfigurable en modo OR.	36
5.2. Respuesta con incremento lineal en voltajes de permutación de compuerta lógica reconfigurable.	36
5.3. Tabla de verdad para un medio sumador.	59
5.4. Tabla para un medio sumador.	61
5.5. Tabla de verdad para un medio restador.	61
5.6. Tabla para un medio restador.	62
5.7. Valores de resistencias y voltajes en circuito medio sumador reconfigurable.	63
5.8. Tabla de verdad para un <i>decodificador</i>	66
5.9. Tabla de relación entre voltajes y permutaciones en un <i>decodificador</i>	68
5.10. Valores de resistencias y voltajes en un circuito <i>decodificador</i> reconfigurable.	68
5.11. Tabla de verdad para un <i>decodificador</i>	74
5.12. Tabla para un <i>codificador</i>	76
5.13. Valores de resistencias y voltajes en un circuito <i>codificador</i> reconfigurable.	76
5.14. <i>Valores usados en las resistencias.</i>	83
5.15. <i>Voltajes en M_1.</i>	83
5.16. <i>Tabla de verdad de un comparador de 2 números de 2 bits.</i>	84
5.17. <i>Voltajes para los distintos bloques M_2 del comparador.</i>	85
5.18. Tabla de verdad del demultiplexor 1×4	88
5.19. <i>Voltajes para los distintos bloques M_2 del demultiplexor.</i>	88
5.20. Tabla de verdad del multiplexor 4×1	92
5.21. Voltajes para los distintos bloques M_2 del multiplexor	92
5.22. Tabla de verdad del multiplexor 4×1	98
5.23. Tabla de verdad para un Latch <i>SR</i>	99
5.24. Tabla de selección de funciones para la sección lógica de un ALU.	103
5.25. Tabla de selección de funciones para la sección aritmética de un ALU.	104

5.26. Tabla de verdad para circuito lógico selector de funciones aritméticas. 104
5.27. Tabla de verdad para circuito lógico selector de funciones aritméticas. 107

1

Introducción

La necesidad del hombre por comprender y controlar lo que le rodea lo ha llevado a realizar incontables investigaciones, es así que, se ha propuesto el desarrollo de técnicas y aplicaciones que le permitan cubrir diversas necesidades, de esta forma la humanidad ha pasado de la aplicación de máquinas simples como la rueda, la polea y el plano inclinado a aumentar la complejidad de estas, al diseñar dispositivos como los electrónicos que son utilizados casi para cualquier necesidad, como sistemas de comunicación, sector industrial, sector salud entre otros, los cuales solo representan algunas de las áreas en las que es común el uso de circuitos electrónicos [1].

Cuando hablamos de dispositivo electrónico por ende hablamos de circuito eléctrico. Que se refiere a un camino o conducto que se cierra sobre sí mismo, y en cuyo interior fluye una corriente eléctrica. En los circuitos electrónicos el camino cerrado lo forman materiales y dispositivos capaces de permitir el paso de grandes o pequeñas cantidades de electricidad. Dominando este flujo de electricidad a través de circuitos electrónicos, es posible la generación, transmisión, recepción y almacenamiento de información. Esta información puede ser una señal de voz o música, como en un receptor de radio; o una imagen en una pantalla de televisión; o números u otros datos en una computadora [2].

En general los circuitos electrónicos ofrecen una gran diversidad de funciones para procesar esta información, como las operaciones lógicas que se llevan a cabo en dispositivos tan comunes como teléfonos celulares, computadoras, videojuegos, etc. La adquisición de herramientas electrónicas permite la solución en una gran variedad de problemas [3].

Cabe mencionar que los circuitos electrónicos se pueden dividir en dos categorías: circuitos digitales y circuitos analógicos. La principal característica de un circuito digital es que las señales que emplea presentan valores discretos, es decir solo pueden tomar valores dentro de un conjunto numerable, comúnmente 0 y 5 V. Para el análisis de estos sistemas se utiliza el álgebra de Boole, la cual es una teoría matemática que se aplica en la lógica combinatoria, es así que las variables booleanas solo pueden tomar dos valores posibles: 1 (valor alto - 5 V) ó 0 (valor bajo

- 0 V). Como por ejemplo un interruptor de luz, el cual solo puede estar activado o desactivado, es decir un circuito abierto o cerrado. Caber mencionar que los sistemas digitales se dividen en dos clasificaciones:

- Sistemas combinacionales.- Dentro de sus principales características es que la salida del sistema solo depende de las entradas presentes, es decir realiza las operaciones en tiempo real y no cuenta con ningún banco de memoria.
- Sistemas secuenciales.- La salida de estos sistemas depende directamente de la entrada actual y de los valores de entradas pasadas. Esta situación desemboca en que sea necesario utilizar elementos de memoria, que almacenen la información de la "historia pasada" del sistema.

Por su parte, en la electrónica analógica es posible emplear amplitudes de voltaje con valores continuos, en este caso, señales que por su naturaleza pueden tomar cualquier valor a lo largo de una escala numérica. Como por ejemplo, la amplitud de la señal de salida para un altavoz en un receptor de audio, donde ésta puede variar desde cero hasta su límite más alto. Se dice que una señal continua a diferencia de una discreta, nunca puede ser medida con exactitud, ya que el valor observado depende directamente de la precisión de los instrumentos de medición [4].

Existen diversas ventajas y desventajas que presentan ambos esquemas, sin embargo la tendencia de los últimos años muestra que son más los beneficios en el uso de dispositivos digitales. Dentro de las virtudes de los sistemas digitales se encuentra que son fáciles de diseñar, ya que esta permitido manejar un rango de valores y no un valor exacto; de esta manera el sistema se hace robusto ante posibles cambios de amplitud ocasionados por variaciones de temperatura, perturbaciones a la entrada y ruido en las componentes implementadas [5].

También está el hecho de que en cuanto al almacenamiento de datos, los sistemas digitales presentan una virtud extra sobre los sistemas analógicos; debido a que utilizan dispositivos que pueden guardar información y retenerla por el tiempo que sea necesario (Flip Flops es un dispositivo capaz de permanecer en uno de dos estados posibles durante un tiempo indefinido en ausencia de perturbaciones) [6].

Lamentablemente tienen una pequeña desventaja o limitación, el mundo real es fundamentalmente analógico, ello implica que la mayoría de las cantidades físicas son analógicas, una solución para esta situación es la aproximación que se realiza en cantidades a través de Convertidores Analógico / Digitales (Digital Analogic Converter – DAC). Sin embargo, la implementación digital tiene sus limitaciones en la velocidad de operación de convertidores, la cual depende directamente del ancho de banda y de la velocidad de muestreo a la cual operen [7].

Por su parte, los sistemas analógicos no cuentan con esta desventaja, sin embargo son sistemas susceptibles a ser modificados de forma no deseada por su principal punto débil, el ruido. Desafortunadamente en este esquema, cualquier variación en la información afecta en gran medida al correcto funcionamiento y rendimiento del dispositivo. La solución más trivial ante esta

situación es el incremento de la potencia de operación; siendo así que la calidad en estos sistemas depende de la potencia de operación y de la potencia del ruido a la cual sea vulnerable [8].

Por lo tanto, al utilizar sistemas analógicos y sistemas digitales es posible desarrollar algoritmos que le permitan a un circuito resolver alguna necesidad o problema. Normalmente se selecciona un tipo de función a realizar y se implementa mediante el hardware de un circuito, a estos se les conoce como Circuitos Integrados de Aplicación Específica, por sus siglas en inglés ASIC (Application Specific Integrated Circuit). Los ASIC's son muy rápidos y eficientes al momento de ejecutar la función para la cual fueron diseñados, aunque un inconveniente que presentan es que una vez fabricados no pueden ser alterados, ya que la función que gobierna al sistema ha sido establecida, como en el caso del circuito de una compuerta lógica [9].

El trabajar con esquemas fijos o estáticos, ciertamente permite la optimización del diseño, lo cual implica reducción de espacios y costos. Sin embargo, los crecientes cambios tecnológicos y el cuidado del medio ambiente, exige cada vez más del diseño de circuitos, que sean adaptables y que permitan el rehuso de un mismo sistema para desarrollar funciones distintas [10].

Con esta situación surge la pregunta: ¿Es posible superar la limitación estática del diseño de circuitos de función específica? Este tema ha sido importante durante los últimos años, captando la atención de diversos grupos de investigación al rededor del mundo. Derivando así en la propuesta de diversas técnicas que permitan el diseño de circuitos con una arquitectura reconfigurable. Es decir, generar estructuras flexibles que sean capaces de cambiar su respuesta de acuerdo a sus parámetros de control y reconfiguración de los mecanismos involucrados [11].

Es decir, circuitos dinámicos que pretenden superar las estructuras estáticas convencionales, al dar un mayor nivel de flexibilidad en el hardware. En sí, un circuito con arquitectura dinámica puede cambiar la configuración de su hardware y por ende su lógica, durante el curso de un proceso, de acuerdo a las necesidades del usuario [12, 13].

El uso de sistemas basados en circuitos reconfigurables hacen cada vez más fácil imaginar una futura generación de procesadores que permitan transformarse acorde a diversas funciones o necesidades a cumplir [14].

Actualmente se utiliza una gran cantidad de sistemas que presentan una arquitectura de función de operación dinámica, hay estructuras como los Arreglos de Compuertas de Campo Programables, por sus siglas en inglés FPGA (Field Programmable Gate Array), que se consideran como circuitos con arquitectura dinámica, ya que en su interior se realiza una "reconexión" cada vez que se programa para realizar una función; ello le permite al sistema poder cambiar su lógica, sin embargo es más lento que un ASIC y tiene un mayor consumo de potencia. A pesar de estos inconvenientes destaca su principal ventaja característica, es un sistema reconfigurable [15].

Por otro lado existen también circuitos reconfigurables basados en sistemas caóticos, como es de esperarse las ecuaciones que describen a estos sistemas corresponden a modelos matemáticos de sistemas dinámicos que varían de diversas maneras con la simple variación de sus condiciones iniciales [16, 17, 18, 19, 20]. El realizar pequeñas variaciones en las condiciones iniciales

de estos sistemas, dan como resultado circuitos que varían su función de operación, permitiendo así diseños de compuertas lógicas basados en funciones dinámicas, por ejemplo, los que utilizan ecuaciones como las que rigen el comportamiento del Mapeo Casa de Campaña. Inicialmente se utiliza un circuito que permite generar una condición inicial, acompañado de un circuito que toma esta condición y la utiliza en un sistema que emula la función de Mapeo Casa de Campaña, finalmente y mediante otro circuito, se utiliza un criterio de comparación para determinar el valor de la salida (alto o bajo). Integrando estas tres partes se obtiene un sistema que depende directamente de una condición inicial, de una ecuación dinámica y de un criterio de comparación [23].

Siendo así que la propuesta de este trabajo radica directamente en ofrecer un sistema que no sea sensible a cambios en condiciones iniciales y que tampoco dependa de ecuaciones dinámicas, el objetivo es presentar un sistema que sea reconfigurable y que tenga la bondad de una fácil sintonización de parámetros de reconfiguración. Lo que conlleva a la propuesta de un sistema con arquitectura reconfigurable que sea robusto y flexible en diseño.

Para ello se considerará el esquema de un perceptrón simple, siendo este modelo semejante al propuesto en 1959 por Frank Rosenblatt, el cual corresponde a un modelo de neurona artificial que está constituido por un conjunto de entradas (señales) que linealmente son procesadas para analizar si son capaces de producir cierto estímulo, tal que sobrepase un umbral de activación y produzca una salida [24].

Este trabajo está estructurado en siete capítulos. En este primer capítulo, se presentaron algunos antecedentes de proyectos basados en circuitos reconfigurables, además de algunas necesidades que se presentan en estos dispositivos. Se deja ver que la propuesta de este trabajo de investigación radica en la propuesta de un nuevo esquema, que permita trabajar con sistemas digitales y continuos, además de presentar una estructura sencilla que le permita adaptarse a diferentes aplicaciones.

En el capítulo 2, se presentan definiciones algebraicas que sustentan bajo propiedades de linealidad, la propuesta presentada en este trabajo de investigación.

En el capítulo 3 se define el concepto de una *celda lógica reconfigurable*, también se presentan dos posibles esquemas para obtener este objetivo, uno que depende de un sistema bidimensional y otro que se basa en sistemas lineales por partes. En esta sección solo se presenta el diseño del circuito para el sistema bidimensional, ya que como el trabajo está basado en la metodología de sistemas lineales por partes, este último se presentará con diversas modificaciones en los capítulos posteriores.

En el capítulo 4 se presentan las respuestas *lineal* y *lineal por partes* del sistema propuesto, también se da a conocer una metodología para el diseño de compuertas lógicas y se generaliza para los casos de compuertas AND y OR.

En el capítulo 5 se presentan los circuitos para el diseño de diversos sistemas, se inicia con la propuesta de una *celda lógica reconfigurable*, y se muestra como con un solo sistema es posible

obtener las 16 posibles funciones lógicas para una compuerta de dos entradas. A continuación se presentan las propuestas del diseño de circuitos combinacionales, iniciando con un sistema que desarrolla las funciones de un *medio sumador*, *medio restador*, *decodificador*, *codificador*, *comparador*, *multiplexor*, *demultiplexor*, *latch SR* y una *ALU* de 1 y 2 bits de operación.

En el capítulo 6 se muestran dos aplicaciones, la primera consiste en la presentación del modelo de un perceptrón, el cual corresponde a un modelo de neurona artificial, el cual mantiene una enorme semejanza con el sistema propuesto y por otro lado se expone también la implementación de un convertidor multinivel.

Finalmente en el capítulo 7 se presentan las conclusiones generales.

Conceptos teóricos fundamentales

2.1 Definiciones algebraicas

Para comprender el principio de funcionamiento de los dispositivos que se propondrán en este trabajo de investigación, se recomienda tomar en cuenta las siguientes definiciones [25], inicialmente para un **Espacio Vectorial Real**, el cual dicta que:

Un espacio vectorial real V es un conjunto de objetos, denominados vectores, los cuales junto con los operadores de suma y multiplicación por un escalar cumplen los siguientes axiomas:

1. Si x pertenece a V y y pertenece a V , entonces $x + y$ pertenece a V .
2. Para todo x, y y z en V , $(x + y) + z = x + (y + z)$.
3. Existe un vector 0 que pertenece a V tal que para todo x que pertenece a V , $x + 0 = 0 + x = x$.
4. Si x pertenece a V , existe un vector $-x$ en V tal que $x + (-x) = 0$.
5. Si x y y están en V , entonces $x + y = y + x$.
6. Si x pertenece a V y α es un escalar, entonces αx pertenece a V .
7. Si x y y están en V y α es un escalar, entonces $\alpha(x + y) = \alpha x + \alpha y$.
8. Si x pertenece a V y α y β son escalares, entonces $(\alpha + \beta)x = \alpha x + \beta x$.
9. Si x pertenece a V y α y β son escalares, entonces $\alpha(\beta x) = (\alpha\beta)x$.
10. Para cada vector x que pertenece a V , $1x = x$.

Posteriormente para un **Sub Espacio Vectorial**, siendo su definición la siguiente:

Sub Espacio Vectorial.- Sea H un subconjunto no vacío de un espacio vectorial V y suponga que H es en sí un espacio vectorial bajo las operaciones de suma y multiplicación por un escalar definidas en V . Entonces se dice que H es un **Sub Espacio** de V .

Debido a que existe una gran cantidad de ejemplos de *sub espacio vectorial*, se debe demostrar que el *subconjunto de V* es en realidad *sub espacio* de V , para ello se contempla el siguiente teorema:

Teorema de Sub Espacio.- Un subconjunto no vacío de H de un espacio vectorial V es un sub espacio de V si se cumplen las dos reglas de cerradura:

1. Si $x \in H, y \in H$, entonces $(x + y) \in H$.
2. Si $x \in H$, entonces $\alpha x \in H$ para todo escalar $\alpha \in \mathbb{R}$.

Es decir, mediante este teorema se puede verificar que para comprobar que H es un *sub espacio* de V , basta con verificar que:

$$x + y \text{ y } \alpha x \text{ están en } H \text{ cuando } x \text{ y } y \text{ están en } H \text{ y } \alpha \text{ es un escalar.}$$

Posteriormente se define el concepto de **Función Lineal**.

Función Lineal.- Sean V y W espacios vectoriales sobre K , se dice que $f : V \rightarrow W$ es una **función K -lineal** (o morfismo de K -espacios) si $f : V \rightarrow W$ es homomorfismo de grupos abelianos que preserva el producto por escalar.

$$f(\alpha \cdot v) = \alpha f(v), \forall \alpha \in K, \forall v \in V$$

Cabe mencionar que en una función $f : V \rightarrow W$ es K -lineal si y sólo si $\forall v_1, v_2 \in V, \forall \alpha_1, \alpha_2 \in K, f(\alpha_1 v_1 + \alpha_2 v_2) = \alpha_1 f(v_1) + \alpha_2 f(v_2)$.

Posteriormente, es necesario expresar la definición formal de **Composición de Funciones**, la cual establece que:

Dadas dos funciones f y g , se define como la composición de la función f con la función g , a la función denotada $f \circ g$ (léase g compuesta con f), cuya regla de correspondencia es:

$$(f \circ g)(x) = f[g(x)]. \tag{2.1}$$

donde su dominio está representado por el conjunto:

$$D_{f \circ g} = \{x \mid x \in D_g; g(x) \in D_f\}. \tag{2.2}$$

Cabe mencionar que para obtener la regla de correspondencia de la función $f \circ g$, según la definición anterior, basta con sustituir la función g en la variable independiente de la función f . Por lo tanto el dominio de la función resultante, es un subconjunto del dominio de la función g .

Definición de celda lógica reconfigurable

En este punto es prudente definir el concepto de una Celda Lógica Reconfigurable:

Se denomina Celda Lógica Reconfigurable a aquel sistema que permite obtener diversas compuertas lógicas como las funciones AND, OR, XOR, sus respectivas negaciones y algunas funciones lógicas más. El término celda refiere entonces a un sistema mínimo que puede ser unido con otros idénticos a él, donde un solo sistema tiene la capacidad de emular diversas funciones lógicas, al agrupar varios de estos sistemas se obtiene un sistema de mayor complejidad, como una red neuronal.

Observe que dentro de esta definición se pueden contemplar diversos dispositivos como los FPGA's, DSP's, PLD's y Microcontroladores, manteniendo estos como característica común, el hecho de que para obtener dicha reconfiguración, se requiere pasar por un proceso de programación digital, es decir generar un código que describa la función lógica que se desea obtener, para posteriormente mediante alguna interfaz de lenguaje humano-máquina, cargar estas instrucciones en el dispositivo.

Sin embargo la propuesta de esta investigación consiste en el diseño de un esquema que permita llevar a cabo esta reconfiguración mediante una forma distinta, tratando de omitir la parte de programación y limitando el funcionamiento lógico del circuito, mediante la sintonización de un conjunto de variables de entrada.

3.1 Celda lógica reconfigurable basada en un sistema bidimensional

El objetivo de esta sección es dar a conocer un antecedente previo a la propuesta que se presentará en este trabajo de investigación. Cabe mencionar que existen diversos trabajos en los que se proponen metodologías basadas en sistemas dinámicos [19, 20, 21, 22], sin embargo, el desarrollo de estos conlleva un mayor esfuerzo en el análisis de estos sistemas, además de que por tratarse en algunos casos de sistemas caóticos se depende de una condición inicial. En esta sección se representa como antecedente, el diseño de una celda lógica reconfigurable basada en el espacio de estados de un sistema bidimensional, el cual tiene valores propios distintos.

Esta idea contempla las cuatro soluciones posibles de un sistema bidimensional con dos entradas. Inicialmente se parte de las siguientes ecuaciones:

$$\dot{X} = AX + BU \quad (3.1)$$

$$Y(X) = \begin{cases} 1, & \text{si } |X - C| < K; \\ 0, & \text{de otra forma.} \end{cases} \quad (3.2)$$

La ecuación 3.1 corresponde a una función lineal donde $X, U \in \mathbb{R}^2$ y $A, B \in 2 \times 2$ son matrices reales. Realizando transformaciones se obtienen los siguientes casos para la matriz A:

$$A_1 = \begin{bmatrix} \lambda & 0 \\ 0 & \mu \end{bmatrix}, \quad A_2 = \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}, \quad A_3 = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}$$

Analizando exclusivamente para el caso donde $A = A_1$ y $U \in \{0,1\}$, las únicas entradas binarias permitidas por el sistema. Si λ y μ son valores positivos, entonces podemos reescribir el sistema explícitamente como:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\lambda & 1 \\ 0 & -\mu \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.3)$$

el cual tiene los cuatro casos siguientes:

u_1	u_2	
0	0	caso 1
0	1	caso 2
1	0	caso 3
1	1	caso 4

Tabla 3.1: Combinación de las entradas en el sistema lineal.

los cuales se exponen a continuación.

■ CASO 1:

El sistema (3.3) presenta la siguiente estructura:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\lambda & 1 \\ 0 & -\mu \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3.4)$$

y su solución es

$$x_1 = x_{10}e^{-\lambda t}, \quad x_2 = x_{20}e^{-\mu t}$$

con las condiciones iniciales x_{10} y x_{20} . Su representación en el espacio de estados se muestra en la Figura 3.1 (a), el cual es un nodo estable en el origen.

■ CASO 2:

El sistema (3.3) se escribe como

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\lambda & 1 \\ 0 & -\mu \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{12} \\ b_{22} \end{bmatrix} \quad (3.5)$$

la solución es

$$\begin{aligned} x_1 &= \left(x_{10} - \frac{b_{12}}{\lambda} \right) e^{-\lambda t} + \frac{b_{12}}{\lambda}, \\ x_2 &= \left(x_{20} - \frac{b_{22}}{\mu} \right) e^{-\mu t} + \frac{b_{22}}{\mu}, \end{aligned}$$

por ejemplo, si b_{12} y b_{22} toma valores positivos, su diagrama en el espacio de estados sera el mostrado en la Figura 3.1 (b) con el centro del nodo ubicado en $(b_{12}/\lambda, b_{22}/\mu)$.

■ CASO 3:

Para este caso la Ecuación (3.3) toma la siguiente forma:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\lambda & 1 \\ 0 & -\mu \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix} \quad (3.6)$$

de donde

$$\begin{aligned} x_1 &= \left(x_{10} - \frac{b_{11}}{\lambda} \right) e^{-\lambda t} + \frac{b_{11}}{\lambda}, \\ x_2 &= \left(x_{20} - \frac{b_{21}}{\mu} \right) e^{-\mu t} + \frac{b_{21}}{\mu}, \end{aligned}$$

y si $b_{11} > b_{12}$ y $b_{21} > b_{22}$ se encuentra el siguiente diagrama de fase del nodo con centro en $(b_{11}/\lambda, b_{21}/\mu)$ Figura 3.1 (c).

■ CASO 4:

Podemos escribir (3.3) como

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\lambda & 1 \\ 0 & -\mu \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{11} + b_{12} \\ b_{21} + b_{22} \end{bmatrix} \quad (3.7)$$

así

$$\begin{aligned} x_1 &= \left(x_{10} - \frac{b_{11} + b_{12}}{\lambda} \right) e^{-\lambda t} + \frac{b_{11} + b_{12}}{\lambda}, \\ x_2 &= \left(x_{20} - \frac{b_{21} + b_{22}}{\mu} \right) e^{-\mu t} + \frac{b_{21} + b_{22}}{\mu}, \end{aligned}$$

dando su diagrama fase en la Figura 3.1 (d). Con el centro del nodo ubicado en $((b_{11} + b_{12})/\lambda, (b_{21} + b_{22})/\mu)$.

Como se puede apreciar en la Figura 3.1, dependiendo de la combinación de las entradas se generan cuatro nodos estables en el espacio de estados $x_1 - x_2$, residiendo en una, dos, tres ó cuatro posiciones diferentes según se escojan los valores de los parámetros de la matriz B .

La ecuación 3.2 corresponde a la salida del sistema, esta define una región cerrada del espacio de estados con centro en C .

De esta manera, si se encuentra dentro de esa región, la salida del sistema tomará el valor de uno (nivel alto) y la salida proporcionara un cero (nivel bajo) en caso contrario. Por lo tanto, la estructura entrada-salida de este modelo de celda lógica se podría expresar como el diagrama de la figura 3.2.

u_1	u_2	OR
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 3.2: Tabla de verdad para una función lógica OR de dos entradas.

Por ejemplo, para obtener la función OR, inicialmente se consideran las ecuaciones 3.1 y 3.2 para posteriormente definir una región cerrada de forma circular, con centro en $(b_{11}/\lambda, b_{21}/\mu)$ y radio k menor que las distancias a los nodos estables más cercanos como se aprecia en la Figura 3.3.

Para obtener esta función se define una región cerrada con centro en $(b_{11}/\lambda, b_{21}/\mu)$. Utilizando un círculo bajo las especificaciones siguientes: se toman los valores $b_{ii} > 0$ con $i \in 1, 2$, $b_{12} = b_{22}$, $b_{11} = 2b_{12}$, $b_{21} = 2b_{22}$ y un radio $k < \sqrt{(2b_{12}/\lambda)^2 + (2b_{22}/\mu)^2}$ como se muestra en la Figura 3.3. Por lo tanto la Ecuación 3.2 se expresa como sigue:

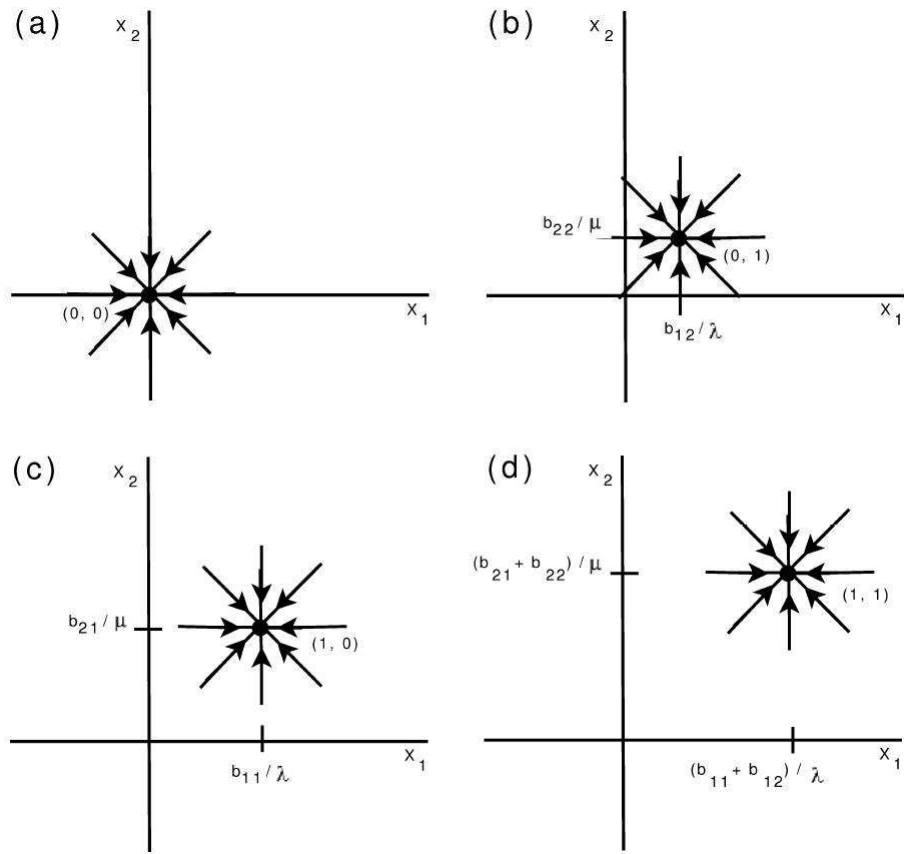


Figura 3.1: Nodo estable en: (a) $(0,0)$, (b) $(b_{12}/\lambda, b_{22}/\mu)$, (c) $(b_{11}/\lambda, b_{21}/\mu)$ y (d) $((b_{11} + b_{12})/\lambda, (b_{21} + b_{22})/\mu)$.

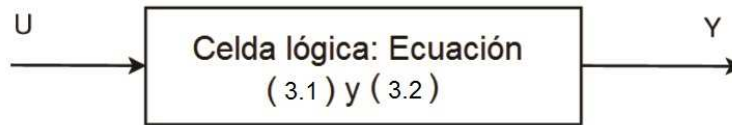


Figura 3.2: Celda Lógica.

$$Y_{OR}(X) = \begin{cases} 1, & \text{si } (x_1 - \frac{b_{11}}{\lambda})^2 + (x_2 - \frac{b_{21}}{\mu})^2 < K; \\ 0, & \text{de otra forma.} \end{cases} \quad (3.8)$$

Obteniendo como resultado la función lógica OR como se puede corroborar con la Tabla II.

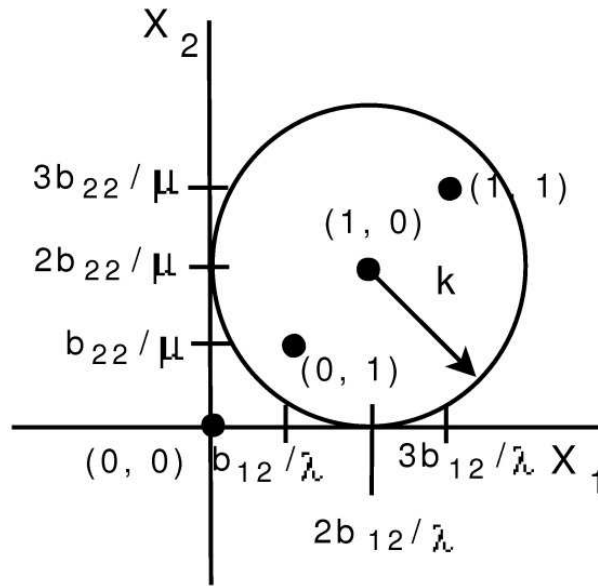


Figura 3.3: Ubicación del nodo y región cerrada para función lógica OR.

3.2 Circuito de celda lógica reconfigurable basada en un sistema bidimensional

En esta sección se aborda el proceso de instrumentación de la celda basada en un sistema bidimensional, presentada anteriormente, la cual cuenta con valores propios distintos. Este diseño esta conformado por un arreglo de resistencias, capacitores y comparadores [26].

Partiendo de la ecuación 3.1, se considera un circuito constituido por 4 resistencias, dos capacitores y dos terminales de entrada (U_1 y U_2) como se muestra en la figura 3.4.

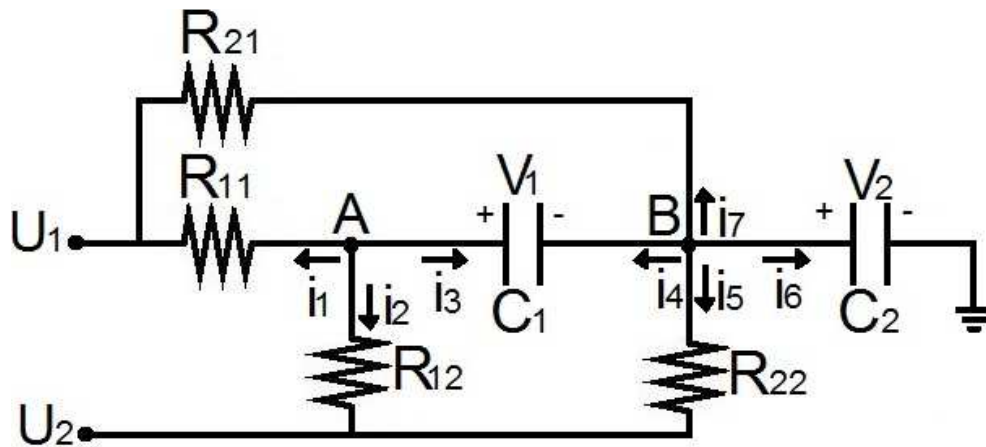


Figura 3.4: Diagrama a bloques de la celda lógica.

Utilizando las leyes de Kirchoff se obtiene la siguiente relación para el nodo **M**:

$$i_1 + i_2 + i_3 = 0. \quad (3.9)$$

de donde:

$$i_1 = \frac{V_A - U_1}{R_{11}}, i_2 = \frac{V_A - U_2}{R_{12}} \text{ e } i_3 = C_1 \frac{d(V_A - V_B)}{dt}$$

Note que $V_A = V_1 + V_2$ y $V_B = V_2$, al sustituir esta relación en (3.17) se llega a:

$$C_1 \frac{dV_1}{dt} = \frac{1}{R_{11}} U_1 + \frac{1}{R_{12}} U_2 - \left(\frac{1}{R_{11}} + \frac{1}{R_{12}} \right) (V_1 + V_2). \quad (3.10)$$

De manera semejante para el nodo **B** se tiene:

$$i_4 + i_5 + i_6 + i_7 = 0. \quad (3.11)$$

donde: $i_4 = C_1 \frac{d(V_B - V_A)}{dt}$, $i_5 = \frac{V_B - U_2}{R_{22}}$, $i_6 = C_2 \frac{dV_B}{dt}$ e $i_7 = \frac{V_B - U_1}{R_{21}}$, se sustituyen en (3.11) y utilizando (3.18) se obtiene:

$$\begin{aligned} C_2 \frac{dV_2}{dt} &= \left(\frac{1}{R_{11}} + \frac{1}{R_{21}} \right) U_1 + \left(\frac{1}{R_{12}} + \frac{1}{R_{22}} \right) U_2 \\ &- \left(\frac{1}{R_{11}} + \frac{1}{R_{12}} \right) V_1 - \left(\frac{1}{R_{11}} + \frac{1}{R_{12}} + \frac{1}{R_{21}} + \frac{1}{R_{22}} \right) V_2. \end{aligned} \quad (3.12)$$

Así (3.18) y (3.12) conforman (3.1). La notación \dot{X} denota que se trata de la derivada con respecto del tiempo, por lo tanto los vectores y las matrices quedan definidos como sigue:

$$\begin{aligned} X &= \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}; A = \begin{bmatrix} -\frac{1}{C_1} \left(\frac{1}{R_{11}} + \frac{1}{R_{12}} \right) & -\frac{1}{C_1} \left(\frac{1}{R_{11}} + \frac{1}{R_{12}} \right) \\ -\frac{1}{C_2} \left(\frac{1}{R_{11}} + \frac{1}{R_{12}} \right) & -\frac{1}{C_2} \left(\frac{1}{R_{11}} + \frac{1}{R_{12}} + \frac{1}{R_{21}} + \frac{1}{R_{22}} \right) \end{bmatrix} \\ U &= \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}; B = \begin{bmatrix} \frac{1}{R_{11}C_1} & \frac{1}{R_{12}C_1} \\ \frac{1}{C_2} \left(\frac{1}{R_{11}} + \frac{1}{R_{21}} \right) & \frac{1}{C_2} \left(\frac{1}{R_{12}} + \frac{1}{R_{22}} \right) \end{bmatrix} \end{aligned} \quad (3.13)$$

Para el circuito equivalente de la ecuación (3.2) se utiliza el circuito ventana [28], el cual se muestra en la figura 3.5, este circuito esta constituido por dos comparadores LM311 y una resistencia.

El circuito comparador LM311 funciona como se describe a continuación: si el potencial en la terminal de entrada no inversora es mayor que el de la terminal de entrada inversora, la terminal de salida se coloca en alta impedancia, con lo cual no fluye corriente por esta terminal. Y si el potencial en la terminal de entrada no inversora es menor que el de la terminal inversora, la terminal de salida se conecta con la terminal aterrizada y el potencial en la salida toma el valor de 0V. El arreglo que se muestra en la figura 3.5 se modela mediante el siguiente conjunto de

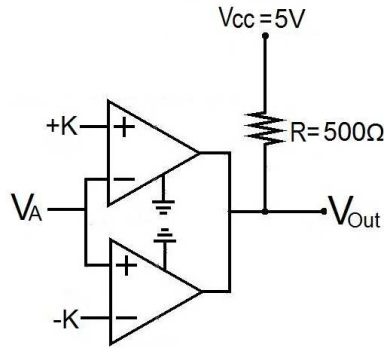


Figura 3.5: Circuito ventana.

ecuaciones:

Para el comparador de la parte superior:

$$V_{Out} = \begin{cases} 1, & \text{si } V_A < +K; \\ 0, & \text{de otra forma.} \end{cases} \quad (3.14)$$

Y para el comparador de la parte inferior:

$$V_{Out} = \begin{cases} 1, & \text{si } -K < V_A; \\ 0, & \text{de otra forma.} \end{cases} \quad (3.15)$$

Por lo tanto combinando (3.14) y (3.15) en una sola ecuación se llega a la ecuación propuesta para (3.2) de la celda lógica, tal como se representa en (3.16) donde el valor de $V_{CC} = 5V$ se toma como un uno lógico y $0V$ como 0 lógico.

$$V_{Out} = \begin{cases} 1, & \text{si } |V_A| < K; \\ 0, & \text{de otra forma.} \end{cases} \quad (3.16)$$

Cabe mencionar que el centro C de la ecuación (3.2) se calcula mediante los valores de $\pm K$ como $C = (-K + +K)/2$.

Por ejemplo, para realizar la función lógica OR, se proponen los valores de $R_{11} = R_{12} = R_{21} = R_{22} = R$, $C_1 = 2C$ y $C_2 = C$, con esto la matriz A arroja los siguientes valores propios.

$$\begin{aligned} \lambda &= \frac{-5}{2RC} + \frac{\sqrt{17}}{2RC} \\ \mu &= \frac{-5}{2RC} - \frac{\sqrt{17}}{2RC} \end{aligned} \quad (3.17)$$

y B queda constituido como:

$$B = \begin{bmatrix} \frac{1}{2RC} & \frac{1}{2RC} \\ \frac{1}{RC} & \frac{1}{RC} \end{bmatrix} \quad (3.18)$$

De donde $b_{11} = \frac{1}{2RC}$, $b_{12} = \frac{1}{2RC}$, $b_{21} = \frac{2}{RC}$ y $b_{22} = \frac{2}{RC}$.

Así para las diferentes combinaciones de las entradas (Tabla 3.3) se tienen cuatro casos, donde la solución de cada uno es dado en las columnas X_1 y X_2 respectivamente:

U_1	U_2		X_1	X_2
0	0	Caso 1	$X_{10}e^{\lambda t}$	$X_{20}e^{\mu t}$
0	1	Caso 2	$\left(V_{10} + \frac{1}{-5+\sqrt{17}}\right)e^{\lambda t} - \frac{1}{-5+\sqrt{17}}$	$\left(V_{20} + \frac{4}{-5+\sqrt{17}}\right)e^{\mu t} - \frac{4}{5+\sqrt{17}}$
0	1	Caso 3	$\left(V_{10} + \frac{1}{-5+\sqrt{17}}\right)e^{\lambda t} - \frac{1}{-5+\sqrt{17}}$	$\left(V_{20} - \frac{4}{5+\sqrt{17}}\right)e^{\mu t} + \frac{4}{5+\sqrt{17}}$
0	1	Caso 4	$\left(V_{10} + \frac{2}{-5+\sqrt{17}}\right)e^{\lambda t} - \frac{2}{5-\sqrt{17}}$	$\left(V_{20} + \frac{8}{-5-\sqrt{17}}\right)e^{\mu t} - \frac{8}{5+\sqrt{17}}$

Tabla 3.3: Soluciones de X_1 y X_2 para los distintos casos.

La ubicación de estas soluciones se encuentran en la figura 3.6.

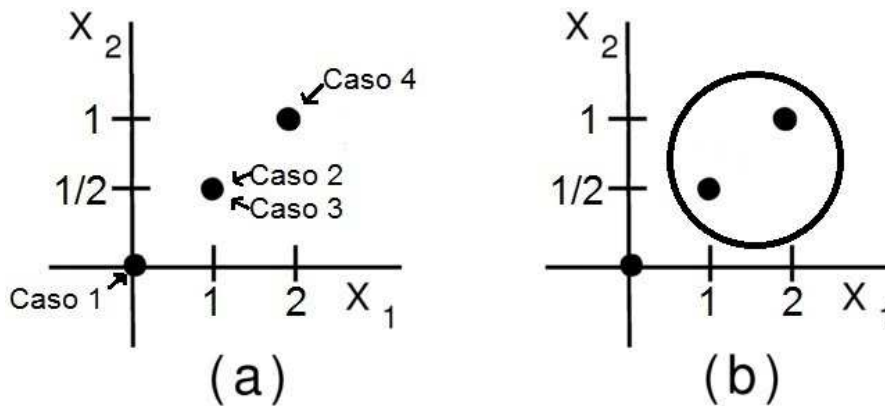


Figura 3.6: Nodos estables.- a) Representación de los cuatro casos, b) Función OR.

Como solo el Caso 1 está fuera del área definida, se tiene entonces que la respuesta obtenida corresponde a la tabla 3.2, la cual es la tabla de verdad de una función lógica OR.

De manera experimental este circuito puede constituirse por componentes de diferentes valores, sin embargo en este proyecto de investigación se replicaron los resultados utilizando resistencias de $10K\Omega$, capacitores de $10nF$ y $20nF$, además se utilizó el comparador LM311 en la configuración ventana, obteniendo los resultados mostrados en la figura 3.7

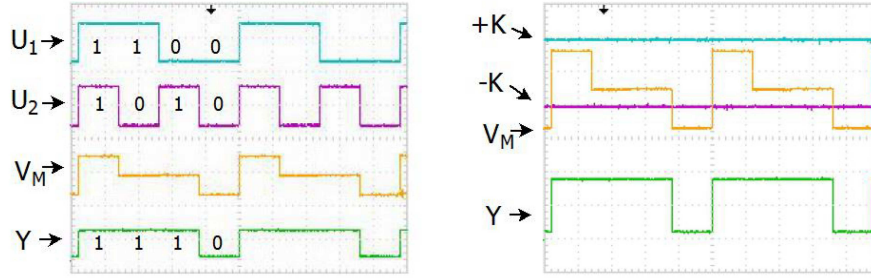


Figura 3.7: Función OR. (a) Señales de entrada a las terminales U_1 y U_2 , potencial en el nodo M y salida Y de la celda lógica. (b) Potenciales de umbral $\pm K$, en el nodo M y de salida Y .

3.3 Celda lógica reconfigurable basada en sistema lineal por partes

En esta sección se expone la idea base de *celda lineal* para este trabajo de investigación, la cual consiste en la metodología de diseño para una celda reconfigurable, basándose en funciones lineales por partes. Para ello, se considera un sistema descrito por las siguientes ecuaciones.

$$F(X) = A^T X + B \quad (3.19)$$

de donde

$$A^T = [a_0, a_1, \dots, a_{N-1}] \quad (3.20)$$

$$X = \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ \dots \\ x_{N-1} \end{bmatrix} \quad (3.21)$$

Con $A^T, X \in R^n$ y $B \in R$, donde A y B contienen los parámetros del sistema a sintonizar y X es el vector de N -entradas, X_i es el número de posibles combinaciones ($i = 2^N$) lógicas y además cada $x_i \in \{0, 1\}$; la salida del sistema esta dada por la siguiente ecuación:

$$g(F(X)) = \begin{cases} 1, & F(X) \in (m_{inf}, m_{sup}); \\ 0, & \text{de otra forma.} \end{cases} \quad (3.22)$$

donde $F(X), m_{inf}, m_{sup} \in R$.

De esta forma la ecuación (3.19) tiene como dominio el conjunto $\{0, 1\}$ y como imagen R , por otro lado, la ecuación (3.22) tiene como dominio el conjunto de los reales contenidos en la región limitada por las constantes m_{sup} y m_{inf} ; y como imagen el conjunto $\{0, 1\}$. De manera general, una celda lógica solo tiene una salida, sin embargo, usando la ecuación (3.22) se contempla $G = [g_1, g_2, \dots, g_j]$; donde j representa el número total de funciones de salida que componen al sistema, siendo así se establece la siguiente descripción de una celda lógica lineal:

Celda lógica lineal: Dadas las funciones $F:M \rightarrow N$ y $G:N \rightarrow Q$ defina una celda lógica como $(GoF):M \rightarrow Q$ tal que para todos los elementos de X de M se tienen Y elementos de Q que satisfacen la ecuación (3.24). De forma general:

$$F(X) = a_0x_0 + a_1x_1 + \dots + a_{N-1}x_{N-1} + b \quad (3.23)$$

$$y_j = (GoF)(X) = \begin{cases} 1, & F(x_i) \in (m_{jInf}, m_{jSup}); \\ 0, & \text{de otra forma.} \end{cases} \quad (3.24)$$

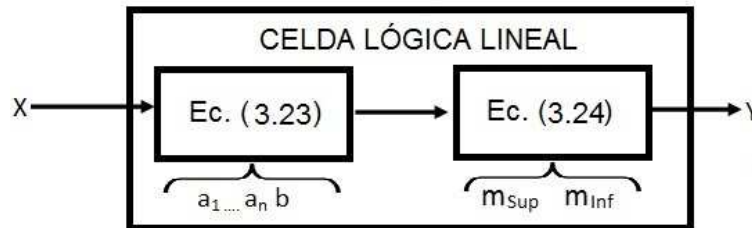


Figura 3.8: Celda lógica propuesta

El diagrama de la figura (3.8) representa a una celda lógica lineal, cabe destacar que se contempla como un sistema simple, el cual tiene la característica de solo contar con una salida independientemente de el numero de entradas, su funcionamiento es descrito mediante el siguiente algoritmo:

1. Definir los valores A^T y la constante b (offset) en la ecuación (3.23) y evalué F para todas las combinaciones posibles.
2. Defina los valores m_{Sup} y m_{Inf} .
3. Con el valor de $F(x_i)$ obtenido del paso anterior evaluarlo en la ecuación (3.24), de donde si $F(x_i)$ está dentro de la región acotada por m_{Sup} y m_{Inf} se tiene que $y_j = 1$ (nivel alto), y si se localiza fuera de la región acotada $y_j = 0$ (nivel bajo).

En la figura 3.9a se representan los posibles valores que la celda lógica puede tomar en base a las permutaciones de las entradas, las líneas punteadas representan con las cotas m_{Inf} y m_{Sup} , las cuales delimitan el umbral de operación. Por otro lado, en la figura 3.9b se muestra la salida y_j para cada permutación en los estados de entrada.

Resumiendo, este tipo de celda lógica reconfigurable se compone básicamente de las funciones descritas por las ecuaciones (3.23) y (3.24) como se representa en la figura 3.9. Hay que tener en mente que los sistemas más sencillos que se pretenden emular son compuertas lógicas y posteriormente incrementar el orden de complejidad pasando por diversos circuitos combinatoriales.

Por lo tanto cabe mencionar que como tal una compuerta lógica puede tener una cantidad "n" de entradas, pero una característica muy peculiar es que solo cuenta con una salida. Convencionalmente se suelen encontrar en presentaciones de 2, 3 y hasta 4 entradas como se observan

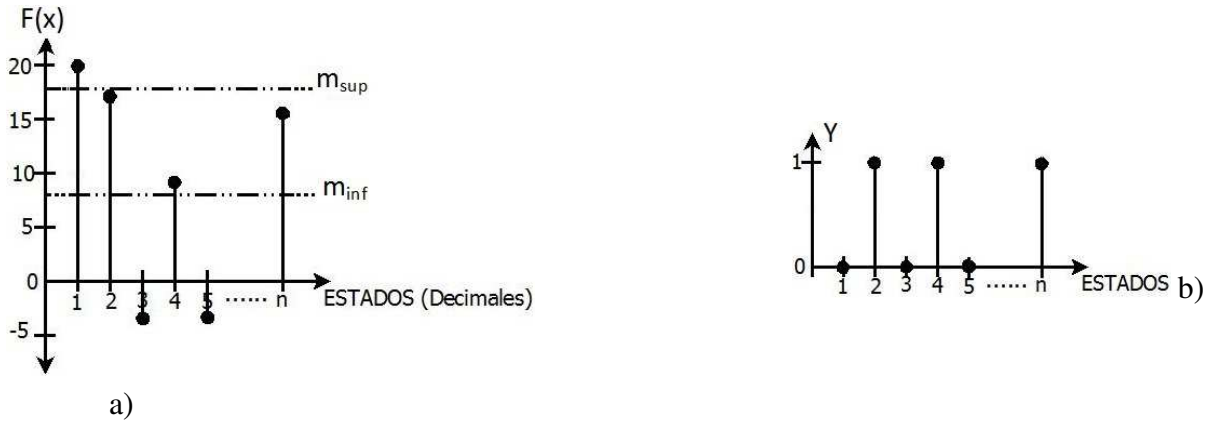


Figura 3.9: a) Posibles valores para las diferentes permutaciones de la ecuación (3.23); b) Representación gráfica de la ecuación (3.24) por estados

en los diagramas de los circuitos TTL de la figura 3.10. Para los casos de compuertas con un mayor número de entradas, es necesario combinar varias de estas, de tal manera que se hacen arreglos triviales en cascada como los que se muestran en la figura 3.11. Las compuertas que se emularán mediante el diseño de la celda lógica reconfigurable son de solo dos entradas.

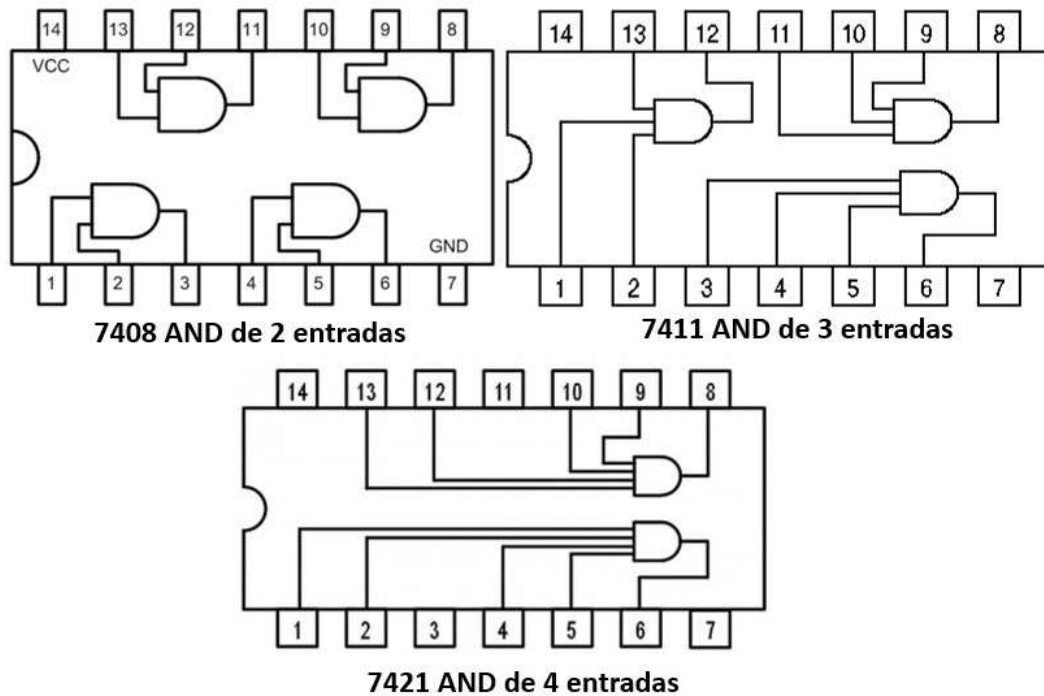


Figura 3.10: Compuertas AND comerciales.

En la sección 5 se presenta el circuito equivalente de éste sistema.

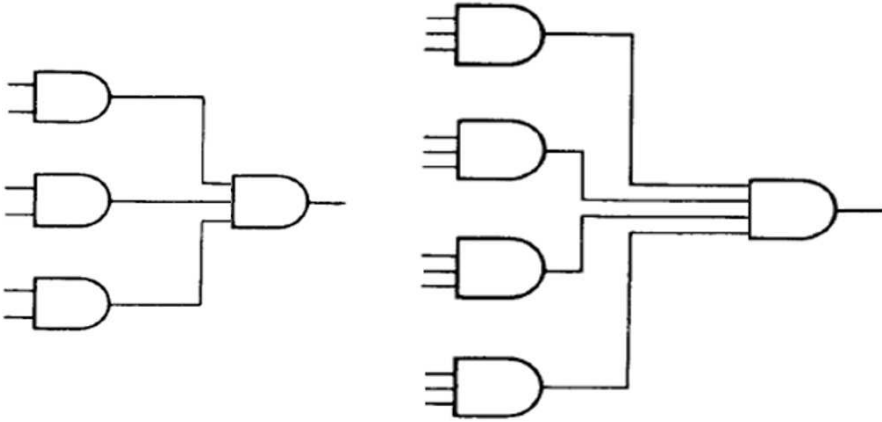


Figura 3.11: Arreglos en cascada para formar compuertas de múltiples entradas.

4

Diseño teórico de compuertas lógicas

En esta sección se presentan las consideraciones que se tomaron para el diseño de compuertas lógicas utilizando el modelo de la celda lógica reconfigurable presentado en la sección 3.3.

Retomando el esquema propuesto en la figura 3.8 y considerándolo para una compuerta AND de dos entradas, se observa que existirán 4 permutaciones en las entradas, lo que consecuentemente conlleva a 4 posibles estados en la función descrita por la ecuación 3.23. Estos estados dependen directamente de los valores que se asignen en A^T y del valor que se le de a B , existen múltiples opciones para esta asignación, siendo la manera más trivial $B = 0$ y el distribuir los demás valores en potencias de dos tal que:

$$A^T = [2^0, 2^1] = [1, 2] \tag{4.1}$$

Mediante esta asignación se generan los estados mostrados en la tabla 4.1, en la figura 4.1 se pueden observar los cuatro estados para las permutaciones de entrada, siendo el último de ellos el que debe estar dentro del umbral de operación, para este caso delimitado por la región entre $m_{Sup} = 3.5$ y $m_{Inf} = 2.5$, y así emular el comportamiento de una compuerta AND de dos entradas.

En la figura 4.1 se presentan los estados que se generan por la ecuación (3.23) para este caso particular, mientras que en la figura 4.2 se muestran las respuestas de salida en la ecuación (3.24) para cada uno de estos estados.

Permutaciones	$a_1 = 2^1$	$a_0 = 2^0$	$F(X) = a_0x_0 + a_1x_1 + B$	Salida AND
Estado 0	$x_1 = 0$	$x_0 = 0$	0	0
Estado 1	$x_1 = 0$	$x_0 = 1$	1	0
Estado 2	$x_1 = 1$	$x_0 = 0$	2	0
Estado 3	$x_1 = 1$	$x_0 = 1$	3	1

Tabla 4.1: Combinación de las entradas en el sistema lineal.

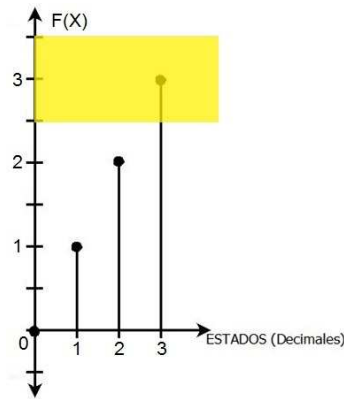


Figura 4.1: Estados para una compuerta de dos entradas.

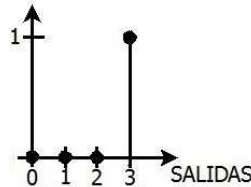


Figura 4.2: Salidas de la celda lógica.

En esencia es relativamente simple extrapolar para el caso de una compuerta AND de "n" entradas, ya que independientemente de los parámetros de sintonización que se elijan, al final solo se debe considerar el último estado, por lo tanto las siguientes reglas funcionan para esta asignación:

- n = Número de entradas.
- 2^n = Número de estados.
- $i = 0, 1, 2, \dots, n - 1$.
- $i, n \in \mathbb{R}$.
- $B = a_0$.
- $a_{i+1} = 2a_i$.
- $\pm m = a_0 * [2^n \pm 0.5]$.

En las figuras 4.3 y 4.4 se muestra el caso general de una compuerta AND de "n" entradas, en la figura 4.3 se representan los estados que se generan para la ecuación (3.23) y en la figura 4.4 las respuestas de salida en la ecuación (3.24).

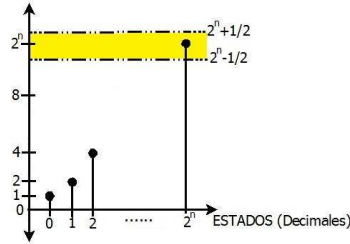


Figura 4.3: Estados para una compuerta AND de n entradas.

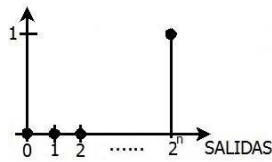


Figura 4.4: Salidas de la celda lógica en modo AND de n entradas.

De una manera semejante se puede generalizar este concepto para una compuerta OR de "n" entradas, la diferencia radicaría en la condición de los límites del umbral de operación, el cual se definiría como:

- n = Número de entradas.
- 2^n = Número de estados.
- $i = 0, 1, 2, \dots, n - 1$.
- $i, n \in \mathbb{R}$.
- $B = a_0$.
- $a_{i+1} = 2a_i$.
- $m_{Sup} = a_0 * [2^n + 0.5]$.
- $m_{Inf} = 3a_0/2$.

En las figuras 4.5 y 4.6 se presenta el caso general de una compuerta OR de "n" entradas, en la figura 4.5 se representan los estados que se generan para la ecuación (3.23) y en la figura 4.6 las respuestas de salida en la ecuación (3.24). Observe como en comparación del caso AND de "n" entradas, la diferencia consiste en la apertura del umbral de operación.

Esta técnica es sencilla de implementar, sin embargo no es posible obtener una gama completa de compuertas lógicas manejando solo este concepto, ya que los miniterminos que componen las diversas funciones no pueden ser contemplados con un solo umbral de operación.

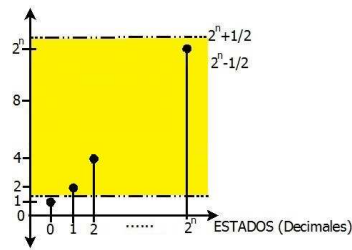


Figura 4.5: Estados para una compuerta OR de n entradas.

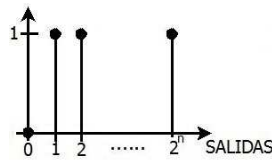


Figura 4.6: Salidas de la celda lógica en modo OR de n entradas.

A continuación, suponga el caso donde se desea que una sola celda lógica reconfigurable pueda emular el comportamiento de todas las funciones booleanas de dos entradas.

En la tabla 4.2 se presentan las 16 funciones booleanas correspondientes a un sistema de dos entradas, donde se aprecia la relación entrada-salida de cada una de estas. Por otro lado, en la tabla 4.3 se exponen las funciones booleanas de manera tal que sean vistas como una operación lógica.

X_1	X_0	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Tabla 4.2: Funciones booleanas que se pueden obtener con una compuerta de dos entradas.

Función	Función equivalente
F_0	Función Zero, todas las permutaciones generan cero a la salida.
F_1	Función X_0 NOR X_1 .
F_2	Función $X_0 > X_1$.
F_3	Función $\overline{X_1}$.
F_4	Función $X_1 > X_0$.
F_5	Función $\overline{X_0}$.
F_6	Función X_0 XOR X_1 .
F_7	Función X_0 NAND X_1 .
F_8	Función X_0 AND X_1 .
F_9	Función X_0 XNOR X_1 .
F_{10}	Función X_0 .
F_{11}	Función $\overline{X_1} > X_0$.
F_{12}	Función X_1 .
F_{13}	Función $X_0 > \overline{X_1}$.
F_{14}	Función X_0 OR X_1 .
F_{15}	Función One, todas las permutaciones generan uno a la salida.

Tabla 4.3: Operaciones booleanas para compuertas de dos entradas.

Analizando las tablas 4.2 y 4.3 se observan que las funciones están divididas en dos conjuntos, uno de estos conjuntos corresponde a las funciones que se pueden realizar tomando la respuesta lineal arrojada por la asignación de parámetros que se propuso en la ecuación (4.1), en la figura 4.7 se puede observar una gráfica donde se representa su respectivo incremento lineal.

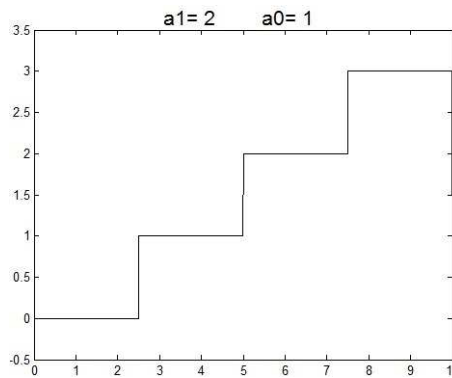
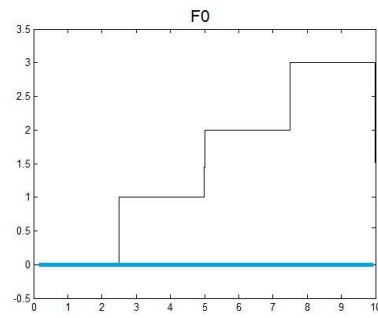
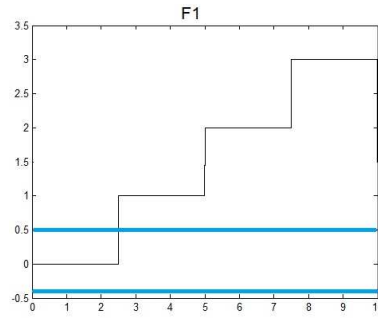


Figura 4.7: Respuesta con incremento lineal de la ecuación (3.23) utilizando la asignación de la ecuación (4.1).

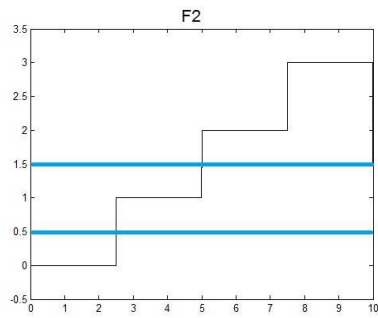
En la Figura 4.7 se muestra la respuesta con incremento lineal que se obtiene en la ecuación (3.23) si se asignan los valores de acuerdo a la ecuación (4.1), observe que en los diversos casos presentados en la figura 4.8 se utiliza la misma respuesta y con variaciones del umbral de operación. De esta manera es posible obtener las funciones $F_0, F_1, F_2, F_3, F_4, F_6, F_7, F_8, F_{12}, F_{14}$ y F_{15} .



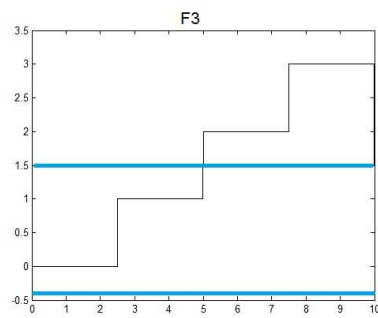
(a) F_0



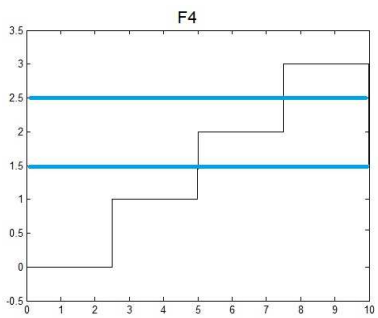
(b) F_1



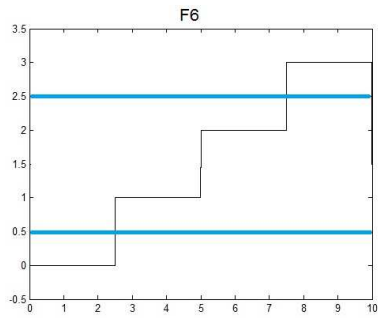
(c) F_2



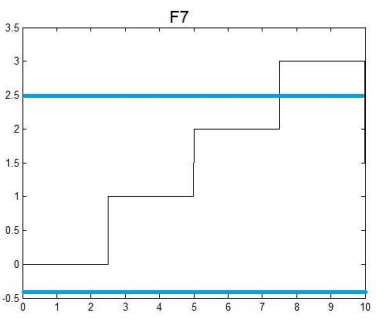
(d) F_3



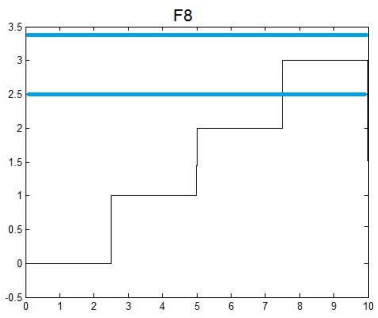
(e) F_4



(f) F_6



(g) F_7



(h) F_8

Para obtener el conjunto de funciones faltantes existen dos posibles vías para conseguirlo:

- Utilizar múltiples umbrales de operación.

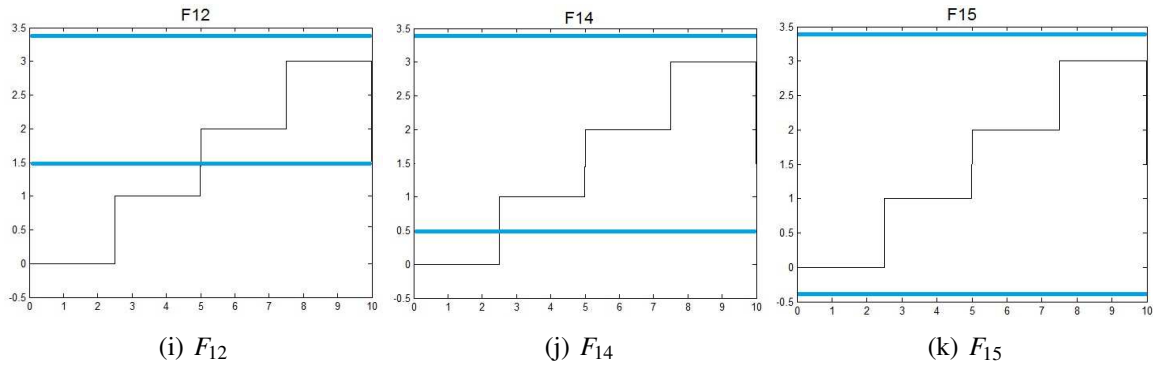


Figura 4.8: Funciones que son obtenidas utilizando un solo umbral de operación y una respuesta lineal creciente en la ecuación (3.23).

- Variar los valores del vector en la ecuación (4.1) obteniendo una respuesta lineal por partes en la ecuación (3.23).

En esta sección se profundizará en el manejo de una respuesta lineal por partes en (3.23), para ello se modifica el vector de la ecuación (4.1) de la siguiente manera:

$$A^T = [2^0, -2^1] = [1, -2] \tag{4.2}$$

De esta manera se obtiene la respuesta presentada en la figura 4.9, donde se aprecia que la función toma tanto valores positivos como negativos, formando así una función lineal por partes.

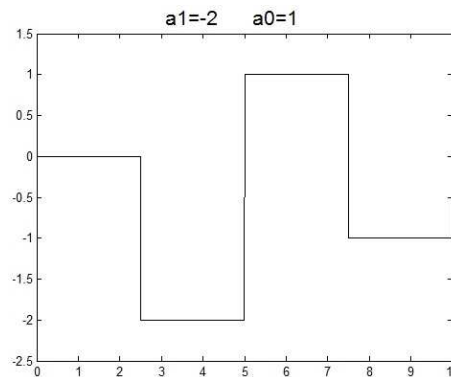


Figura 4.9: Respuesta lineal a trozos para la ecuación (3.23) utilizando la asignación de la ecuación (4.2).

De manera complementaria al caso anterior se presentan en la figura 4.10 como es posible obtener las funciones F_5 , F_9 , F_{10} , F_{11} y F_{13} , con solo variar un solo umbral de operación.

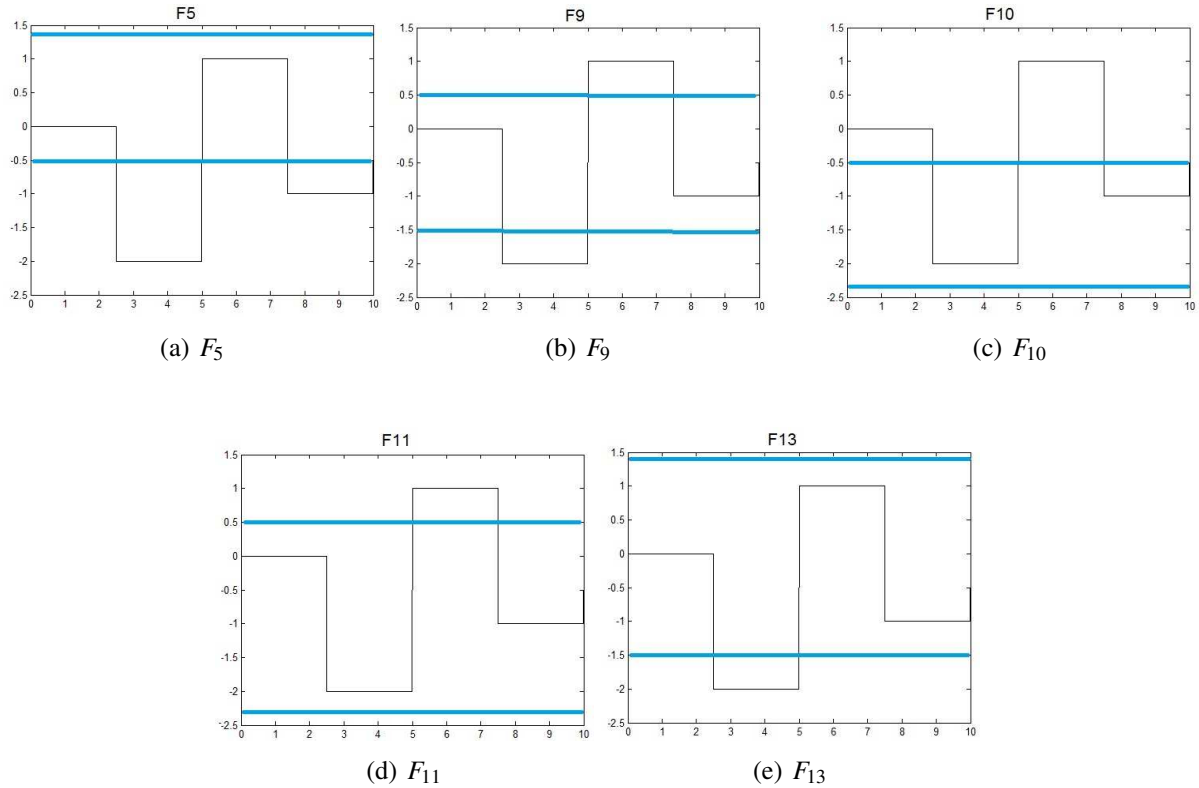


Figura 4.10: Funciones que son obtenidas utilizando un solo umbral de operación y una respuesta lineal a trozos en la ecuación (3.23).

Este método es sensible al momento de incrementar el número de entradas en la ecuación (3.23), ya que aumentan las posibilidades de asignación de los valores A^T , complicando así la selección de parámetros, en el sentido de que sea posible obtener todas las configuraciones posibles mediante un solo umbral de operación; ecuación (3.24).

Una propuesta que garantiza la solución de esta situación, es el manejo de una cantidad mínima de umbrales, tal que para:

- n = Número de entradas.
- 2^n = Número de estados (permutaciones).
- 2^{2^n} = Número de posibles funciones a realizar con n entradas.
- 2^{n-1} = Número de umbrales necesarios para una compuerta de n entradas.

Al tener una cantidad 2^{n-1} de umbrales, es posible representar cualquier función, ya que independientemente del número de entradas, el caso más complejo sucede cuando los miniterminos toman valores intercalados en forma de una secuencia de $0, 1, 0, 1, \dots, 0, 1$ ó $1, 0, 1, 0, \dots, 1, 0$. Cumpliendo así que solo la mitad de todas las permutaciones deban tomar un valor de 1 a la salida.

Es prudente mencionar que la asignación de los parámetros de configuración (A^T en la ecuación (3.20), m_{inf} y m_{sup} en la ecuación (3.22)), se asignaron inicialmente de una manera arbitraria sin embargo la sección 6.1.3 se menciona sobre el algoritmo de obtención de los mismos.

Circuitos reconfigurables

Uno de los objetivos principales en el área de la arquitectura de computadoras es desarrollar nuevos paradigmas que sean capaces de disminuir el tiempo de procesamiento y el consumo de energía del sistema. En este sentido se genera el desarrollo de dispositivos digitales programables y la creación de lenguajes de descripción de hardware como VHDL, que es un lenguaje de alto nivel y se utiliza para describir distintos circuitos digitales que pueden ser programados en un FPGA [27].

Un FPGA es un arreglo de bloques lógicos programables, el cual proporciona la ventaja de un procesador de propósito general y la de un circuito especializado que se puede reconfigurar una gran cantidad de veces. Si se mide la densidad de los elementos lógicos programables en compuertas lógicas equivalentes se podría decir que en un FPGA se encuentran del orden de millones de estas.

Basados en esta idea de reconfiguración, se abre la posibilidad de crear circuitos de propósitos específicos con la finalidad de ofrecer un mayor rendimiento para ciertas aplicaciones y además de convertirse en una fuente importante de investigación para el diseño de hardware-software. Así la arquitectura de computadoras y los sistemas digitales cobran gran importancia en esta área de investigación.

Partiendo de esta filosofía y de forma general, refiriéndose al diseño de chips electrónicos, surge la pregunta ¿Será posible realizar un sistema físico que genere diferentes funciones lógicas, a través de sintonizar los parámetros del mismo?

Buscando dar una respuesta a esta situación en este capítulo se presentan varios de los resultados obtenidos en este trabajo de investigación. Se parte desde una estructura básica que permite emular diversas compuertas lógicas, posteriormente mediante la agrupación de varias estructuras de este mismo tipo, se presentan los modelos algunos circuitos lógicos combinatoriales, flip-flops y finalmente se presenta la propuesta para el diseño de una unidad

lógica aritmética (ALU).

5.1 Compuerta lógica reconfigurable

Los diseños se implementaron sobre una tarjeta de circuito impreso (PCB). Para el circuito experimental se utilizó el amplificador operacional TL081, el comparador LM311 en la configuración ventana, resistencias con $\pm 15\%$ de tolerancia con los valores dados en las figuras para cada circuito combinacional; para alimentar los circuitos integrados una fuente de $\pm 15\text{V}$, 3A modelo PS280 de la compañía Tektronix, en cuanto a la captura de las señales se utilizó un osciloscopio Tds2014 de la compañía Tektronix, finalmente para generar las secuencias de entrada y comprobar su correcto funcionamiento de forma experimental se emplea un microcontrolador PIC16f877A de la compañía Motorola.

En todos los experimentos realizados se considera como nivel alto el potencial de 2.3V para las entradas (x_i), mientras que el nivel bajo corresponde a 0V; con el objetivo de obtener los parámetros indicados en la sección 4, se considera multiplicar el voltaje de entrada por las ganancias que se asignan en cada diseño mediante los ajustes en las diferentes resistencias de cada uno de los circuitos lógicos combinacionales.

En la figura 5.1 se presenta un arreglo de tres amplificadores operacionales; un amplificador en modo sumador inversor (TL081) y los otros dos en configuración de comparador ventana (LM311). Este diseño corresponde a una **compuerta lógica reconfigurable** de dos entradas, la cual es contemplada para emular la función OR.

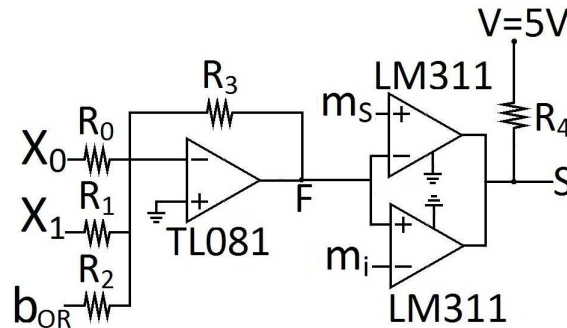


Figura 5.1: Diagrama de compuerta lógica reconfigurable.

Este diseño esta basado en las ecuaciones 3.23 y 3.24, donde se observa que el potencial en el nodo F esta dado por la siguiente expresión:

$$F_{OR}(X) = -\frac{R_3}{R_0}x_0 - \frac{R_3}{R_1}x_1 - \frac{R_3}{R_2}b_{OR} \quad (5.1)$$

La ecuación 5.1 es análoga a la ecuación 3.23, observe que las ganancias de los pesos en A^T están dadas por las relaciones entre las resistencias, como se muestra a continuación:

$$a_0 = -\frac{R_3}{R_0}; a_1 = -\frac{R_3}{R_1} \text{ y } b = -\frac{R_3}{R_2}$$

En esta parte del sistema es donde se define que se trata de una compuerta de dos entradas (x_0, x_1). En el caso de donde sea necesario contemplar más entradas, sería necesario agregar una resistencia más en esta sección, de tal manera que el peso a_n estaría definido por:

$$a_n = -\frac{R_N}{R_n}$$

Donde R_N correspondería a la resistencia de referencia y n a la n -ésima resistencia de cada entrada.

La segunda parte del diagrama presenta un par de amplificadores operacionales (*LM311*) en modo de circuito ventana, esta configuración recibe este nombre ya que acota un umbral de operación mediante las variables m_i y m_s tal como se describe en la siguiente ecuación:

$$S = \begin{cases} 5, & F(x_i) \in (m_{jInf}, m_{jSup}); \\ 0, & \text{de otra forma.} \end{cases} \quad (5.2)$$

Hay que mencionar que el valor de 5V que se asigna en la resistencia R_4 , es meramente con objetivo de obtener una salida de 5V, al igual que compuerta lógica convencional, sin embargo este voltaje puede provenir de diversas fuentes y no estrictamente de una fuente constante como se presenta en este caso.

Algunos valores que permiten obtener la función OR son:

- $x_0a_0 = -1$
- $x_1a_1 = -2$
- $b = 3.5$

Para obtener estos valores inicialmente se tomaron las salidas de un PIC (16F877A) para las señales de conmutación de x_0 y x_1 , posteriormente para obtener la constante de b se tomo también un pin del mismo PIC, pero para obtener el valor negativo se hizo pasar esta señal por un inversor. Esto para poder complementar el cambio de signo que conlleva la configuración utilizada para el amplificador (TL081).

De tal manera que los valores en las resistencias empleadas son:

R_0	23k Ω
R_1	11.5k Ω
R_2	6.5k Ω
R_3	10k Ω
R_4	500 Ω

Tabla 5.1: Valores de resistencias en una compuerta reconfigurable en modo OR.

En la tabla 5.2 se presentan el conjunto de permutaciones que tiene el sistema y los voltajes que toma el nodo F_{OR} para cada caso.

x_1	x_0	$F(X)$	F_{OR}	S
0	0	b	3.5	0
0	1	$x_0a_0 + b$	2.5	1
1	0	$x_1a_1 + b$	1.5	1
1	1	$x_0a_0 + x_1a_1 + b$	0.5	1

Tabla 5.2: Respuesta con incremento lineal en voltajes de permutación de compuerta lógica reconfigurable.

En la figura 5.2, se presenta la respuesta con incremento lineal (*línea morada*) que produce esta parte de la compuerta lógica. Al tratarse de una compuerta lógica OR, solo la primera permutación es la que debe estar fuera del umbral de operación, debido a ello se asignan $m_{sup} = 3V$ (*línea amarilla*) y $m_{inf} = 0V$ (*línea azul*), obteniendo a la salida del umbral de operación la respuesta S (*línea verde*).

Hay que denotar que el sistema tiene dos formas de reconfiguración:

- Modificación de los parámetros del umbral de operación.
- Modificación de los pesos en las entradas mediante la variación de las resistencias.

Al realizar cambios en el *umbral de operación*, este se puede ampliar al momento de tomar dos constantes que permitan envolver más permutaciones y en el caso contrario, acercar estas constantes de tal manera que solo una permutación sea contemplada.

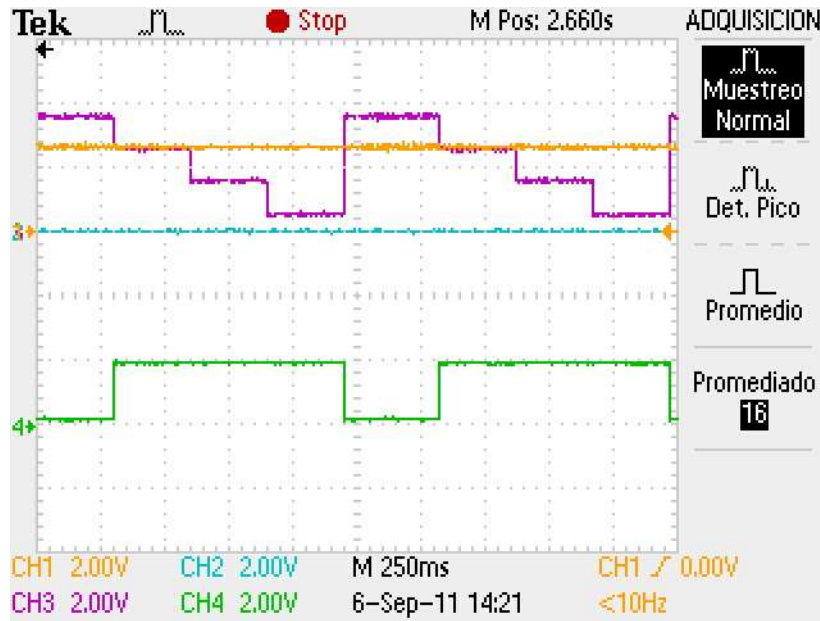


Figura 5.2: Respuesta de compuerta lógica reconfigurable en modo OR.

La limitante que se presenta con esta opción sucede al contemplar el caso de la compuerta XNOR, ya que al estar solo activos los miniterminos correspondientes a la primer y ultima permutación, resulta imposible considerarlos con un solo umbral de operación, siendo la mejor solución para esta situación el *cambio en los pesos de las entradas*.

Este cambio se puede llevar de dos formas, un caso en que se contempla que los valores que se asignen a los pesos de las entradas, sean solo valores positivos o negativos y un segundo caso donde estos valores correspondan a todos los elementos del conjunto de los números reales.

Por lo tanto, la combinación de estas dos soluciones permite contemplar un esquema donde sea posible utilizar valores positivos y negativos en los pesos de las entradas además de la implementación de un mínimo de dos umbrales de operación.

Modificando así, el esquema inicial por el que se presenta en la figura 5.3, donde se puede percibir que los valores en las entradas a_n toman una ganancia negativa por la terminal inversora del amplificador operacional, mientras que para las entradas b_n , la ganancia es positiva. Por tratarse de una configuración de sumador-restador la ecuación que describe las ganancias es:

$$F(X) = -\frac{R_{A4}}{R_{A1}}a_1 - \frac{R_{A4}}{R_{A2}}a_2 - \frac{R_{A4}}{R_{A3}}a_3 + \frac{R_{A4}}{R_{B1}}b_1 + \frac{R_{A4}}{R_{B2}}b_2 + \frac{R_{A4}}{R_{B3}}b_3 \quad (5.3)$$

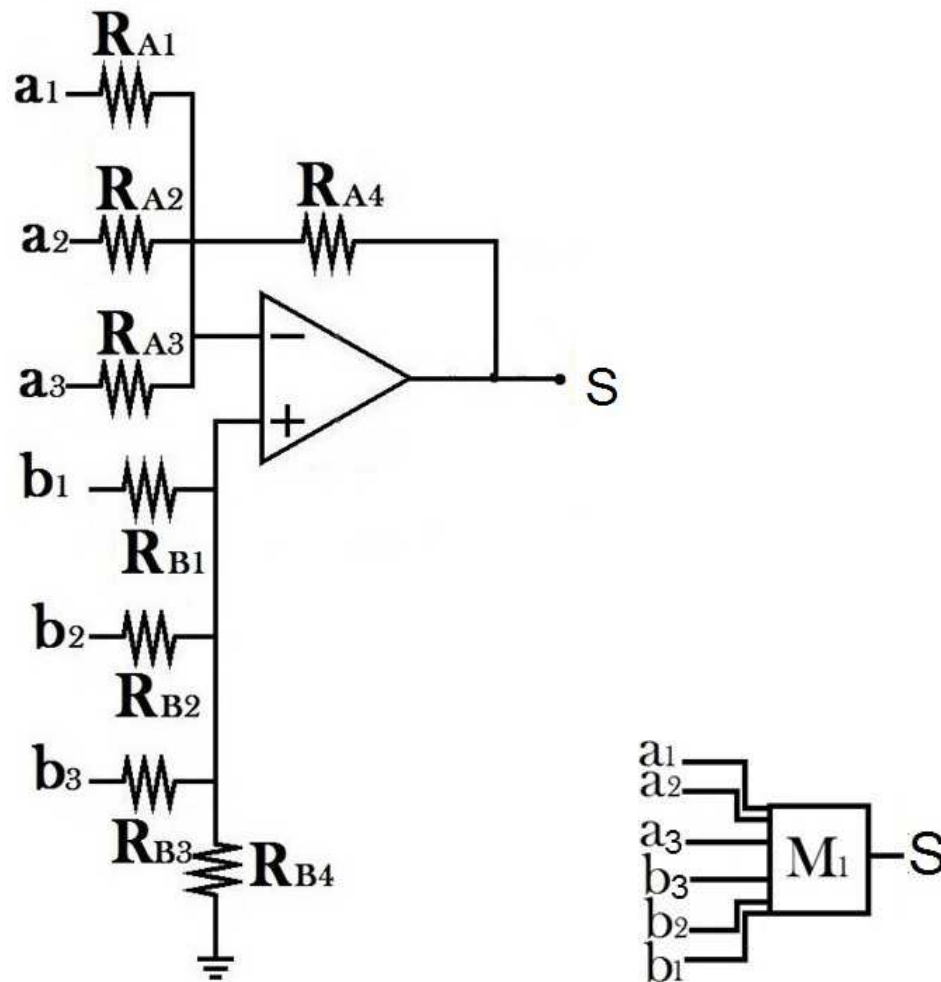


Figura 5.3: Amplificador operacional en modo de sumador restador.

Ahora bien, contemplando un esquema de un solo umbral de operación y que permite el manejo de pesos con valores positivos y negativos, es posible obtener todas las compuertas mostradas en la tabla 4.3. Durante éste trabajo de investigación se diseñó un prototipo en una placa de montaje superficial, el cual corresponde a esta propuesta, en la figura 5.5 se presenta el diagrama de bloques y en la figura 5.6 una fotografía de éste prototipo .

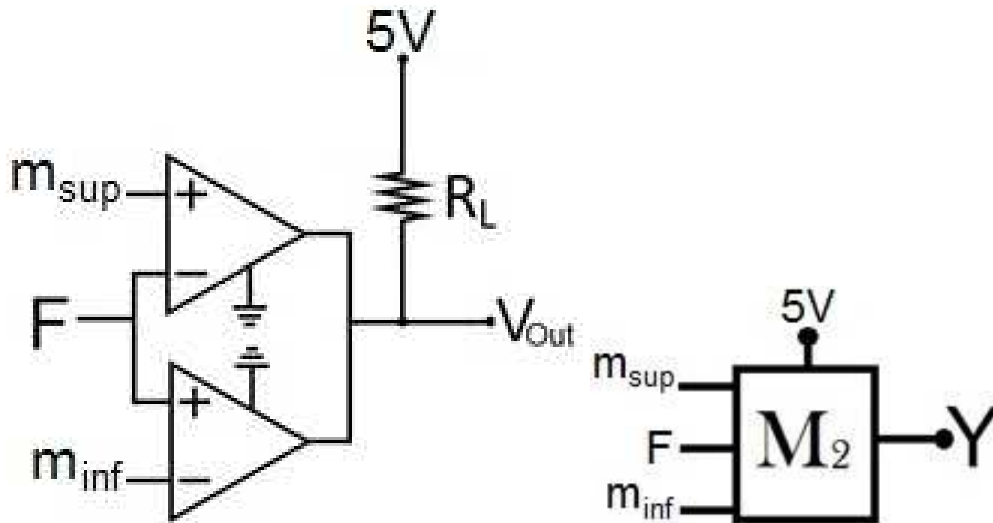


Figura 5.4: Arreglo de amplificadores operacionales en modo de circuito ventana.

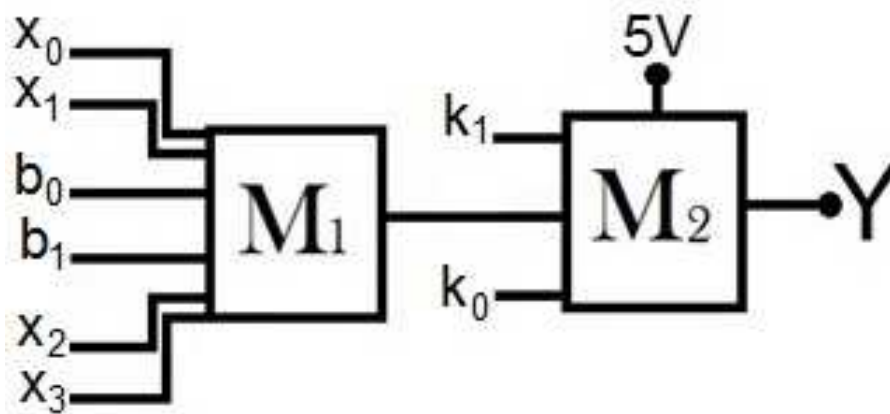


Figura 5.5: Diagrama de bloques para diseño de celda lógica reconfigurable.

En el lado izquierdo de la imagen aparecen las entradas del sistema, con X_1 , X_0 y B_0 siendo los terminales con pesos negativos de -2, -1 y -1 respectivamente; con X_3 , X_2 y B_1 como terminales con pesos positivos de 2, 1 y 1. En este mismo lado aparecen los pines K_0 y K_1 , los cuales corresponden a las variables de los límites del umbral de operación m_{inf} y m_{sup} respectivamente.

En el lado opuesto del circuito se encuentran los pines de salida del sistema, indicados en V_{out1} y V_{out2} , siendo el primero el inverso del segundo. También aparecen los pines de alimentación de los circuitos, $-V_{cc}(-12V)$, $+V_{cc}(+12V)$ y GND .

Como se vió en la sección 4, para una compuerta de 2 entradas existen un total de un conjunto de 16 funciones booleanas, las cuales pueden ser obtenidas utilizando una *respuesta con incremento lineal* (figura 4.7) o mediante una *respuesta lineal a trozos* (figura 4.9), a continuación se presentan los resultados obtenidos con este prototipo.

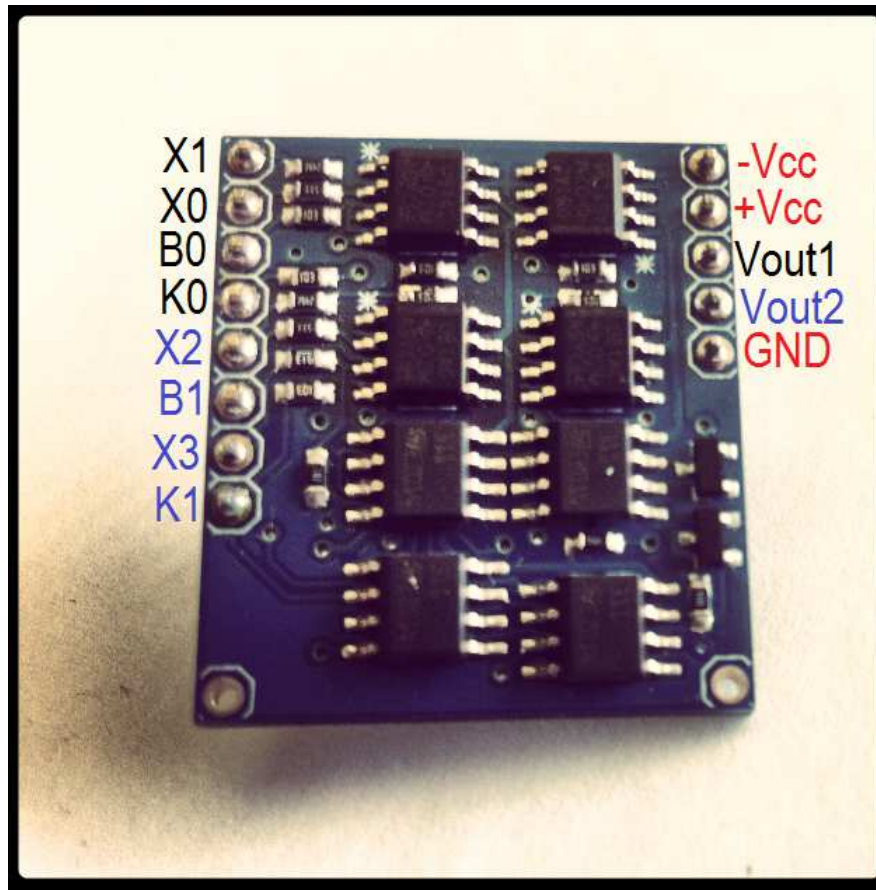


Figura 5.6: Circuito de montaje superficial de celda lógica reconfigurable.

En primera instancia se presenta la respuesta *con incremento lineal* que se obtiene con el circuito de la figura 5.6, en la figura 5.7 se presenta la señal de esta respuesta, de donde en colores amarillo y azul se pueden apreciar las señales de entrada x_1 y x_0 respectivamente, de color morado se representan las cuatro permutaciones que estas generan (00, 01, 10, 11).

Los resultados de las compuertas lógicas se interpretan mediante dos imágenes, las imágenes correspondientes en color azul se muestra la cota superior del umbral de operación $k+$, en color morado la cota inferior del umbral de operación $k-$, de color amarillo las permutaciones de la celda lógica F y en color verde la salida Y del sistema, mientras que en *b*) se representan los estados de entrada x_1 y x_0 , en colores amarillo y azul respectivamente, y en la parte inferior la salida Y del sistema.

En las figuras 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.17 y 5.18 se presentan los resultados de las funciones $F_0, F_1, F_2, F_3, F_4, F_6, F_7, F_8, F_{12}, F_{14}$ y F_{15} .

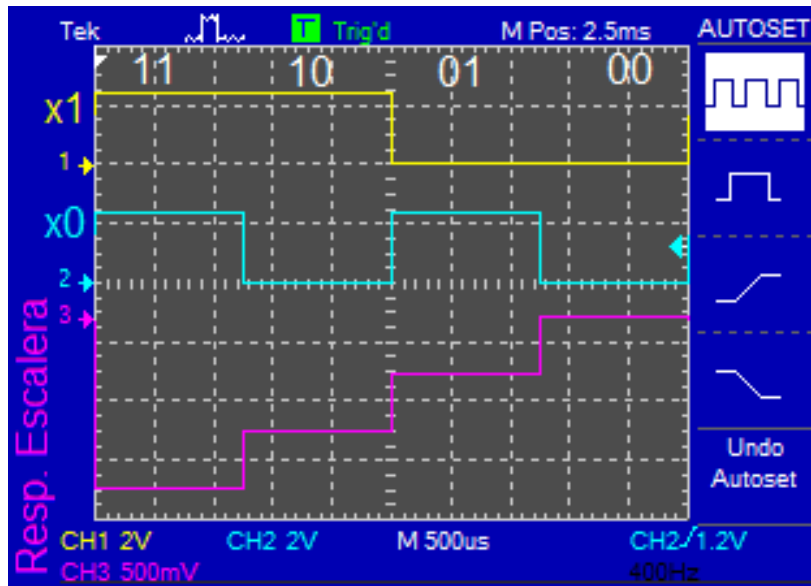
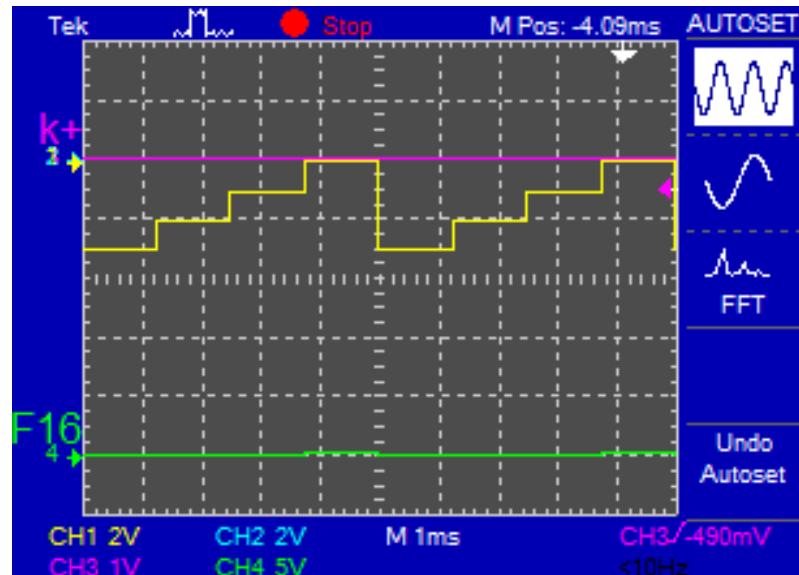
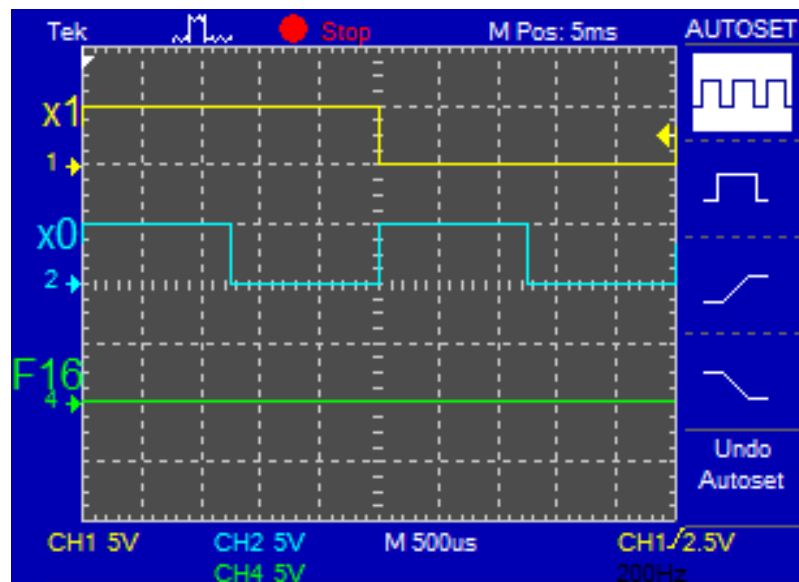


Figura 5.7: Respuesta con incremento lineal.

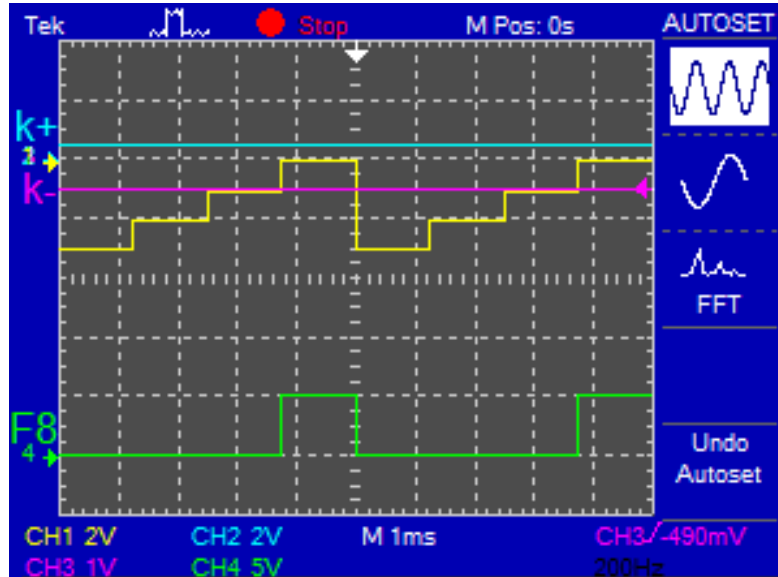


a)

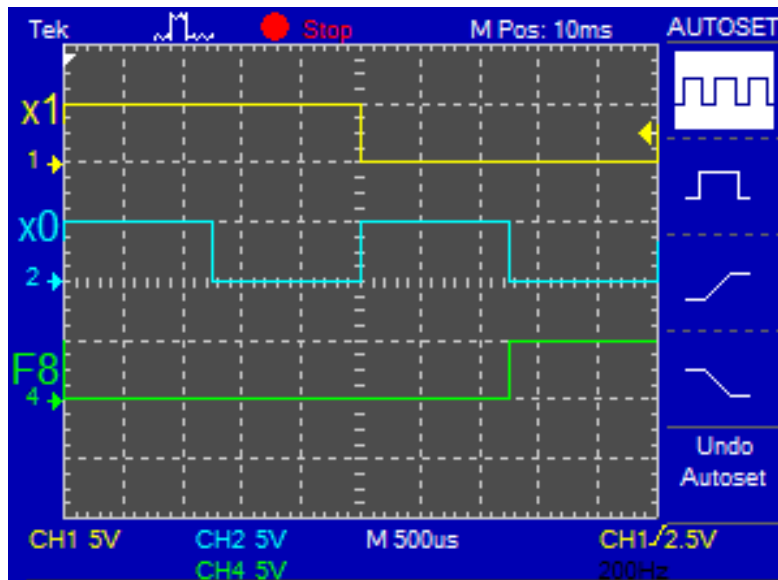


b)

Figura 5.8: Función F_0 .

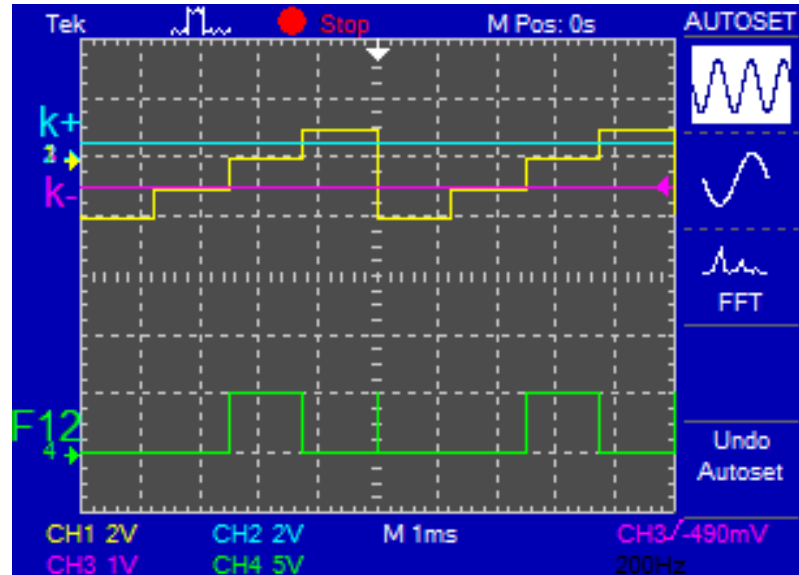


a)

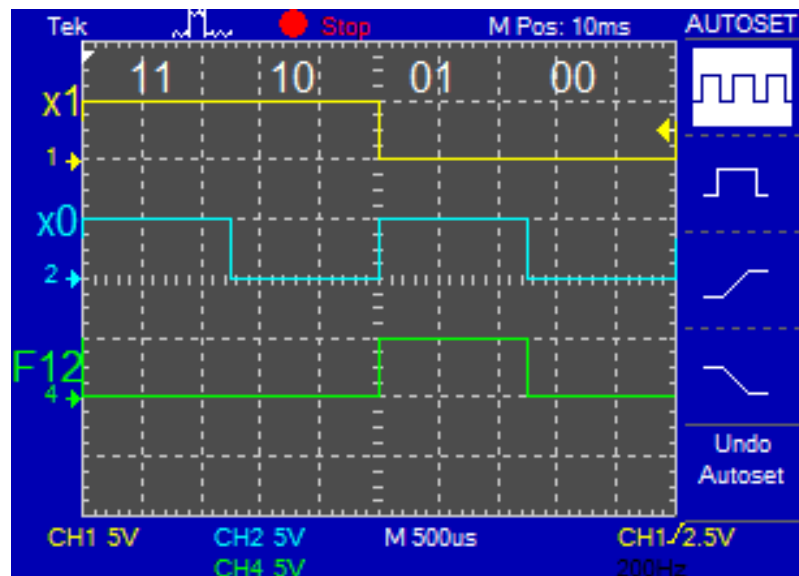


b)

Figura 5.9: Función F_1 .

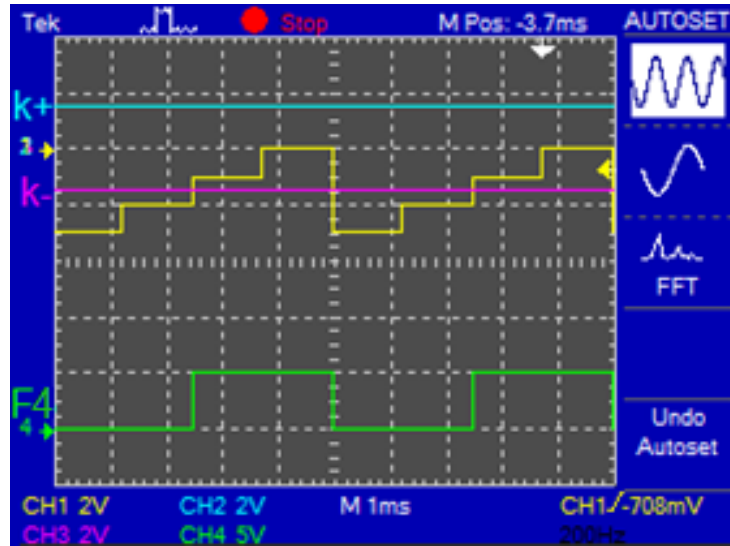


a)

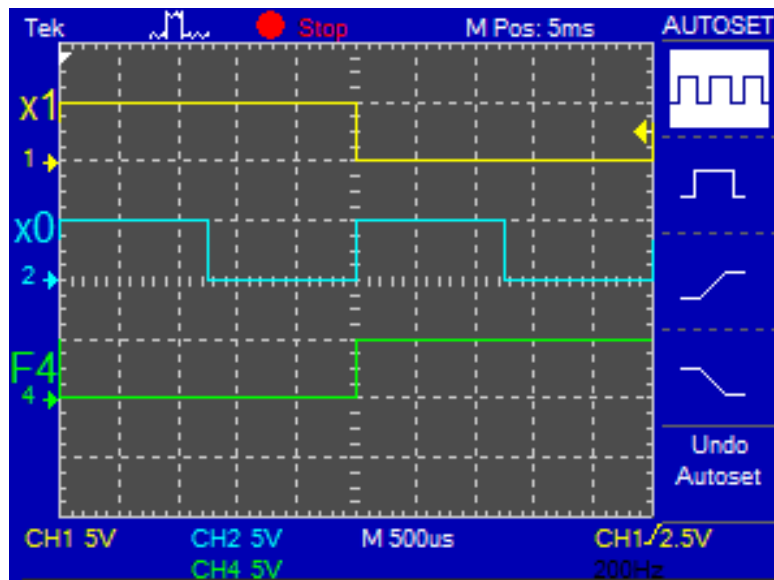


b)

Figura 5.10: Función F_2 .

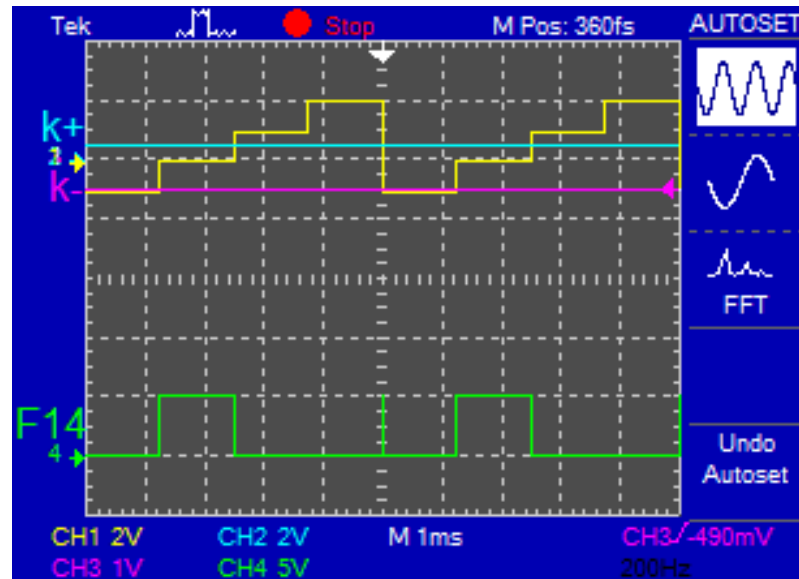


a)

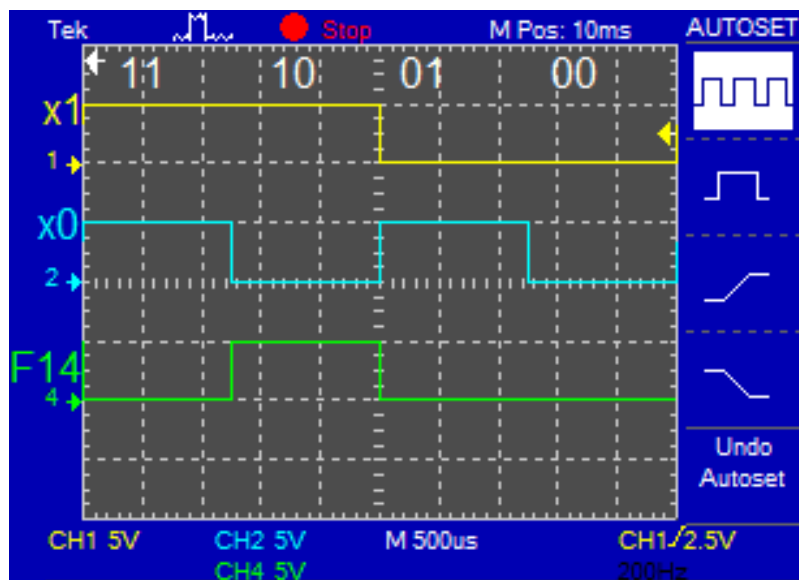


b)

Figura 5.11: Función F_3 .

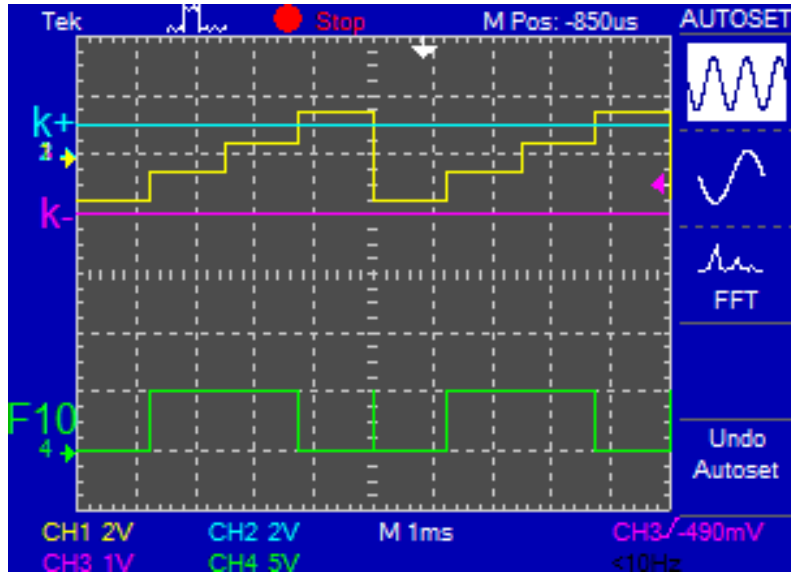


a)

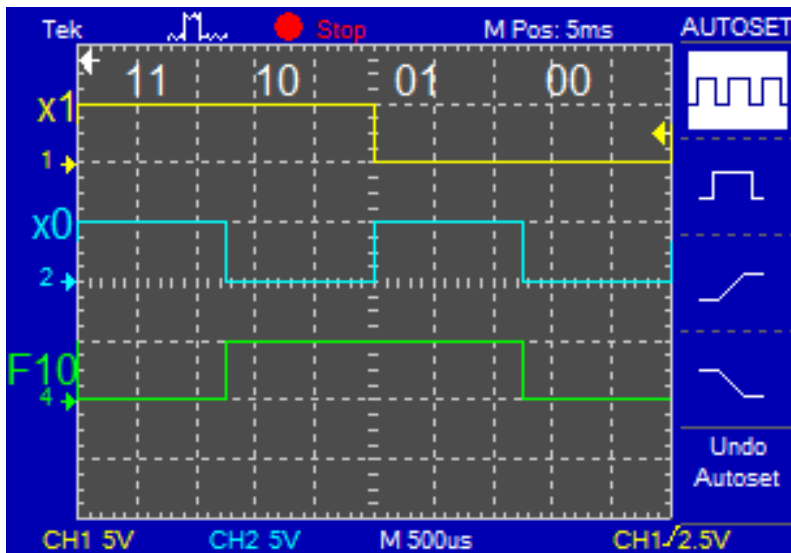


b)

Figura 5.12: Función F_4 .

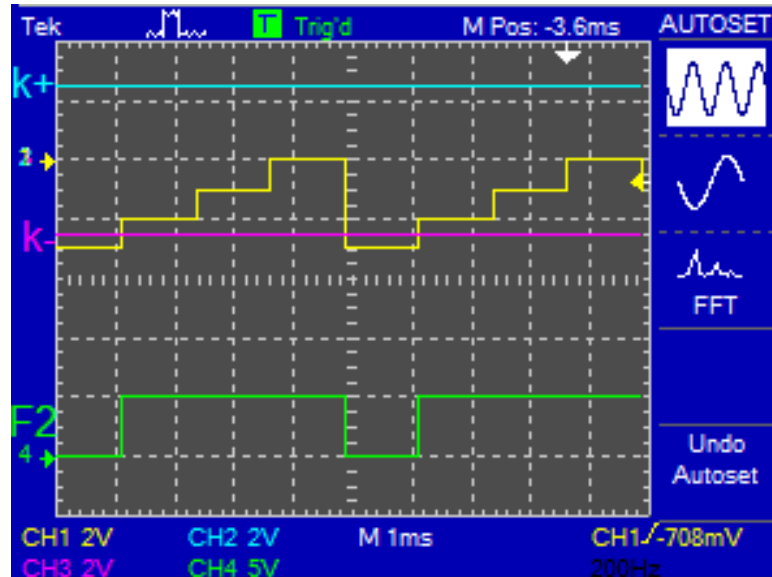


a)

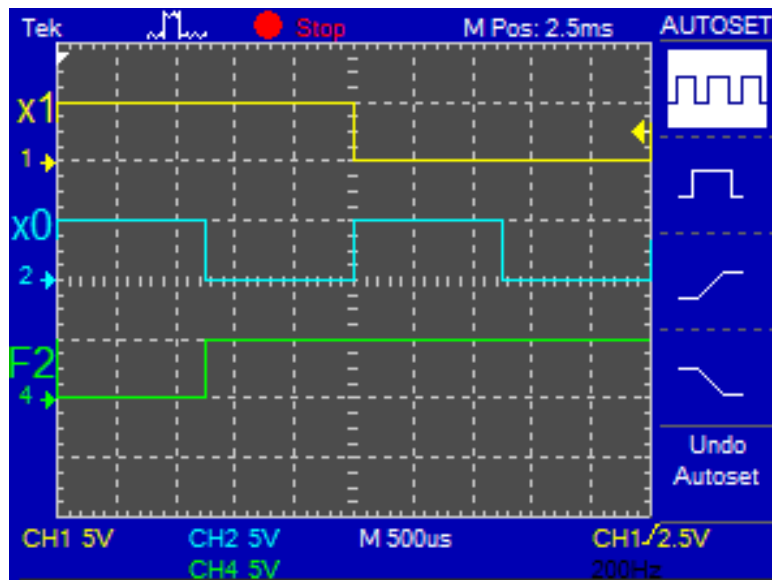


b)

Figura 5.13: Función F_6 .

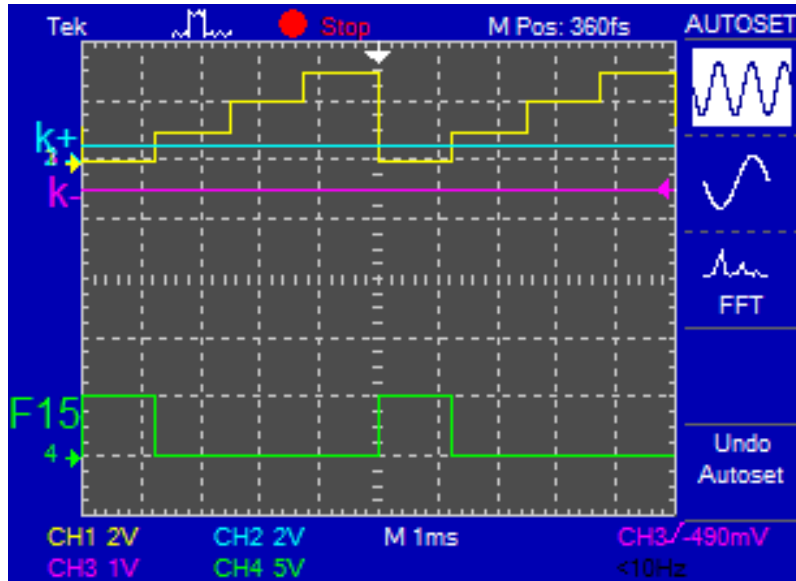


a)

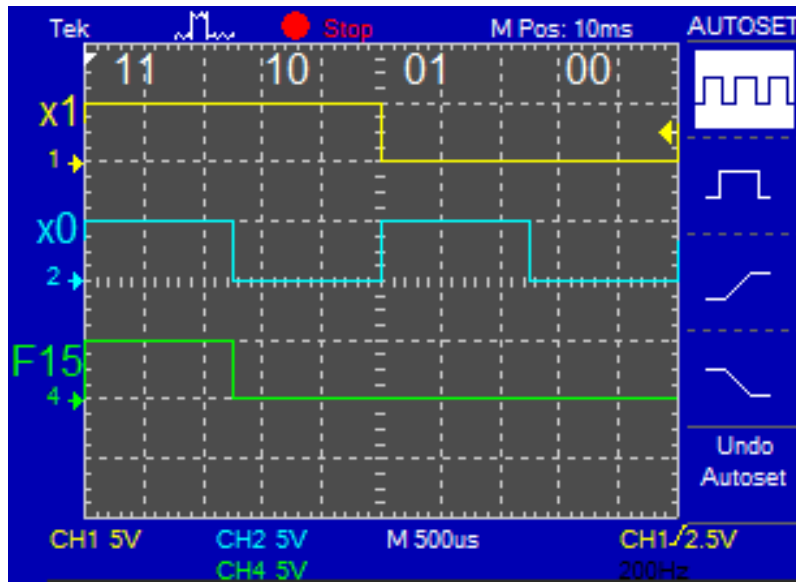


b)

Figura 5.14: Función F_7 .

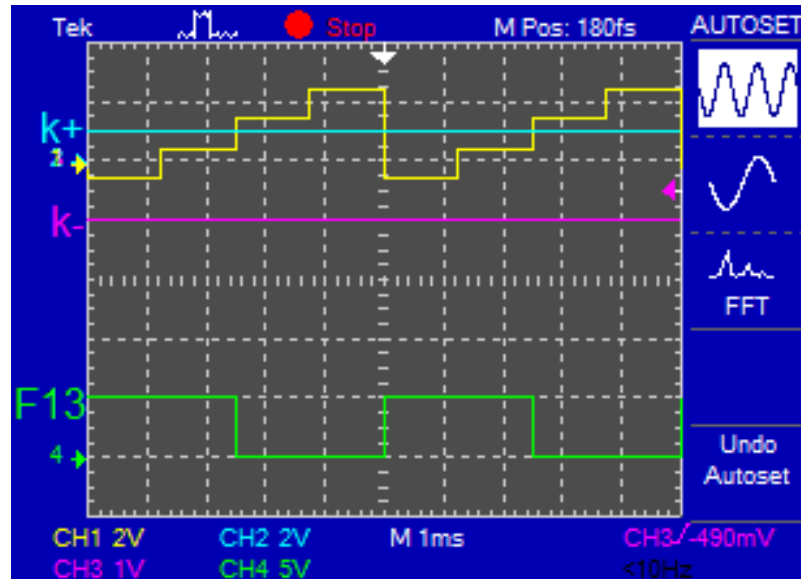


a)

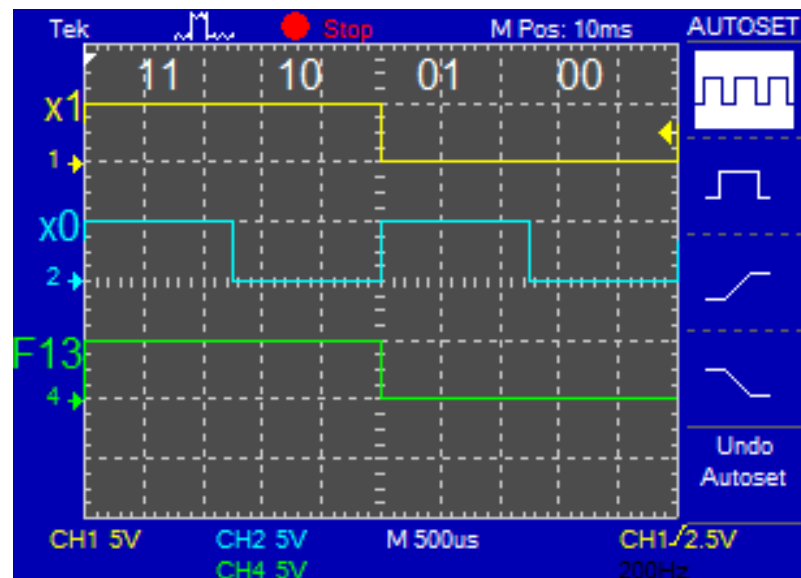


b)

Figura 5.15: Función F_8 .

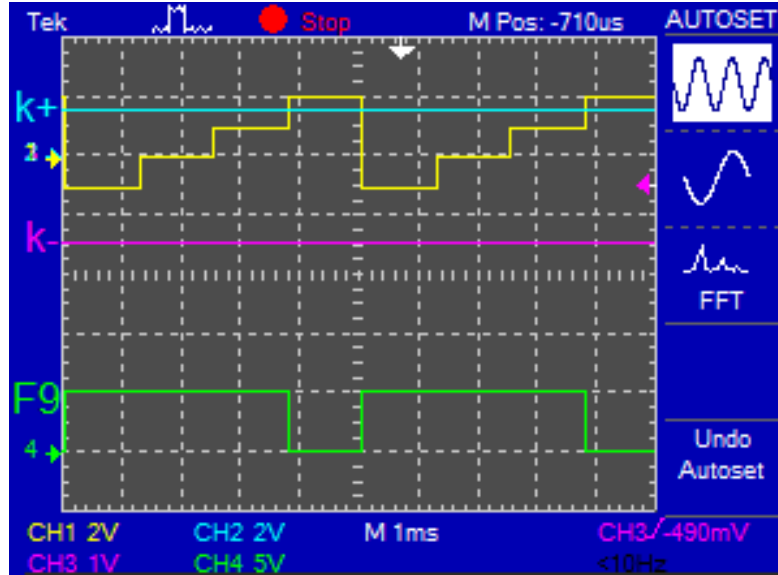


a)

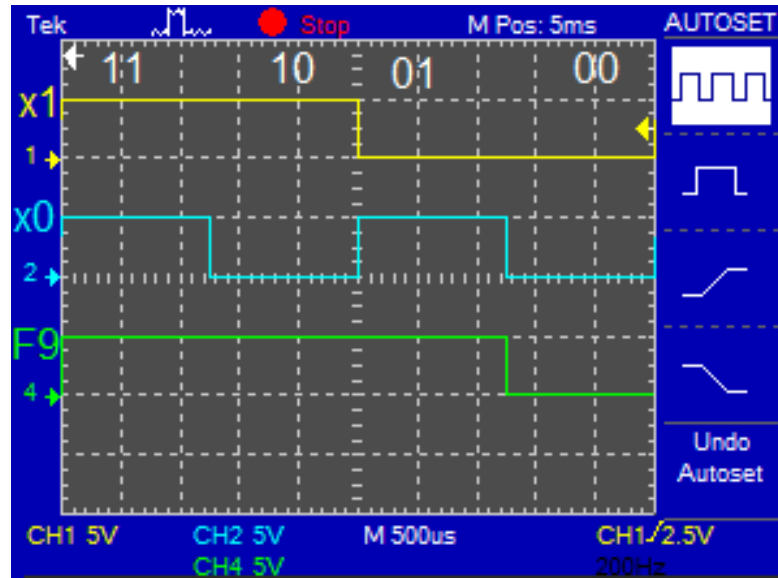


b)

Figura 5.16: Función F_{12} .

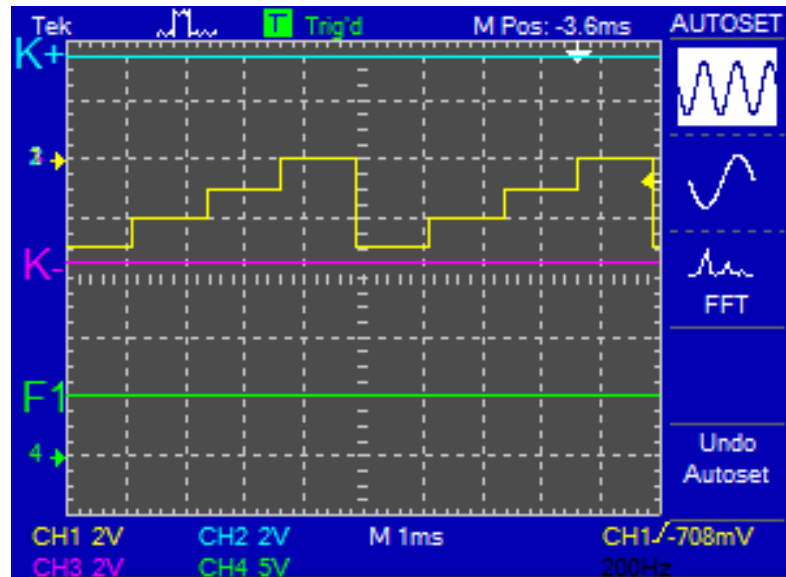


a)

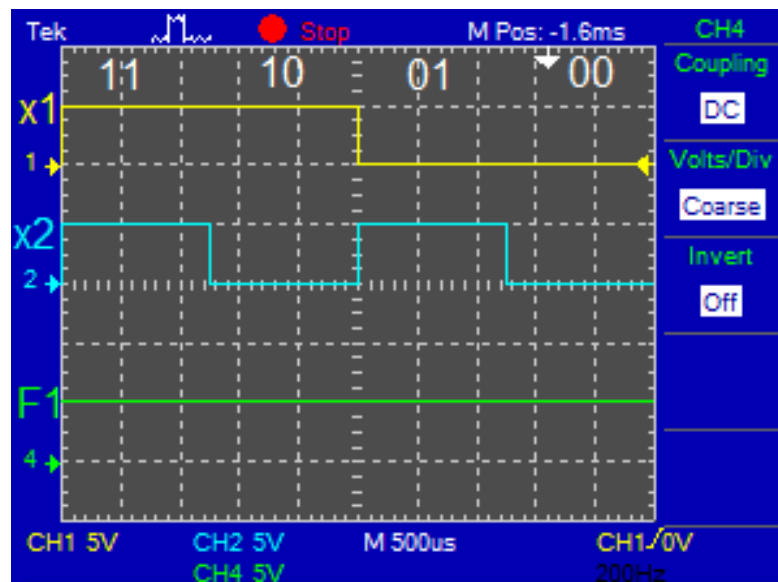


b)

Figura 5.17: Función F_{14} .



a)



b)

Figura 5.18: Función F_{15} .

En la figura 5.19 se presenta la respuesta lineal a trozos que se obtiene del circuito de la figura 5.6, en colores amarillo y azul se pueden apreciar las señales de entrada x_1 y x_0 respectivamente, de color morado se representan las cuatro permutaciones que estas generan (00, 01, 10, 11).

De igual manera que con los resultados anteriores, estos resultados se interpretan mediante dos imágenes, las imágenes correspondientes muestran en color azul la cota superior del umbral de operación $k+$, en color morado la cota inferior del umbral de operación $k-$, de color amarillo las permutaciones de la celda lógica F y en color verde la salida Y del sistema, mientras que en $b)$ se representan los estados de entrada x_1 y x_0 en colores amarillo y azul respectivamente, y en la parte inferior la salida Y del sistema.

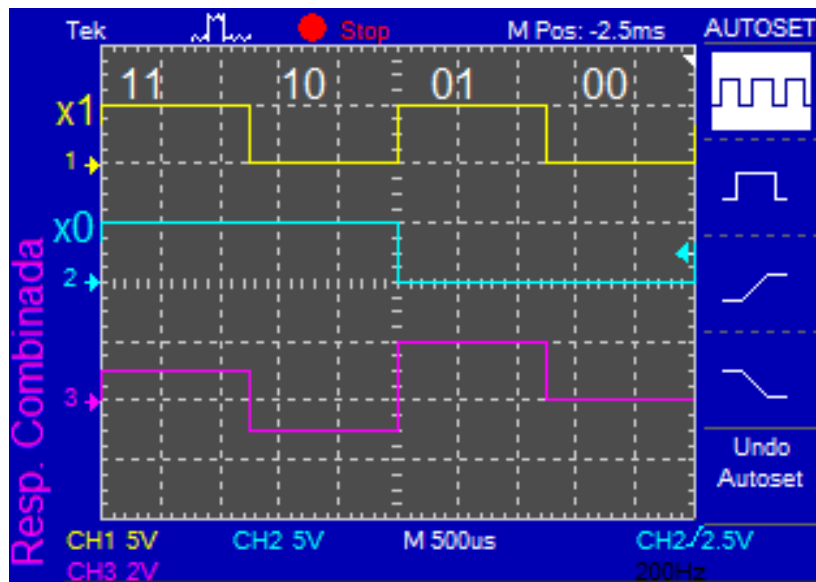
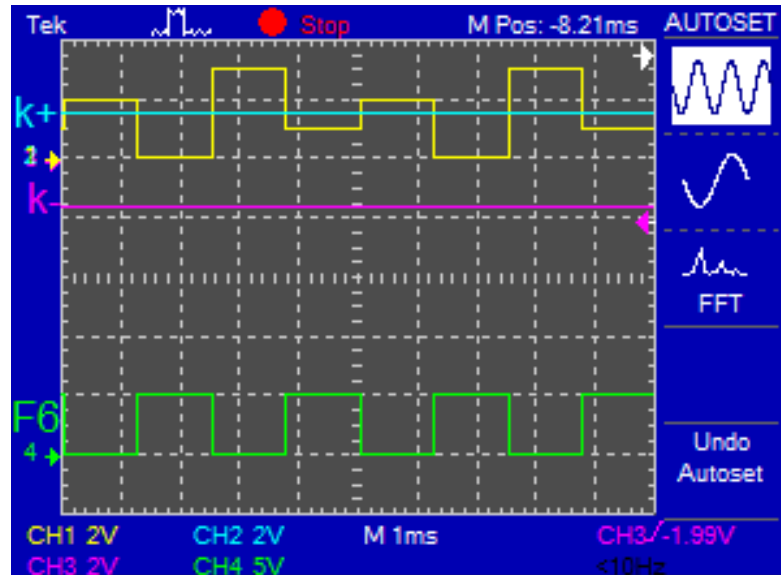
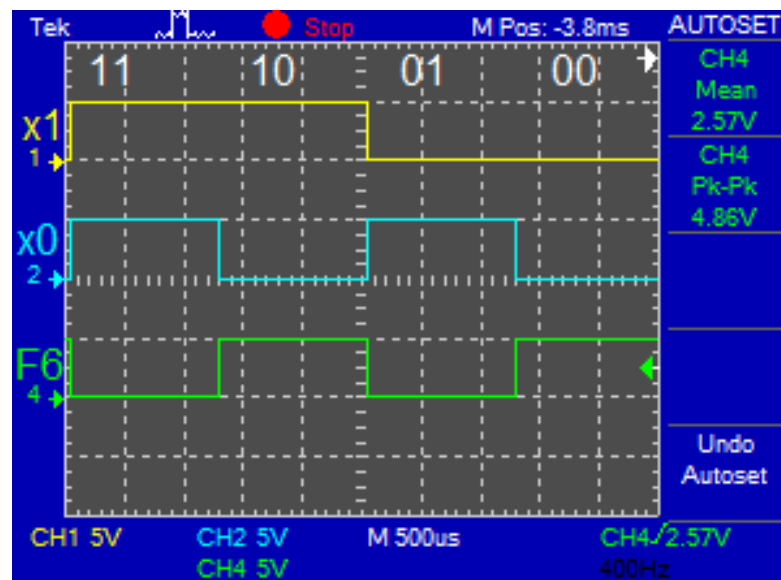


Figura 5.19: Respuesta lineal a trozos.

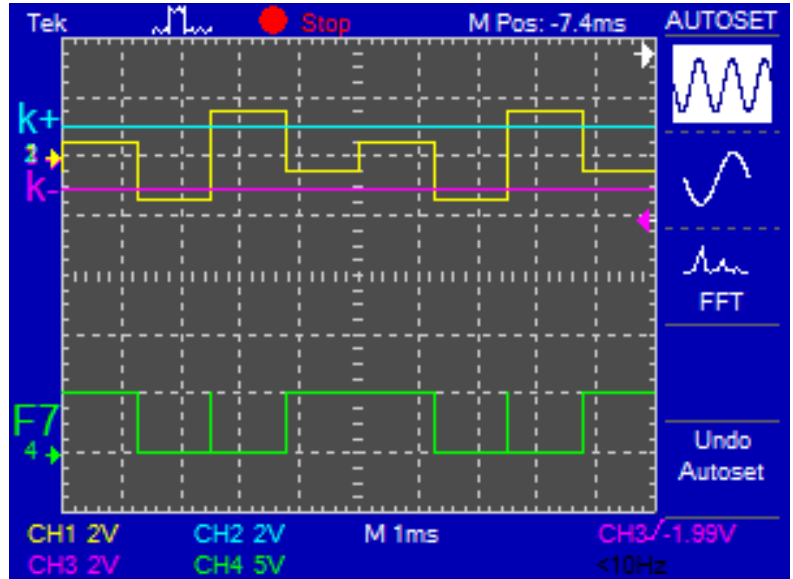


a)

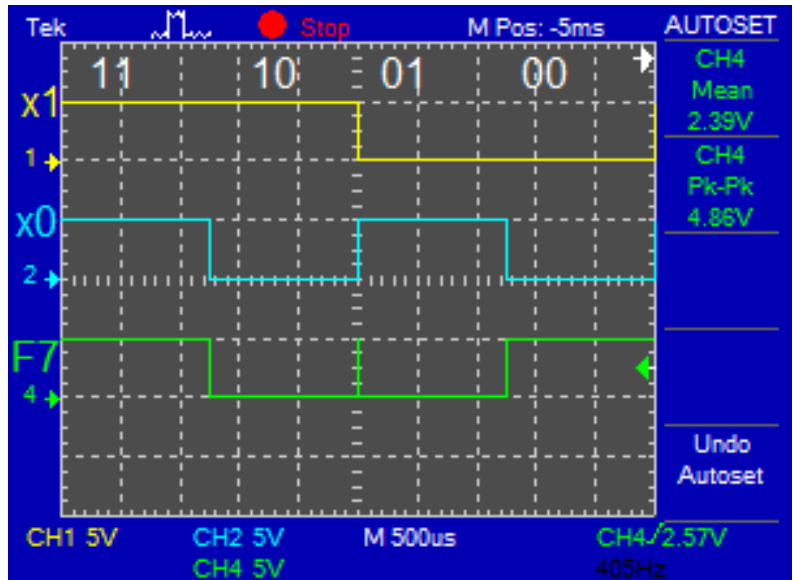


b)

Figura 5.20: Función F_5 .

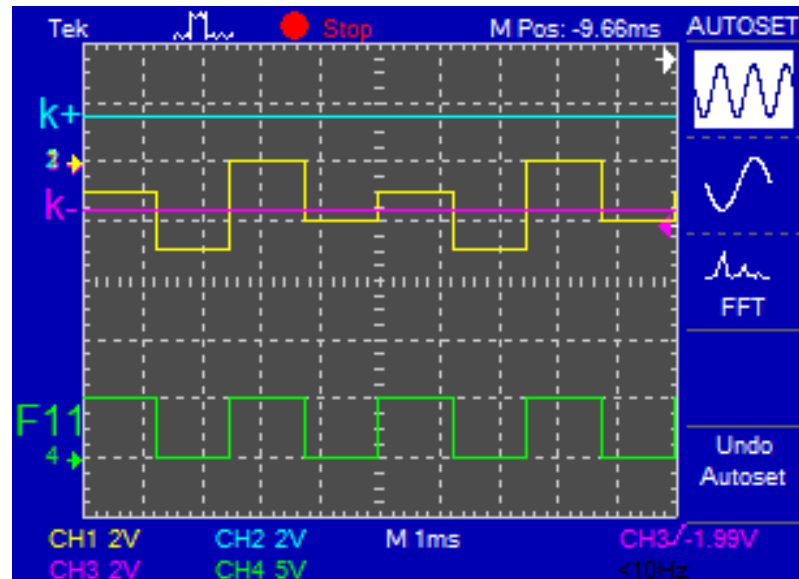


a)

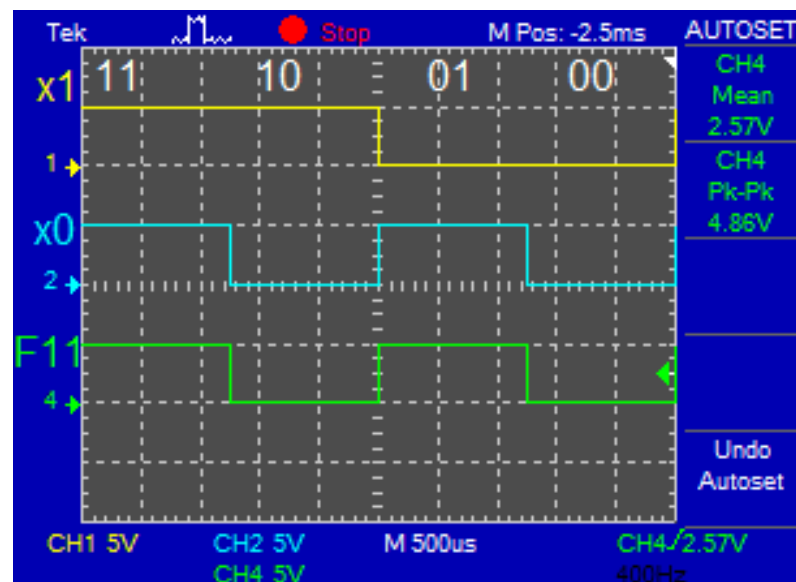


b)

Figura 5.21: Función F_9 .

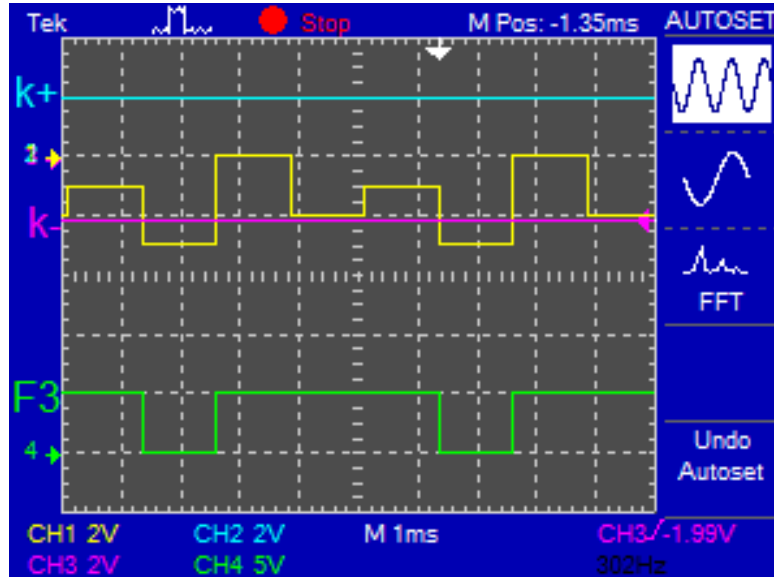


a)

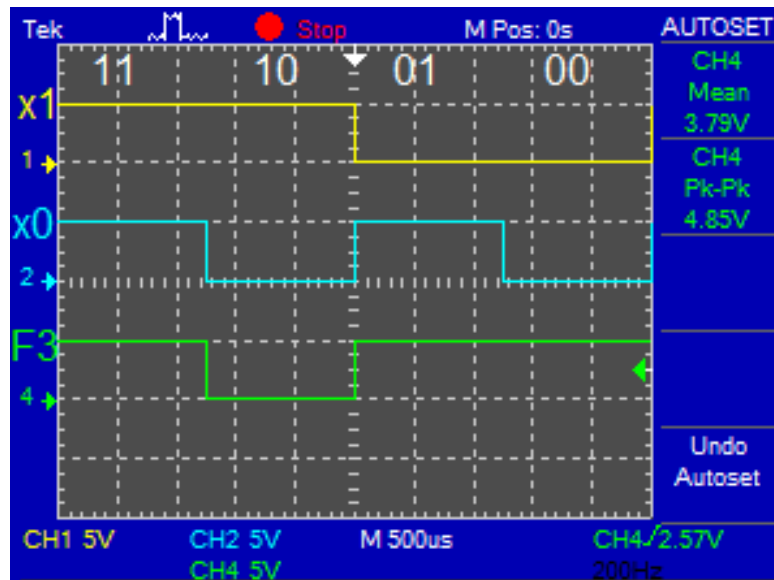


b)

Figura 5.22: Función F_{10} .

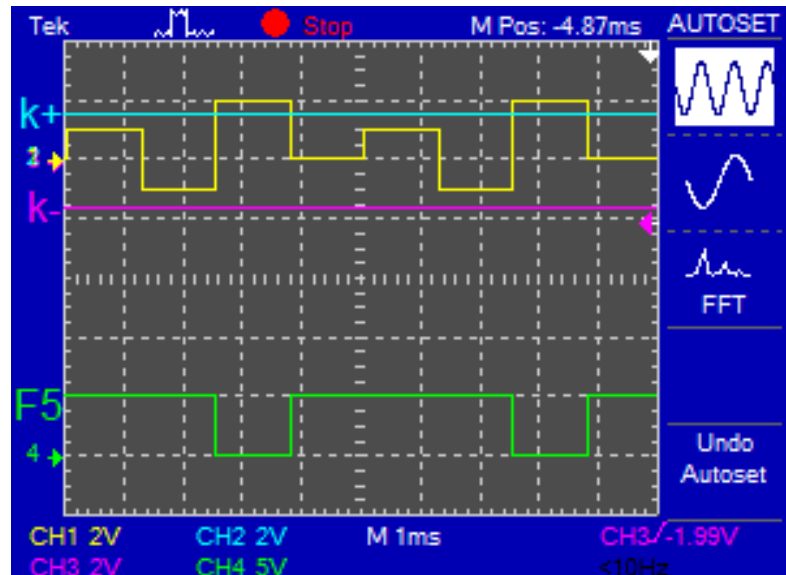


a)

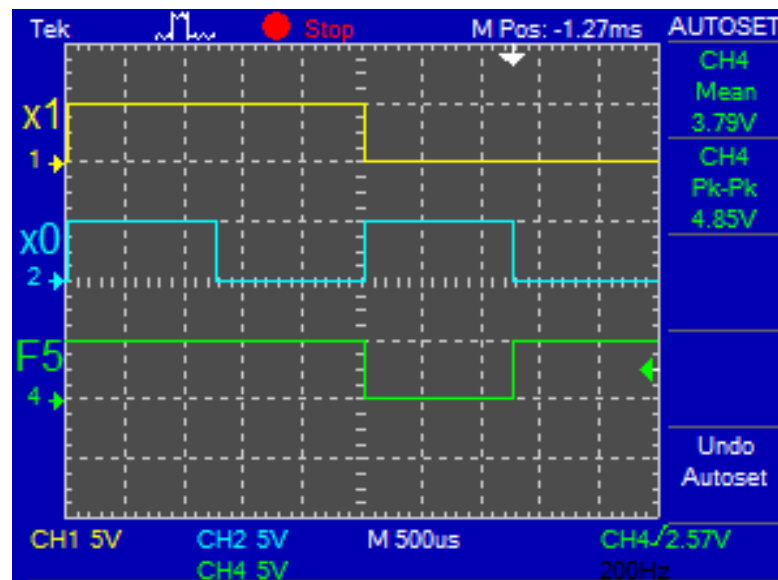


b)

Figura 5.23: Función F_{11} .



a)



b)

Figura 5.24: Función F_{13} .

5.2 Circuitos combinacionales

Un circuito combinacional se compone de compuertas lógicas cuyas salidas dependen solamente de la combinación que este presente en las entradas. Estos circuitos realizan operaciones que se pueden especificar lógicamente con un conjunto de funciones booleanas [29].

Tomando esta filosofía de diseño, se propusieron los siguientes circuitos combinacionales utilizando la misma metodología que con circuitos lógicos reconfigurables.

5.2.1 Medio sumador - Medio Restador

Actualmente las computadoras digitales y diversos circuitos lógicos realizan operaciones aritméticas, siendo la más común la suma de dos dígitos binarios. En la tabla 5.3 se presentan las permutaciones que conlleva esta operación.

Las primeras tres permutaciones corresponden a una suma de un dígito, pero para el caso en que ambos bits son iguales a 1, la suma binaria requiere un bit más para representar el resultado. El bit más significativo de este resultado es el bit de acarreo. El circuito combinacional que efectúa la operación de suma en dos bits se conoce como *medio sumador*, el circuito que permite contemplar la suma de tres bits (dos bits significativos y un acarreo previo) se denomina *sumador completo*. Cabe mencionar que los nombres que reciben estos circuitos vienen del hecho de que mediante el uso de dos circuitos *medio sumador* se puede formar un *sumador completo*.

Entradas		M. Sumador	
x_1	x_0	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Tabla 5.3: Tabla de verdad para un medio sumador.

En la figura 5.25 se presenta el diagrama de un circuito medio sumador usando compuertas lógicas. Como se puede observar se requieren dos compuertas lógicas, una compuerta AND que realice la función del acarreo (C) y una compuerta XOR que realice la suma (S) de dos bits.

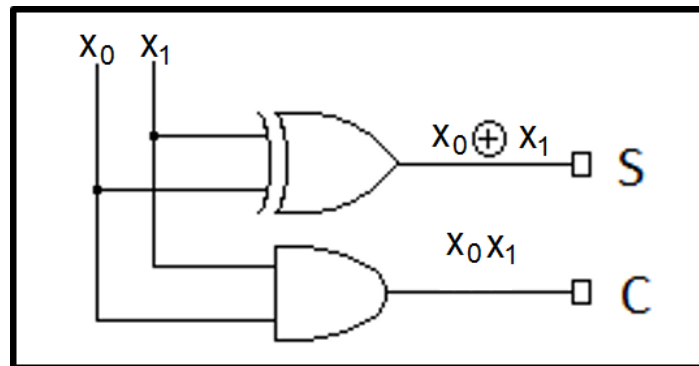


Figura 5.25: Diagrama de un circuito combinacional medio sumador .

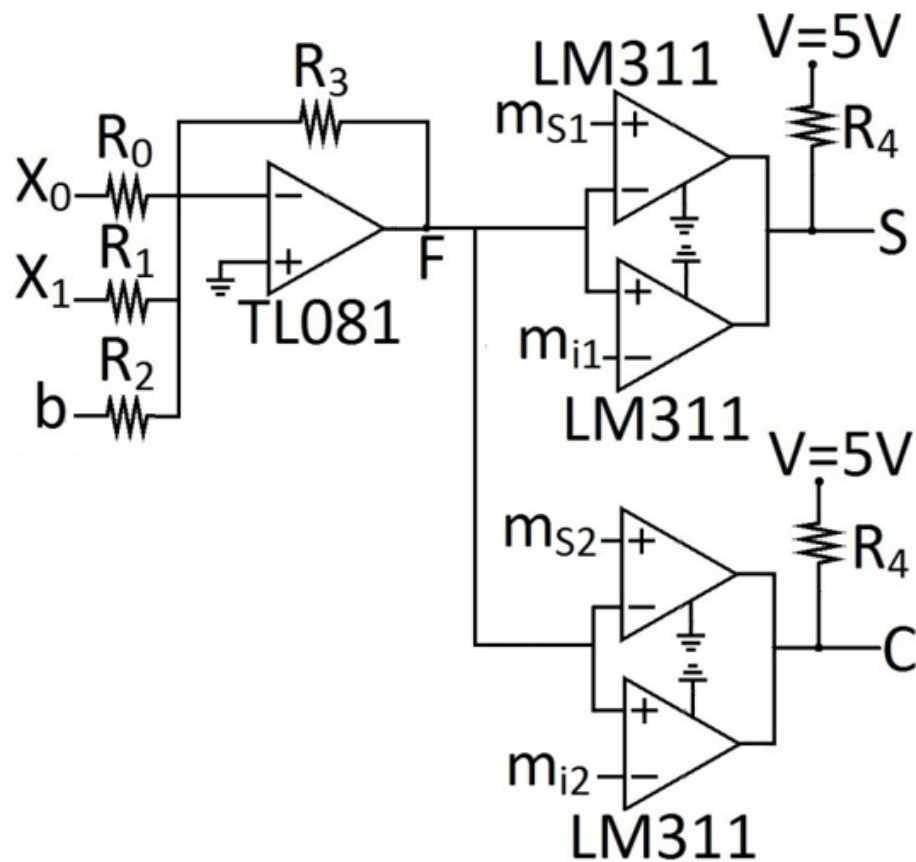


Figura 5.26: Diagrama de un circuito combinacional medio sumador usando un esquema basado en estructuras reconfigurables.

La propuesta que se realizó en este trabajo consiste en tomar el circuito que se presenta en la figura 5.26, al requerir dos salidas en el sistema, se propuso utilizar el mismo esquema del circuito sumador inversor de la figura 5.1 pero ahora combinado con dos circuitos ventana como los de la figura 5.4.

Mediante un análisis usando las ecuaciones de Kichhoff en el diagrama presentado en la figura 5.26, se determina que el voltaje en el nodo F esta dado por la siguiente expresión:

$$F_{Sum}(X) = -\frac{R_3}{R_0}x_0 - \frac{R_3}{R_1}x_1 - \frac{R_3}{R_2}b \quad (5.4)$$

Donde las ganancias en las entradas son $a_0 = -\frac{R_3}{R_0}$; $a_1 = -\frac{R_3}{R_1}$ y $b = -\frac{R_3}{R_2}$.

En este circuito el objetivo consiste en que un solo conjunto de parámetros permitan realizar ambas funciones, los parámetros que se propusieron son $a_0 = -1$, $a_1 = -1$, $b = 2.5$, $m_{S1} = 2$, $m_{i1} = 1$, $m_{S2} = 1$ y $m_{i2} = -1$. De esta manera los valores que toman las permutaciones son los que se muestran en la tabla 5.4

Entradas				Salidas	
x_1	x_0	$F(x)$	Voltaje en nodo F	C	S
0	0	b	2.5V	0	0
0	1	$a_0 + b$	1.5V	0	1
1	0	$a_1 + b$	1.5V	0	1
1	1	$a_0 + a_1 + b$	0.5V	1	0

Tabla 5.4: Tabla para un medio sumador.

Utilizando este mismo circuito es posible reconfigurar el sistema para obtener un medio restador. La forma en que funciona una resta binaria es semejante a la suma, se tienen cuatro permutaciones en donde el bit de acarreo indica cuando se produce un cambio de signo, de esta manera un valor 1 en el bit de acarreo (C) significa un número negativo, mientras que 0 se refiere a un valor positivo.

En la tabla 5.5 se muestra la tabla de verdad para un medio restador de dos bits de entrada.

Entradas		M. Restador	
x_1	x_0	C	R
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Tabla 5.5: Tabla de verdad para un medio restador.

En la figura 5.27 se muestra el diagrama para un circuito medio restador usando compuertas lógicas. Este circuito, al igual que el caso del sumador esta conformado por una compuerta AND para la función que representa el acarreo (C) y una compuerta XOR para la resta (R), la diferencia con el circuito del sumador radica en el inversor que lleva una de las entradas de la compuerta AND.

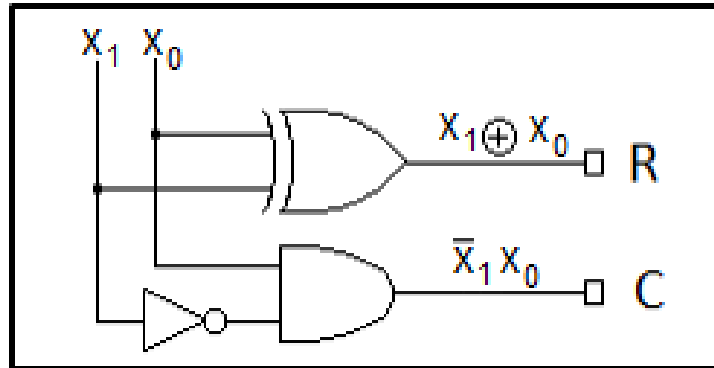


Figura 5.27: Diagrama de un circuito combinacional medio restador.

Para su realización los parámetros que se propusieron son $a_0 = -2$, $a_1 = -1$, $b = 3.5$, $m_{S1} = 3$, $m_{i1} = 1$, $m_{S2} = 3$ y $m_{i2} = 2$. Siendo así, que los valores que ahora tomarán las permutaciones son los que se muestran en la tabla 5.6.

Entradas				Salidas	
x_1	x_0	$F(x)$	Voltaje en nodo F	C	S
0	0	b	3.5V	0	0
0	1	$a_0 + b$	2.5V	1	1
1	0	$a_1 + b$	1.5V	0	1
1	1	$a_0 + a_1 + b$	0.5V	0	0

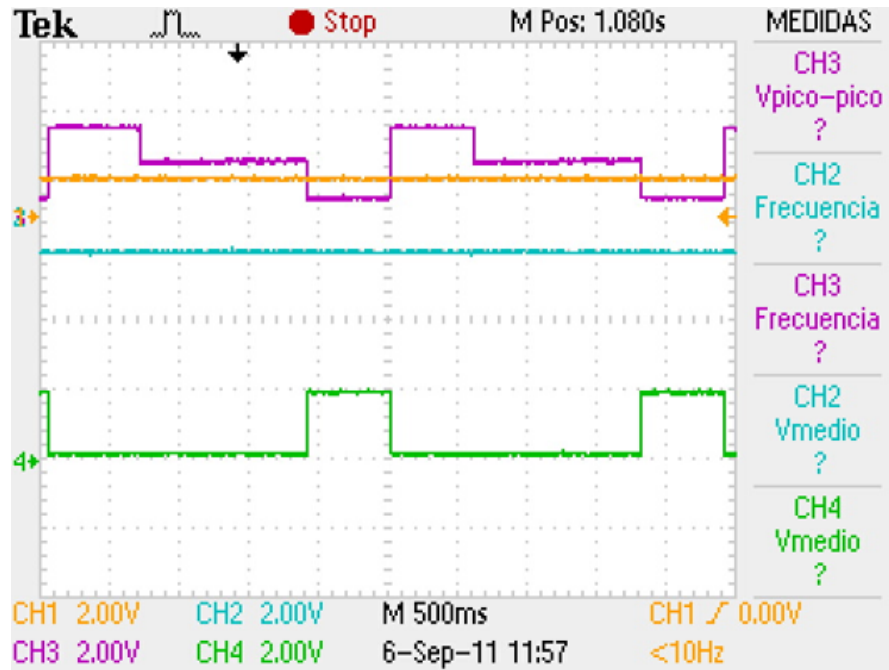
Tabla 5.6: Tabla para un medio restador.

Los valores en las resistencias y en los voltajes empleados, para que este circuito se comporte tanto como un circuito medio sumador y como un circuito medio restador, se presentan en la tabla 5.7.

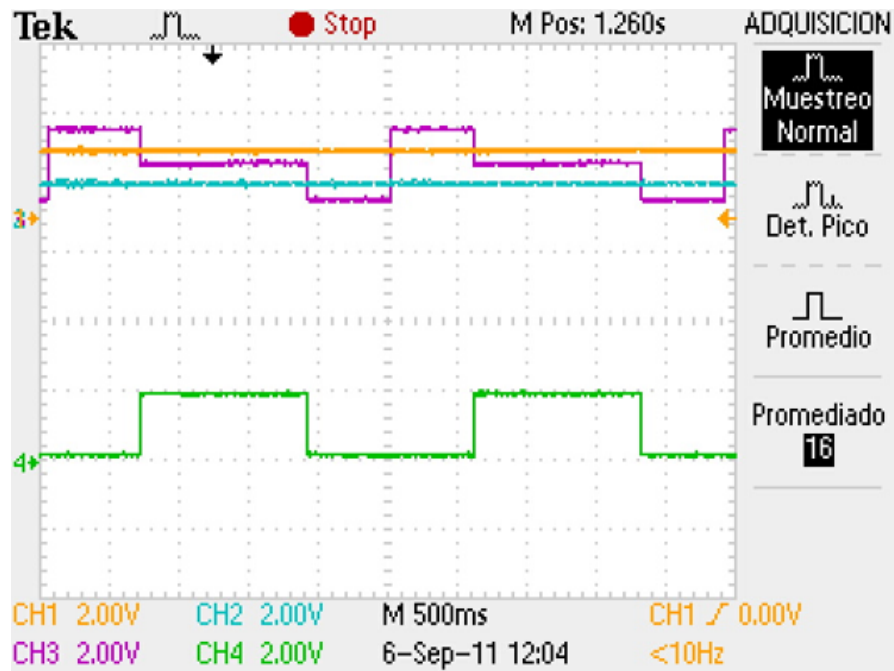
En la figura 5.28 se muestran los resultados de este circuito en modo de medio sumador, mientras que en 5.29 en modo de medio restador, en ambas imágenes, tanto a) como b), en color morado se muestran los niveles de voltaje para las permutaciones F , en color amarillo el límite superior del umbral de operación (m_s), en color azul el límite inferior del umbral de operación (m_i) y en color verde la salida del sistema, mostrando en: a) una función AND (con una entrada negada para el caso del *medio restador*) y en b) una función XOR.

	<i>SumadorMedio</i>	<i>RestadorMedio</i>
R_0	23k Ω	11.5k Ω
R_1	23k Ω	23k Ω
R_2	9k Ω	6.5k Ω
R_3	10k Ω	10k Ω
R_4	500 Ω	500 Ω
m_{s1}	2V	3V
m_{i1}	1V	1V
m_{s2}	1V	3V
m_{i2}	-1V	2V
	$b_{Sum}=-2.5V$	$b_{Res}-3.5V$

Tabla 5.7: Valores de resistencias y voltajes en circuito medio sumador reconfigurable.

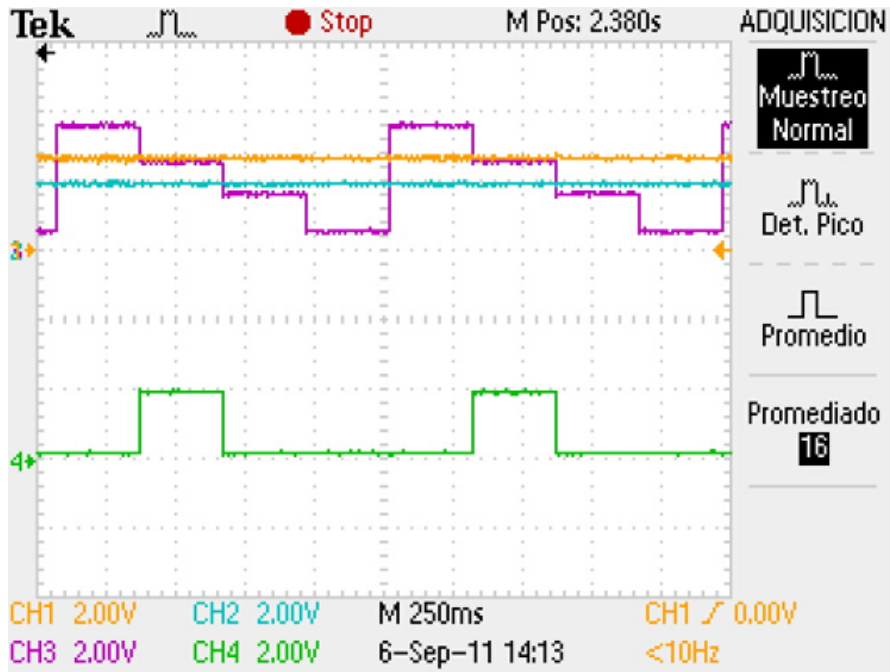


a)

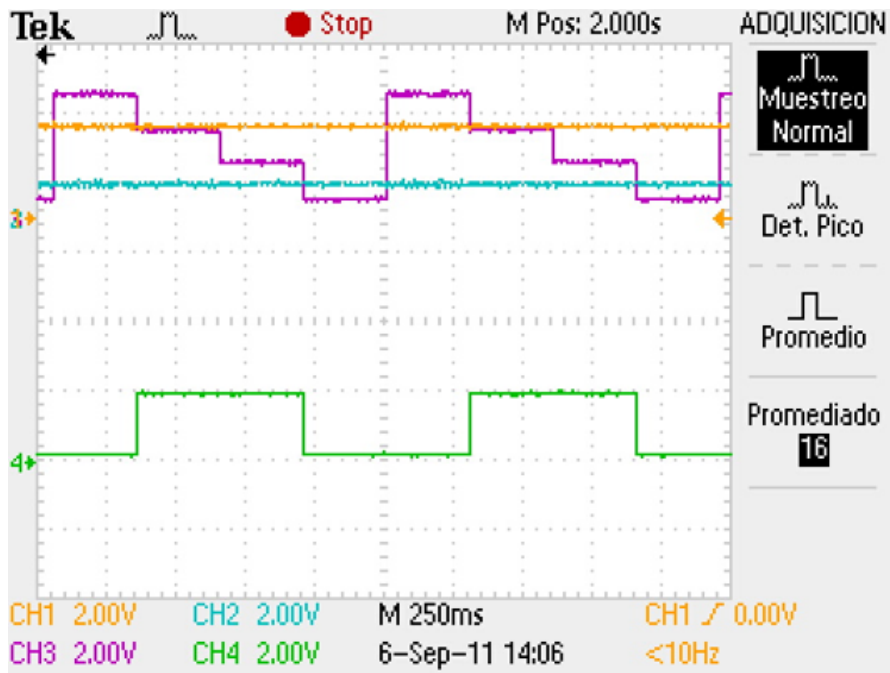


b)

Figura 5.28: Resultados de circuito reconfigurable en modo *medio sumador*.



a)



b)

Figura 5.29: Resultados de circuito reconfigurable en modo *medio restador*.

5.2.2 Decodificador

Cuando se quiere representar una variable discreta usando sistemas digitales se utiliza el código binario. Un sistema de código binario de n bits es capaz de representar hasta 2^n diferentes elementos. Un *decodificador* es un circuito combinacional que permite convertir información binaria de n líneas de entrada en información binaria de máximo 2^n líneas de salida. La tabla 5.8 corresponde a la tabla de verdad de un decodificador de 3 a 8, tres entradas y 8 salidas.

Entradas			Salidas							
x_2	x_1	x_0	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Tabla 5.8: Tabla de verdad para un *decodificador*.

En la figura 5.30 se presenta el diagrama para un decodificador de 3 a 8 usando compuertas lógicas, el cual esta conformado por tres compuertas NOT y ocho compuertas AND.

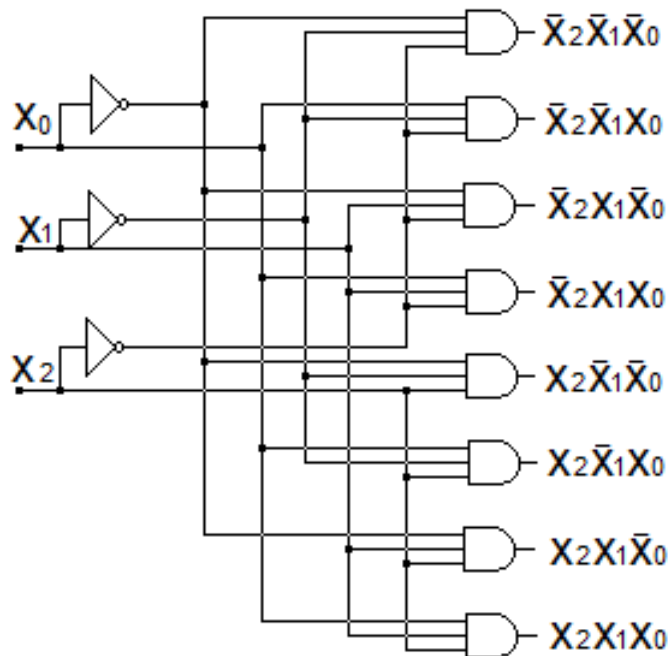


Figura 5.30: Diagrama de un circuito combinacional *decodificador*.

De manera semejante al caso anterior el sistema presenta más de una salida, siendo una solución utilizar más de un circuito ventana (figura 5.4) y un solo circuito sumador (figura 5.4). El circuito propuesto para este caso se presenta en la figura 5.31.

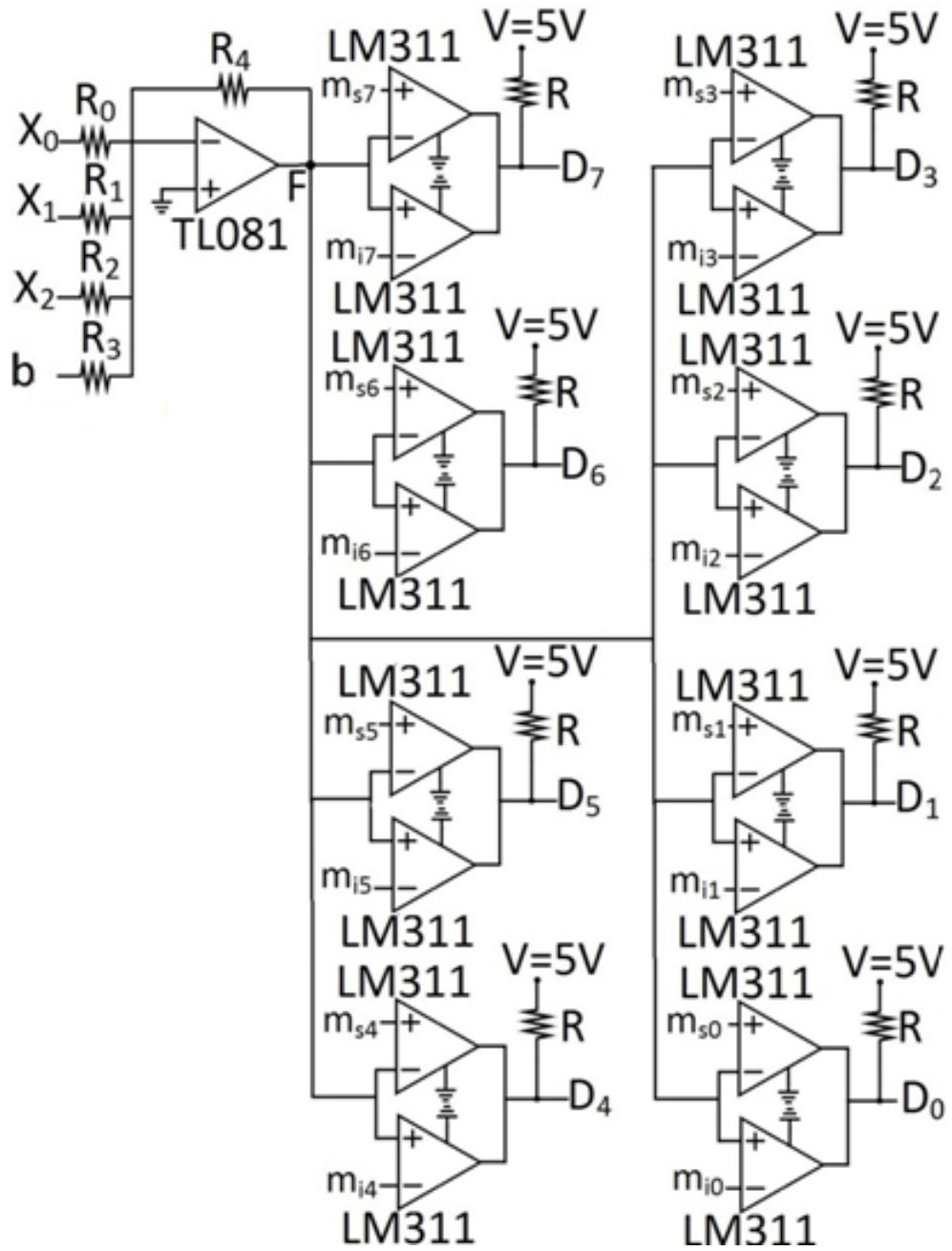


Figura 5.31: Diagrama de un circuito combinacional *decodificador* usando un esquema basado en estructuras reconfigurables.

Del diagrama presentado en la figura 5.31 se observa que el potencial en el nodo F semejante al caso del sumador medio, con ello:

$$F_{Res}(X) = -\frac{R_4}{R_0}x_0 - \frac{R_4}{R_1}x_1 - \frac{R_4}{R_2}x_2 - \frac{R_4}{R_3}b_{Dec} \quad (5.5)$$

Donde: $a_0 = -\frac{R_4}{R_0}$; $a_1 = -\frac{R_4}{R_1}$; $a_2 = -\frac{R_4}{R_2}$ y $b = \frac{R_4}{R_3}$

En la tabla 5.9 se presenta una relación de las permutaciones y los voltajes que se obtienen en el nodo F del circuito mostrado en la figura 5.31.

Entradas			Salidas									
x_2	x_1	x_0	$F(x)$	Voltaje en nodo F	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	b	7.5V	1	0	0	0	0	0	0	0
0	0	1	$a_0 + b$	6.5V	0	1	0	0	0	0	0	0
0	1	0	$a_1 + b$	5.5V	0	0	1	0	0	0	0	0
0	1	1	$a_0 + a_1 + b$	4.5V	0	0	0	1	0	0	0	0
1	0	0	$a_2 + b$	3.5V	0	0	0	0	1	0	0	0
1	0	1	$a_2 + a_0 + b$	2.5V	0	0	0	0	0	1	0	0
1	1	0	$a_2 + a_1 + b$	1.5V	0	0	0	0	0	0	1	0
1	1	1	$a_2 + a_1 + a_0 + b$	0.5V	0	0	0	0	0	0	0	1

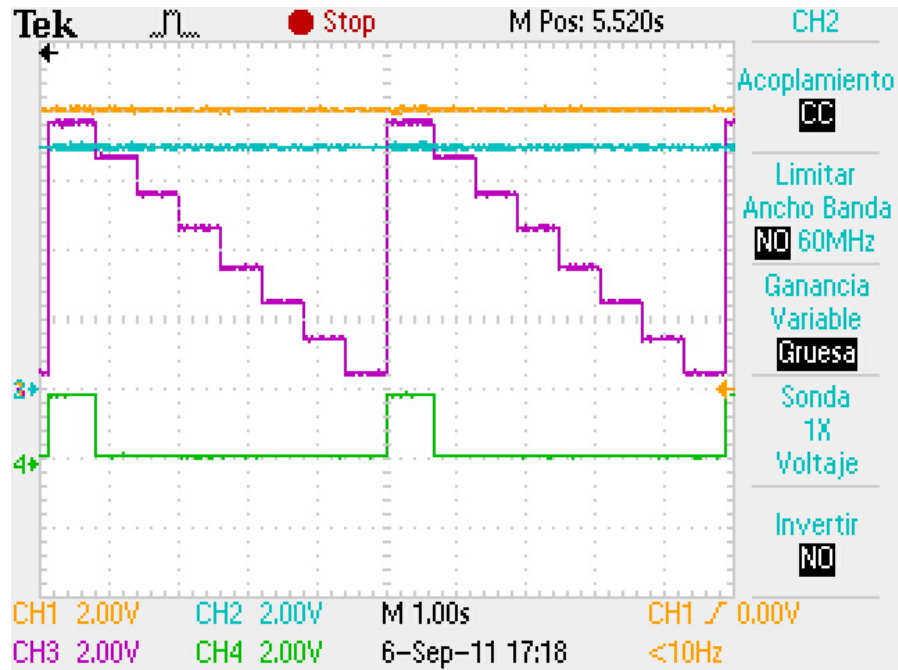
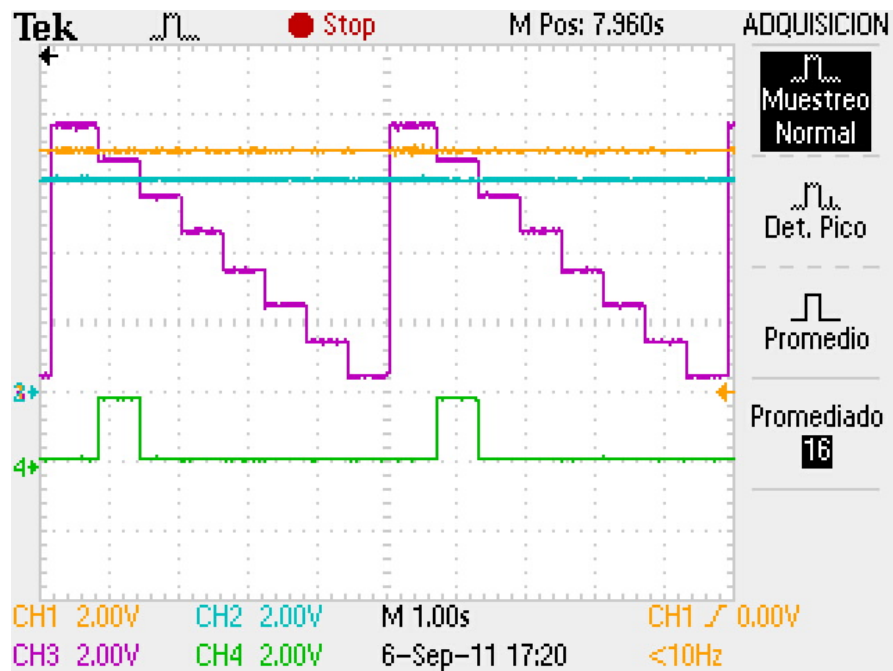
Tabla 5.9: Tabla de relación entre voltajes y permutaciones en un *decodificador*.

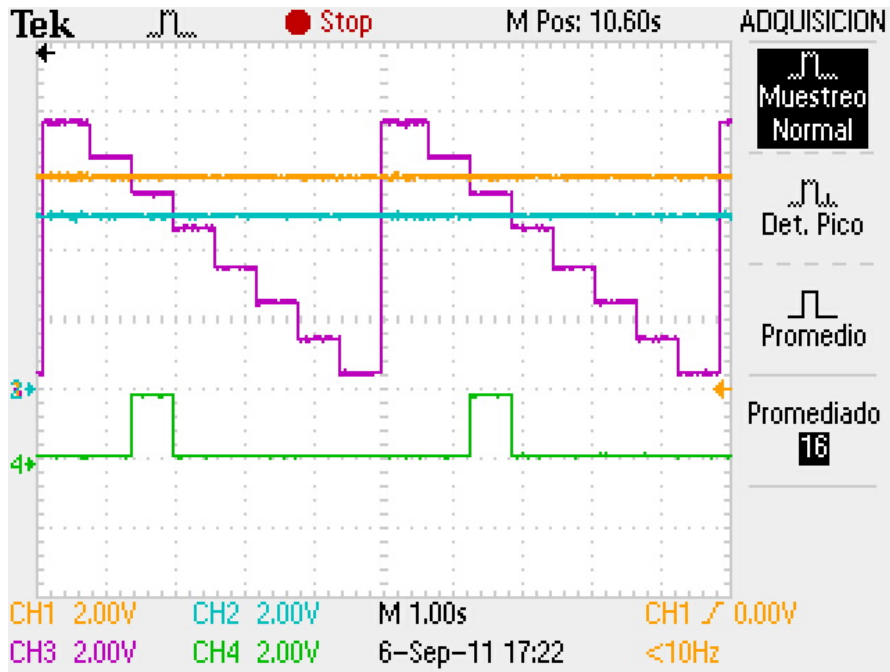
Las ganancias que se propusieron para las entradas de este sistema son $a_0 = -1$, $a_1 = -2$, $a_2 = -4$, $b = 7.5$; las cotas de los umbrales de operación m_{Sup} y m_{Inf} se definen para las 8 salidas de tal forma que $m_{s0} = 8$, $m_{i0} = m_{s1} = 7$, $m_{i1} = m_{s2} = 6$, $m_{i2} = m_{s3} = 5$, $m_{i3} = m_{s4} = 4$, $m_{i4} = m_{s5} = 3$, $m_{i5} = m_{s6} = 2$, $m_{i6} = m_{7s} = 1$, $m_{i7} = 0$.

R_0	5.7k Ω	m_{s7}	1V	m_{s3}	5V
R_1	11.5k Ω	m_{i7}	0V	m_{i3}	4V
R_2	23k Ω	m_{s6}	2V	m_{s2}	6V
R_3	3.06k Ω	m_{i6}	1V	m_{i2}	5V
R_4	10k Ω	m_{s5}	3V	m_{s1}	7V
R	500 Ω	m_{i5}	2V	m_{i1}	6V
b_{Dec}	-7.5V	m_{s4}	4V	m_{s0}	8V
-	-	m_{i4}	3V	m_{i0}	7V

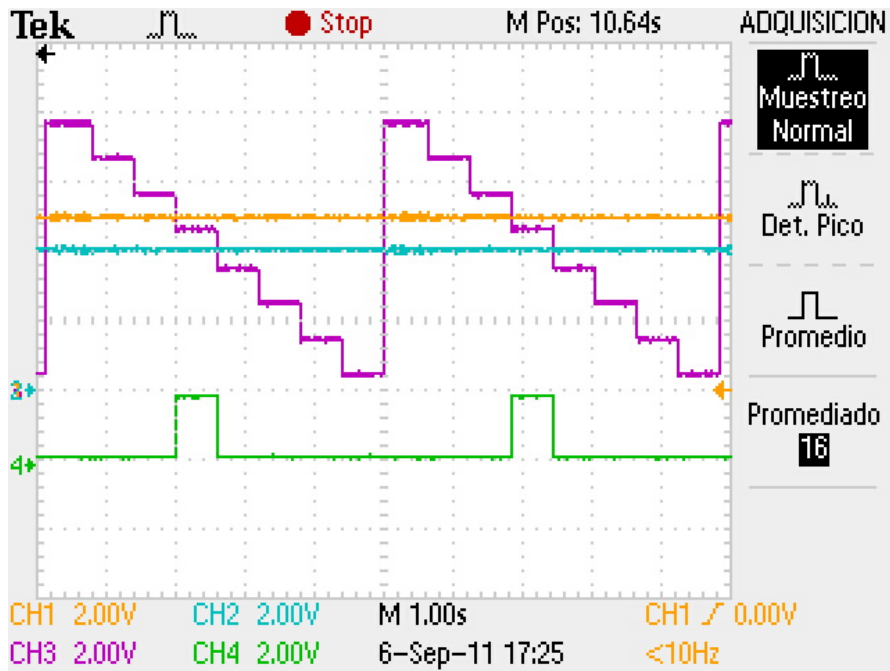
Tabla 5.10: Valores de resistencias y voltajes en un circuito *decodificador* reconfigurable.

En la tabla 5.10 se muestran los valores de las resistencias y voltajes empleados para obtener las ganancias que permiten que el circuito opere como un decodificador, consecuentemente en las figuras 5.32, 5.33, 5.34 y 5.35 se presentan los resultados para este circuito, cada figura corresponde a la relación entre las entradas y un específico valor de umbral de operación, en todas las figuras la señal en color morado representa los niveles de voltaje generados por las permutaciones de las entradas, en colores amarillo y azul los límites superior e inferior del umbral de operación respectivamente y en color verde la salida.

a) Función de salida D_0 de un decodificador reconfigurable.b) Función de salida D_1 de un decodificador reconfigurable.Figura 5.32: Resultados de circuito reconfigurable en modo *decodificador* (Salidas D_0 y D_1).

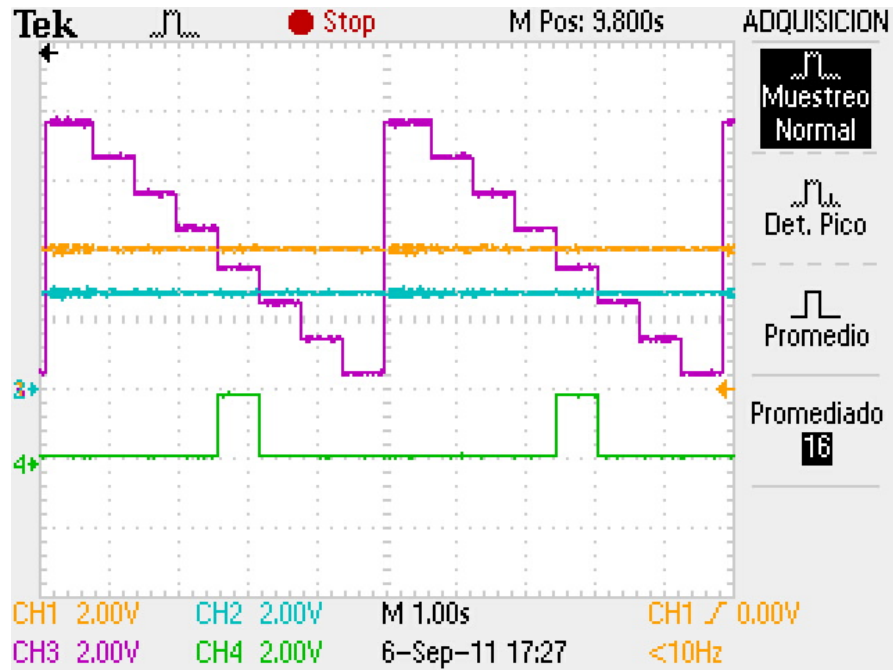
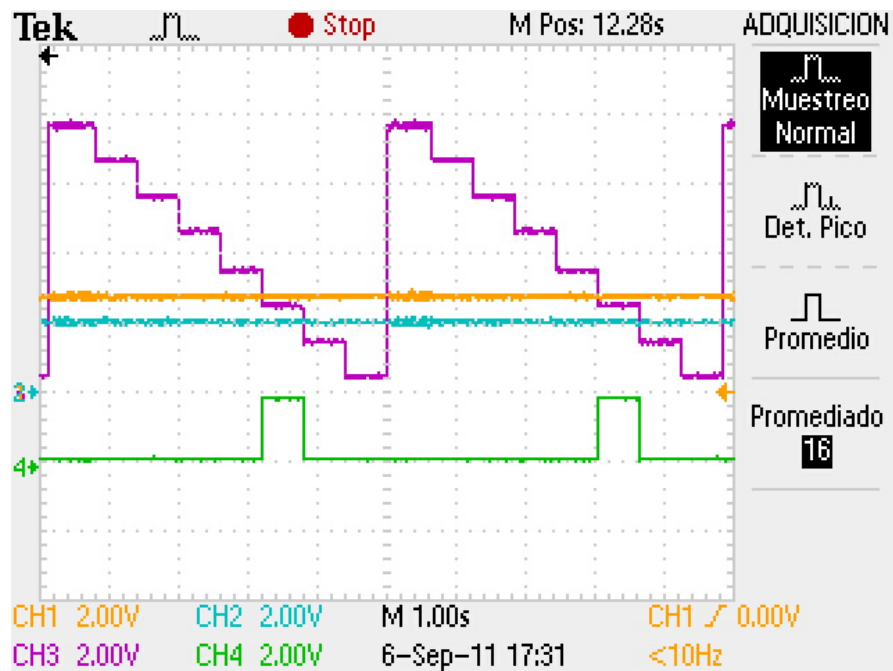


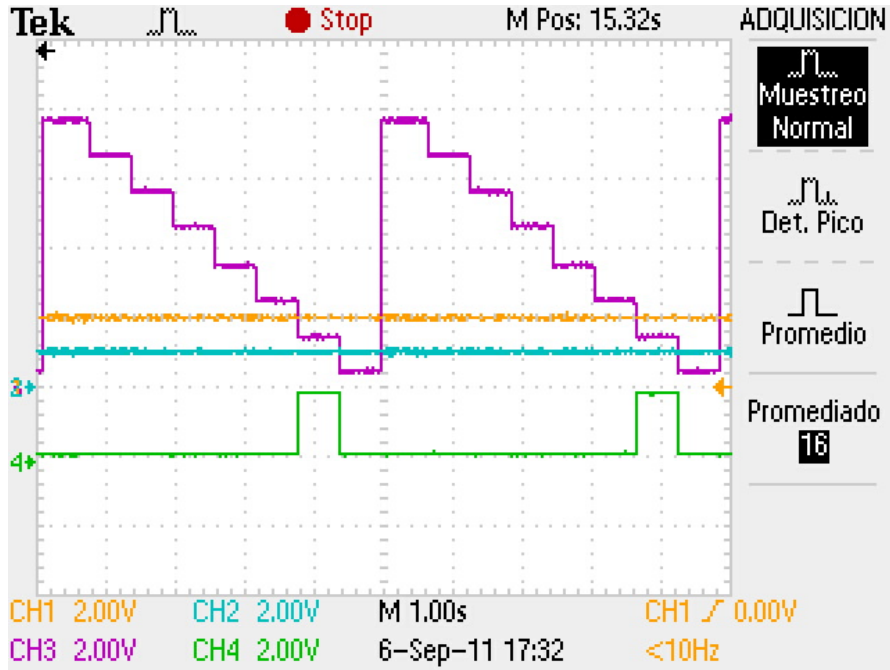
a) Función de salida D_2 de un decodificador reconfigurable.



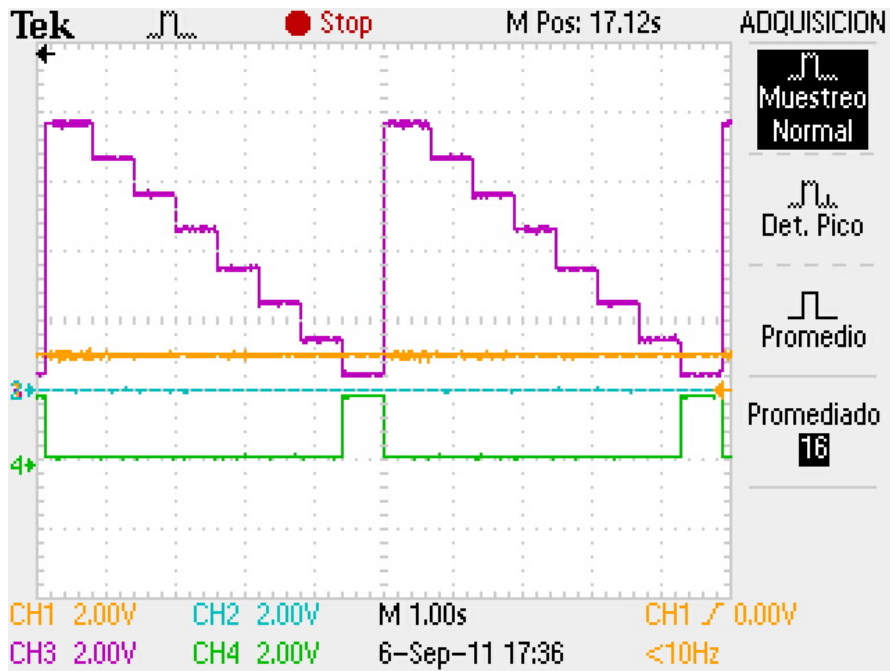
b) Función de salida D_3 de un decodificador reconfigurable.

Figura 5.33: Resultados de circuito reconfigurable en modo *decodificador* (Salidas D_2 y D_3).

a) Función de salida D_4 de un decodificador reconfigurable.b) Función de salida D_5 de un decodificador reconfigurable.Figura 5.34: Resultados de circuito reconfigurable en modo *decodificador* (Salidas D_4 y D_5).



a) Función de salida D_6 de un decodificador reconfigurable.



b) Función de salida D_7 de un decodificador reconfigurable.

Figura 5.35: Resultados de circuito reconfigurable en modo *decodificador* (Salidas D_6 y D_7).

5.2.3 Codificador

Un codificador es una función digital que produce la operación inversa de un decodificador. Un codificador tiene 2^n líneas de entrada y n líneas de salida. En este circuito, las salidas generan el código binario correspondiente al valor de la entrada. En la tabla 5.11 se presenta la tabla de verdad correspondiente para este circuito.

Entradas								Salidas		
x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	D_0	D_1	D_2
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Tabla 5.11: Tabla de verdad para un *decodificador*.

En la figura 5.36 aparece el diagrama combinacional para un codificador de 8 a 3, el cual esta conformado por tres compuertas lógicas OR.

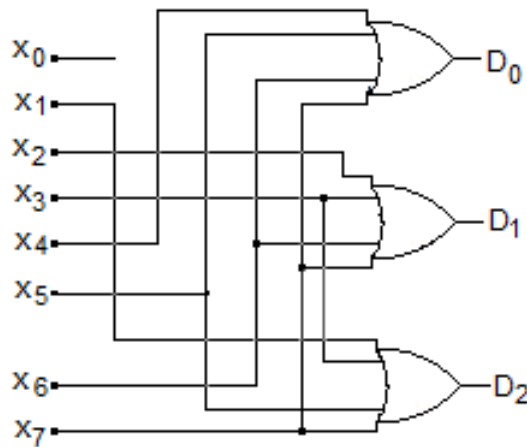


Figura 5.36: Diagrama de un circuito combinacional *codificador*.

La propuesta que se realizó para este sistema se muestra en la figura 5.37, al tratarse de tres compuertas lógicas que tienen entradas en común se optó por utilizar tres esquemas independientes como el mostrado en 5.1, en la figura 5.37 se indica cuales son las entradas para cada una de estas compuertas.

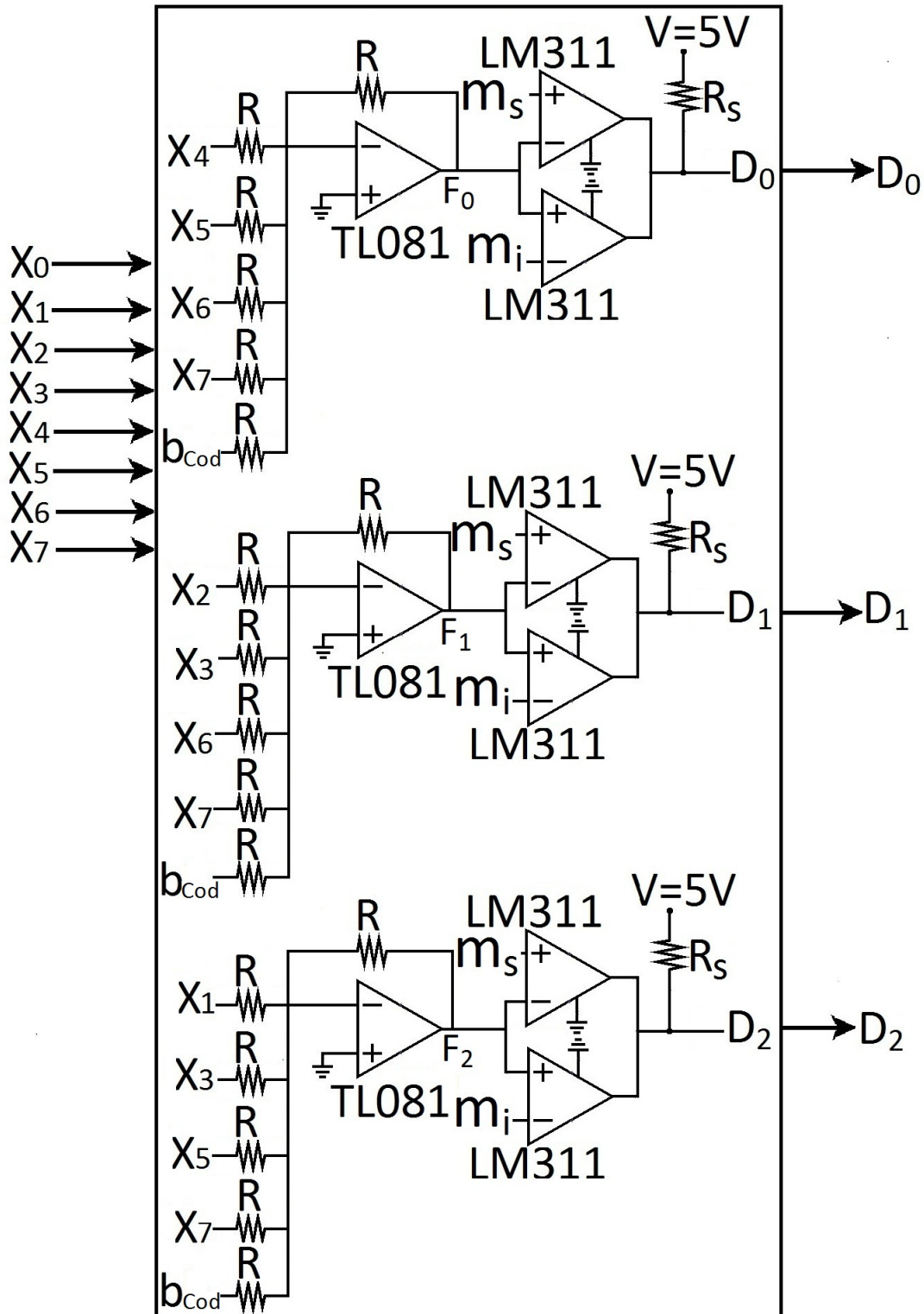


Figura 5.37: Diagrama de un circuito combinacional *codificador* usando un esquema basado en estructuras reconfigurables.

Para este circuito se contempló todas las entradas con la misma ganancia, es decir $a_i = -1$ (con $i = 0, 1, \dots, 7$), $b = 1.5$, $m_s = 1$ y $m_l = 0$, generando así la tabla de verdad mostrada en 5.11. Es necesario mencionar que en el diseño de circuitos que utilizan codificadores sólo se emplean las entradas permitidas de su tabla de verdad y no todo el conjunto de posibles permutaciones.

Del diagrama eléctrico de la figura 5.37 se observa que el potencial en los nodos F_0, F_1, F_2 está dado por una expresión que contempla cinco entradas, debido a que las resistencias de este circuito son de igual magnitud la razón entre ellas es unitaria por lo tanto:

$$F_0(X) = -x_4 - x_5 - x_6 - x_7 - b_{Cod} \quad (5.6)$$

$$F_1(X) = -x_2 - x_3 - x_6 - x_7 - b_{Cod} \quad (5.7)$$

$$F_2(X) = -x_1 - x_3 - x_5 - x_7 - b_{Cod} \quad (5.8)$$

En la tabla 5.12

Entradas								Salidas					
x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	F_0	F_1	F_2	D_0	D_1	D_2
1	0	0	0	0	0	0	0	b	b	b	0	0	0
0	1	0	0	0	0	0	0	b	b	b+1	0	0	1
0	0	1	0	0	0	0	0	b	b+1	b	0	1	0
0	0	0	1	0	0	0	0	b	b+1	b+1	0	1	1
0	0	0	0	1	0	0	0	b+1	b	b	1	0	0
0	0	0	0	0	1	0	0	b+1	b	b+1	1	0	1
0	0	0	0	0	0	1	0	b+1	b+1	b	1	1	0
0	0	0	0	0	0	0	1	b+1	b+1	b+1	1	1	1

Tabla 5.12: Tabla para un *codificador*.

Atendiendo a las ganancias anteriormente mencionadas, fue que se asignaron los valores de resistencias y voltajes de alimentación que se presentan en la tabla 5.13. Para poder evaluar este caso, se programó un microcontrolador con la siguiente secuencia hexadecimal: 01, 00, 02, 00, 04, 00, 08, 00, 10, 00, 20, 00, 40, 00, 80, 00, ya que son las únicas entradas permitidas, de esta manera a continuación se presentan las secuencias de entrada que habilitan cada una de las salidas.

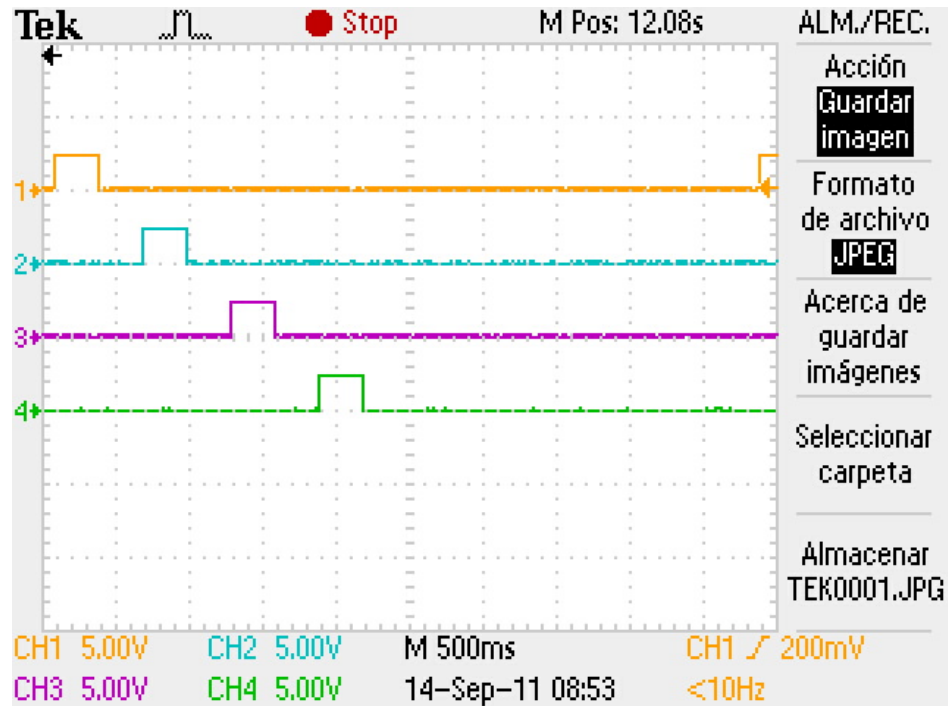
R	10k Ω
R_S	500 Ω
b_{Cod}	-1.8V
m_S	1V
m_l	0V

Tabla 5.13: Valores de resistencias y voltajes en un circuito *codificador* reconfigurable.

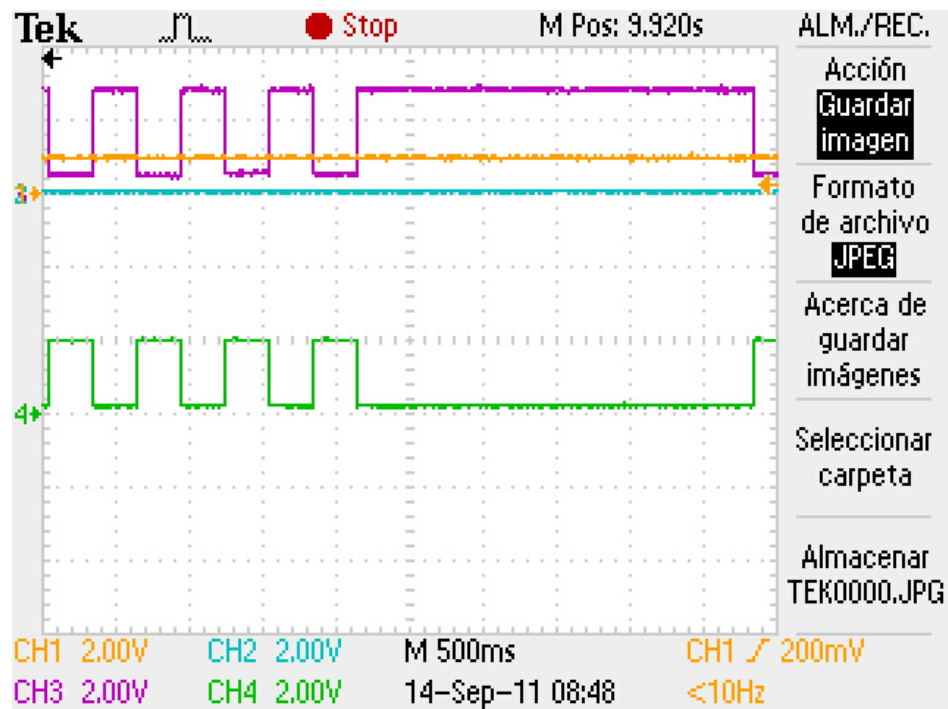
En la figura 5.38 se presenta el resultado del codificador, en a) se muestran las entradas de F_0 (x_4 - amarillo, x_5 - azul, x_6 - morado, x_7 - verde), en b) se muestra la salida D_0 , donde el canal uno (amarillo) corresponde a la cota superior, el canal dos (azul) a la cota inferior, el canal tres (morado) a la señal F_0 y el canal cuatro (verde) a la salida D_0 .

De manera analoga en la figura 5.39, en a) se muestran las señales de entrada (x_2 - amarillo, x_3 - azul, x_6 - morado, x_7 - verde) para la función D_1 y en b) el canal uno (amarillo) es la cota superior, el canal dos (azul) es la cota menor, el canal tres (morado) la señal F_1 y el canal cuatro (verde) es la salida D_1 .

Por otro lado, para la figura 5.40, en a) se muestran las señales de entrada (x_1 - amarillo, x_3 - azul, x_5 - morado, x_7 - verde) que activan F_2 , mientras que en b) el canal uno (amarillo) corresponde a la cota superior, el canal dos (azul) a la cota inferior, el canal tres (morado) a la señal F_2 y el canal cuatro (verde) a la salida D_2 .

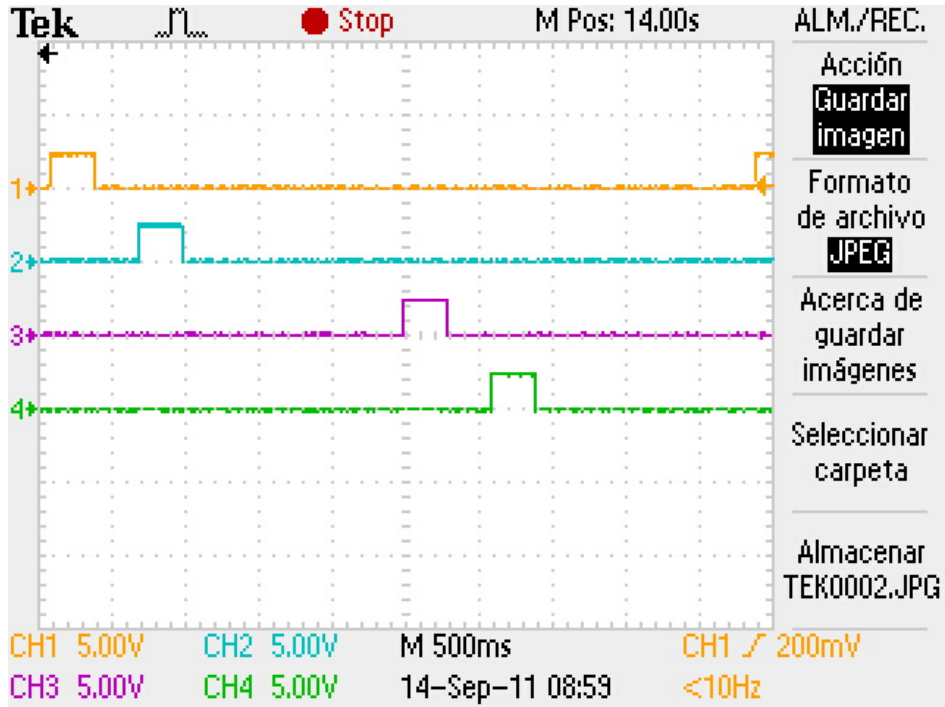


a) Entradas que habilitan la salida D_0 (x_4, x_5, x_6, x_7) en un codificador reconfigurable.

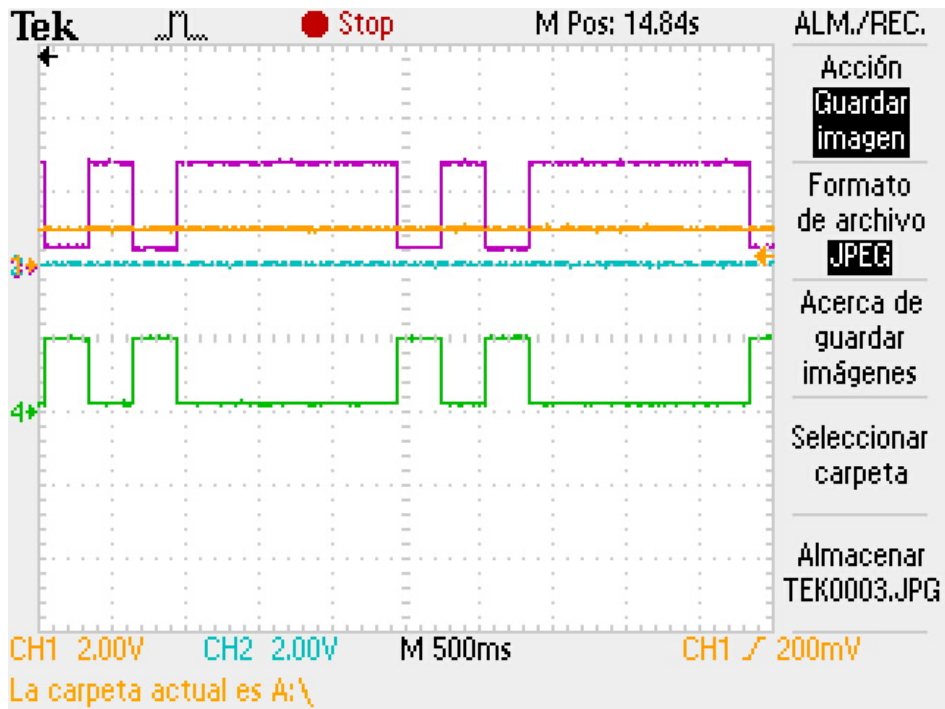


b) Salida D_0 en un codificador reconfigurable.

Figura 5.38: Resultado D_0 para circuito reconfigurable en modo *codificador*.

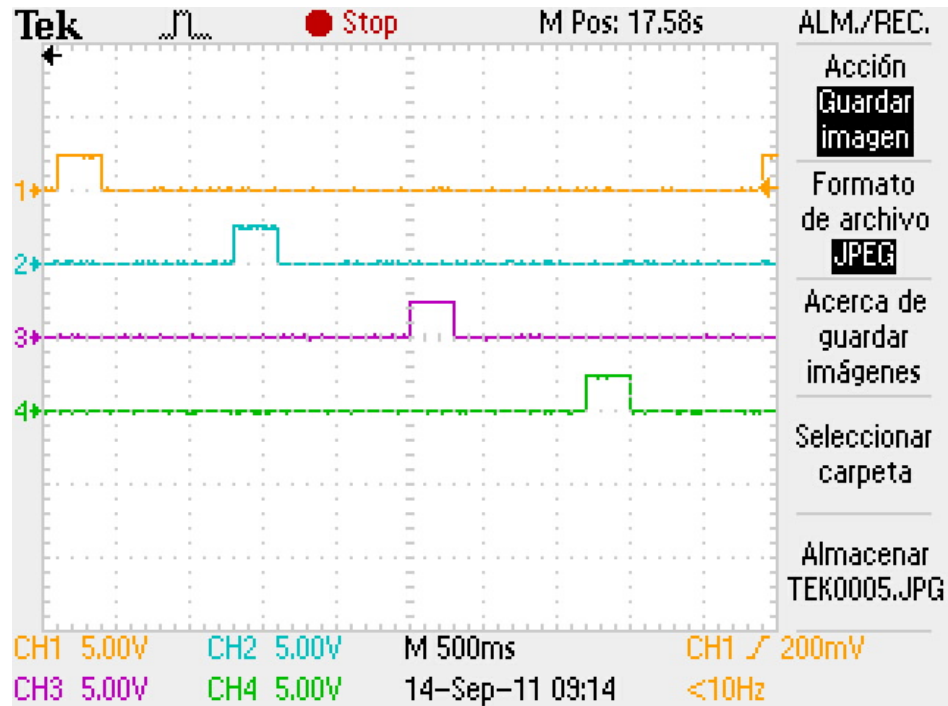


a) Entradas que habilitan la salida D_1 (x_2, x_2, x_6, x_7) en un codificador reconfigurable.

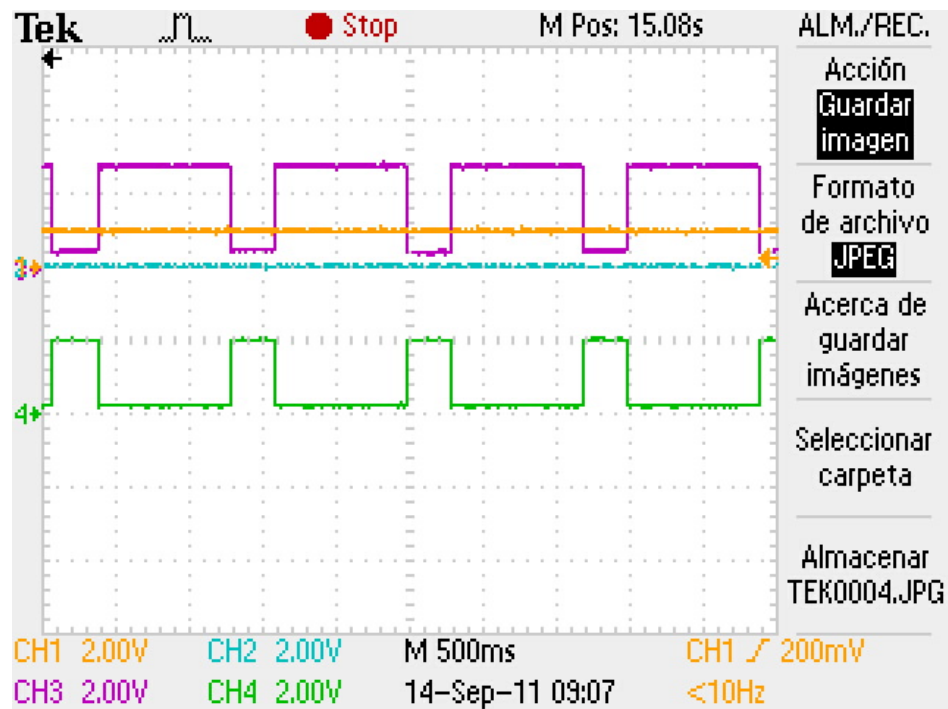


b) Salida D_1 en un codificador reconfigurable.

Figura 5.39: Resultado D_1 para circuito reconfigurable en modo *codificador*.



a) Entradas que habilitan la salida D_2 (x_1, x_3, x_5, x_7) en un codificador reconfigurable.



b) Salida D_2 en un codificador reconfigurable.

Figura 5.40: Resultado D_2 para circuito reconfigurable en modo *codificador*.

5.2.4 Comparador / Multiplexor / Demultiplexor

Para llevar a cabo un mejor control sobre un determinado proceso es necesario capturar una serie de datos que generalmente tienen un carácter analógico, mientras que su tratamiento, almacenamiento y análisis son más eficaces cuando se hace digitalmente. Esto implica una serie de módulos electrónicos que permitan llevar a cabo una transformación de los datos desde el campo analógico al campo digital, a este conjunto de módulos se le denomina sistema de adquisición de datos (SAD); su estructura general está compuesta por diferentes elementos entre los cuales se encuentran comparadores, multiplexores y demultiplexores[30].

Un circuito comparador combinatorio compara dos entradas binarias para indicar la relación de igualdad o desigualdad entre ellas, por medio de "tres banderas lógicas" que corresponden a las relaciones igualdad, *mayor que* y *menor que*.

En cambio un circuito multiplexor consiste de un arreglo de llaves electrónicas conectadas en paralelo con una línea de salida común. Las llaves se activan una por vez. Su diseño incluye un decodificador que activa la llave de acuerdo a la palabra binaria presente en sus entradas. Existen multiplexores analógicos y digitales; en ocasiones son utilizados para la obtención de funciones booleanas[31] e incluso en implementaciones de sistemas basados en organismos biológicos[32], por lo que es una herramienta ideal para el desarrollo de aplicaciones en el área biomédica[33].

Por otro lado el circuito demultiplexor es un sistema que complementa al multiplexor, conecta la información de una línea de entrada hacia una de varias líneas de salida de acuerdo a un código de selección. La idea fundamental de utilizar multiplexores y demultiplexores radica en el ahorro de líneas de comunicación, es decir, el uso de una sola línea para realizar múltiples funciones o para conectar a través de ella diversas fuentes de información.

Para los circuitos del comparador, multiplexor y demultiplexor, se determinó que mediante un solo sistema era posible llevarlos a cabo, para ello se consideraron inicialmente las opciones que se presentaron previamente en 5.3 y 5.4 y un sumador más con ganancias unitarias como se representan en la figura 5.41.

Desde un punto de vista experimental, es posible instrumentar estos arreglos de amplificadores operacionales como se ilustra en la figura 5.41, cada diagrama eléctrico es un bloque de la estructura propuesta. Los valores de las ganancias de a_0, \dots, a_5 son asignados en el bloque M_1 ; para el caso de M_3 , el cual corresponde al sumador con ganancia unitaria, por lo que las resistencias en el diagrama eléctrico se colocan todas del mismo valor R y finalmente en el bloque M_2 corresponde al circuito ventana que define el umbral de operación.

Es así que para el diseño de este dispositivo se propuso un esquema que se muestra en la figura 5.42, el cual considera un bloque M_1 , cuatro bloques M_2 y un bloque M_3 .

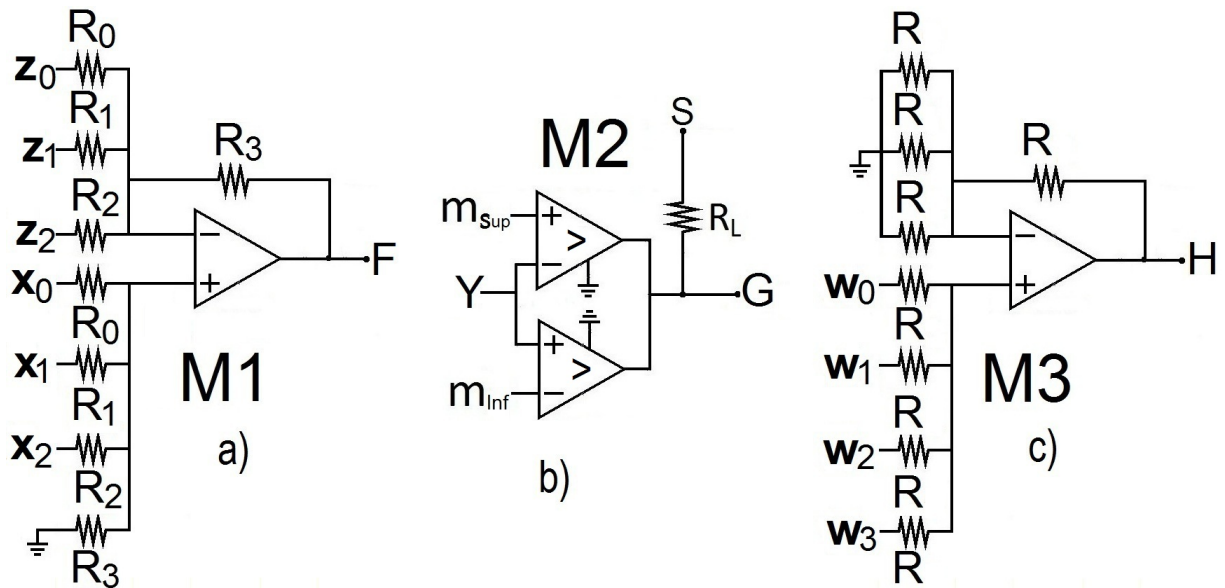


Figura 5.41: Circuitos empleados para sistema reconfigurable comparador / multiplexor / demultiplexor.- a) Bloque M_1 , modelo para sumador restador, b) Bloque M_2 , modelo para circuito ventana, c) Bloque M_3 , modelo para sumador con ganancia unitaria.

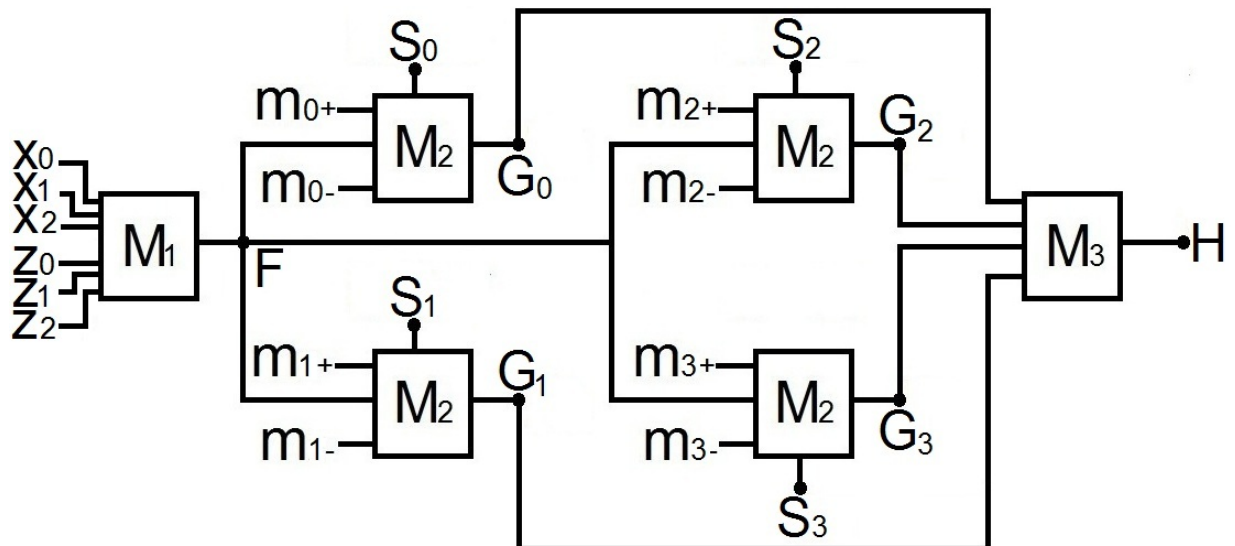


Figura 5.42: Circuitos empleados para sistema reconfigurable comparador / multiplexor / demultiplexor.- a) Bloque M_1 , modelo para sumador restador, b) Bloque M_2 , modelo para circuito ventana, c) Bloque M_3 , modelo para sumador con ganancia unitaria.

El diseño se implementó sobre una tarjeta de circuito impreso (PCB). Para el circuito experimental se utilizó el amplificador operacional LM124 (M_1 y M_3), el comparador LM311 (M_2), resistencias, fuentes de poder de $\pm 15V$, 3A modelo PS280 de la compañía Tektronix, para la adquisición de las señales un osciloscopio TDS2014 de la compañía Tektronix, finalmente para generar las secuencias de entrada y comprobar su funcionamiento se emplea un microcontrolador PIC16F877A de la compañía Microchip, el cual maneja señales de salida de 2.3V, en base a este nivel de voltaje se han propuesto los valores para las resistencias, las cuales se muestran en la tabla 5.14, estos valores hacen que las entradas del bloque M_1 presenten la relación mostrada en tabla 5.15.

	<i>Mux/Demux/Comp</i>
R_0	23k Ω
R_1	11.5k Ω
R_2	5.7k Ω
R_3	10k Ω
R_L	500 Ω
R	10k Ω

Tabla 5.14: Valores usados en las resistencias.

a_0x_0	a_1x_1	a_2x_2	a_3z_0	a_4z_1	a_5z_2
1V	2V	4V	1V	2V	4V

Tabla 5.15: Voltajes en M_1 .

La figura 5.43 representa el esquema para un comparador de 2 números de 2 bits cada uno, la tabla de verdad que describe su comportamiento se muestra en la tabla 5.16.

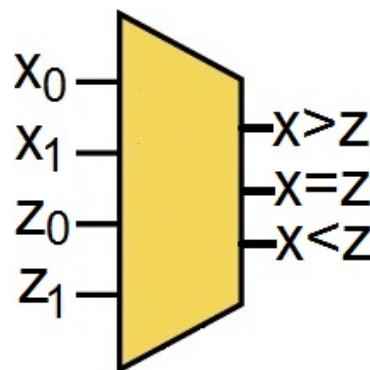


Figura 5.43: Esquema de entradas y salidas presentes en un comparador de 2 números de 2 bits.

Entradas				Salidas		
x_1	x_0	z_1	z_0	$G_0 \rightarrow x > z$	$G_1 \rightarrow x = z$	$G_2 \rightarrow x < z$
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

Tabla 5.16: Tabla de verdad de un comparador de 2 números de 2 bits.

Para este modelo se utilizó el sistema mostrado en la figura 5.44, el cual corresponde a la estructura *GoF*, observe que en comparación con el sistema propuesto en la figura 5.42 este descarta el uso de dos entradas, lo que implica de manera experimental que serán conectadas a 0V como se observa en su diagrama a bloques, en el sistema sólo se consideran los elementos mostrados en la figura 5.44, observe que tiene tres salidas constituidas por 3 bloques M_2 , los parámetros de configuración para estos bloques se presentan en la tabla 5.17.

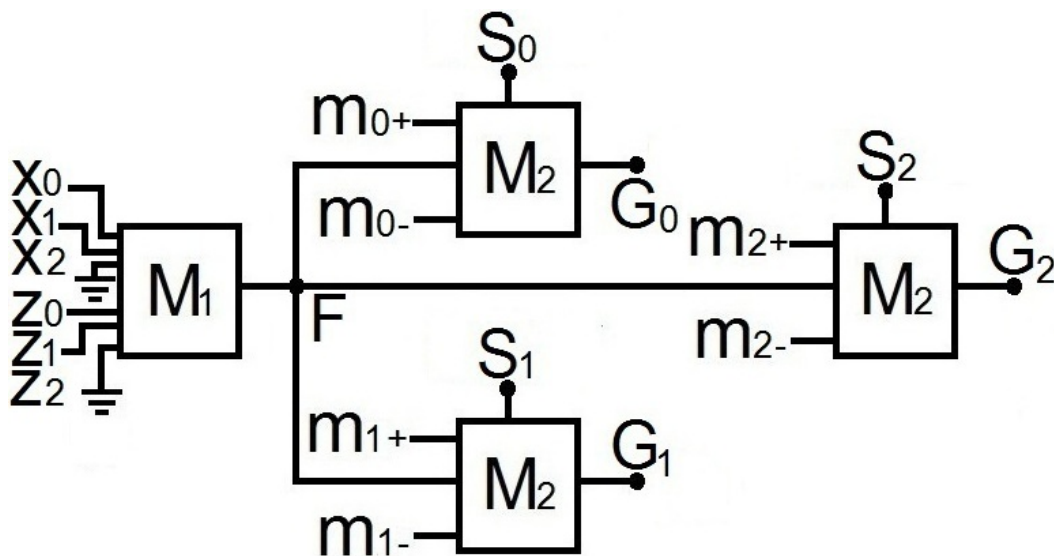
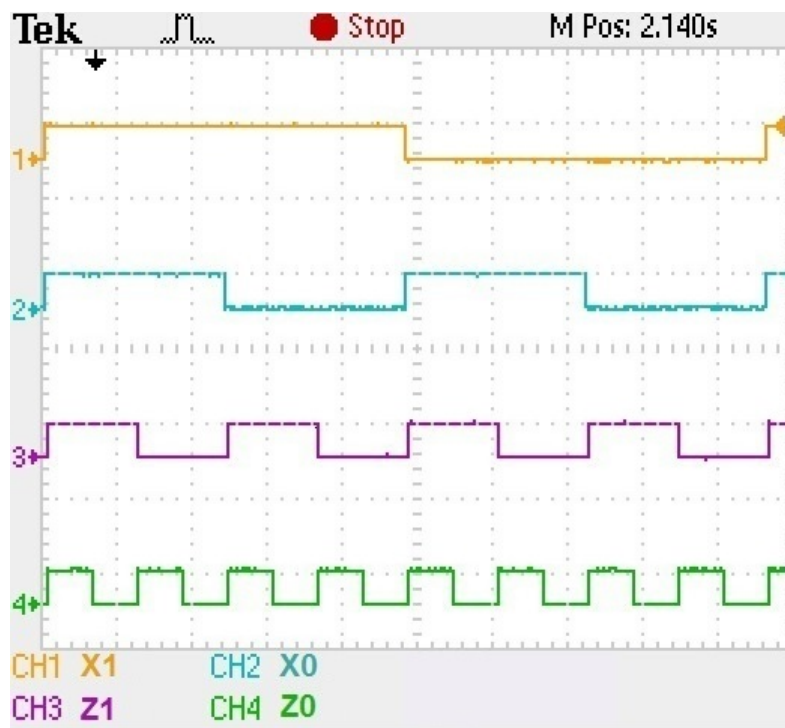


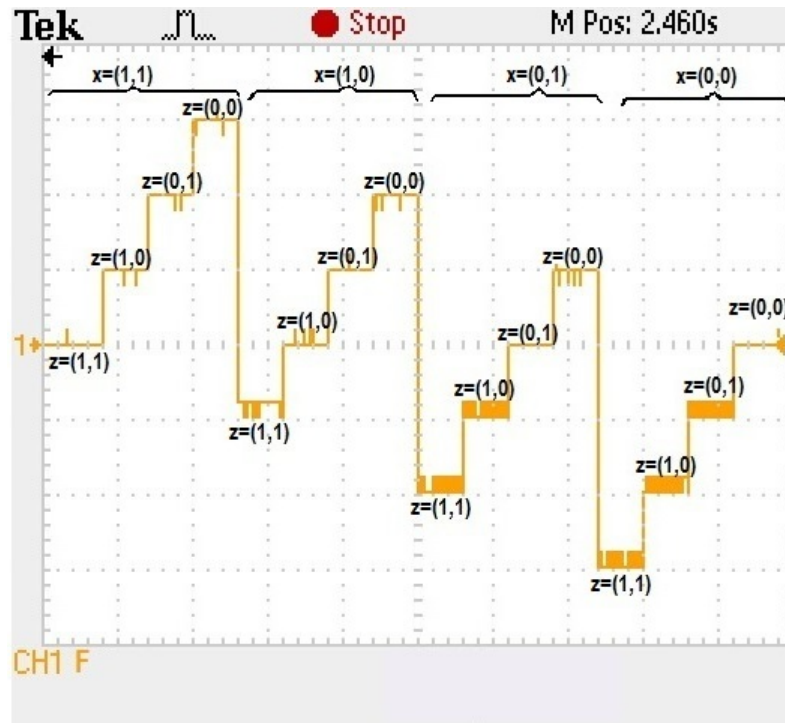
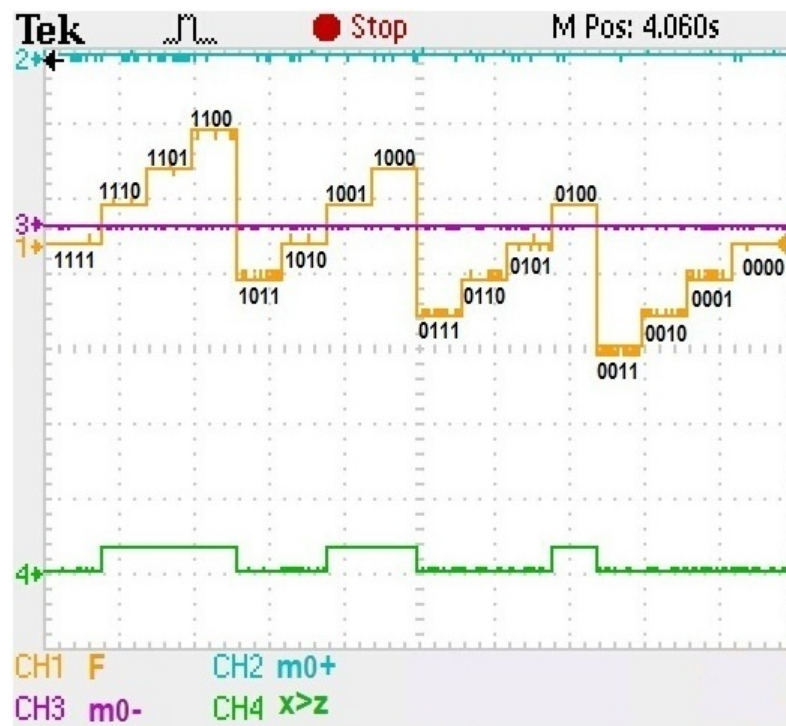
Figura 5.44: Diagrama a bloques para un comparador de 2 números de 2 bits.

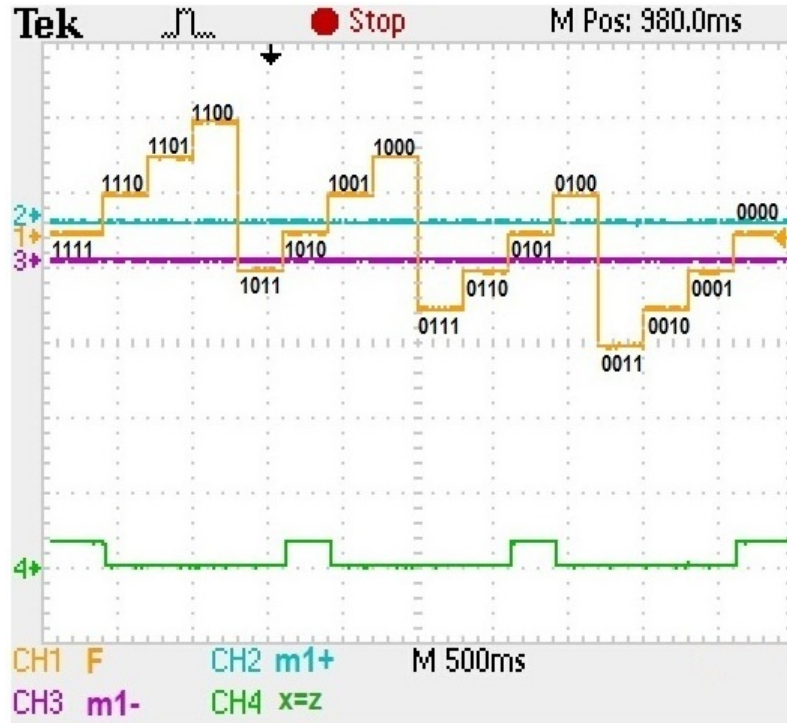
x_0	x_1	x_2	z_0	z_1	z_2	m_{0+}	m_{0-}	m_{1+}	m_{1-}	m_{2+}	m_{2-}
2.3V	2.3V	0V	2.3V	2.3V	0V	5V	0.5V	0.5V	-0.5V	-0.5V	-5V

Tabla 5.17: Voltajes para los distintos bloques M_2 del comparador.

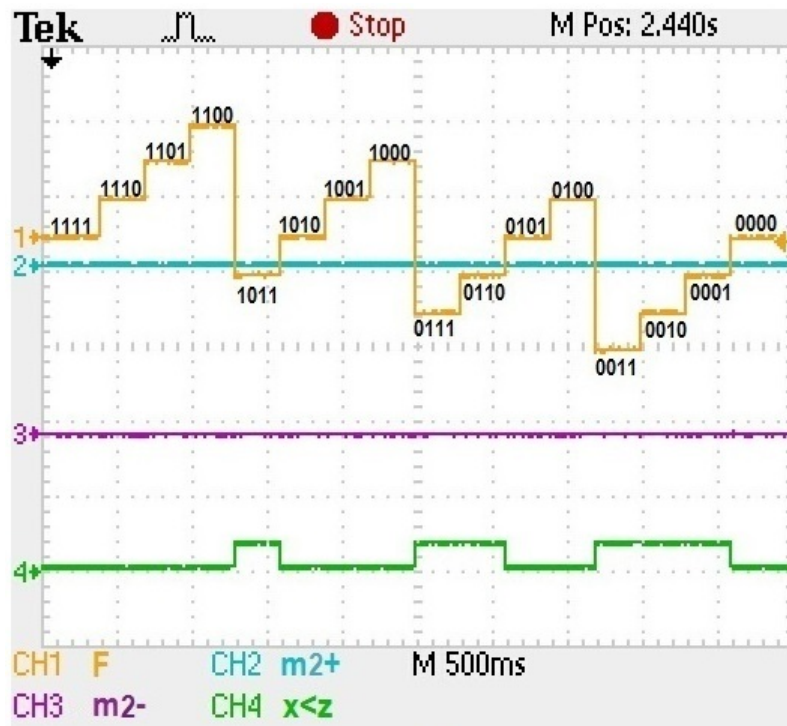
En las figuras 5.45, 5.46 y 5.47 se presentan los resultados obtenidos para este modelo; en la figura 5.45 a) las señales de entrada x_1 , x_0 , z_1 y z_0 son mostradas en los canales CH1:(amarillo), CH2:(azul), CH3:(morado) y CH4:(verde) respectivamente; la figura 5.46 a) corresponde a la variación de voltaje producida por la combinación de las entradas (salida de M_1), observe que las cuatro entradas producen un total de 16 combinaciones que van desde $x = (0,0)$, $z = (0,0)$ hasta $x = (1,1)$, $z(1,1)$; en las figuras 5.46 b), 5.47 a) y b) aparecen cuatro señales en cada imagen, en las 3 figuras se representa en el canal 1 la salida de M_1 ; los canales 2 y 3 corresponden a los parámetros m_+ y m_- de cada bloque M_2 ; por su parte en el canal 4 es la salida, es decir en c) el caso cuando $X > Z$, d) cuando $X = Z$ y e) cuando $X < Z$. Los valores de S_i se han fijado en 5V, por lo que la salida del sistema trabaja en estado binario, 5V equivale a un 1 lógico y 0V a 0 lógico.

a) Señales de entrada x_0, x_1, z_0, z_1 .Figura 5.45: Resultados de circuito reconfigurable en modo *comparador*.

a) Señal de salida del bloque M_1 .b) Función de salida ($x > z$).Figura 5.46: Resultados de circuito reconfigurable en modo *comparador*.



a) Función de salida ($x = z$).



b) Función de salida ($x < z$).

Figura 5.47: Resultados de circuito reconfigurable en modo *comparador*.

En la figura 5.48 se observa el bloque esquemático de un demultiplexor de 1 señal de entrada, 4 de salida y 3 líneas que sirven como llave de selección; en la tabla 5.18 se muestra su tabla de verdad, la cual describe su comportamiento.

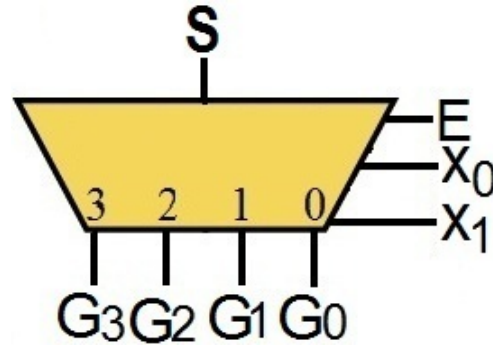


Figura 5.48: Esquema de entradas y salidas presentes en un demultiplexor 1×4 .

Señal de entrada	Llave de selección			Salidas			
S	$E(x_2)$	x_1	x_0	G_0	G_1	G_2	G_3
1	0	X	X	0	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	0	1	0	0
1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1

Tabla 5.18: Tabla de verdad del demultiplexor 1×4 .

En este caso se utiliza el sistema representado por la figura 5.42, pero con la diferencia que el último bloque M_3 se desconecta, de tal forma que el sistema se reduce como se muestra en la figura 5.49, note que las entradas z_1 y z_2 están conectadas a 0V ya que no se consideran en este caso; las salidas $G_0 \dots G_3$ están directamente ligadas a los valores S_i que para este caso serán de 5V, en este diseño se utiliza la entrada z_0 como un nivel de offset que siempre esta presente y que toma un valor de 3.5V; siendo así que los parámetros de configuración se muestran en la tabla 5.19.

x_0	x_1	x_2	z_0	z_1	z_2	m_{0+}	m_{0-}	m_{1+}	m_{1-}	m_{2+}	m_{2-}	m_{3+}	m_{3-}
2.3V	2.3V	2.3V	8V	0V	0V	1V	0V	2V	1V	3V	2V	4V	3V

Tabla 5.19: Voltajes para los distintos bloques M_2 del demultiplexor .

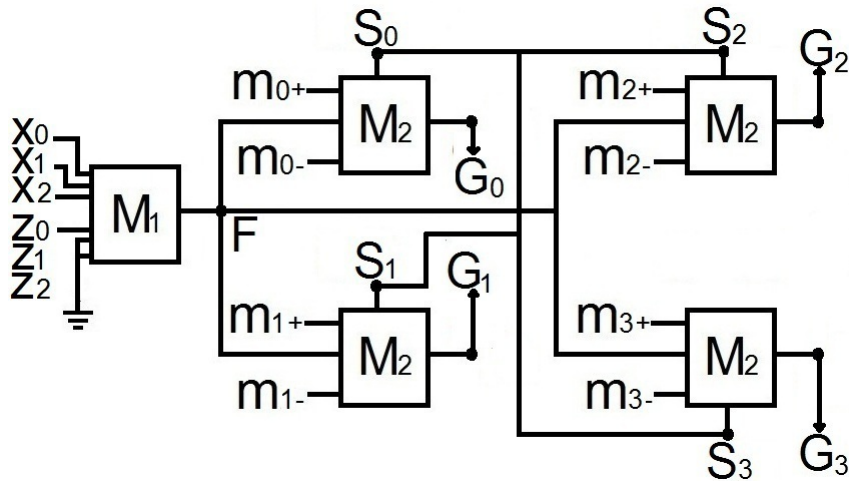


Figura 5.49: Diagrama a bloques para un demultiplexor 1×4 .

Los resultados de esta configuración se muestran en las figuras 5.50, 5.51 y 5.52. En la figura 5.50 a) se observan cuatro señales donde los canales 1, 2, y 3 corresponden a las señales de la llave de selección x_0 , x_1 y $E(x_2)$, por su parte en el canal cuatro se presenta la salida de M_1 la cual varía según las posibles combinaciones que van desde $E = 0$, $x_1 = 0$, $x_0 = 0$ hasta $E = 1$, $x_1 = 1$, $x_0 = 1$. En la figura 5.51 a) y b) y en la figura 5.52 a) y b) se pueden observar 4 señales en cada imagen; en cada una el canal 1 es la respuesta del bloque M_1 , los canales 2 y 3 son los parámetros m_+ y m_- de cada bloque M_2 ; el canal 4 es la salida de cada uno de los bloques M_2 donde para b) corresponde a G_0 , c) a G_1 , d) a G_2 y e) a G_3 . Note como ninguna salida opera cuando la señal E se mantiene en un nivel de 0 lógico.

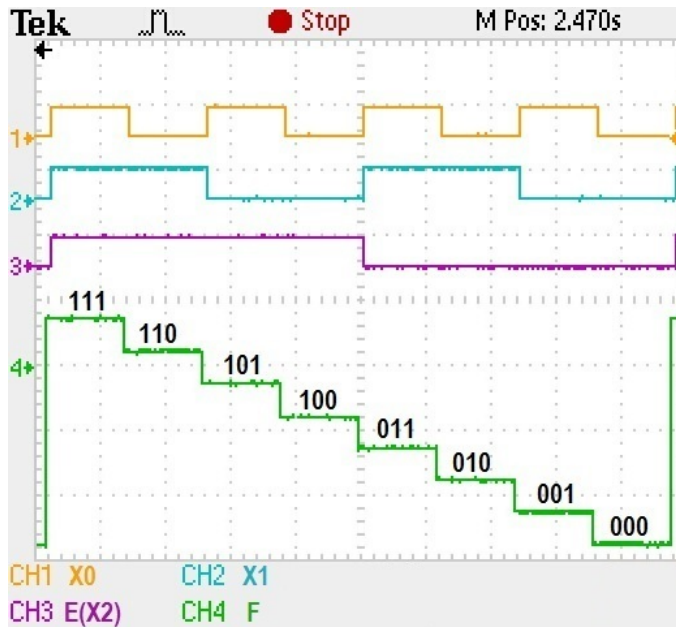
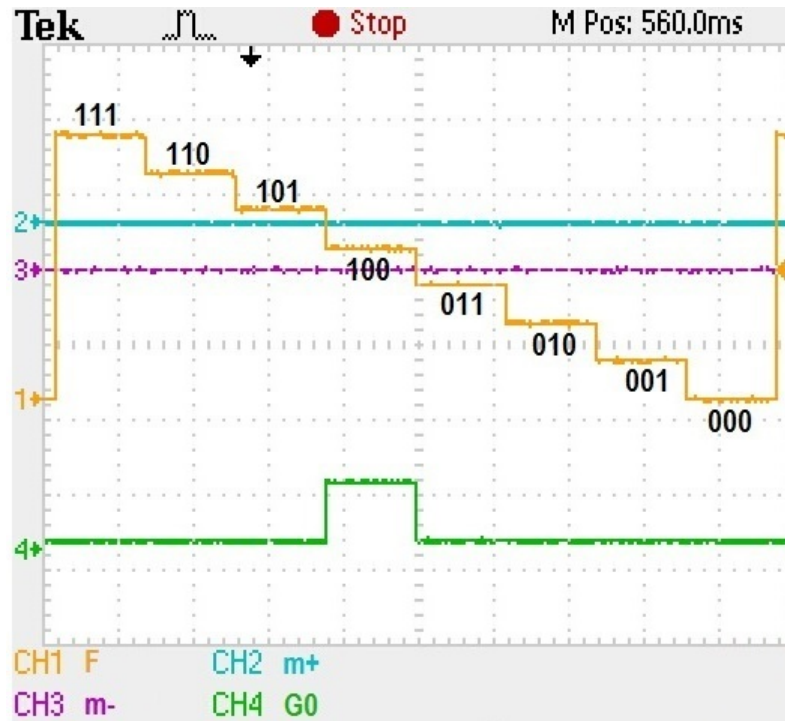
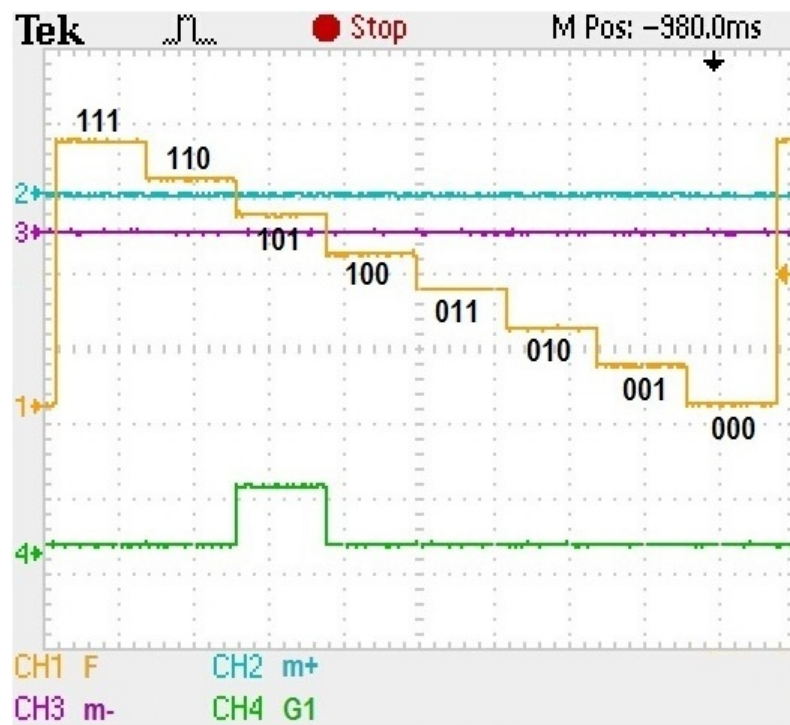
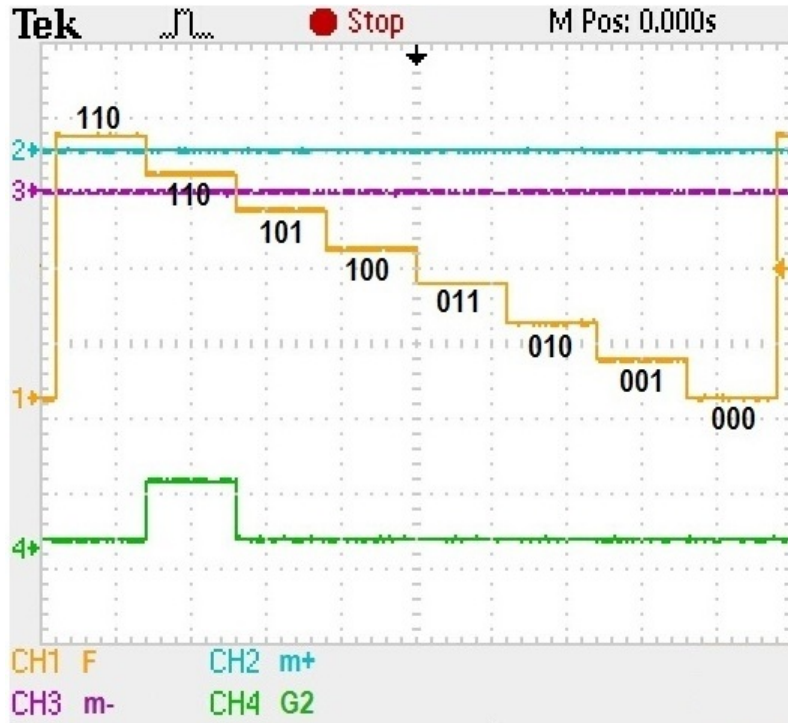
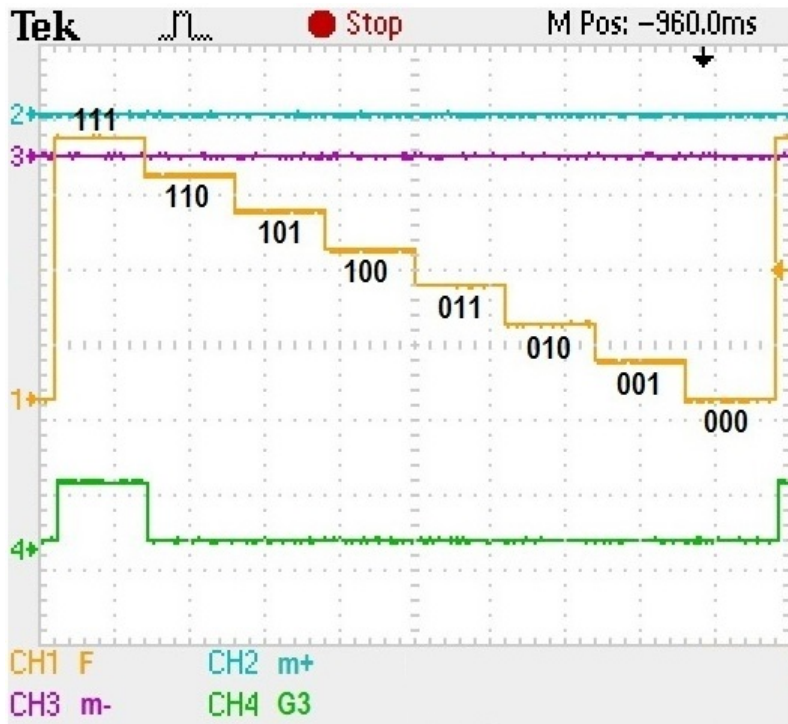


Figura 5.50: Resultados de circuito reconfigurable en modo *Demultiplexor*. a) Señales de entrada x_0, x_1, x_2 y función F .

a) Función de salida G_0 .b) Función de salida (Función de salida G_1).Figura 5.51: Resultados de circuito reconfigurable en modo *comparador*.



a) Función de salida G_2 .



b) Función de salida (Función de salida G_3).

Figura 5.52: Resultados de circuito reconfigurable en modo *comparador*.

En la figura 5.53 se observa el bloque esquemático de un multiplexor convencional de 4 señales de entrada, 1 salida y 2 líneas que funcionan como llave de selección, la función que describe su comportamiento se muestra en (5.9), la tabla 5.20 corresponde a su tabla de verdad.

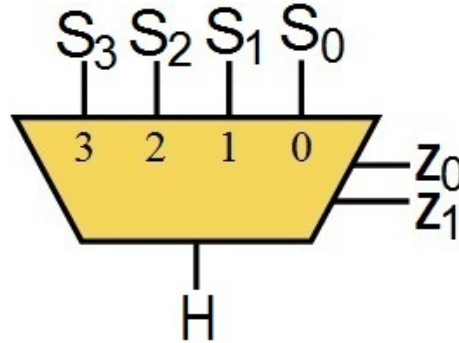


Figura 5.53: Bloque esquemático que muestra las entradas, salidas y entradas de control en un multiplexor 4×1 .

$$G = (S_1\bar{x}_0\bar{x}_1) + (S_2x_0\bar{x}_1) + (S_3\bar{x}_0x_1) + (S_4x_0x_1). \tag{5.9}$$

Llave de selección		
z_1	z_0	H
0	0	S_1
0	1	S_2
1	0	S_3
1	1	S_4

Tabla 5.20: Tabla de verdad del multiplexor 4×1 .

Para realizar el multiplexor se consideró el circuito propuesto por la función compuesta $HoGoF$ en la figura 5.42, dejando activas solo las entradas z_0 y z_1 para el control de la llave de selección, mientras que x_0 como un nivel de offset de 3.5V; las demás entradas se mantienen inactivas (0V) como se muestra en la figura 5.54.

Observe que el sistema presenta una reconfiguración doble, ya que se puede tomar como primer reconfiguración el alambrado entre bloques, mientras que su segunda reconfiguración se lleva a cabo mediante la sintonización paramétrica, en la tabla 5.21 se presentan dichos parámetros.

x_0	x_1	x_2	z_0	z_1	z_2	m_{0+}	m_{0-}	m_{1+}	m_{1-}	m_{2+}	m_{2-}	m_{3+}	m_{3-}
8V	0V	0V	2.3V	2.3V	0V	4V	3V	3V	2V	2V	1V	1V	0V

Tabla 5.21: Voltajes para los distintos bloques M_2 del multiplexor .

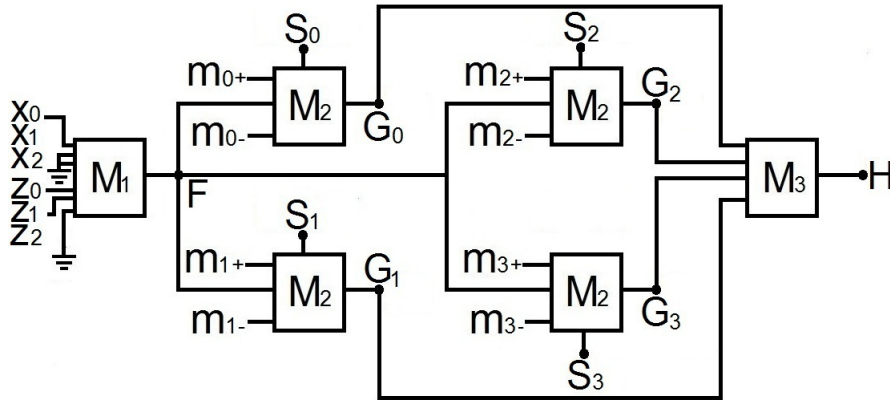


Figura 5.54: Diagrama a bloques para el multiplexor 4×1 .

Los resultados de esta configuración se presentan en las figuras 5.56, 5.57 y 5.58; en a) aparecen las señales de selección z_0 y z_1 , las cuales generan 4 posibles combinaciones que van desde $z_0 = 0$ y $z_1 = 0$ hasta $z_0 = 1$ y $z_1 = 1$, mientras que en el canal 3 se representa la salida del bloque M_1 . En b) se presentan las señales (S_i) que pasan a través del multiplexor, siendo S_0 una señal de tipo triangular, S_1 un tren de pulsos, S_2 una señal tipo senoidal y S_3 una constante de 5V. Observe que esto implica directamente que el sistema es capaz de trabajar con señales digitales y continuas. Por su parte en c), d), e) y f) se aprecian cuatro señales en cada imagen, donde el canal 1 es la salida del bloque M_1 , los canales 2 y 3 corresponden a los parámetros m_+ y m_- de cada bloque M_2 , en la parte posterior de cada imagen aparece el canal 4 el cual representa la salida de cada bloque M_2 , observe que de esta forma se asigna una señal S_i a cada una de las combinaciones dadas por la llave de paso; finalmente en g) se muestran dos señales donde el canal 1 representa la salida del bloque M_1 y el canal 2 la respuesta total del sistema a las 4 posibles combinaciones, salida del bloque M_3 .

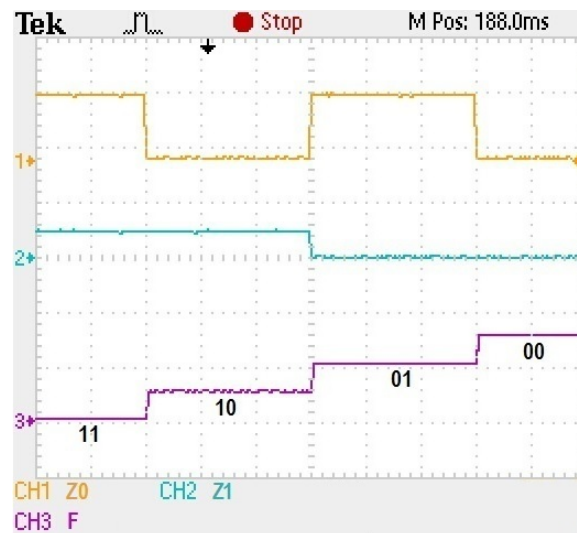
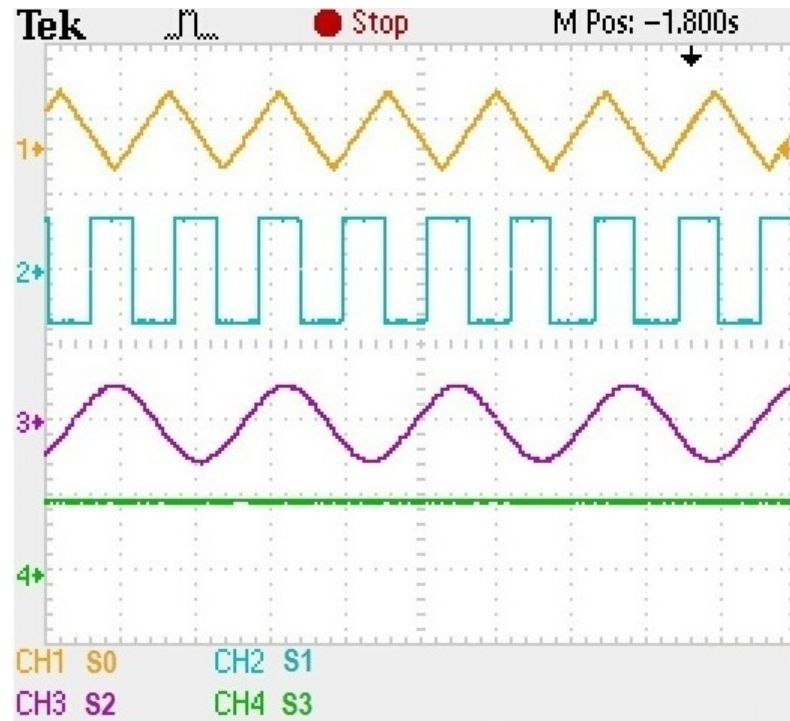
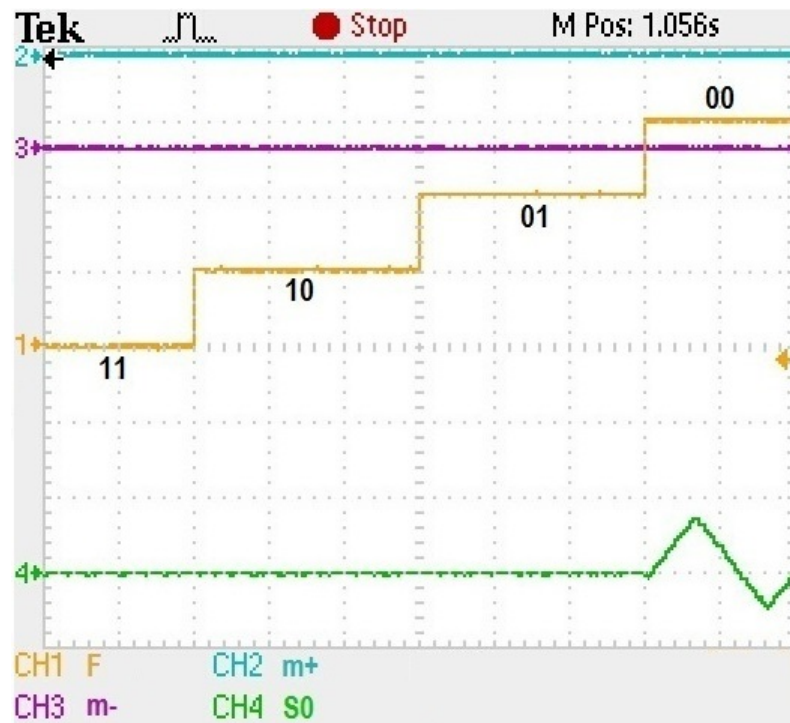
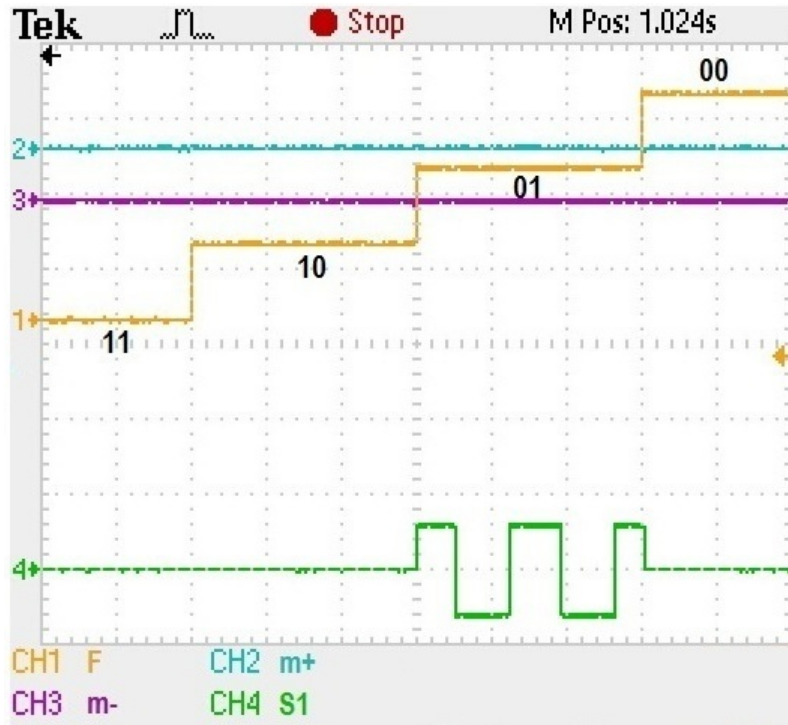
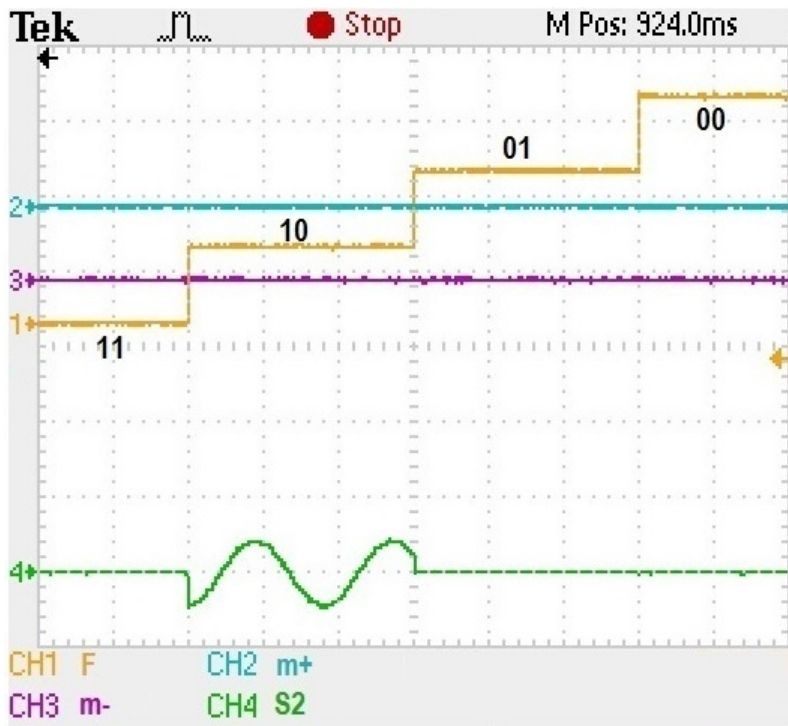


Figura 5.55: Resultados de circuito reconfigurable en modo *Multiplexor*. a) Señales de selección z_0, z_1 y función F .

a) Señales de entrada S_i .b) Multiplexado de S_0 .Figura 5.56: Resultados de circuito reconfigurable en modo *multiplexor*.

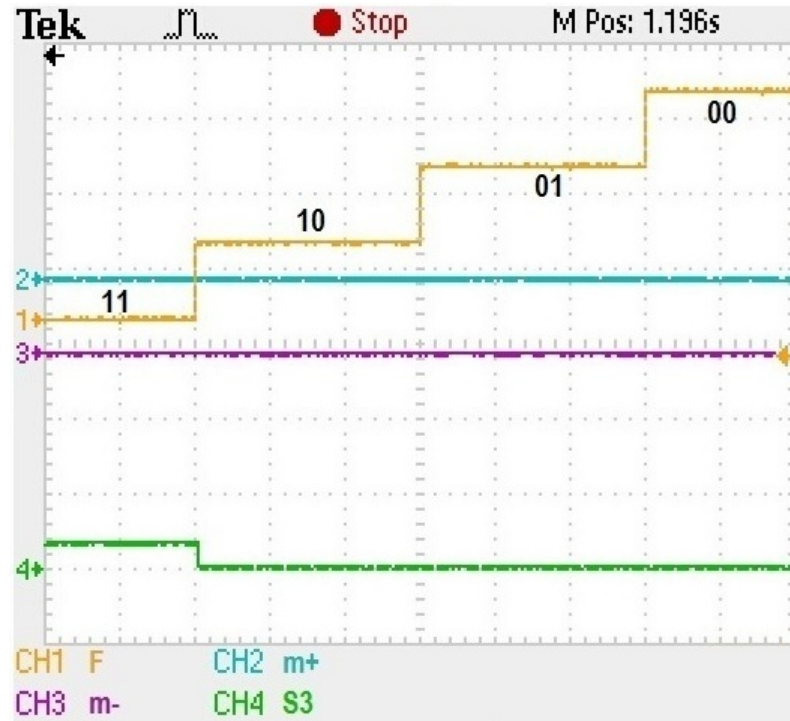
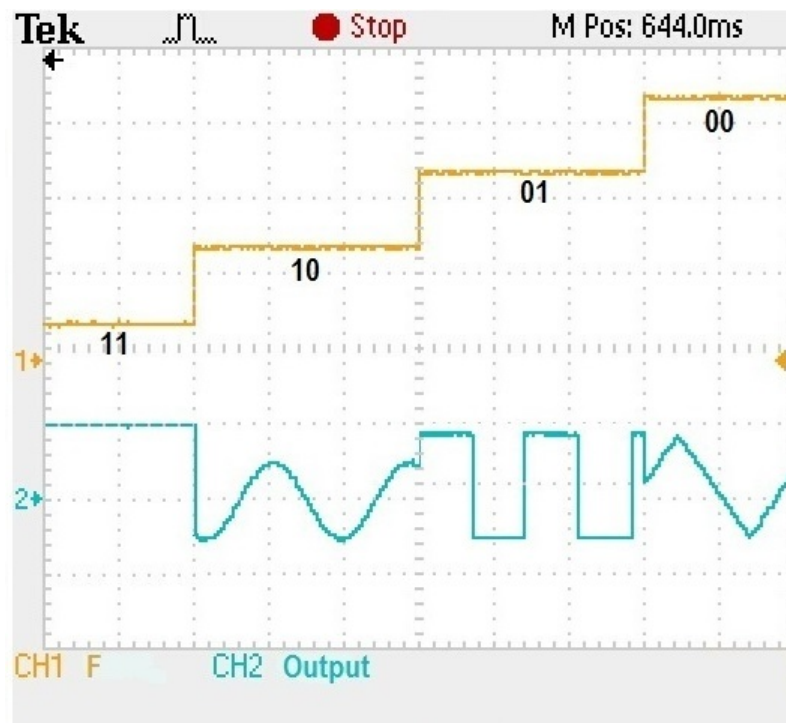


a) Multiplexado de S_1 .



b) Multiplexado de S_2 .

Figura 5.57: Resultados de circuito reconfigurable en modo *multiplexor*.

a) Multiplexado de S_3 .

b) Salida general del multiplexor.

Figura 5.58: Resultados de circuito reconfigurable en modo *multiplexor*.

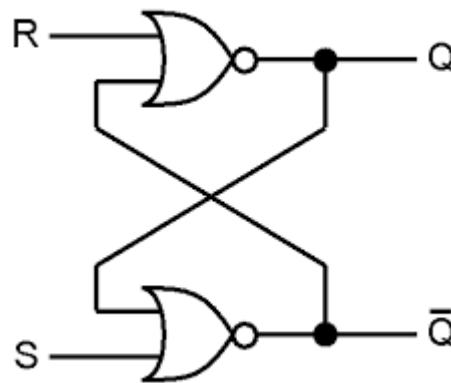
5.2.5 Elementos de almacenamiento: Latches

En un circuito digital, un dispositivo de almacenamiento puede mantener un estado binario por tiempo indefinido (mientras se mantenga un voltaje de alimentación en el circuito), hasta que una señal de entrada le indique que cambie de estado. Dentro de las importantes diferencias que se presentan entre varios dispositivos de almacenamiento destacan el número de entradas que poseen y la manera en que estas afectan al sistema.

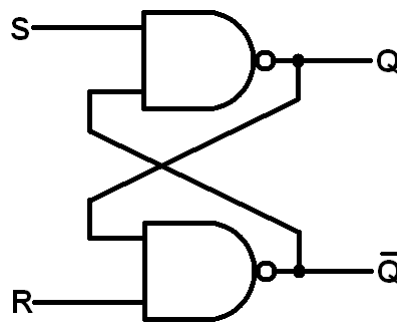
A los circuitos de almacenamiento que trabajan con niveles de señal se les conoce como *latches*, mientras que los que son controlados por una transición de reloj son llamados *flip-flop*.

Ambos dispositivos de almacenamientos están estrechamente relacionados ya que se considera que los latches son los circuitos básicos con los que se construyen todos los flip-flop [29].

El **Latch SR** es un circuito que convencionalmente se puede formar con compuertas NOR o NAND acopladas de forma cruzada como se muestra en la figura 5.59. En la tabla 5.22 se presenta la tabla de verdad de este dispositivo y en la figura 5.60 se presenta su respectivo cronograma.



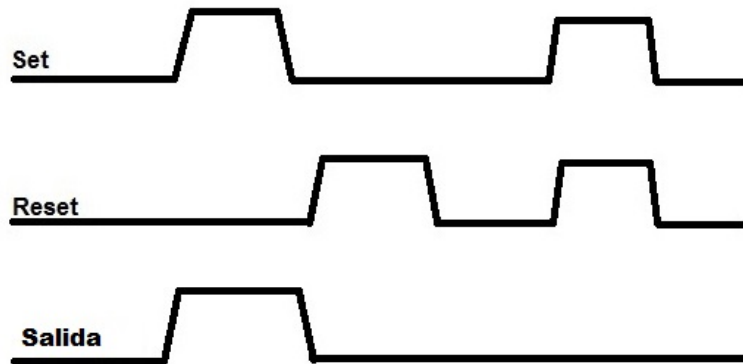
a) Diagrama usando compuertas NOR.



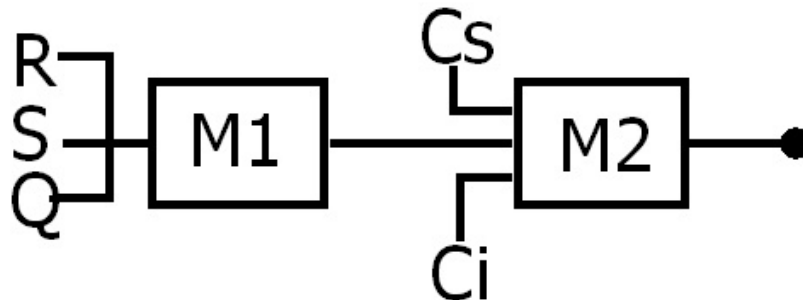
b) Diagrama usando compuertas NAND.

Figura 5.59: Diagramas combinacionales para *Latch SR*.

S	R	Q	Q'	Acción
0	0			Mantiene el dato
0	1	0	1	RESET
1	0	1	0	SET
1	1			Estado prohibido

Tabla 5.22: Tabla de verdad del multiplexor 4×1 .Figura 5.60: Cronograma (diagrama de tiempo) de un *Latch SR*.

Realizando un ejercicio de simulación, se propuso inicialmente un esquema donde la variable Q representa una entrada más al esquema reconfigurable utilizado anteriormente, en la figura 5.61 se representa este sistema mediante un diagrama de bloques. Al asignar ganancias de $S = 2$, $R = 4$ y $Q = 1$ y un umbral de operación acotado por las variables $C_s = 3.5V$ y $C_i = 0.5V$ se contemplo manejar una tabla de verdad como en 5.23, donde se pueden apreciar todas las permutaciones que pueden existir en las entradas.

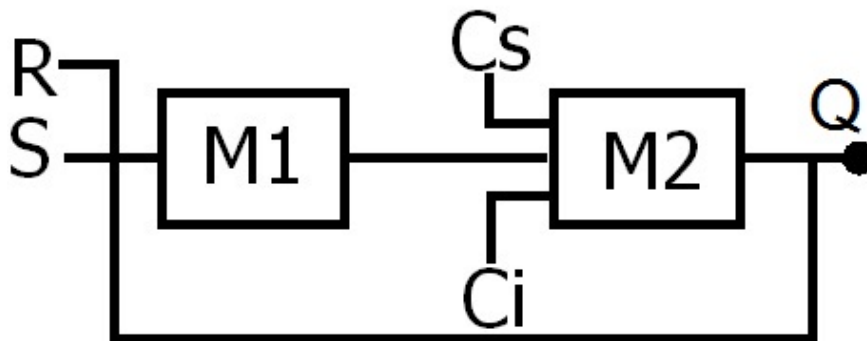
Figura 5.61: Diagrama a bloques para un *Latch SR*.

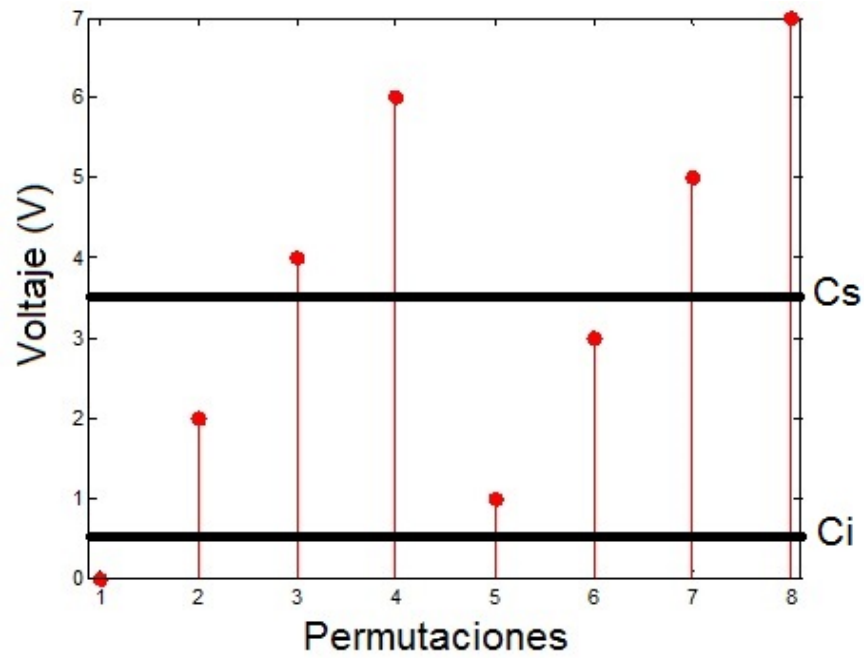
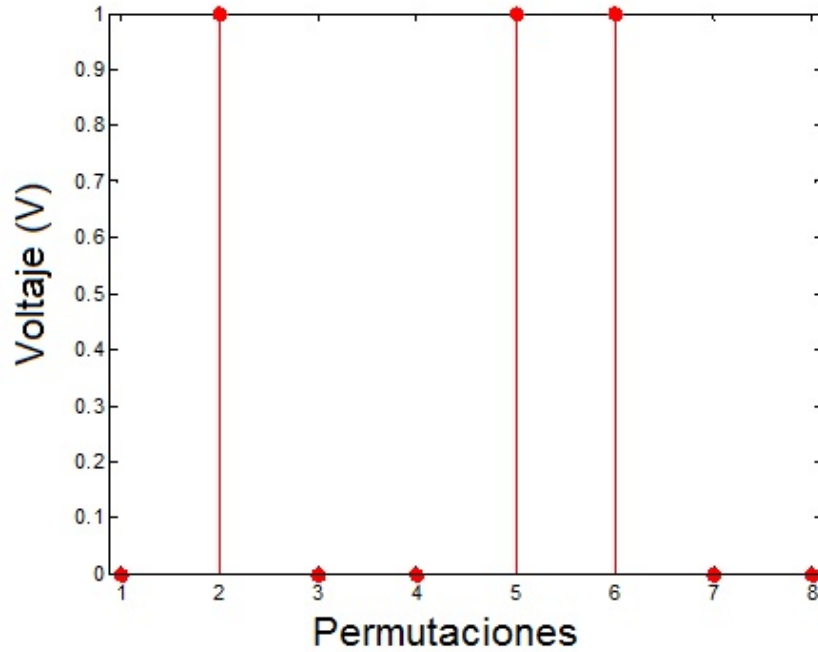
Q	R	S	Qn
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	X
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	X

Tabla 5.23: Tabla de verdad para un Latch SR.

En la figura 5.63 a), se presentan mediante una señal discretizada los valores que toma la salida del bloque M_1 , mientras que en b) se puede observar la respuesta de salida del bloque M_2 , es decir la señal Q .

Una vez obtenidos estos resultados, se optó por considerar un esquema que tuviera un lazo de retroalimentación como el que se presenta en la figura 5.62. Por su parte, en la figura 5.64 se muestra la respuesta de la simulación realizada para esta configuración. Los canales en verde y rojo corresponden a las señales que emulan las entradas R y S respectivamente. Se debe mencionar que para este caso se contemplo inicialmente un esquema de lazo abierto sin embargo a pesar de utilizar un sistema con retroalimentación, los estados oscilatorios del amplificador operacional son controlados por las variables R y S .

Figura 5.62: Diagrama a bloques para un *Latch SR* utilizando un lazo de retroalimentación.

a) Salida del bloque M_1 del diagrama 5.61.b) Salida del bloque M_2 del diagrama 5.61.Figura 5.63: Diagramas combinacionales para *Latch SR*.

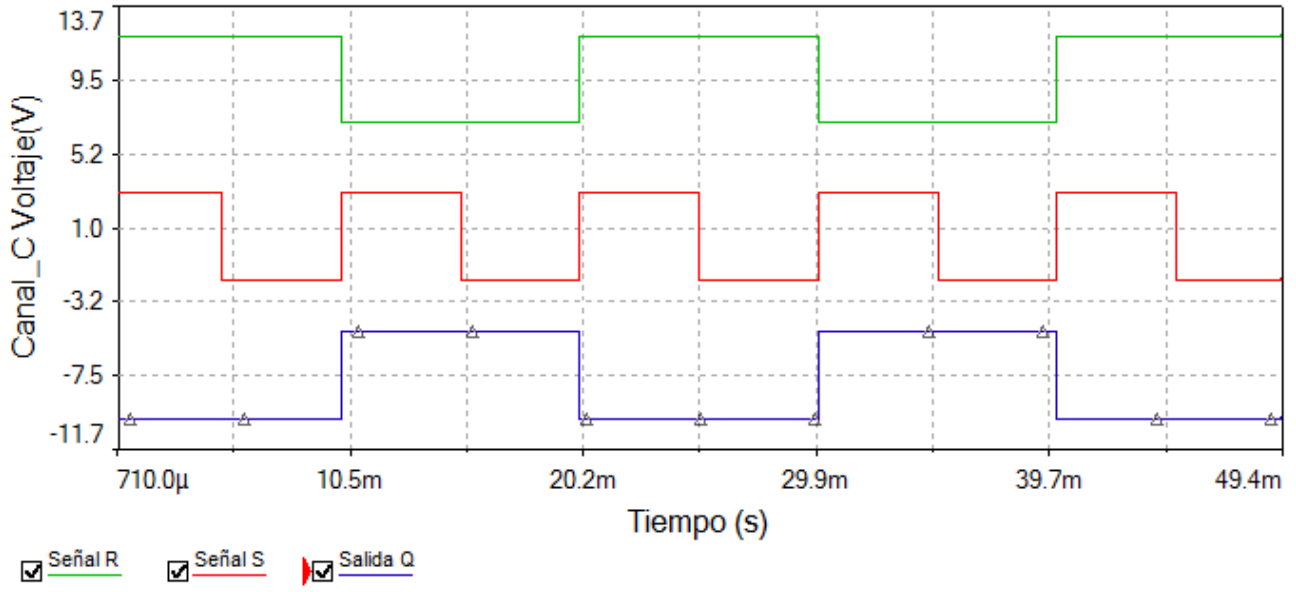


Figura 5.64: Resultado de simulación para un *Latch SR*.

5.2.6 Unidad Aritmética - Lógica

Las operaciones aritméticas son esenciales en la mayoría de las computadoras y calculadoras, para realizarlas es necesario implementar circuitos donde mediante la combinación de múltiples compuertas lógicas sea posible poder realizar operaciones como suma, resta, multiplicación y división de números binarios, a este tipo de circuitos se les conoce como *Unidad Aritmética - Lógica (ALU)* [4].

El funcionamiento de un ALU depende de una unidad de control, en esta se determina que operación aritmética o lógica se ejecutará. Es posible diseñar un ALU para cualquier operación, sin importar lo compleja que esta sea, sin embargo es necesario considerar la relación de compromiso que existe, ya que en cuanto más operaciones se realicen en el circuito, más costosa será y más espacio ocupará [34].

En la figura 5.65 se presenta el esquema de un ALU de dos bits, para este circuito las entradas del puerto *A* se combinan con las del puerto *B*, teniendo como salidas los bits del puerto *F*. La entradas de selección S_0 y S_1 determinan que operación se llevará a cabo, esta puede ser una función aritmética o lógica, no esta de más mencionar que los acarrees de entrada y salida solo tienen sentido cuando se tratan operaciones aritméticas.

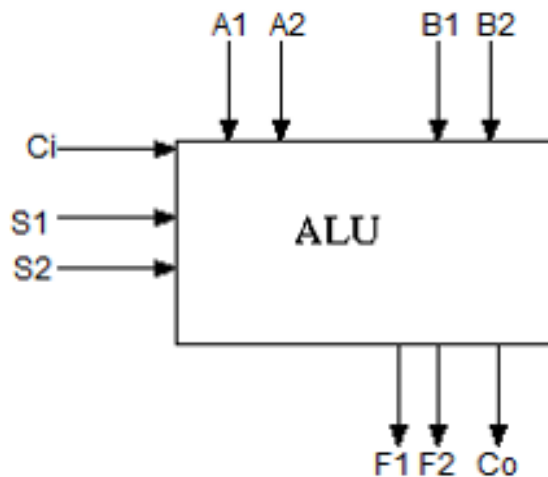


Figura 5.65: Diagrama de entradas y salidas para un ALU. Los puertos *A* y *B* corresponden a los datos de entrada que se van a operar; los bits *C* corresponden a los bits de acarreo de entrada y salida; en el puerto *S* están los bits que corresponden a las señales de control que permiten seleccionar que función se realizará; el puerto *F* contiene los bits de salida del sistema.

Convencionalmente, existen diversas formas para conformar un ALU, siendo la más sencilla el manipular la sección aritmética independientemente de la sección lógica, considerando en todo momento unir posteriormente ambas secciones. Inicialmente se contempla la sección lógica, para este trabajo de investigación, por lo que se propone partir de un esquema combinacional como el que se presenta en la figura 5.66.

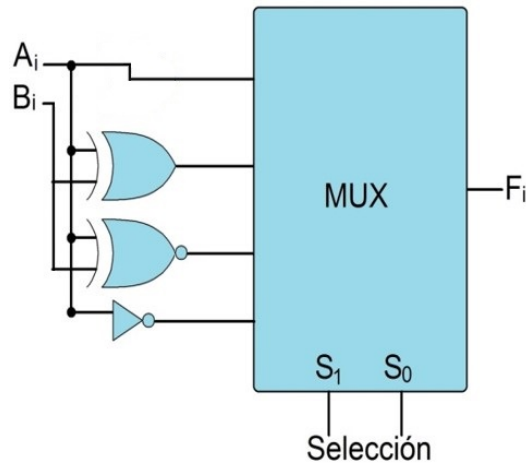


Figura 5.66: Diagrama de la sección lógica del *ALU* (1 bit).

Como se puede observar en el diagrama de la figura 5.66, los datos se manipulan de manera separada, es decir se opera bit a bit. Las funciones lógicas contempladas para el diseño que se propone en este trabajo de investigación son: BUFFER, XOR, XNOR y NOT. Hay que notar que es posible considerar cualquier función lógica.

La tabla de verdad 5.24 muestra el control de selección de las funciones lógicas.

S_1	S_0	Salida	Función F_1
0	0	$F = A$	BUFFER
0	1	$F = A \oplus B$	XOR
1	0	$F = \overline{A} \oplus B$	XNOR
1	1	$F = \overline{A}$	NOT

Tabla 5.24: Tabla de selección de funciones para la sección lógica de un ALU.

Para la sección aritmética se contemplaron las funciones mostradas en la tabla 5.25; hay que mencionar que estas operaciones se realizan tomando un número binario en el *puerto A* y otro en el *puerto B*, mientras que el acarreo de entrada C_i , toma valor lógico dependiendo de la función que se vaya a implementar.

Sin embargo, para poder llevar a cabo esta implementación, es necesario diseñar un sistema lógico bajo el concepto de intervenir cada bit del puerto *B*, en la tabla 5.26 se presenta el comportamiento de este subsistema siendo contemplado para un solo bit (B_i), mientras que en la figura 5.67 se presenta el diagrama lógico que cumple con esta función de selección.

Selección de función			Salida del selector N	Función F	Descripción
S_1	S_0	C_i	N	F	
0	0	0	0	A	Transferir A
0	0	1	0	$A + 1$	Incrementar A
0	1	0	B	$A + B$	Sumar A más B
0	1	1	B	$A + B + 1$	Sumar A más B más 1
1	0	0	\overline{B}	$A + \overline{B}$	Agregar el complemento de 1 de B a A
1	0	1	\overline{B}	$A + \overline{B} + 1$	Agregar el complemento de 2 de B a A
1	1	0	Todos uno	$A - 1$	Decrementar A
1	1	1	Todos uno	A	Transferir A

Tabla 5.25: Tabla de selección de funciones para la sección aritmética de un ALU.

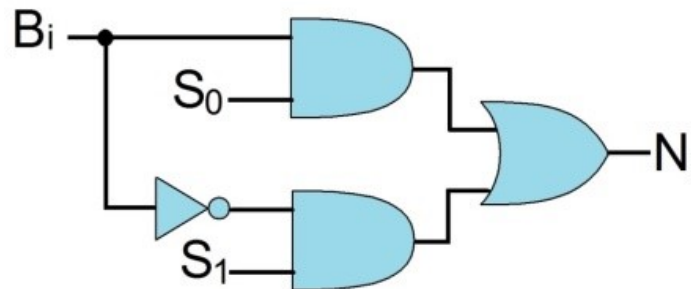


Figura 5.67: Diagrama lógico para la selección de funciones aritméticas.

Selector		Salida
S_1	S_0	N_i
0	0	0
0	1	B_i
1	0	\overline{B}_i
1	1	1

Tabla 5.26: Tabla de verdad para circuito lógico selector de funciones aritméticas.

De esta manera y utilizando el subsistema propuesto en la figura 5.67, resulta posible implementar las funciones propuestas en la tabla 5.25, siendo una solución agregar el número N_i a la entrada A mediante el uso de un sistema sumador en paralelo, repitiendo esta operación para cada bit que sea necesario considerar.

En la figura 5.68 se presenta el diagrama lógico para el arreglo anteriormente mencionado, un ALU considerando solo un bit de operación, mientras que en la figura 5.69 el caso para dos bits de operación.

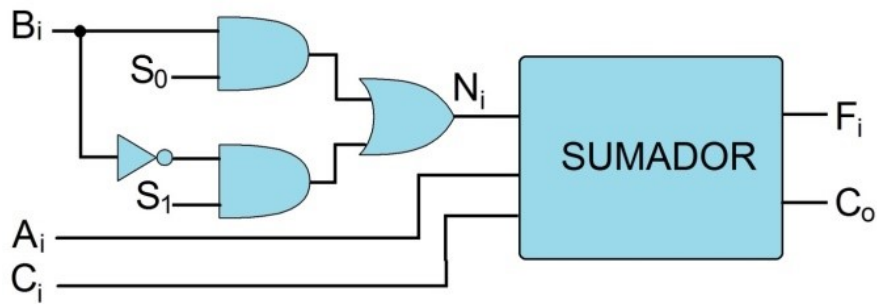


Figura 5.68: Diagrama lógico de un ALU de 1 Bit de operación.

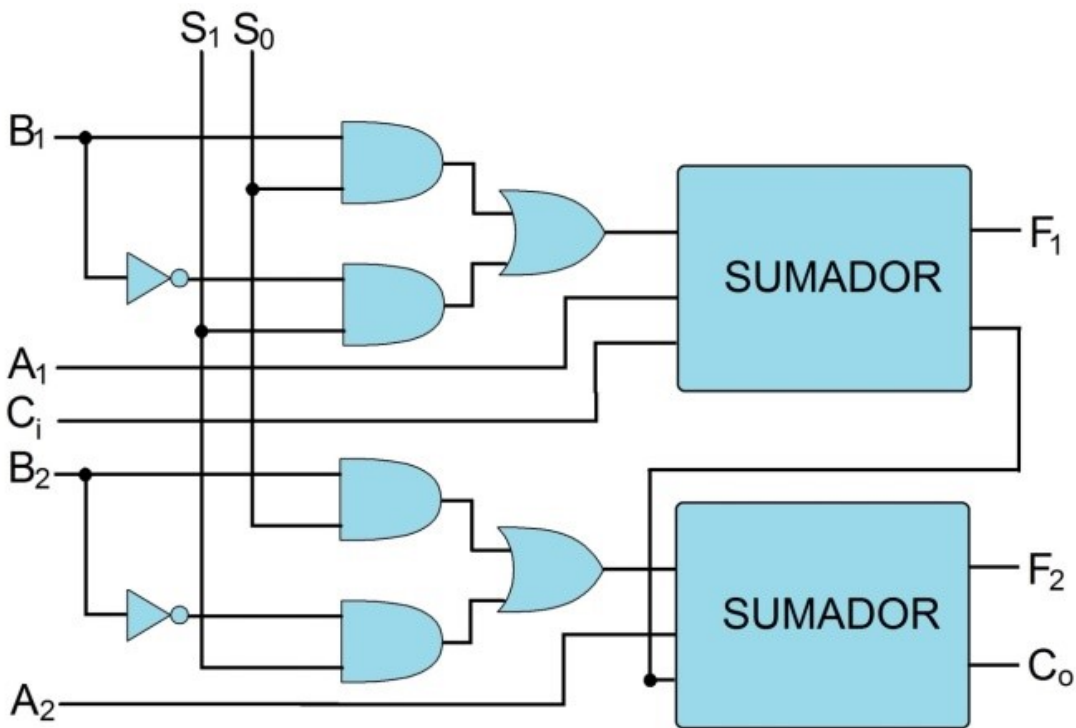
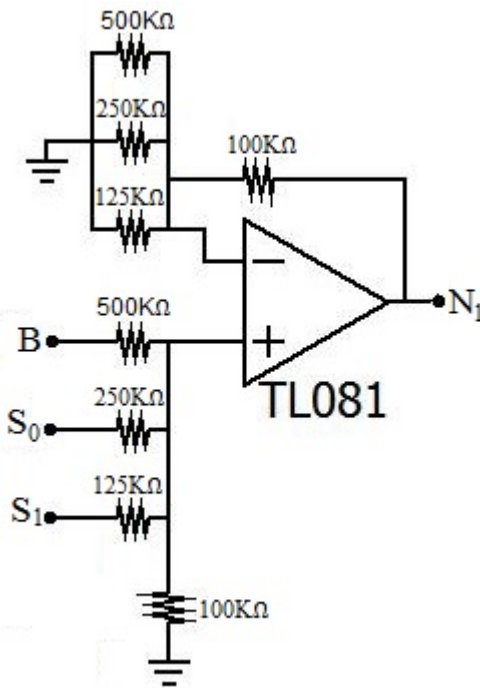
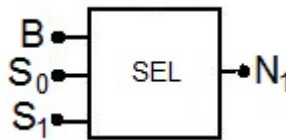


Figura 5.69: Diagrama lógico de un ALU de 2 Bits de operación.

Partiendo del diagrama mostrado en la figura 5.68, se propuso dividir en dos partes su diseño, la primera parte se presenta en la figura 5.70 a), mientras que 5.70 b) se presenta su bloque de simulación equivalente, hay que mencionar que se mantiene la misma estructura del diagrama presentado en 5.41 a).



a) Diagrama electrónico.



b) Representación en bloque.

Figura 5.70: Diagrama de la primera sección del seleccionador aritmético.

En la tabla 5.27 se presentan todas las permutaciones posibles de la tabla 5.26, esta tabla cobra sentido y permite detectar cuales son los niveles que activan un valor de "1" lógico a la salida de este sistema. La solución propuesta para este circuito fue utilizar dos circuitos ventana, como los utilizados en la figura 5.41 b), posteriormente usar el sumador de ganancia unitaria presentado en la figura 5.41 c). En la figura 5.71 se presenta el esquema del circuito ventana y su equivalente en diagrama de bloque.

Selector		Dato	Salida
S_1	S_0	B_i	N_i
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Tabla 5.27: Tabla de verdad para circuito lógico selector de funciones aritméticas.

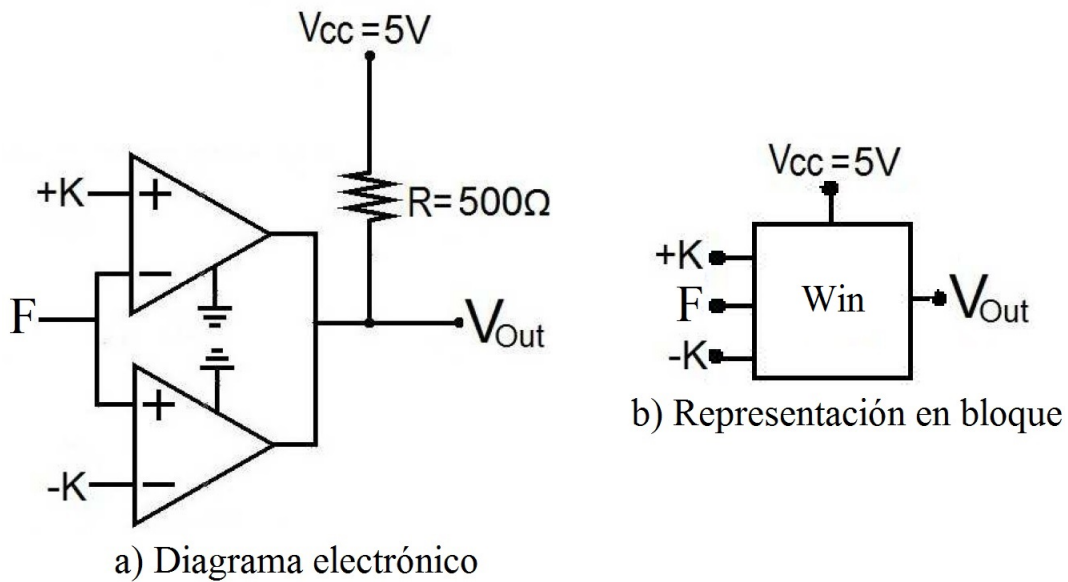


Figura 5.71: Diagrama de la segunda sección del seleccionador aritmético.

En la figura 5.72 se presenta la respuesta de la primera sección del circuito seleccionador, las regiones de color verde indican los umbrales de acción que seleccionan que niveles activan la salida.

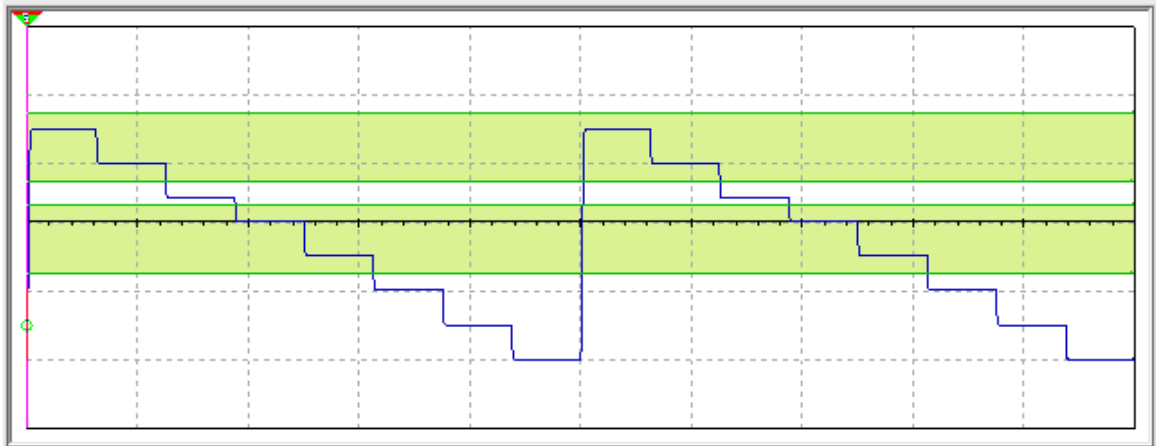


Figura 5.72: Respuesta en tiempo del sistema de la figura 5.70 (color azul) y regiones de umbral de acción (color verde).

Por lo tanto, el circuito completo para este seleccionador aritmético implica el uso de dos circuitos ventana, que operen como los dos umbrales de acción mostrados en la figura 5.72. De esta forma se propone un sistema como el que se presenta en la figura 5.73, el cual tiene como salida la suma de los dos circuitos ventana, es decir la combinación de los umbrales de acción presentados en la figura 5.72. En la figura 5.74 se presenta la respuesta en serie de tiempo para este sistema.

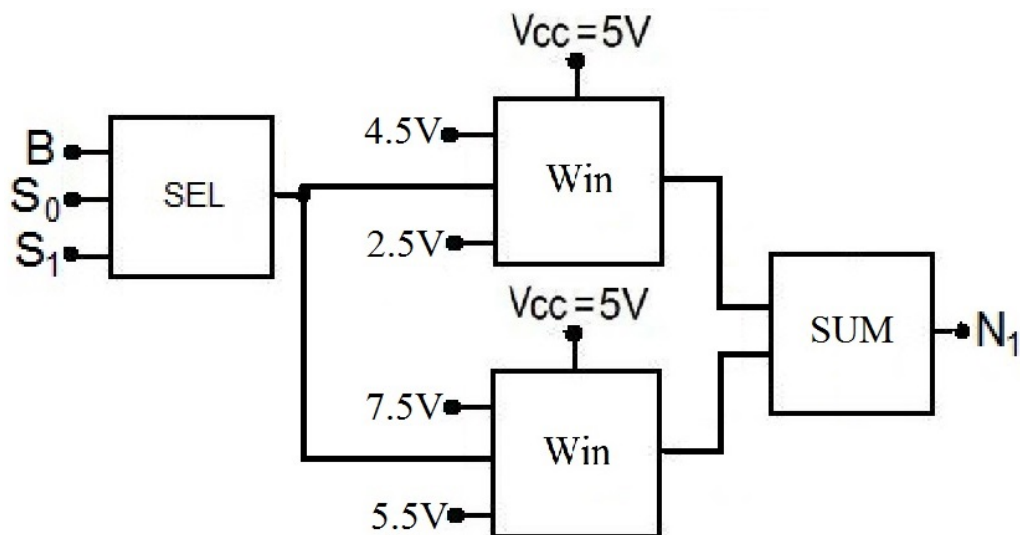


Figura 5.73: Diagrama de bloques de un circuito seleccionador aritmético.

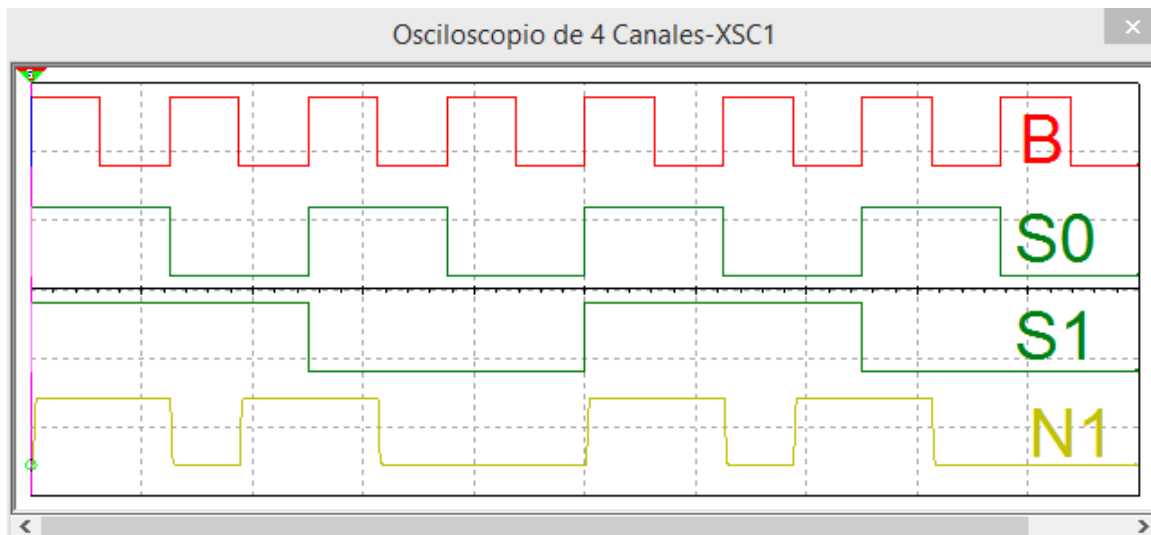


Figura 5.74: Respuesta en tiempo para circuito seleccionador aritmetico presentado en figura 5.73.

De esta forma es posible conformar un ALU de 1 bit, contemplando un bit de acarreo en entrada y otro de salida, para la implementación de este primer sistema se propuso el diagrama a bloques presentado en la figura 5.75, donde se aprecia que se utiliza el circuito seleccionador aritmético de la figura 5.73 pero ahora considerando agregar un circuito sumador como el presentado en 5.41 a) y tres circuitos ventana como los de la figura 5.41 b), dos de estos detectan los casos en que se obtienen niveles de 1 y 2 volts, mientras que el tercero detecta cuando es necesario implementar un bit de acarreo en la salida.

En la figura 5.76 se presentan los resultados para este ALU, tomando como entradas las series A , B , C , S_0 y S_1 , y como salidas F y C_0 .

Para formar un ALU de dos bits de operación y un bit de acarreo de entrada y otro de salida, se tomó la misma estructura presentada en la figura 5.75 y conectando de forma paralela las variables correspondientes al selector de operación (S_0 y S_1); en cuanto al acarreo, inicialmente se considera el acarreo de salida correspondiente al bit menos significativo, siendo este el que representa ahora el bit de acarreo de entrada en la sección del segundo bit, de esta manera el arreglo que se tiene es en serie para el bit de acarreo y en paralelo para el selector de función. El diagrama a bloques de este sistema se presenta en la figura 5.77 mientras que su respectiva respuesta se muestra en la figura 5.78.

Es necesario considerar que en cuanto a los resultados de las funciones lógicas, son operaciones que se realizan bit a bit, por lo tanto no utilizan una señal de acarreo. Esta condición conlleva que realice la propuesta de mantener la misma estructura física y explotar las propiedades de reconfiguración, ya que con solo utilizar un circuito ventana para cada bit de salida se obtiene este comportamiento, es así que los voltajes de los demás umbrales de detección se sintonizan en 0V. En la figura 5.79 se representan las funciones lógicas que se propusieron en la tabla 5.24 para el caso de 1 solo bit de operación.

Finalmente utilizando el mismo esquema propuesto en la figura 5.77 se llevan a cabo las operaciones lógicas de la tabla 5.24, pero ahora contemplando un sistema de dos bits de operación, los resultados de este sistema se presentan en la figura 5.80.

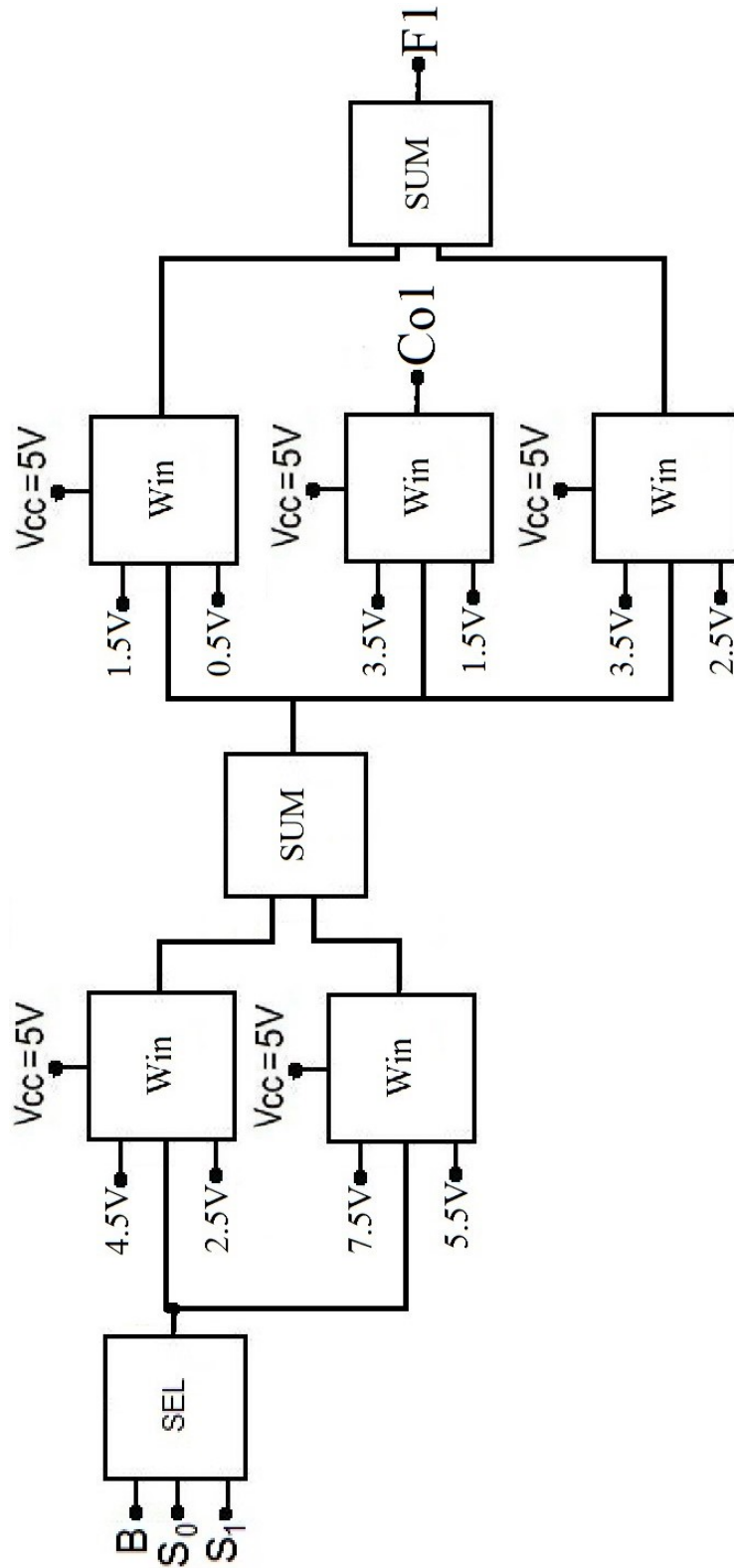


Figura 5.75: Diagrama a bloques de un ALU 1 Bit

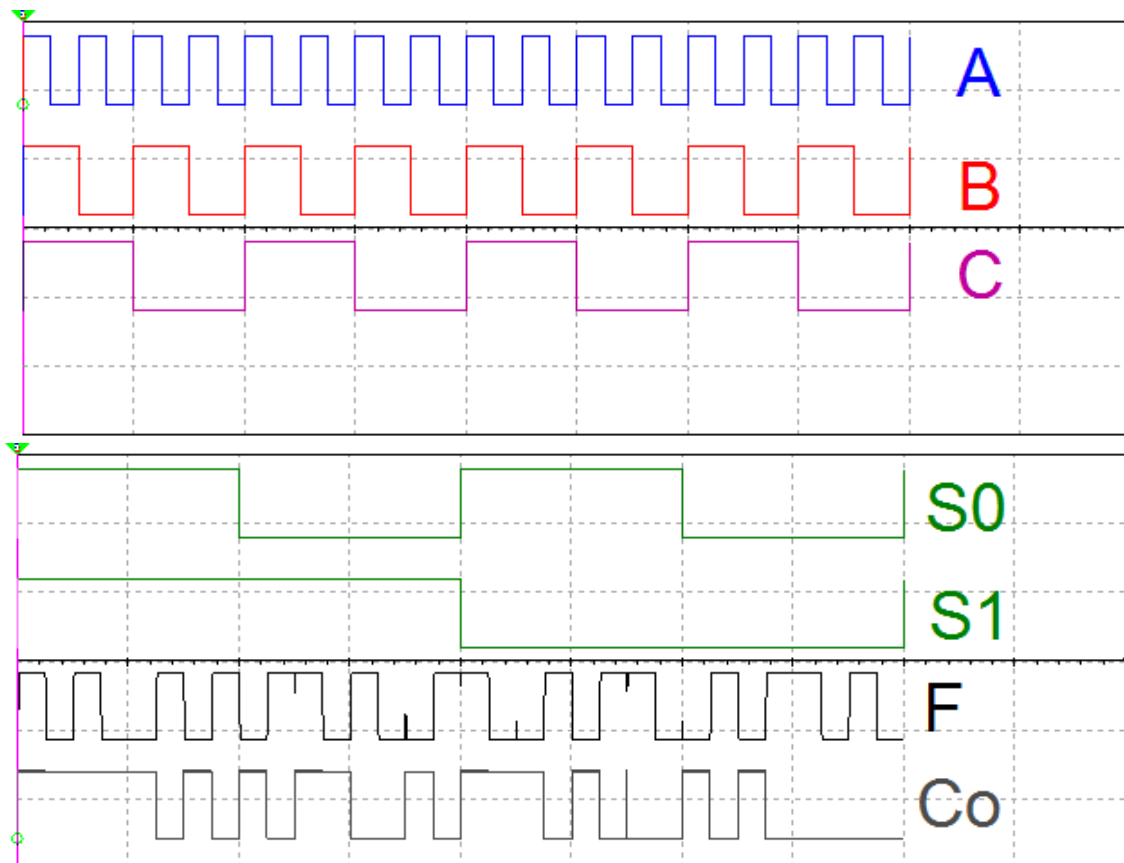


Figura 5.76: Respuesta de funciones aritméticas para un ALU 1 Bit

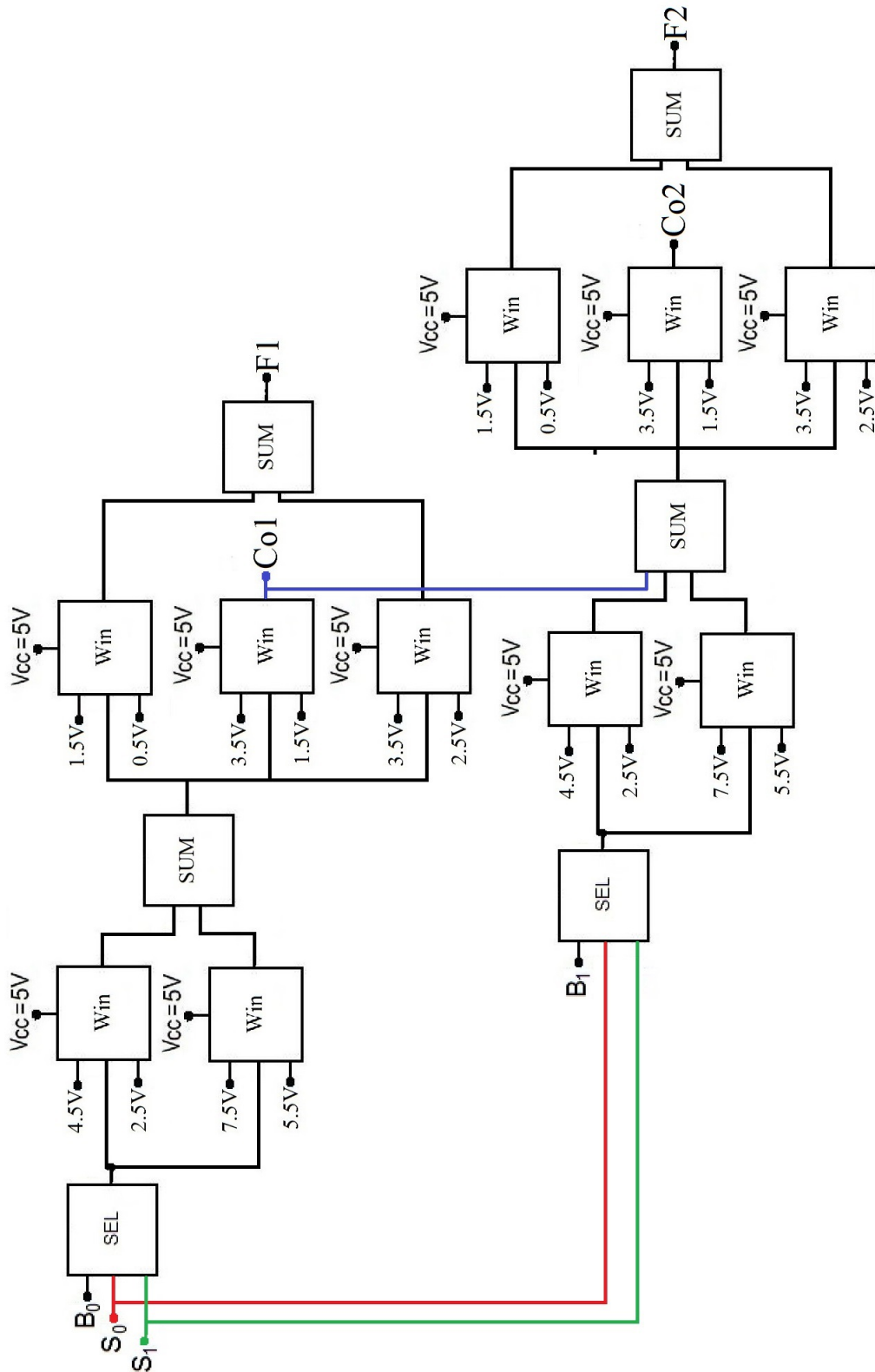


Figura 5.77: Diagrama a bloques de un ALU 2 Bit

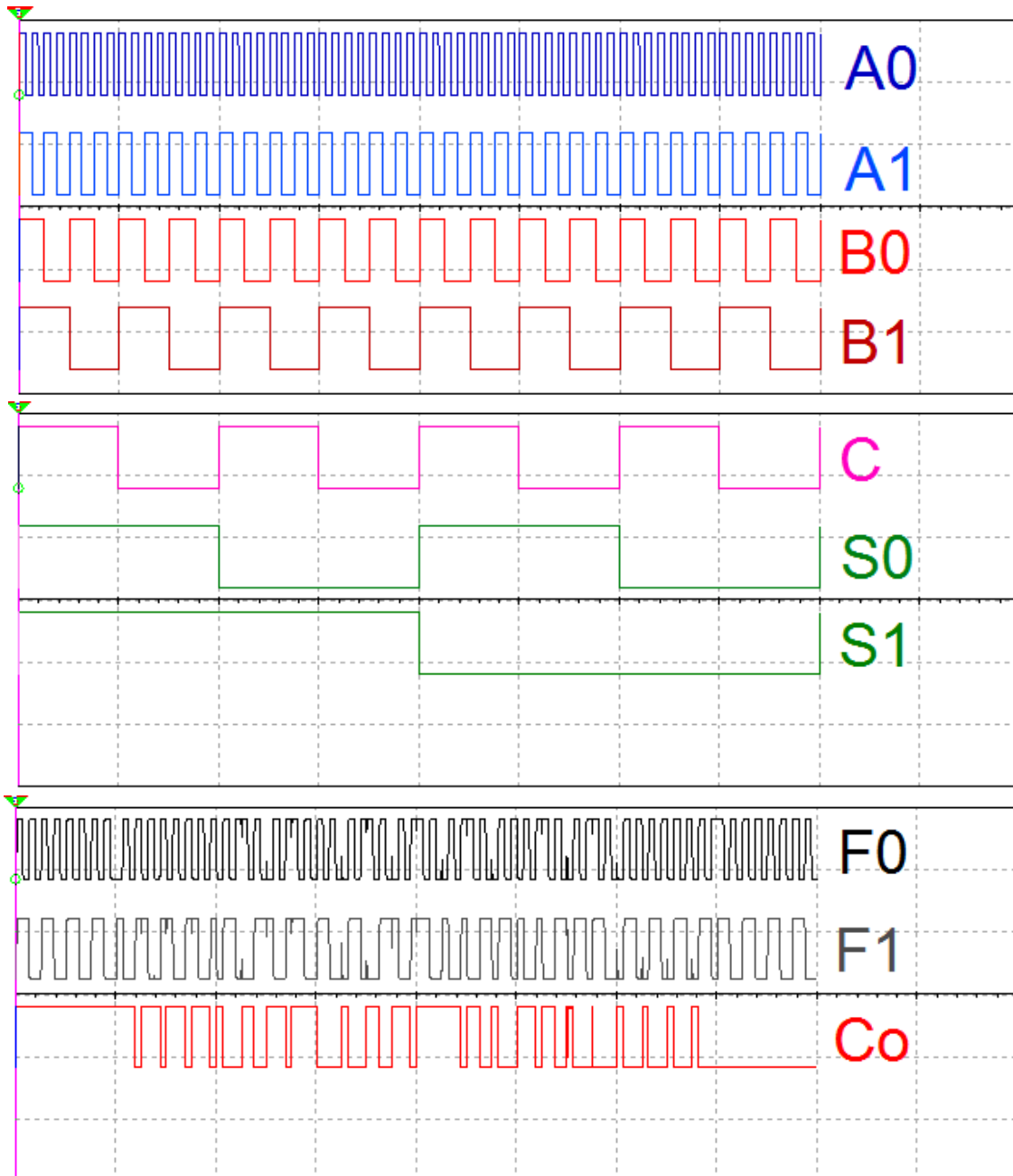


Figura 5.78: Respuesta de funciones aritméticas para un ALU 2 Bit

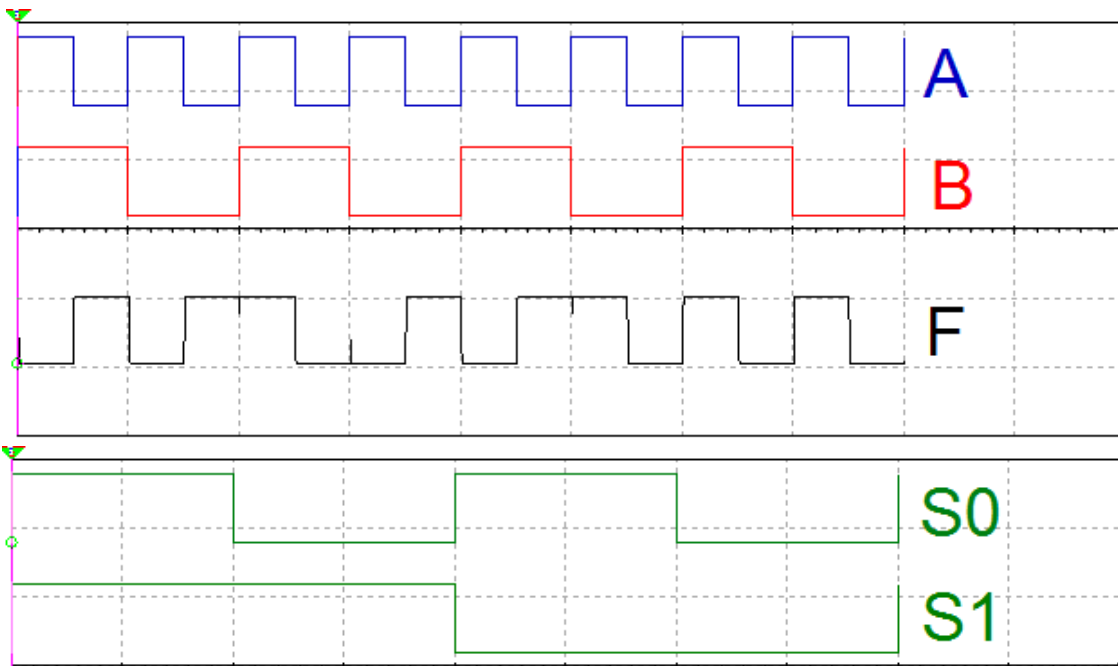


Figura 5.79: Respuesta de funciones lógicas para un ALU 1 Bit

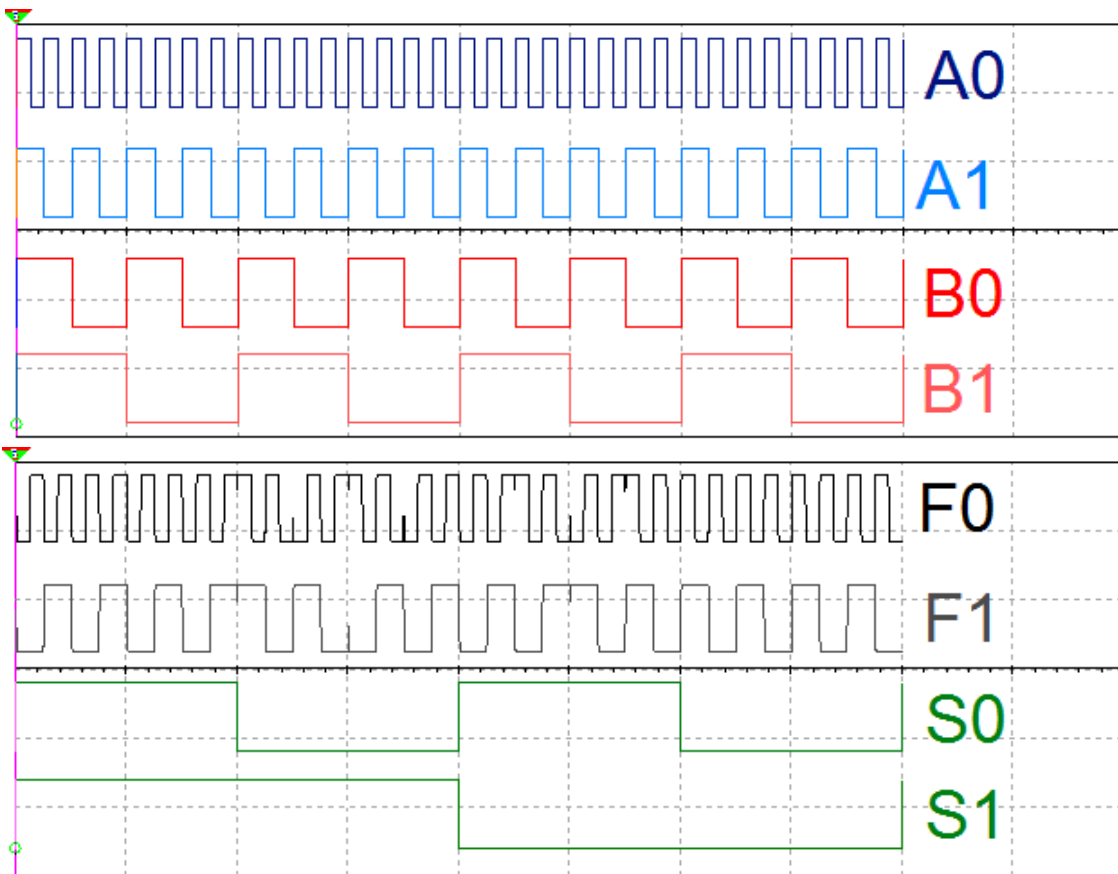


Figura 5.80: Respuesta de funciones lógicas para un ALU 2 Bit

Aplicaciones

6.1 Diseño de una neurona artificial usando circuitos reconfigurables

Desde los primeros años del siglo XX se han propuesto diversos modelos que han intentado emular el comportamiento del cerebro humano [35]. Aunque existe una gran cantidad de modelos propuestos, la mayoría se basa en el hecho de ver a las neuronas como procesos numéricos que involucran los estados de otros según sus interconexiones [36].

El primer modelo de neurona artificial propuesto por McCulloch y Pitts, presentaba un modelo de estructura donde su funcionamiento consideraba a las neuronas como dispositivos con n entradas con una única salida y sólo dos estados posibles: activa o inactiva. Por lo tanto, este primer modelo propuesto consiste de un dispositivo binario donde existe un umbral de operación fijo que hay que superar para que se realice un cambio de estado.

Para realizar dicha tarea se propone considerar el uso de una estructura reconfigurable. La estructura utilizada corresponde a una celda lógica reconfigurable [26].

6.1.1 Neurobiología

Una neurona típica posee el aspecto que se presenta en la figura 6.1, en este esquema se pueden observar las partes básicas que la conforman: Dendritas, Cuerpo celular, Núcleo y el Axón.

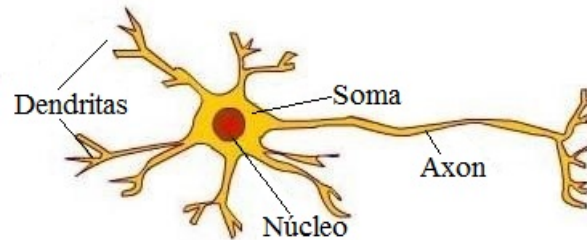


Figura 6.1: Partes de una neurona.

La mayoría de las neuronas codifican sus salidas como una serie de breves pulsos periódicos, llamados potenciales de acción, que se originan cercanos al soma de la célula y se propagan a través del axón. Luego este pulso llega a las sinapsis y de ahí a las dendritas de la neurona siguiente.

Una sinapsis es una interconexión entre dos neuronas, en ella, el tipo más común es la sinapsis química, donde una señal neural eléctrica pre-sináptica, llega al botón sináptico, figura 6.2; allí esta hace que las vesículas sinápticas (en color azul) se rompan, liberándose así una sustancia llamada neurotransmisor. Esta sustancia se difunde a través del espacio entre las neuronas. Luego, es captada por la dendrita, en donde estimula la emisión de un nuevo impulso eléctrico. Así vemos que las dendritas son las zonas receptoras de una neurona y por su parte el axón una línea de transmisión que llega a los botones, los cuales son como terminales que comunican los impulsos a otras neuronas.

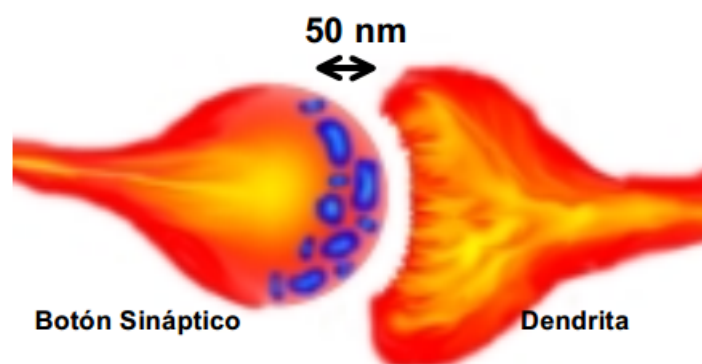


Figura 6.2: Unión entre neuronas, Sinapsis.

Dos comportamientos que destacan en este modelo son:

- El impulso que llega a una sinapsis y el que sale de ella no son iguales en general ya que una sinapsis modifica el pulso, ya sea reforzándolo o debilitándolo.
- En el soma se suman las entradas de todas las dendritas. Si las entradas sobrepasan un cierto umbral, entonces se envía el pulso a lo largo del axón, en caso contrario no transmitirá.

Considerando estas características se propone el siguiente modelo neuronal.

6.1.2 Modelo Perceptrón

Cabe mencionar que solo se desea introducir un modelo sencillo de la neurona, esto con el fin de tener una primera aproximación en un circuito reconfigurable. No se pretende modelar exactamente el comportamiento fisiológico de la neurona sino solo sus características más generales.

El primer modelo de red neuronal artificial fue desarrollado por el científico Rosenblatt en 1950, este modelo tomo mucho revuelo en los años 60 por su capacidad de aprendizaje y reconocimiento de patrones sencillos. Un perceptrón formado por varias neuronas es capaz de determinar cuándo una entrada activa o desactiva un potencial de acción a la salida del sistema, se observa como a cada una de las entradas se les asigna un respectivo peso, observe como la única neurona de salida realiza la suma ponderada de las entradas y pasa el resultado a una función de transferencia de tipo escalón. La regla de decisión es responder +1 si el patrón presentado pertenece a la clase A y con -1 si el patrón pertenece a la clase B.

En el diagrama de la figura 6.3 se presenta el modelo teórico de una neurona artificial. En principio se aplica un conjunto de entradas a la neurona, donde cada entrada se multiplica por un "peso" o ponderación (a_n) análogicamente al grado de conexión de la sinapsis.

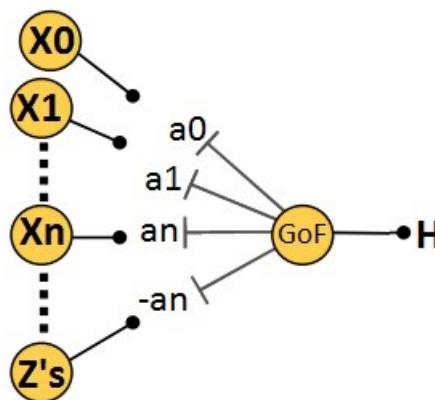


Figura 6.3: Esquema de neurona.

Posteriormente todas las entradas ponderadas se suman y se determina el nivel de excitación o activación de la neurona. Es entonces cuando la neurona se activa si la entrada total supera un cierto umbral, esta función de activación se define utilizando un arreglo de amplificadores operacionales.

En el modelo realizado por McCulloch las neuronas son de tipo binario 0,1, tienen una respuesta como la que se muestra en la figura 6.4, los umbrales y las sinapsis se mantienen fijas y la función de activación es del tipo escalón. En este modelo se demostró que varias funciones lógicas se pueden describir mediante la combinación de varias neuronas, lo que en principio daría pie a la teoría de que cualquier función lógica podría ser obtenida utilizando una red de neuronas artificiales.

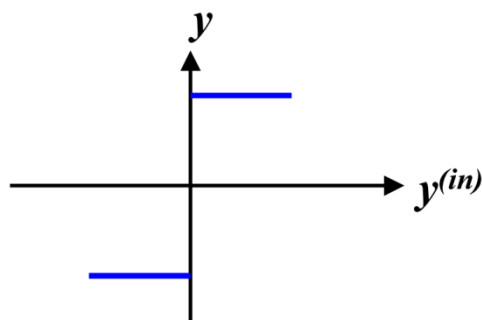


Figura 6.4: Función de activación tipo escalón.

Este modelo sólo es capaz de discriminar patrones que sean linealmente separables. En el caso de neuronas de dos entradas, esta separación se lleva a cabo mediante una línea recta, tal como puede apreciarse en la figura 6.5. Para el caso de tres entradas, la separación se realiza mediante un plano en el espacio tridimensional, y así sucesivamente hasta el caso de N entradas, en el cual el espacio N -dimensional es dividido en un hiperplano.

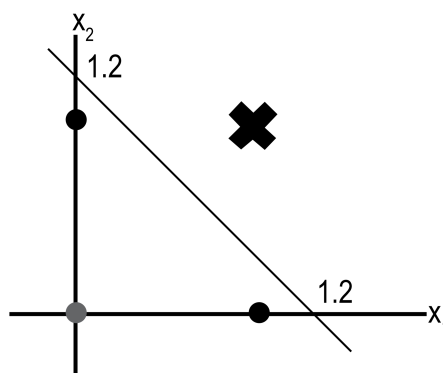


Figura 6.5: Plano para función de 2 entradas (OR).

Para el caso específico de la función XOR, se debe considerar una red de perceptrones, ya que este problema no es linealmente separable. En la figura 6.6 se presenta la solución a esta situación.

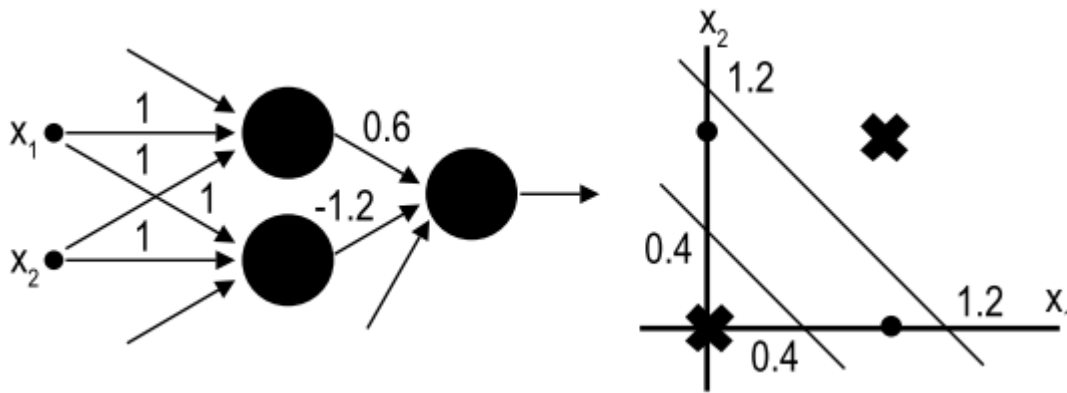


Figura 6.6: Solución con red de perceptrones para el caso XOR.

6.1.3 El aprendizaje en las redes neuronales artificiales

El problema principal al trabajar con redes neuronales artificiales es el determinar los pesos en las conexiones para que la red realice una función específica. En algunos casos es posible encontrar algunos problemas en donde los pesos se tienen de manera casi intuitiva, sin embargo, existen casos específicos donde es necesario enseñar a la red a ejecutar los cálculos por ajustes iterativos de los pesos w_{ij} [37].

Existe un método de aprendizaje por corrección de error, el cual consiste en presentar al sistema un conjunto de datos, comparando la entrada y la salida deseada. El objetivo de este conjunto de entrenamiento consiste en minimizar el error entre la salida deseada y la salida actual, en la figura 6.7 se muestra un diagrama de este proceso [37].

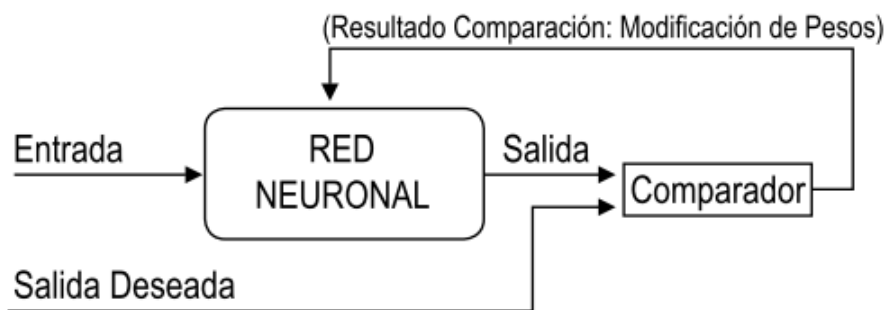


Figura 6.7: Aprendizaje por corrección de error.

En este algoritmo los pesos se ajustan en función de la diferencia entre los valores deseados y los obtenidos en la salida.

$$\Delta w_{ij} = \alpha x_i (d_j - x_j) \quad (6.1)$$

donde:

- Δw_{ij} .- Variación en el peso de la conexión entre el i -ésimo nodo y el j -ésimo.
- α .- Umbral en el aprendizaje que regula la velocidad y precisión ($0 < \alpha \leq 1$).
- x_i .- Salida del i -ésimo nodo.
- x_j .- Salida del j -ésimo nodo.
- d_j .- Valor de salida deseado del j -ésima unidad de procesamiento.

6.1.4 Metodología de diseño

Primero considere que el sistema esta dado por la ecuación 3.19, que repetimos a continuación.

$$F(X) = A^T X + B \quad (6.2)$$

Donde

$$A^T = [a_0, a_1, \dots, a_{N-1}] \quad (6.3)$$

$$X = \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ \dots \\ x_{N-1} \end{bmatrix} \quad (6.4)$$

Con $A^T, X \in R^n$ y $B \in R$, B es un parámetro para sintonizar, A corresponde a los pesos que se agregan a cada entrada mediante la sinapsis y X es un vector de N entradas, las cuales tienen su analogía directa con las dendritas de la neurona; los valores que puede tomar x son binarios, es decir $x_i \in \{0, 1\}$; y la función que describe la activación esta dada por la ecuación 3.22.

$$g(z) = \begin{cases} 1, & z \in (m_{inf}, m_{sup}); \\ 0, & \text{de otra forma.} \end{cases} \quad (6.5)$$

En general las ecuaciones que modelan a esta neurona artificial propuesta son las ecuaciones 3.23 y 3.24 presentadas para una *Celda Lógica Reconfigurable* de la sección 3. Por lo que el circuito correspondiente sería el presentado en la figura 5.1 de la sección 5.

6.2 Diseño e implementación para un convertidor multinivel

Los convertidores multinivel están conformados por arreglos de semiconductores de potencia y capacitores que son utilizados como fuentes de voltaje, el voltaje generado de salida tiene una forma de onda escalonada, esto se debe a que los interruptores se cierran y abren en tiempos diferentes produciendo diferentes señales de switcheo; en un caso ideal, un convertidor con un número infinito de niveles permite obtener una onda sinusoidal [38] [39]. La figura 6.8 presenta un sistema de 11 niveles.

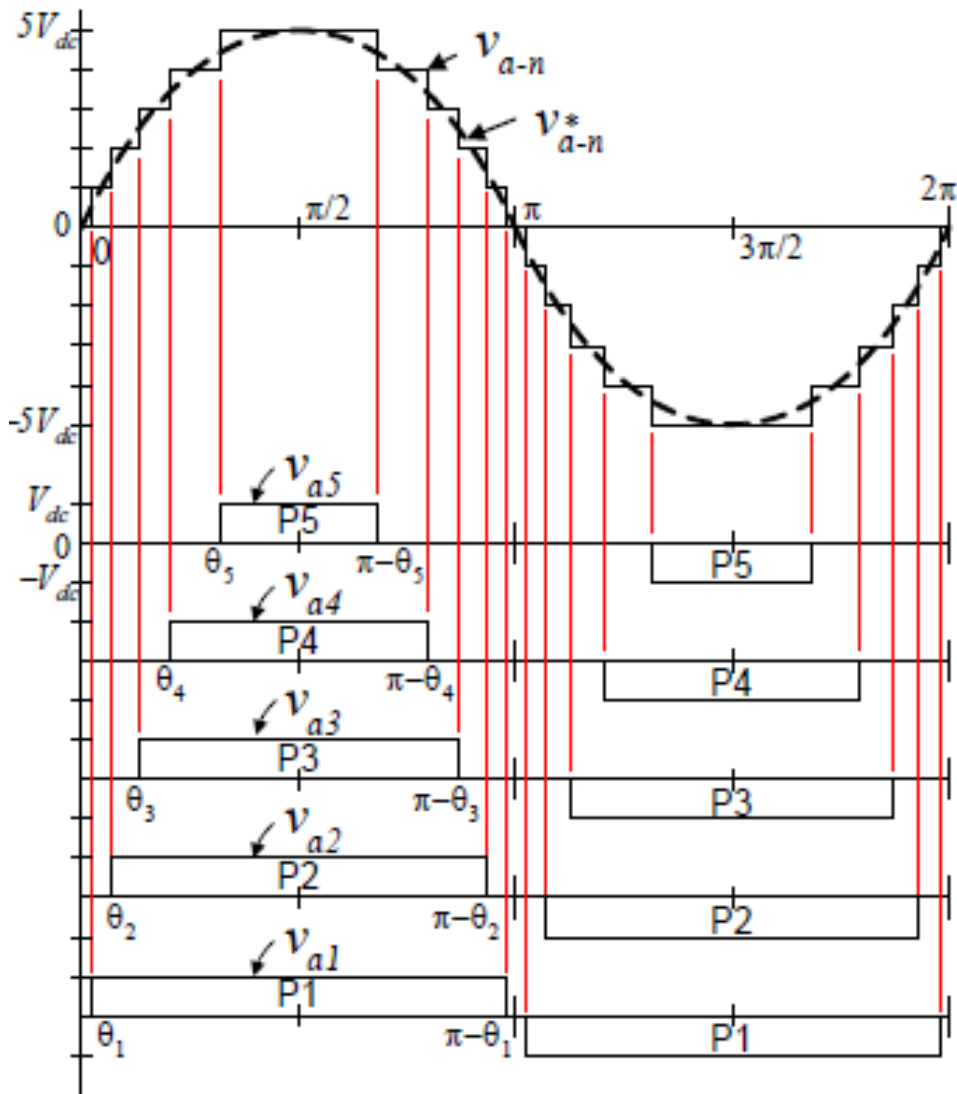


Figura 6.8: Señales de switcheo y comparación entre señal escalonada y señal sinusoidal.

En esta sección se presenta el desarrollo de un circuito de baja potencia usando el esquema de circuitos reconfigurables, tratando de emular dispositivos convertidores multinivel. Aplicando este dispositivo es posible la transformación y mejora de diferentes señales, que son equivalentes a las generadas por los convertidores de potencia multinivel convencionales. El circuito está compuesto por los esquemas utilizados en las secciones pasadas, este sistema permite conseguir una emulación de la típica forma de onda escalonada mostrada en dispositivos generadores de señales de varios niveles, los cuales son utilizados generalmente en sistemas de energía renovable.

Los convertidores son apropiados para la aplicación de alta potencia y alta tensión, debido a su capacidad de sintetizar las formas de onda con un mejor espectro armónico. La tecnología de los convertidores multinivel es muy atractiva para aplicaciones de voltajes de mediano y alto rango, como lo son sistemas de accionamiento de motores, distribución de energía, calidad de energía y aplicaciones de acondicionamiento de potencia.

En la figura 6.8 se presenta un diagrama significativo que presenta los elementos necesarios para realizar un convertidor multinivel. Como primer elemento se requiere el manejo de fuentes VCD, las cuales serán utilizadas para formar los niveles del convertidor, posteriormente se observa la etapa de topología de conversión multinivel; existen diferentes topologías pero las 3 más utilizadas son la de capacitor flotante en la figura 6.9, diodo anclado en la figura 6.10 y puentes H en la figura 6.11.

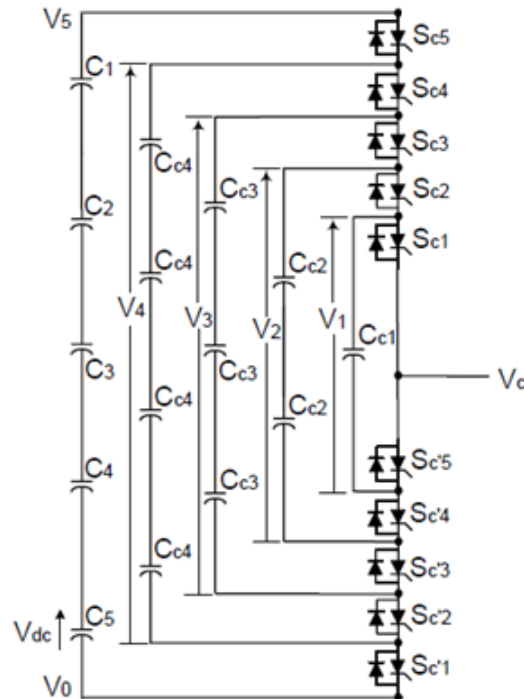


Figura 6.9: Topología de capacitor flotante.

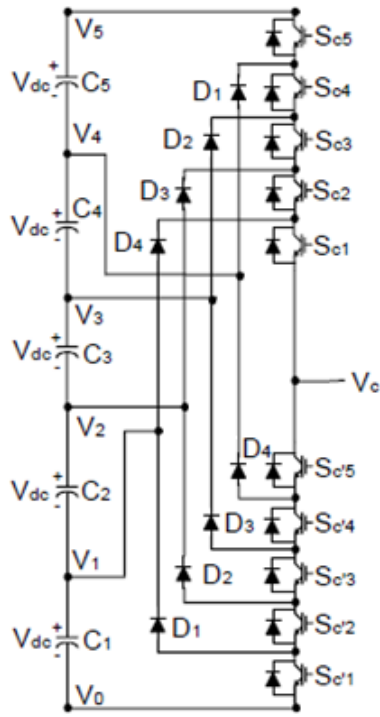


Figura 6.10: Topología de diodo anclado.

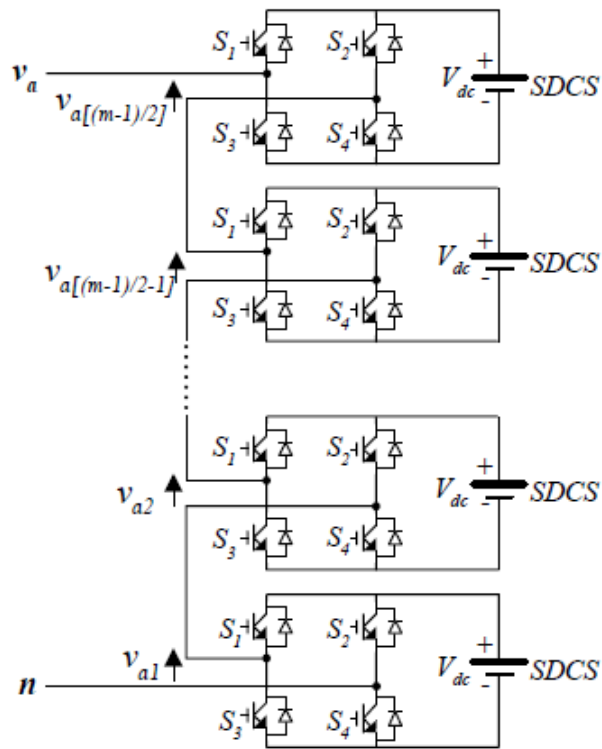


Figura 6.11: Topología de puente H.

Dichas topologías deben ser controladas por alguna técnica de conmutación, en la figura 6.12 se presenta un diagrama a bloques que incluye diferentes técnicas y remarcada la que se utiliza en este trabajo.

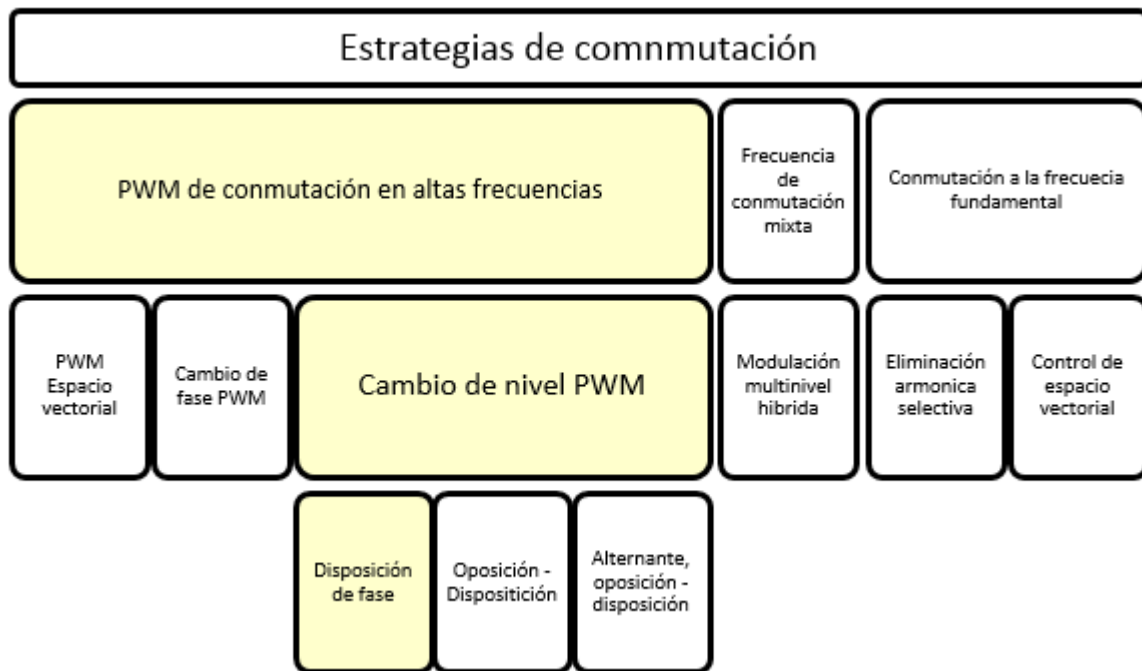


Figura 6.12: Técnicas de conmutación.

La modulación PWM por disposición de fase es una técnica de modulación manejando altas frecuencias, esta consiste en obtener diferentes señales de switcheo, las cuales se generan a través de la comparación de una señal moduladora con diferentes señales portadoras; posteriormente se selecciona una de las portadoras y se compara directamente con la señal moduladora, cuando la señal moduladora es mayor se genera una señal de switcheo PWM; haciendo el mismo proceso para todas las señales portadoras es como se obtienen múltiples señales PWM, al sumar todas estas señales es cuando se genera la forma escalonada, en las figuras 6.13, 6.14, 6.15 y 6.16 se representan las comparaciones que se realizan en esta técnica de modulación.

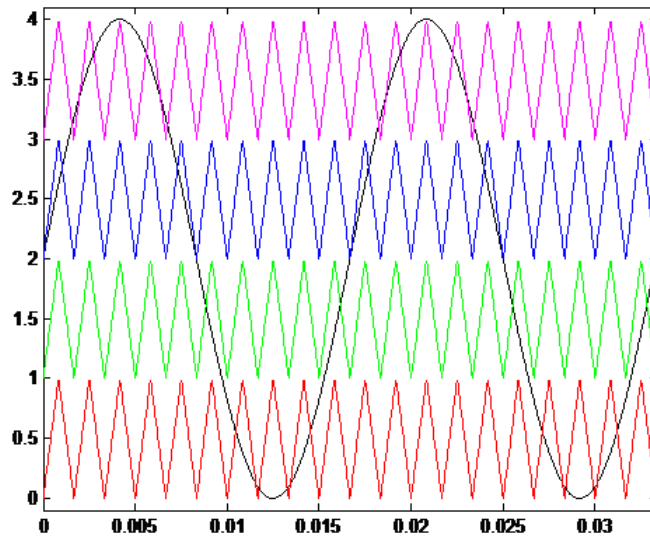


Figura 6.13: Señales portadoras (rosa, azul, verde, rojo) y señal moduladora (negro).

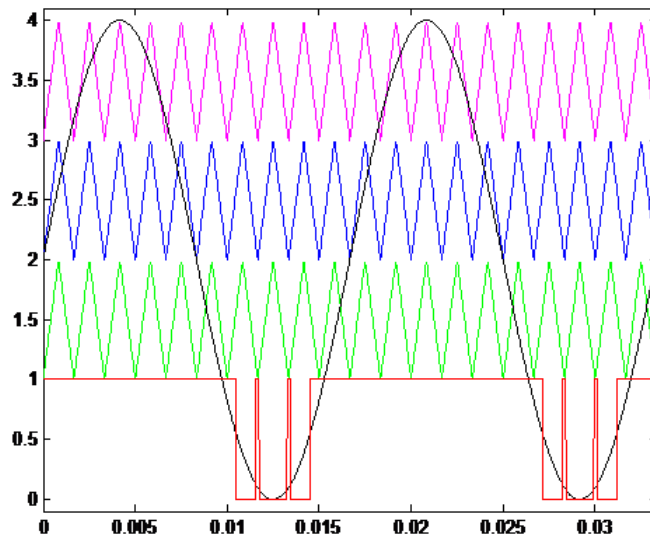


Figura 6.14: Señal de switcheo obtenida tras comparar la portadora de nivel más bajo (rojo).

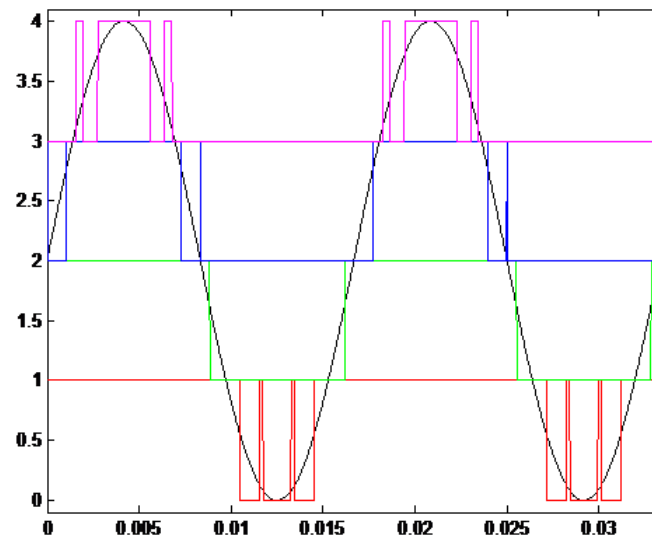


Figura 6.15: Señal de switcheo obtenida tras comparar cada portadora.

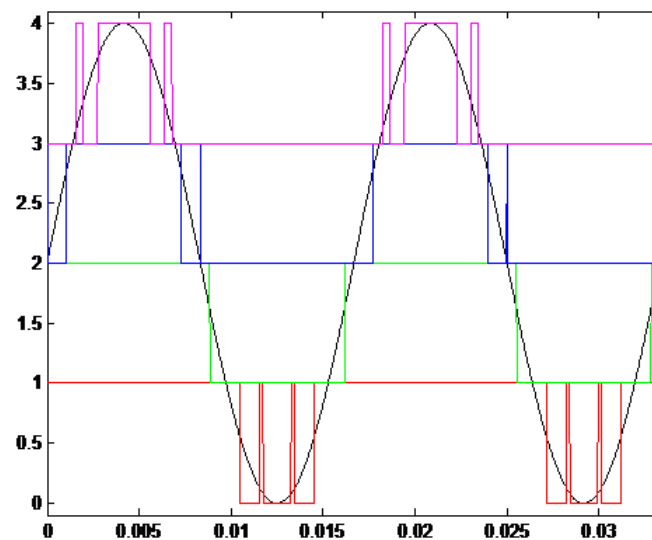


Figura 6.16: Sumatoria de todas las señales de switcheo (señal multinivel azul) y señal moduladora (negro).

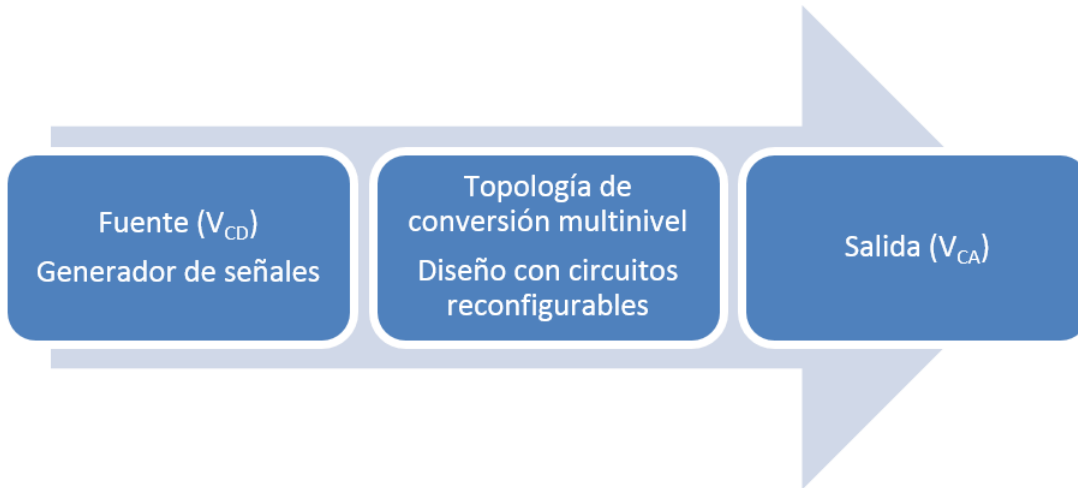
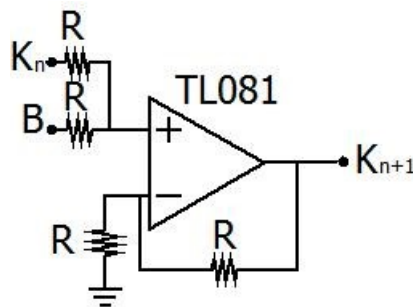


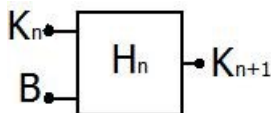
Figura 6.17: Diagrama del proceso de emulación para un convertidor multinivel.

En la figura 6.17 se presenta un diagrama bloques del procedimiento de emulación, en la primera sección aparecen fuentes de voltaje V_{CD} y un generador de señales, en la segunda parte se muestran la topología de conversión multinivel, para ello es necesario considerar las siguientes configuraciones para los circuitos reconfigurables.

Para conseguir las "n" portadoras se propone utilizar un amplificador en configuración sumador no inversor, como se muestra en la figura 6.18. Los parámetros de entrada son la portadora K_n (60 Hz) y un nivel de offset B , la salida se expresa como K_{n+1} y corresponde a una portadora montada sobre un voltaje de offset.



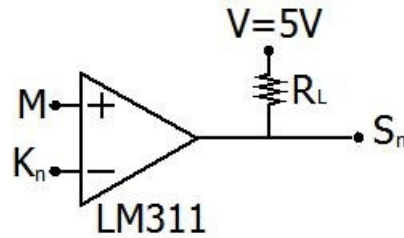
a)



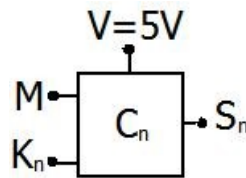
b)

Figura 6.18: a) Configuración sumador no inversor. b) Diagrama a bloque

El siguiente circuito permite realizar la comparación cada una de las señales portadoras con la señal moduladora; genera un valor de 5V cuando la portadora es mayor y 0V para el otro caso.



a)

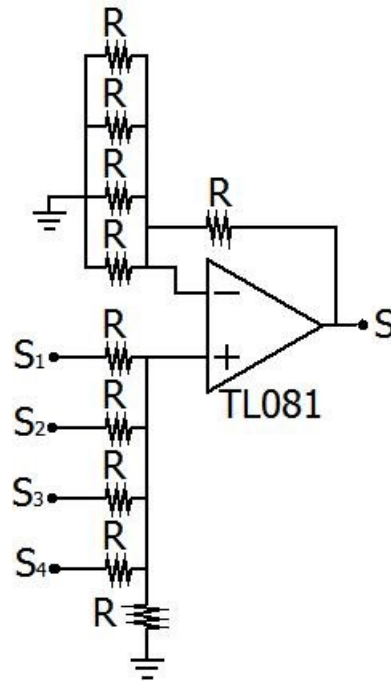


b)

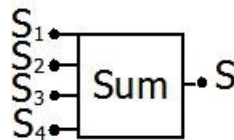
Figura 6.19: a) Configuración comparador. b) Diagrama a bloque

Posteriormente se suman las señales de switcheo mediante un circuito sumador no inversor.

Para los resultados físicos se consideraron como parámetros una señal moduladora $M = 60\text{Hz}$, Portadoras senoidal, triangular, cuadrada, K_n de 1.2KHz , también señales constante, ruido blanco y el estado Z del *oscilador caótico de Lorenz*, utilizando niveles de offset de 1, 2 y 3V, los posibles arreglos para obtener las señales portadoras se presentan en la figura 6.21.



a)



b)

Figura 6.20: a) Configuración sumador. b) Diagrama a bloque

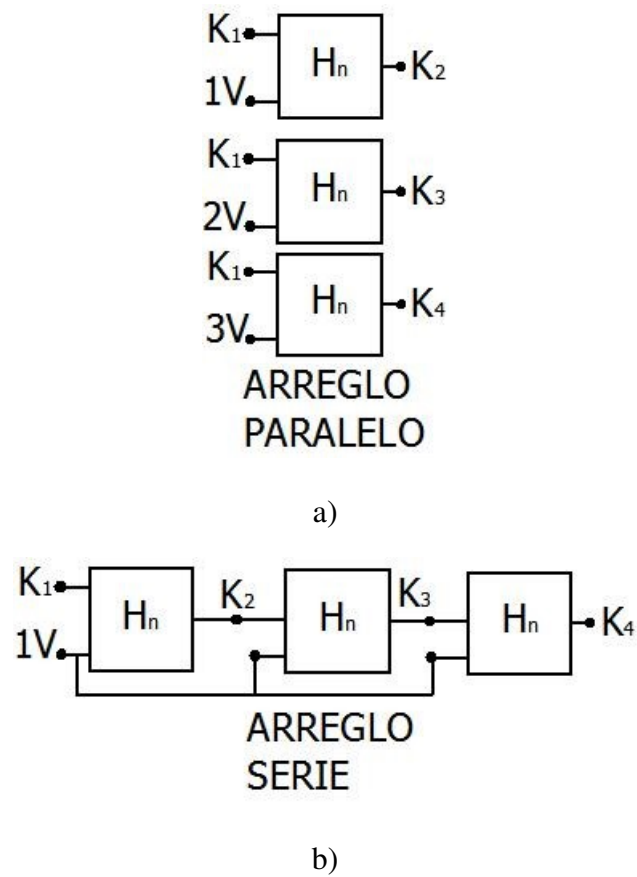


Figura 6.21: Tipos de arreglo para obtener señales portadoras.- a) Paralelo, b) Serie.

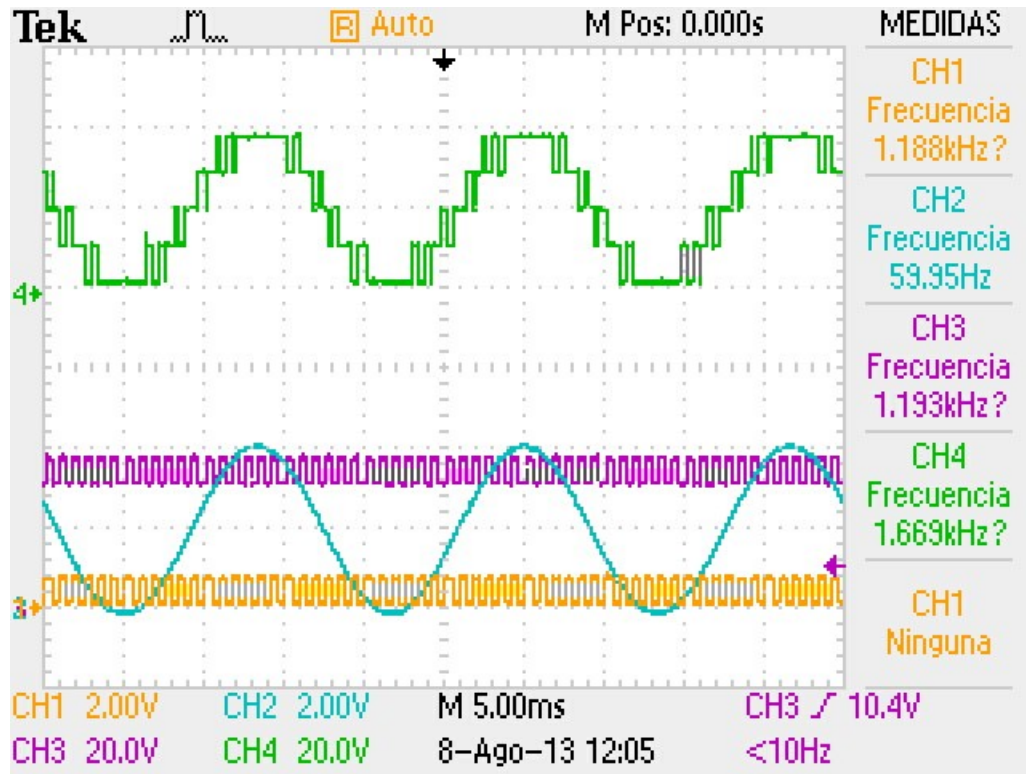


Figura 6.22: Resultado experimental usando señal cuadrada $K_n = 1.2KHz$.

Posteriormente se obtienen los espectros de Fourier de las señales generadas y se comparan los valores de Distorción Total Harmonica obtenidos con el analizador de harmonicos marca fluke modelo 43B

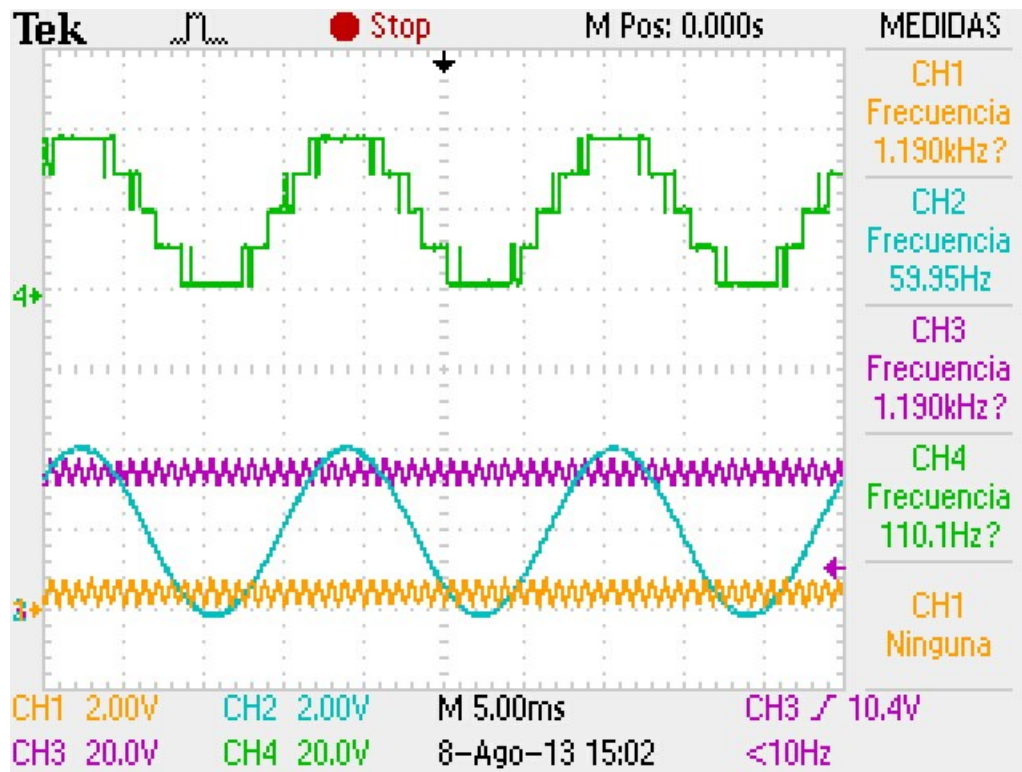


Figura 6.23: Resultado experimental usando señal triangular $K_n = 1.2\text{KHz}$.

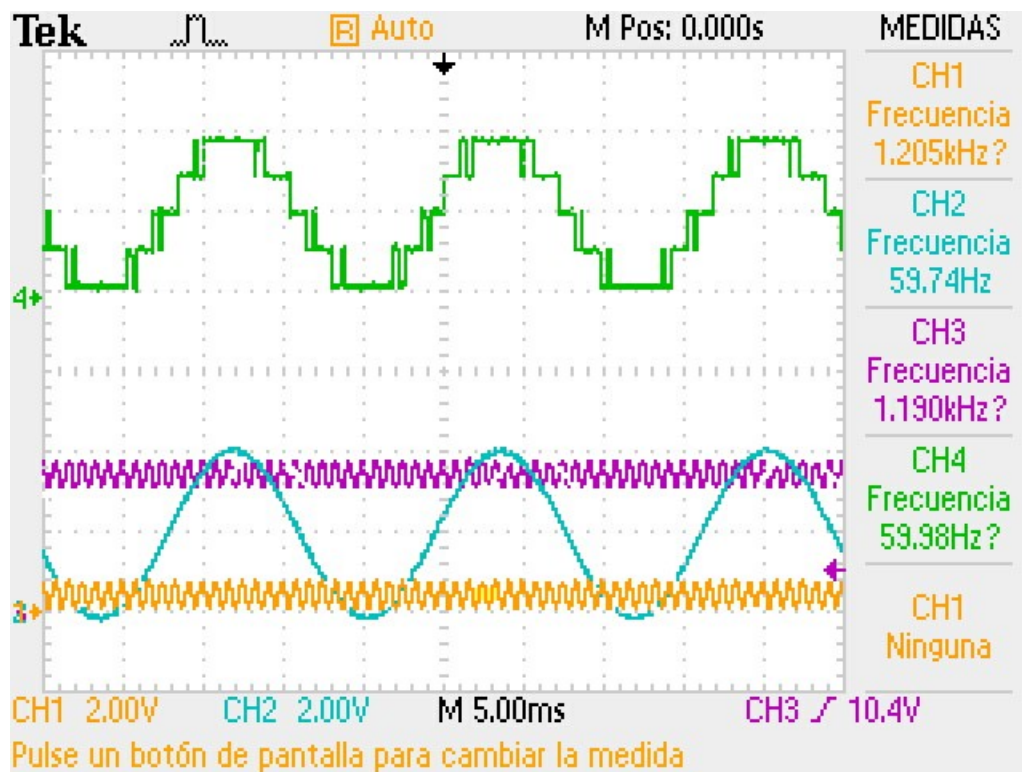


Figura 6.24: Resultado experimental usando señal senoidal $K_n = 5.5\text{KHz}$.

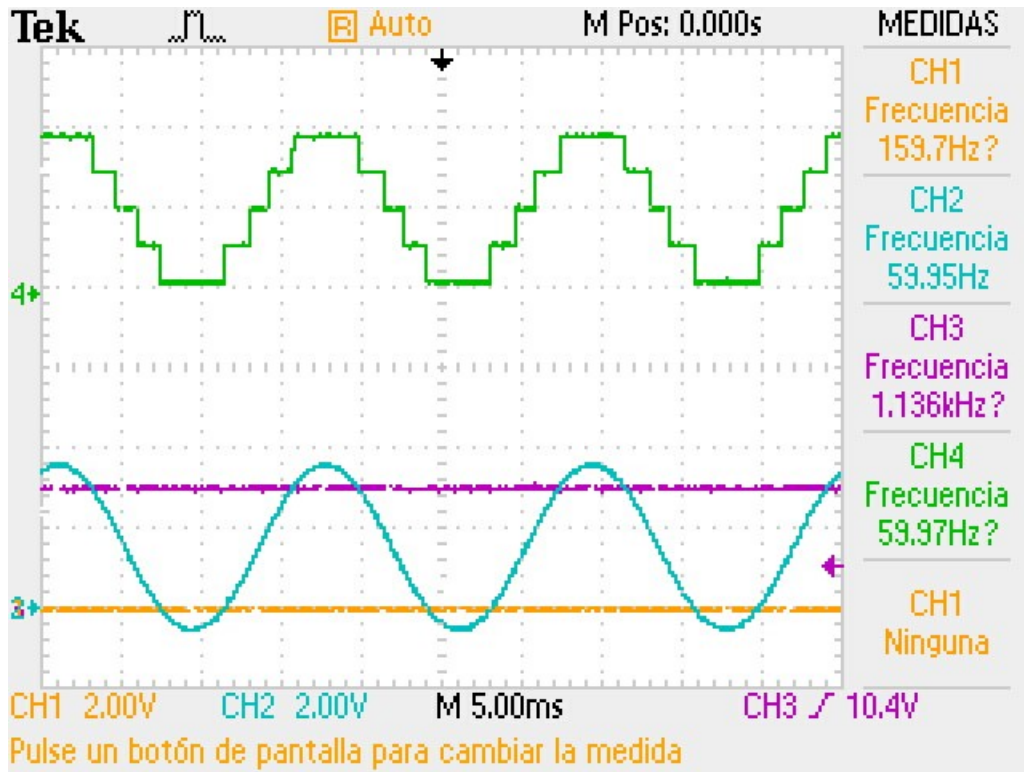


Figura 6.25: Resultado experimental usando señal constante.

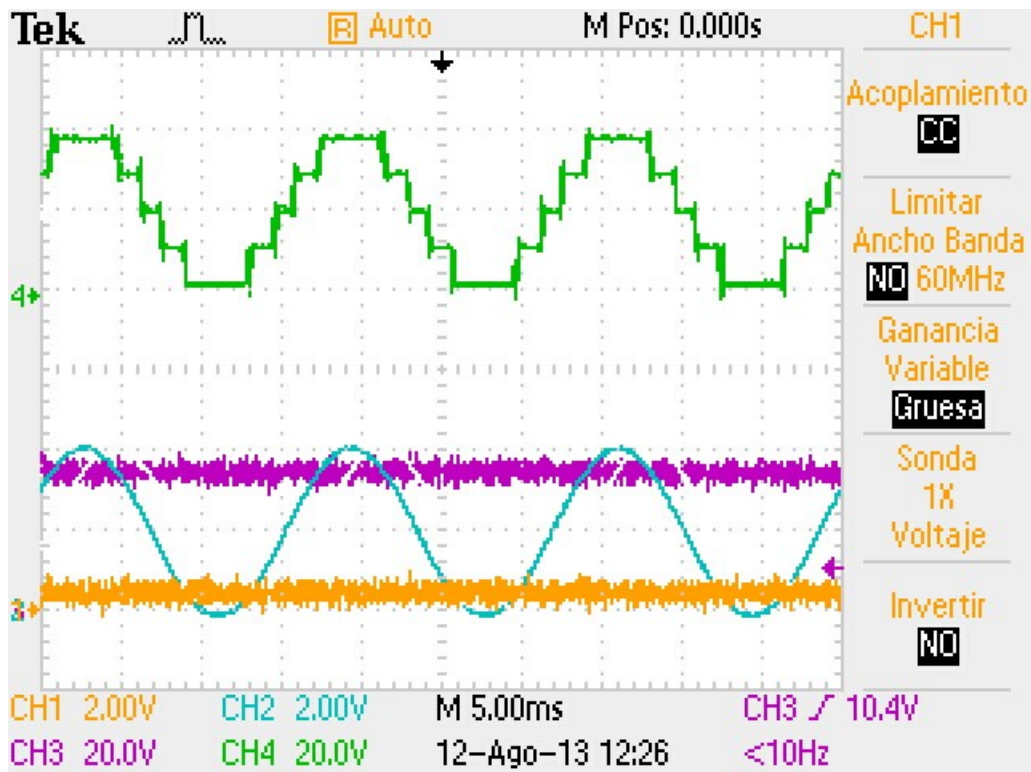


Figura 6.26: Resultado experimental usando señal de ruido blanco.

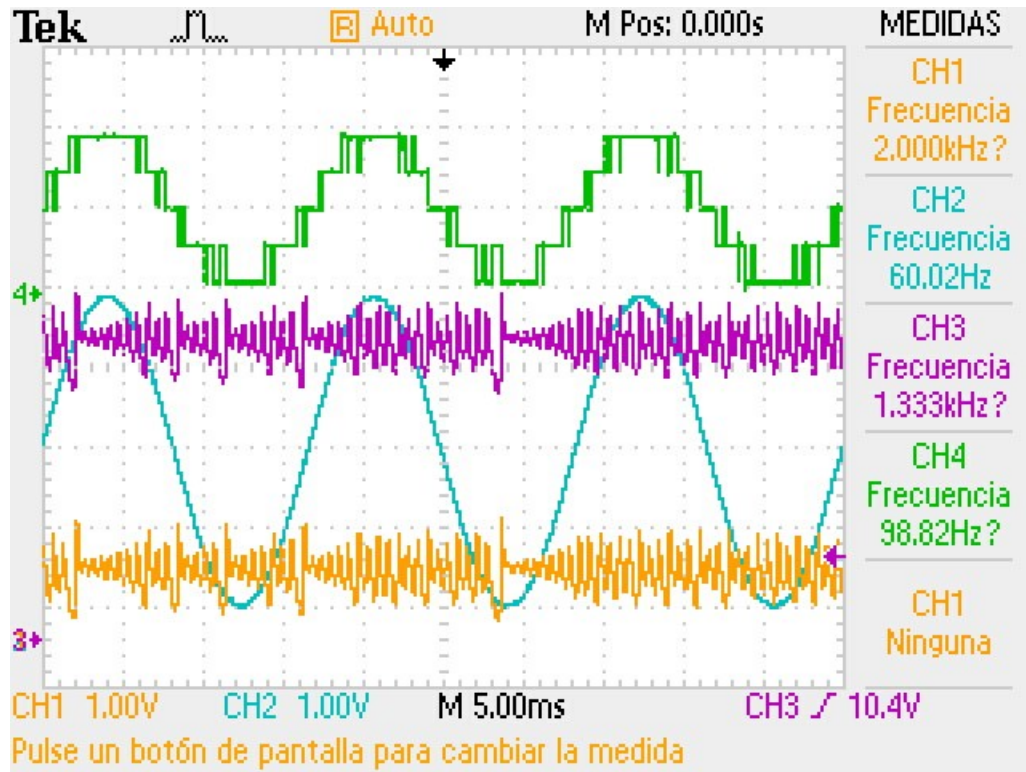


Figura 6.27: Resultado experimental usando estado Z del *oscilador de Lorenz*.

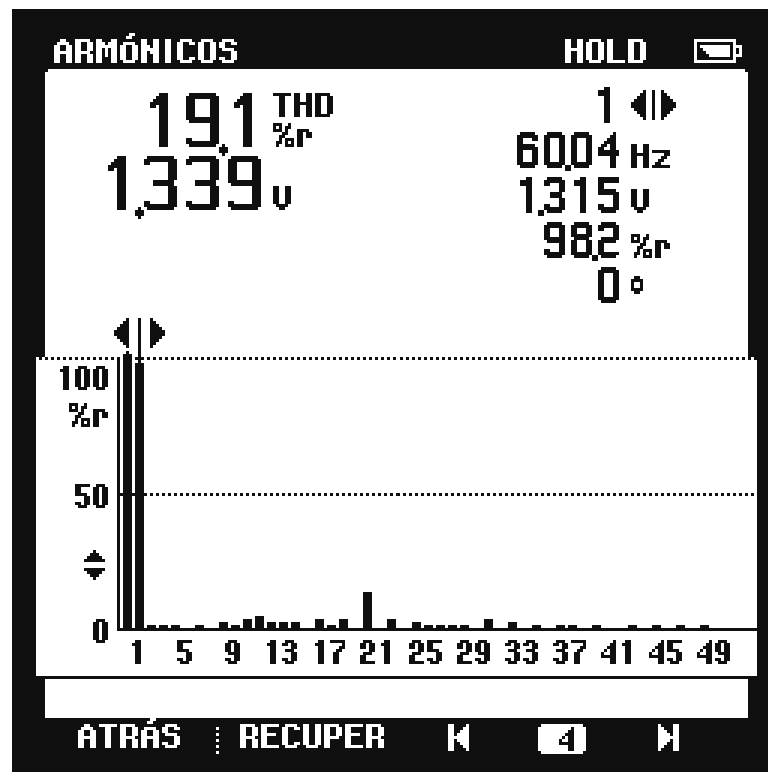


Figura 6.28: Resultado experimental usando señal cuadrada $K_n = 1.2\text{KHz}$.



Figura 6.29: Resultado experimental usando señal triangular $K_n = 1.2\text{KHz}$.



Figura 6.30: Resultado experimental usando señal senoidal $K_n = 1.2\text{KHz}$.



Figura 6.31: Resultado experimental usando señal constante.



Figura 6.32: Resultado experimental usando señal de ruido blanco.



Figura 6.33: Resultado experimental usando señal del estado Z del *oscilador de Lorenz*.



Conclusiones

Se desarrollo la posibilidad de construir funciones lógicas binarias, bajo el uso de sistemas basados en estructuras con comportamiento lineal por partes, utilizando diversas configuraciones de amplificadores operacionales.

Se definió el termino de celda lógica reconfigurable, refiriendose a un sistema con la capacidad de obtener diversas funciones lógicas como las compuertas *AND*, *OR*, *XOR* y sus respectivas negaciones.

Como casos específicos se generalizaron las compuertas lógicas *AND* y *OR* para los caso de "n" entradas.

Se estudiaron los casos de respuesta con incremento lineal y respuesta lineal a trozos, permitiendo así proponer el diseño las 16 funciones lógicas presentes en una compuerta de dos entradas.

Con el esquema de entrada-salida propuesto en el capítulo 4 y utilizando amplificadores operacionales fue posible instrumentar las diferentes funciones posibles, para una compuerta lógica con dos entradas. El prototipo físico se muestra en el capítulo 5.

Es necesario mencionar que gracias a que este sistema esta basado en el uso de amplificadores operacionales y resistencias, su diseño permite que el sistema sea robusto ante condiciones iniciales, además este tipo de estructura electrónica garantiza una escalabilidad tanto en el número de entradas y salidas.

Aprovechando esta propiedad, se obtuvieron esquemas de sistemas basados en circuitos lógicos combinacionales como se presento en el capítulo 5, donde se presentaron tanto el diseño como la implementación para los circuitos *medio sumador - medio restador*, *decodificador*, *codificador*, *comparador*, *multiplexor*, *demultiplexor*, *latches* y una *ALU* de 1 y 2 bits de operación.

Este modelo de ALU utilizando celdas lógicas reconfigurables, permite el manejo de las operaciones aritméticas básicas, además la implementación de este circuito no limita su funcionamiento solo al procesamiento de señales digitales, ya que una de sus mayores virtudes radica en el manejo de señales continuas.

Cabe hacer mención que los circuitos presentados están basados en una celda reconfigurable, la cual es un sistema que se puede configurar a través de dos vías, una mediante la reconexión de los bloques presentados y la otra mediante la sintonización de los parámetros a_i , m_+ y m_- . La libertad de asignar estos parámetros se ve reflejada directamente en la posibilidad de obtener funciones específicas, es decir existe la posibilidad de generar una función arbitraria en base a sus maxitérminos, es decir, el usuario podría asignar que combinaciones de entrada generen una respuesta en alto con sólo configurar los parámetros de m_+ y m_- , posteriormente se suman las salidas de los bloques M_2 en el bloque M_3 , esto equivale a seleccionar los maxitérminos de una función, e incluso resulta posible el obtener distintas compuertas lógicas por cada bloque M_2 , de esta forma una misma celda lógica reconfigurable permite comprender múltiples circuitos de aplicación específica (ASIC) ello permite incrementar considerablemente el número de posibles aplicaciones en el diseño de circuitos lógicos combinatoriales e incluso algunos sistemas que trabajen con señales continuas.

Los resultados presentados se obtuvieron con el uso de componentes electrónicas comerciales, ya que el objetivo de esta investigación radica en sentar las bases de diseño de funciones lógicas a través de una nueva técnica la cual mantiene un núcleo reconfigurable analógico, si bien no se pretende suplir los diferentes esquemas usados hoy en día, si se pretende realizar una aportación en el área de la electrónica al tener un complemento que sea reconfigurable por dos vías (reconexión y sintonización paramétrica) como se presenta en el capítulo 5.

En general el sistema se describe como una unidad de configuración analógica la cual pretende desempeñar ciertas funciones propias de un SAD, pero manteniendo en su núcleo la posibilidad de reconfigurar su comportamiento.

En el capítulo 6, se presentó inicialmente una analogía de este sistema con el modelo del perceptrón, el cual corresponde a uno de los primeros modelos de una neurona artificial, dando pie a poder continuar con la parte de investigación desarrollando el estudio de redes neuronales artificiales basadas en celdas lógicas reconfigurables.

Finalmente se presentan los resultados de la implementación de un convertidor multinivel de cinco niveles. El sistema realizado es de bajo costo y permite fácilmente el incremento de niveles.

Los resultados obtenidos son semejantes a los obtenidos usando topologías de mediana y alta potencia. Además el diseño resulta práctico y permite la implementación de distintas técnicas de conmutación.

Dado que el sistema puede operar con señales continuas, se abre la posibilidad de estudiar más adelante la implementación de circuitos electrónicos que no operen bajo un dominio totalmente binario, es decir poder realizar circuitos que puedan adaptar modelos de otras bases numéricas, en concreto circuitos ternarios.

Uno de los trabajos a futuro de esta investigación se enfoca a obtener un circuito equivalente mediante el uso de transistores, de esta forma se diseñaría un circuito a nivel semiconductor pero con una estructura dinámica, además de obtener un circuito reconfigurable de menor tamaño.

Bibliografía

- [1] Ernest E. Doebelin, *Sistemas de medición e instrumentación*, 2005 McGraw Hill. ISBN 13:9789701049778.
- [2] Jorge Raúl Villaseñor Gómez, Fredy Alberto Hernández Aguirre, *Circuitos eléctricos y aplicaciones digitales*, 2013 Pearson. ISBN 978-607-32-1515-2.
- [3] M. M. Mano, *Computer System Architecture*, Prentice-Hall, Englewood Cliffs, NJ, USA, 3rd edition, 1993.
- [4] Ronald Tocci, *Sistemas Digitales*, 1993 Prentice-Hall.
- [5] T. C. Bartee, *Computer Architecture and Logic Design*, McGraw-Hill, New York, NY, USA, 1991.
- [6] Steven H. Strogatz, *Nonlinear Dynamics and Chaos*, 2001 Westview Press. ISBN 9780738204536.
- [7] Mark D Hahm, Eby G. Friendman and Edward L. Titlebaum, *Analog vs digital: A comparison of circuit implementations for low-power matched filters*, 1996, IEEE International Symposium, Conference: Circuits and Systems.
- [8] Fan, Y., and A. O. Verview. *Signal Processing, Communications and Computing (ICSPCC)*. IEEE International Conference on. 2011.
- [9] Lan. K. R., S. H. Gopisetty and R. Dharmarajan. *Custom Integrated Circuits Conference*. Proceedings of the IEEE (1993): 3.
- [10] K. Compton, S. Hauck, *An Introduction to Reconfigurable Computing*, 1999 Northwestern University, Dept. of ECE Technical Report.
- [11] Mangione-Smith, W.H.; Hutchings, B.; Andrews, D.; DeHon, A.; Ebeling, C.; Hartenstein, R.; Mencer, O.; Morris, J.; Palem, K.; Prasanna, V.K.; Spaanenburg, H.A.E., *Seeking solutions in configurable computing*, 1997 IEEE, Computer, vol. 30, No.12, pp. 38-43.
- [12] Peng, H., Hu, G., Li, L., Yang, Y., & Xiao, J. *Constructing dynamic multiple-input multiple-output logic gates*. Mathematical Problems in Engineering, 2011.
- [13] D. Kuo, *Chaos and its computing paradigm*, IEEE Potentials, vol. 24, no. 2, pp. 13-15, 2005

- [14] D. Tabak, *Dynamic architecture and LSI modular computer systems*, IEEE Micro, vol. 4, no. 2, pp. 48-66, 1984.
- [15] Vemuri, R.R.; Harr, R.E., *Configurable computing: technology and applications*, 2000 IEEE, Computer, vol. 33, No.4, pp. 39-40.
- [16] S. Sinha and W. L. Ditto, *Dynamics based computation*, Physical Review Letters, vol. 81, no. 10, pp. 2156-2159, 1998.
- [17] S. Sinha, W. Ditto, Phys. Rev. E 60 (1999) 363.
- [18] D. Graham-Rowe, *Logic from chaos: new chips use chaos to produce potentially faster, more robust computing*, Technology Review, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2006.
- [19] T. Munakata, S. Sinha, and W. L. Ditto, *Chaos computing: implementation of fundamental logical gates by chaotic elements*, IEEE Transactions on Circuits and Systems. I: Fundamental Theory and Applications, vol. 49, no. 11, pp. 1629-1633, 2002.
- [20] K. Murali, S. Sinha, W. L. Ditto, and A. R. Bulsara, *Reliable logic circuit elements that exploit nonlinearity in the presence of a noise floor*, Physical Review Letters, vol. 102, no. 10, p. 104101, 2009.
- [21] I. Campos-Canton, E. Campos-Cantón, J.A. Pecina-Sanchez y H.C. Rosu, Int. J. Bif. and Chaos 20 (2010) 2547; to appear.
- [22] I. Campos-Cantón, S. Tesheira, C. Jiménez-Calvillo y E. Campos-Cantón, Rev. Mex. F'ys. 54 (2008) 468.
- [23] Luis Alfonso González Bravo *Funciones lógicas reconfigurables con mapeos en una dimensión*. Tesis para obtener el grado de Maestro en Ciencias Aplicadas, UASLP, 2011.
- [24] Jean-Pierre Aubin, *Neural Networks and Qualitative Physics: A Viability Approach*, Cambridge University Press, 1996.
- [25] Stanley I. Grossman, *Álgebra lineal*, McGrawHill, Sexta edición. 2156-2159, 1998.
- [26] I. Campos Cantón, R.E. Lozoy Ponce, y R.O. Lozoya Ponce, Rev. Mex. Fys. 59 (2013) 359.
- [27] FPGA's 101, Gina R., ISBN 978-1-85617-706-1 (2010).
- [28] Joel Uriel Cisneros Parra, El amplificador operacional. ISBN-970-93886-6-5, 2006.
- [29] M. Morris Mano, Michael D. Ciletti, *Diseño digital*, Pearson, Quinta edición, 2013.
- [30] Y. Fan and A. O. Verview, *Signal Processing, Communications and Computing (ICSPCC)*, 2011 IEEE International Conference on, (2011) 1.
- [31] K. R. Lan, S.H.; Gopisetty, R.; Dharmarajan, *Custom Integrated Circuits Conference*, 1993., Proceedings of the IEEE 1993, (1993) 3.

-
- [32] G. Tempesti, D. Mange, and A. Stauffer. *A robust multiplexer-based FPGA inspired by biological systems*. Journal of Systems Architecture 43.10 (1997): 719-733.
- [33] R. H. Olsson, D. L. Buhl, A. M. Sirota, G. Buzsaki, and K. D. Wise, *Biomedical Engineering, IEEE Transactions on*, **52**, (2005) 1303.
- [34] William Stallings, *Computer organization and architecture, designing for performance* Pretince Hall, Eighth edition, 2010.
- [35] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biology, 5 (4): 115-133, 1943.
- [36] Howard Demuth, Mark Beale, Martin Hagan, Orlando Jesús *Neural Network Design*, Martin Hagan, 2014.
- [37] Dowla F, Rogers L. Solving problems in environmental engineering and geosciences with artificial neural networks. Cambridge. MIT Press. 504p. 1996.
- [38] Geethalakshmi, B and DelhiBabu, K, An advanced modulation technique for the cascaded multilevel inverter used as a shunt active power filter, Power Electronics (IICPE), 2010 India International Conference on, pag=1-6, (2011),IEEE.
- [39] Tolbert, Leon M and Chiasson, John and McKenzie, Keith and Du, Zhong, Elimination of harmonics in a multilevel converter with nonequal DC sources, Applied Power Electronics Conference and Exposition, 2003.
- [40] K. Chlouverakis and M. J. Adams, *Optoelectronic realization of NOR logic gate using chaotic two section lasers*, Electronics Letters, vol. 41, no. 6, pp. 359-360, 2005.
- [41] K. Murali, A. Miliotis, W.L. Ditto y S. Sinha, Phys. Lett. A 373 (2009) 1346.

