

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Sumador Carry Look-Ahead basado Dual Mode Pass Logic

Cristhopher Rafael Mosquera Montero

Electrónica y Automatización

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero en Electrónica

Quito, 24 de diciembre de 2021

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN DE TRABAJO DE FIN DE CARRERA

Sumador Carry Look-Ahead basado en Dual Mode Pass Logic

Cristhopher Rafael Mosquera Montero

Nombre del profesor, Título académico

Edison Ramiro Taco Lasso, PhD.

Nombre del profesor, Título académico

Luis Miguel Prócel Moya, PhD.

Quito, 24 de diciembre de 2021

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en la Ley Orgánica de Educación Superior del Ecuador.

Nombres y apellidos: Cristhopher Rafael Mosquera Montero

Código: 00204095

Cédula de identidad: 0954223046

Lugar y fecha: Quito, 24 de diciembre de 2021

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en <http://bit.ly/COPETHeses>.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on <http://bit.ly/COPETHeses>.

RESUMEN

En el presente trabajo se diseñan tres sumadores *Carry Look Ahead* de 32 bit en el nodo tecnológico de 65 nm con la librería de TSMC. Se realizan un sumador CMOS, un *Dual Mode Logic* y una combinación entre *Dual Mode Logic* y *Dual Mode Pass Logic* (DMPL). Las dos últimas son familias que admiten dos modos de operación. Un modo de operación dinámico, que dirigido por una señal de reloj logra un menor tiempo de retardo que CMOS y otro modo estático, que tiene menor consumo energético que CMOS. Para estudiar y analizar el comportamiento de la familia DMPL se realiza una explicación detallada de la misma, indicando tanto sus ventajas como desventajas, en especial sus limitaciones de uso para no hacer un circuito completamente DMPL. Para la comparación entre las otras familias lógicas se ejecutan simulaciones donde se obtiene el tiempo de retardo y el consumo energético del sumador, para distintos voltajes, temperaturas y *process corners*. Se presentan resultados *post-layout* obteniendo un 23,7% menos de consumo con DMPL estático en comparación a CMOS y un 24.9% más de velocidad con DMPL en modo dinámico.

Palabras clave: sumador, CLA, circuitos digitales, señal de reloj, DML, DMPL.

ABSTRACT

In this work, three 32-bit Carry Look Ahead adders are designed in the 65 nm technology node with the TSMC library. A CMOS adder, a Dual Mode Logic and a combination between Dual Mode Logic and Dual Mode Pass Logic (DMPL) are realized. The last two are families that support two modes of operation. A dynamic operation mode, which is driven by a clock signal, achieves a shorter delay time than CMOS and another static mode, which has lower power consumption than CMOS. To study and analyze the behavior of the DMPL family, a detailed explanation of it is made, indicating both its advantages and disadvantages, especially its limitations of use for not making a completely DMPL circuit. For the comparison between the other logic families, simulations are carried out where the delay time and the power consumption of the adder are obtained for different voltages, temperatures, and process corners. Post-layout results are presented where 23.7% less power consumption is obtained with static DMPL compared to CMOS, and 24.9% more speed with DMPL in dynamic mode.

Keywords: adder, CLA, digital circuits, clock signal, DML, DMPL.

TABLA DE CONTENIDO

Introducción	10
Dual Mode Logic	11
Dual Mode Pass Logic	14
Desarrollo del Tema.....	16
Sumador Carry Look-Ahead.....	16
Dimensionamiento	18
Layout	19
Simulaciones y resultados.....	21
Process corners.....	24
Conclusiones	26
Referencias bibliográficas	28

ÍNDICE DE TABLAS

Tabla 1. Comparación de transistores y área entre los sumadores	21
---	----

ÍNDICE DE FIGURAS

Figura 1. Topología Compuertas DML: (a) Tipo A <i>footed</i> , (b) tipo A <i>unfooted</i> , (c) tipo B <i>unheaded</i> , (d) tipo B <i>headed</i>	12
Figura 2. Esquemático del bloque B.	17
Figura 3. Esquemático del sumador de 32 bits.	18
Figura 4. <i>Layout</i> compuerta XOR CMOS.	19
Figura 5. <i>Layout</i> XOR tipo A <i>footed</i> familia DML.	20
Figura 6. Retardo de los sumadores en función del voltaje.	22
Figura 7. Energía de los sumadores en función del voltaje.	23
Figura 8. (a) Tiempo de retardo en función de la temperatura a 0.5V. (b) Energía en función de la temperatura a 0.5V.	24
Figura 9. (a) Retardo en función de los <i>process corners</i> a 0.5 V. (b) Energía en función de los <i>process corners</i> a 0.5V.	25

INTRODUCCIÓN

El avance tecnológico de los últimos años exige trabajar con una gran cantidad de datos. Con el fin de satisfacer las exigencias actuales del diseño digital han surgido diversos sistemas de mayor o menor complejidad, que además de su alto rendimiento reducen el tamaño de los transistores. No obstante, el consumo energético ocasionado por las altas frecuencias de operación en estos sistemas sigue siendo un problema. Para enfrentarlo se presentan dos alternativas que, combinadas, permiten un alto rendimiento con menor consumo energético. Se trata del empleo de nuevas familias lógicas, junto con la variación de voltajes inferiores y cercanos al del umbral del transistor.

En este sentido, la familia *Complementary Metal-Oxide Silicon* (CMOS) es la más empleada en los circuitos integrados debido a sus características. Otras familias lógicas como *Dual Mode Logic* (DML) y *Dual Mode Pass Logic* (DMPL) han sido incorporadas con el fin de superar las cualidades de CMOS. En el caso de DML, ésta surge de la estructura CMOS y permite operar en dos modos según se requiera consumir menos energía o una respuesta rápida. Por otra parte, DMPL se forma a partir de DML y *Pass Transistor Logic* (PTL) y también posee ambos modos de operación. Pese a que esta última familia busca un menor consumo, su reciente creación (2020) dificulta su integración en circuitos complejos.

El presente trabajo estudia el funcionamiento y rendimiento de las familias DMPL y DML a través de un sumador *Carry Look-Ahead* de 32 bits con la librería de TSMC en el nodo tecnológico de 65nm, con la finalidad de analizar su variación en factores como frecuencia, energía y área. El sumador es un circuito que anticipa el bit de acarreo con el fin de aumentar la velocidad, lo que provoca una mayor complejidad en su estructura. Para realizarlo se emplearon las familias CMOS, DML y un circuito mixto de DML y DMPL. La necesidad de realizar una operación combinada con DMPL se debe a sus limitaciones de uso, especificadas

más adelante. Se empleó el método de diseño parcialmente medida, es decir, el armado del esquemático, el dimensionamiento de los transistores y los *layouts*, fueron realizados uno por uno. Se dice que fue un diseño parcialmente por que se emplearon varias veces algunas medidas y compuertas. En última instancia se presentan resultados *post-layout* de los tres sumadores para comparar su rendimiento y proporcionar el área esperada del circuito a través de los *layouts* realizados.

Cabe destacar que en el contexto ecuatoriano el diseño de circuitos integrados es mínimo y su investigación escasa. La exploración de nuevas tecnologías y la profundización en el diseño de circuitos a nivel de *layout* contribuye al desarrollo de la electrónica en el país y lo ayuda a posicionarse a nivel internacional.

Dual Mode Logic

Dual Mode Logic es una familia que se forma a partir de la *Pull-up Network* (PUN) y *Pull-down Network* (PDN) de una compuerta CMOS y se le añade uno o dos transistores. Al gate de estos transistores adicionales llega una señal de reloj (CLK) [1]. Según la ubicación de los transistores se obtiene una u otra topología cada una con sus propias características. Si el transistor se ubica en paralelo a la PUN es una compuerta tipo A *unfooted*. Si se le añade un transistor en serie a la PDN es una compuerta tipo A *footed*. En cambio, si el transistor en paralelo se ubica en paralelo a la PDN es una compuerta tipo B. Si además tiene un transistor en serie a la PUN es *headed*, si no lo tiene es *unheaded* [2]. Las cuatro topologías se pueden apreciar de mejor forma en Fig 1.

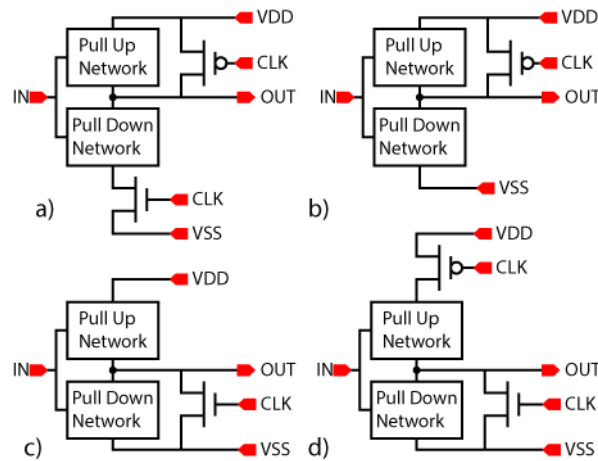


Figura 1. Topología Compuertas DML: (a) Tipo A *footed*, (b) tipo A *unfooted*, (c) tipo B *unheaded*, (d) tipo B *headed*.

El transistor en paralelo es imprescindible y permite a las compuertas DML operar en dos modos. La familia puede operar en estático y dinámico, el primer modo tiene un menor consumo energético, mientras que el segundo alcanza frecuencias superiores a las de CMOS. Para operar en estático el transistor de reloj debe recibir una entrada constante. La señal de reloj debe ser un 1 lógico para las compuertas tipo A y un 0 lógico si es tipo B [3]. Esto hace que los transistores en paralelo estén en corte y no dejen pasar la señal de VDD o VSS a la salida. Por otra parte, el transistor en serie, si es que existe, se mantiene encendido lo que permite la conexión entre la *network* y el voltaje de polarización. Al tener el transistor en paralelo apagado y el que está en serie encendido, la compuerta funciona como una compuerta CMOS evaluando únicamente el valor de la salida según las otras entradas [3]. En este caso el transistor en serie no tiene un gran impacto en el comportamiento de la compuerta, únicamente provoca un aumento mínimo el tiempo de respuesta y el consumo energético.

Para operar en modo dinámico la señal de reloj debe ser una señal cuadrada que cambie entre su valor lógico alto y bajo. El periodo de esta señal debe ser del ancho de las señales de entrada. Si es tipo A debe empezar en 0 y pasar a 1 y si es tipo B se invierte el

orden. Al pasar de un valor a otro el modo de operación dinámico tiene dos etapas. Durante el primer valor la compuerta se encuentra en la etapa de precarga [4]. Al empezar en un valor lógico 0/1 para el tipo A/B el transistor en paralelo este prendido lo que hace que la salida precargue a 1/0. Tras finalizar la precarga se da la etapa de evaluación. La señal de reloj pasa al siguiente valor, se desactiva el transistor en paralelo y hace que la compuerta evalúe el valor a la salida según las entradas. Es en este modo de operación que la topología escogida tiene un mayor impacto.

La topología tipo A/B precarga a VDD/VSS, lo que hace que sea mejor para compuertas como las NOR/NAND donde se tiene transistores en serie en la PUN/PDN. Es decir que, con la precarga a través del transistor de reloj, ya no afecta en nada el tiempo de retardo de la PUN o PDN. Esto beneficia en velocidad a las compuertas donde el transistor se ubique en paralelo a varios transistores en serie. No obstante, la precarga también puede ser perjudicial si no se escoge una topología correctamente.

Durante la precarga la compuerta toma un valor alto o bajo, este valor puede causar un cortocircuito si las entradas de la compuerta dan el valor contrario a la salida. Por ejemplo, si una compuerta tipo A precarga a 1 y la PDN esta activa la salida se evalúa a 0, generando un cortocircuito. Para evitar esto se usan las compuertas *footed* y *headed*. El transistor en serie esta apagado durante la precarga lo que evita que evalúe al valor contrario del transistor en paralelo. Se debe tener en cuenta que estas topologías tienen mayor retardo y ocupan más área [2]. No obstante, no son de carácter obligatorio, si se ubican compuertas tipo A antes de una tipo B, las salidas de la primera precargan a un valor alto. Este valor pasará a los *gates* de los transistores de la segunda compuerta y su salida será un cero lógico. Por lo que la compuerta tipo B puede ser *unheaded* y no se generaría un cortocircuito. Lo óptimo en una cadena es que las primeras compuertas sean *footed* o *headed*, continúe con algunas sin el transistor en serie y cada cierta cantidad de compuertas se vuelva a usar una *footed* o *headed*.

En cuanto a su dimensionamiento, los transistores en paralelo al transistor de precarga el ancho mínimo de la tecnología [5]. Esto hace que consuman menos energía.

Adicionalmente, al momento de realizar la evaluación si la compuerta evalúa al valor opuesto al que precarga el retardo será menor debido a que los transistores de ancho mínimo emplean menor corriente y la corriente de los otros transistores pueden alcanzar el voltaje objetivo más rápido. La *network* opuesta al transistor de precarga se dimensiona igual a la compuerta CMOS.

Dual Mode Pass Logic

Esta familia lógica es muy parecida a la DML, puesto que su estructura se basa en una DML. La diferencia es que la *network* con el transistor de precarga pasa a ser de lógica PTL. Al emplear PTL en una de sus partes se obtiene un menor consumo energético y se aprovecha su menor cantidad de transistores [6]. Como mantiene el resto de la estructura como DML no pierde mucha velocidad y mantiene valores lógicos fuertes, a diferencia de un PTL que tiene valores lógicos degradados. Puede operar en los dos modos al igual que en DML y su funcionamiento es el mismo.

A pesar de tener tantas ventajas su uso se ve limitado por la lógica PTL. En primer lugar, las compuertas *footed* o *headed* no son óptimas. Debido a su lógica PTL la compuerta puede seguir evaluando a cualquier valor y generar el cortocircuito, se debe tener en consideración si se emplea al inicio de una cadena. Como segundo punto estas compuertas solo pueden ser usadas si el transistor de precarga está en paralelo a transistores en serie. Para generar un DMPL se elimina uno de los transistores en serie y en lugar de conectarla al voltaje de polarización se conecta a la entrada del transistor negada [7]. Esto no se puede hacer, por ejemplo, con una NOR tipo B a causa de que los transistores de la PDN están todos paralelo y no pueden ser eliminados, ya que la tabla de verdad no se cumpliría. Por lo que solo funciona correctamente en compuertas como la NOR-A y la NAND-B. Por último, se necesitan

inversores en el circuito para usar DMPL. Si no se tiene una entrada negada sería necesario añadir un inversor lo que genera mayor consumo. Por ello la mejor forma de emplear DMPL es en combinación con DML.

DESARROLLO DEL TEMA

Sumador Carry Look-Ahead

El sumador empleado se basa en la arquitectura de [8], que presenta un sumador CLA de 32 bits. Este sumador busca reducir el tiempo de propagación del bit de acarreo, que limita la velocidad del circuito [9-10], calculando el valor del bit con anticipación. Para anticipar el valor del bit de acarreo se emplea un mayor número de compuertas y una estructura más compleja que se divide en dos bloques A y B con el fin de simplificar su entendimiento. Se adapta esta arquitectura usando una combinación entre DML y DMPL y se realiza un sumador CMOS y otro DML para su comparación.

Bloque A.

El bloque A realiza la suma entre las dos entradas y el bit de acarreo. Adicionalmente se encarga de obtener el bit de generación (G) y el bit de propagación (P). Estos bits permiten calcular anticipadamente el bit de acarreo de la siguiente suma [8]. Para obtener G se emplea una compuerta NAND-A y un inversor tipo B *unheaded*, una compuerta AND, si los dos bits de la suma son 1 se genera un bit de acarreo. Y no necesita esperar al bit de acarreo de la anterior suma para obtener el siguiente. Por otra parte, P se obtiene con una compuerta OR, se emplea una compuerta NOR-A y un inversor tipo B *unheaded*. Todas estas compuertas son DML, debido a que las compuertas tipo A son las primeras es necesario que sean *footed*, por lo que no pueden ser DMPL al igual que los inversores.

Para realizar la suma de los bits se emplearon dos compuertas XOR en cascada. Los dos bits que se van a sumar ingresan a la primera y a la segunda el resultado de la suma anterior y el bit de acarreo. La primera compuerta es una XOR tipo A *footed* y la segunda es también tipo A DMPL, por lo que es *unfooted*. A pesar de que están una después de la otra, ambas son tipo A debido a que el bit de acarreo sale de una compuerta tipo B. Dada la tabla de verdad de la XOR para evitar el cortocircuito de una tipo A las anteriores deben ser cada

una de distinto tipo. De este modo durante la precarga se tenga un valor alto y un valor bajo en las entradas y la salida de 1. En caso de que fuera tipo B ambas compuertas deben ser del mismo tipo para que la siguiente precargue a 0 correctamente.

Bloque B.

El bloque B calcula los bits de acarreo empleando el P y G obtenido por el bloque A. Para realizar las operaciones se emplean diversas compuertas NAND de 2, 3 y 4 entradas, de tipo B y A, también se emplean inversores de ambos tipos. Estos inversores se aprovechan para usar compuertas DMPL con su entrada negada. Todos los bloques B tienen cadenas de dos compuertas, siendo las primeras tipo A y las últimas tipo B para que se conecten con los otros bloques sin problemas de cortocircuitos. Esto último es tanto para DML como la combinación de DML y DMPL. La Fig. 2 da un mejor entendimiento de la distribución de las compuertas y el cálculo de cada valor surge de las siguientes ecuaciones:

$$P_{out} = P_0P_1P_2P_3$$

$$G_{out} = G_0P_1P_2P_3 + G_1P_2P_3 + G_2P_3 + G_3 = \overline{\overline{G_0P_1P_2P_3} \cdot \overline{G_1P_2P_3} \cdot \overline{G_2P_3} \cdot \overline{G_3}}$$

$$C_1 = G_0 + P_0C_0 = \overline{\overline{G_0} \cdot \overline{P_0C_0}}$$

$$C_2 = G_1 + G_0P_1 + C_0P_0P_1 = \overline{\overline{G_1} \cdot \overline{G_0P_1} \cdot \overline{C_0P_0P_1}}$$

$$C_3 = G_2 + G_1P_2 + G_0P_1P_2 + C_0P_0P_1P_2 = \overline{\overline{G_2} \cdot \overline{G_1P_2} \cdot \overline{G_0P_1P_2} \cdot \overline{C_0P_0P_1P_2}}$$

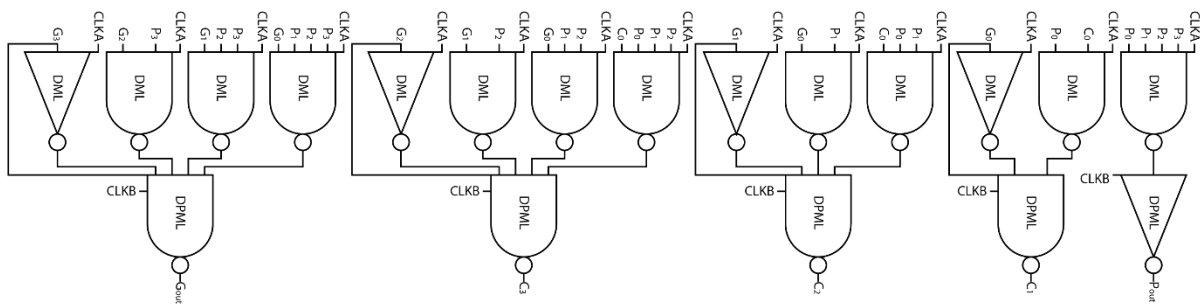


Figura 2. Esquémico del bloque B.

Estos bloques B calculan los bits de acarreo de 3 sumas distintas y generan otro bit G y P. Por lo que se agrupan cada 4 sumas, y se pone otra etapa de bloques B que toman los bits G y P de 4 bloques B de la anterior etapa [8]. Los bloques de otras etapas permiten obtener los bits de acarreo que no se calculan en la primera parte de la jerarquía. Al realizar un sumador de 32 bits se usan 3 etapas de bloques B, no obstante, esta última solo requiere de la mitad de la estructura de un bloque B, puesto que solo requiere generar el acarreo para la suma 16 y para el bit final. Esta estructura se puede ver simplificada en la Fig 3.

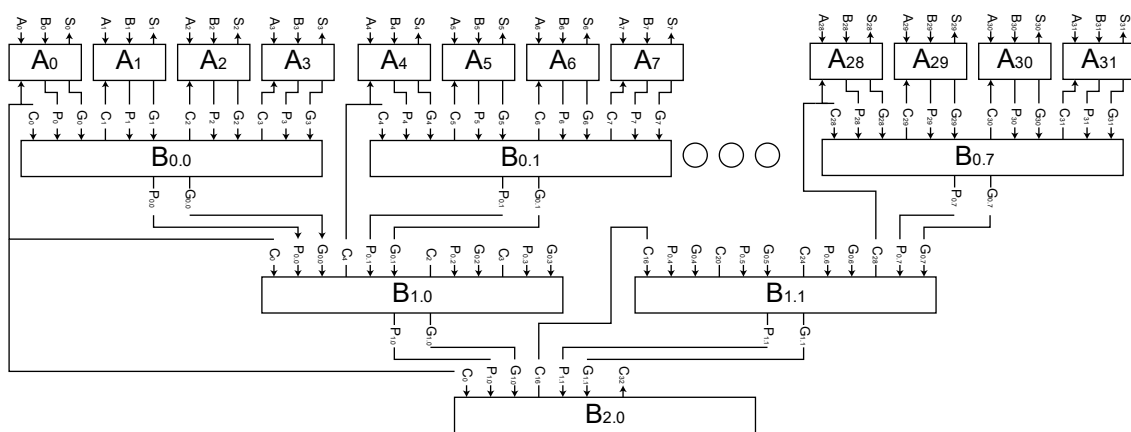


Figura 3. Esquemático del sumador de 32 bits.

Dimensionamiento

Para dimensionar las compuertas se empleó un análisis DC con voltaje nominal de 0.5V, graficando la curva de transferencia. Una posición centrada y una buena pendiente indica una buena transición de un valor lógico a otro. También se observó el margen de ruido alto y bajo, este parámetro indica que tanto ruido permite una compuerta en valores altos y bajos antes de que el resultado se vea comprometido [11]. Se busca obtener la mayor magnitud margen de ruido posible tanto para valores altos como bajos, lo que a su vez daba una curva de transferencia centrada. Según el margen obtenido se determina el ancho del transistor para inversores, NANDs de 2, 3 y 4 entradas, y NORs. Al tener el dimensionamiento de los anchos de estas compuertas se pueden usar los mismos en las

networks de DML y DMPL que no estén en paralelo al transistor de precarga. Los transistores restantes tienen el ancho mínimo, en este caso 200 nm.

Layout

Se realizaron *layouts* de cada compuerta CMOS, DML y DMPL. Para su elaboración se empleó el software de Custom Compiler de Synopsys. Las comprobaciones de reglas de diseño, de las conexiones del *layout* respecto al esquemático y la extracción de parásitos se realiza con Calibre. Para cada *layout* se dio una altura de celda de $2.52\mu\text{m}$, esta altura es fija para todas las familias y el ancho varía dependiendo de la celda. Siendo las compuertas de menor ancho las DML y DMPL en la mayoría de los casos.

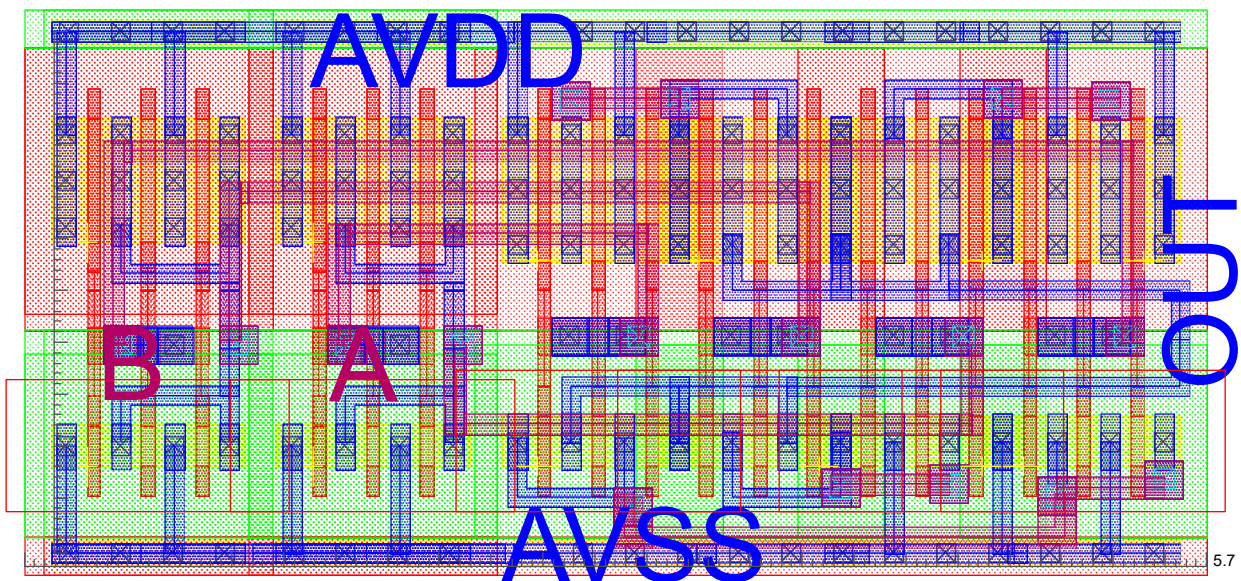


Figura 4. *Layout* compuerta XOR CMOS.

Para todos los *layouts* se empleó la técnica de *fingers*, que consiste en dividir el transistor partes iguales. Se pasa de tener una relación de Ancho(W)/Largo(L) a una relación $N(W_n/L)$, siendo N el número de *fingers* en los que se divide el transistor. Una relación W/L más pequeña es conveniente, produce menos parásitos y se obtiene una menor altura [12]. Además, esto permite tener una mejor polarización de la compuerta al tener menos pérdidas de voltaje. Se utilizó únicamente una capa de metal, a excepción de las compuertas XOR, que

por su complejidad se usó hasta metal 2. Y para reducir el espacio la polarización y la conexión a bulto se ubican sobre una línea en la parte superior e inferior de cada celda.

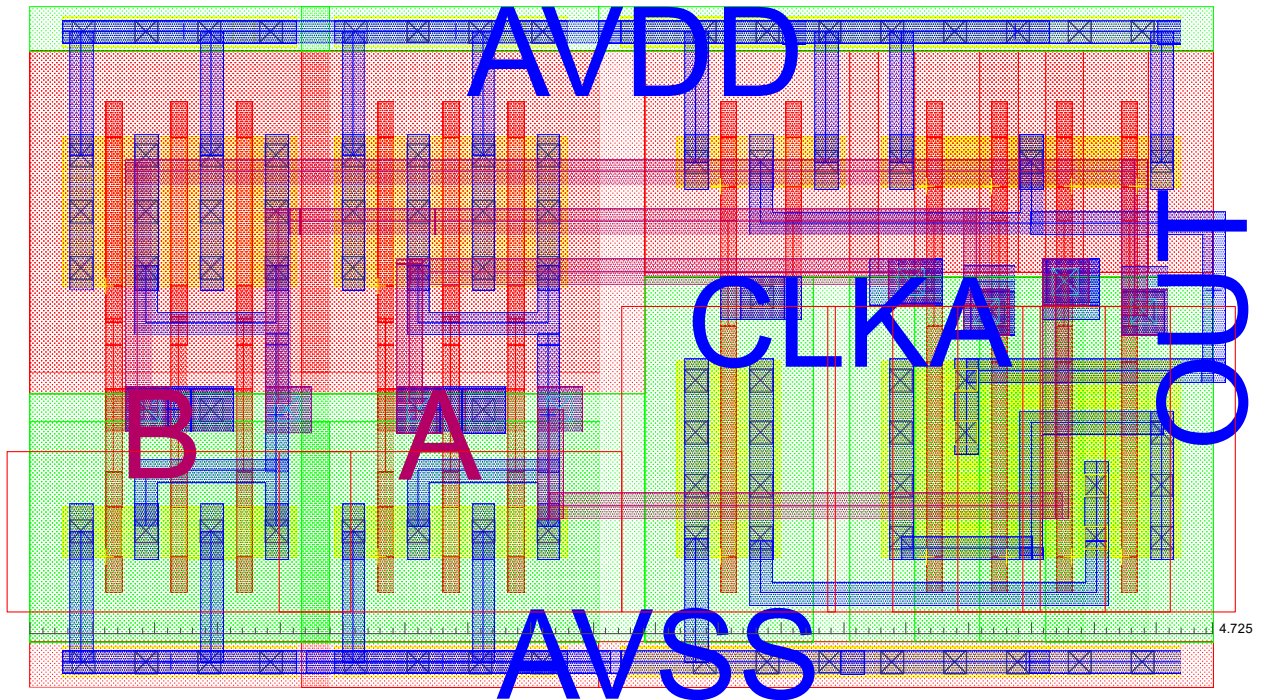


Figura 5. Layout XOR tipo A *footed* familia DML.

En la Fig. 4 se puede ver el *layout* de la compuerta XOR CMOS. Las entradas A y B están conectada a varios *gates*, lo que indica el uso de *fingers*. Las polarizaciones se distinguen claramente en la parte superior e inferior AVDD y AVSS. Este *layout* tiene una altura de $2.52\mu\text{m}$ y un ancho de $5.7\mu\text{m}$. En la Fig. 5 se tiene la XOR tipo A *footed*, que tiene un ancho de $4.725\mu\text{m}$. En la derecha de la imagen cerca de AVDD se puede ver como al ser tipo A los transistores están con el ancho mínimo por lo que ocupan menos espacio y el *layout* no requiere de tantos *fingers*. Finalmente, se realiza la medición de las dimensiones de todas las compuertas y se calcula el área total de cada sumador, los resultados se encuentran en la tabla 1.

Tabla 1. Comparación de transistores y área entre los sumadores

Sumador	Número de transistores nmos/pmos	Área [μm^2]
CMOS	1060/1060	2967.678
DML	1384/1333	2686.8492
DMPL/DML	1342/1269	2686.8492

Simulaciones y resultados

Para las simulaciones se empleó el *Simulation Analysis Enviroment* del *Custom Compiler*. Las librerías son de TSMC de 65 nm. En primer lugar, se realizaron simulaciones para comprobar el funcionamiento del sumador, tanto antes como después del *layout*. Tras comprobar el funcionamiento se realizaron diversas simulaciones para obtener tiempos de retardo, energía y comportamiento a distintas temperaturas. Todas estas simulaciones tenían *buffers* de entrada e inversores a modo de carga a la salida, para obtener resultados más realistas.

Tiempo de retardo

Para obtener el tiempo de retardo se realizó un perfil de testeo que provoque el peor caso de la compuerta. Para un sumador el peor caso es cuando el bit de acarreo se propaga desde la primera hasta la última suma, es decir, en las entradas uno de los bits de suma debe ser 1 y el otro 0, mientras que el primer bit de acarreo es 1. De esta forma la primera suma dará un bit de acarreo a la siguiente y así continuamente hasta la última suma.

Una vez realizada la simulación de este caso, se midió el tiempo de retardo desde una de las entradas la salida de la suma 31. No se considera el bit de acarreo 32, dado que la arquitectura del sumador hace que el bit 32, que se calcula en la tercera etapa del sumador, llegue antes que el bit de acarreo 31. Los valores fueron medidos desde 0.3 a 1.2 V.

Para el modo dinámico se debe medir el tiempo de precarga y de evaluación de la suma 31. La suma de ambos será el retardo total de la compuerta y esto permitirá llevar al sumador a su máxima frecuencia. Los resultados obtenidos se muestran en la Fig. 6.

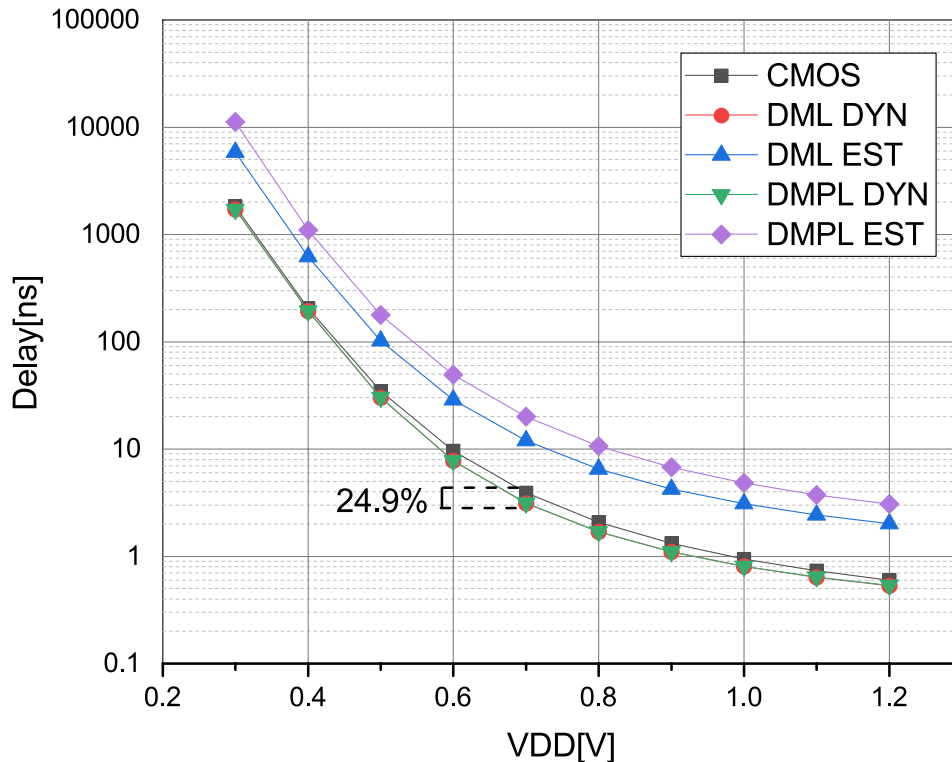


Figura 6. Retardo de los sumadores en función del voltaje.

Como se puede ver DMPL en estático es el sumador más lento, entre un 50 y 90% más lento que DML y 4 veces más lento que CMOS. Por otra parte, DMPL y DML son más rápidos que CMOS, entre el 6% y 25%. Los resultados entre 0.6V y 0.9V son los mejores, superando el 20%. La diferencia de velocidad entre DMPL y DML es mejor al 1% en todos los casos siendo el más rápido DML.

Energía

Para la medición del consumo de energía se llevó el sumador a su máxima frecuencia. Con los tiempos de retardo obtenidos para cada voltaje se obtiene el ancho de pulso de cada señal y el periodo del reloj. Para medir energía se generó un perfil de pruebas con 100

entradas obtenidas de forma aleatoria en MatLab. Se obtuvo la energía y se la promedio para poder obtener la energía por operación. Los resultados se muestran en la Fig. 7.

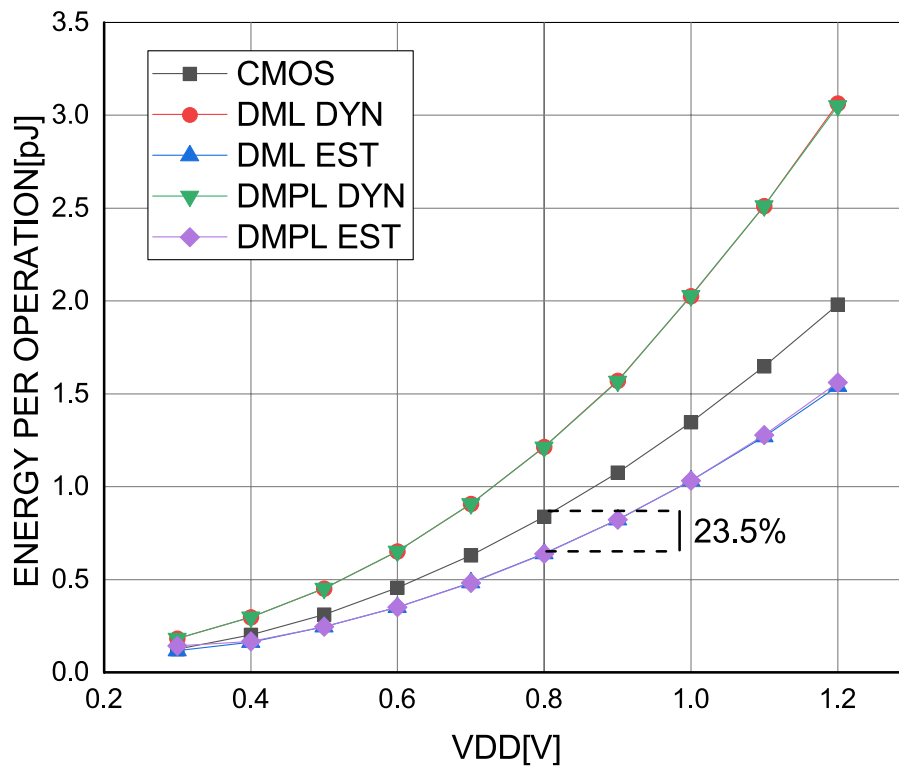


Figura 7. Energía de los sumadores en función del voltaje.

DMPL y DML tienen el menor consumo, entre 20 y 24% menos que CMOS. De 0.6V a 1V el sumador DMPL consume menos que DML. Al emplear tan pocas compuertas DMPL en relación con DML se nota poco el impacto en consumo energético. DMPL consume más a voltajes inferiores a 0.5V, esto se debe a que a voltajes tan bajos las lógicas PTL funcionan peor y el tiempo de retardo es mucho mayor. Al tener mayor tiempo de retardo también se tiene mayor tiempo de transición, esta transición tan lenta en comparación a DML y con un consumo similar genera mayor consumo. No obstante, la disipación de energía es muy parecida en ambos sumadores. El mejor resultado se da a 0.8V donde DML consume 23.5% menos que CMOS y DMPL 23.7% menos. Por otra parte, los modos dinámicos consumen de 80% a 100% más, a excepción de 0.3V donde consume 55% más.

Temperatura

El método de medición del comportamiento del sumador a distintas temperaturas es igual al de temperatura y energía, únicamente se hace una variación de la temperatura de 0 °C a 80 °C, en pasos de 20 °C. En la Fig. 8 se muestra la tendencia de las compuertas respecto a la temperatura con un voltaje de 0.5V.

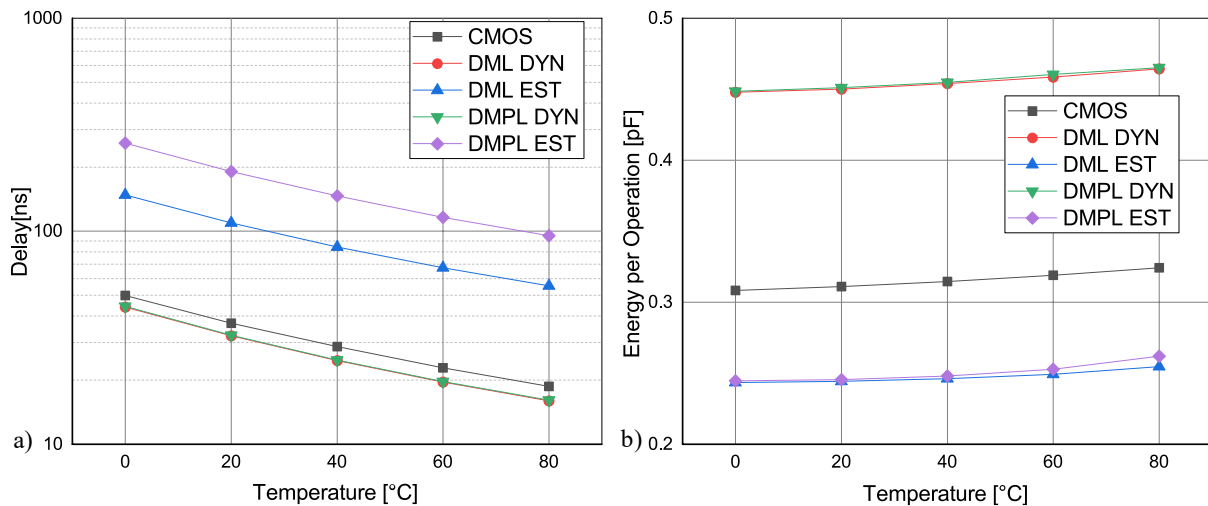


Figura 8. (a) Tiempo de retardo en función de la temperatura a 0.5V. (b) Energía en función de la temperatura a 0.5V.

El sumador mantiene el mismo comportamiento en todas sus familias al disminuir el retardo conforme aumenta la temperatura. Por otra parte, la tendencia de consumo de DMPL estático aumenta ligeramente en las temperaturas más altas. El resto de los modos mantienen la misma tendencia que CMOS, lo que indica un buen comportamiento respecto a las variaciones de temperatura.

Process corners

Estos procesos son variaciones en los transistores, que se pueden dar en transistores por efectos durante el grabado o por variaciones ambientales. Estas variaciones hacen que durante la realización del circuito no se pueda controlar con precisión el dopaje de los materiales semiconductores. La cantidad de dopaje modifica el comportamiento del transistor pudiendo ser Típico (T), Rápido(F) o Lento(S) [11]. Es importante ver el comportamiento del circuito ante estas variaciones de velocidad puesto que son involuntarias y debe comprobarse

que el circuito funciona para cualquier caso. Si no funciona correctamente en alguno de los casos la probabilidad de éxito disminuye y pueden salir una mayor cantidad de circuitos defectuosos o con menor rendimiento del esperado. Para verificar el comportamiento respecto a estas variaciones se midieron tiempo y energía en TT, SS, SF, FS y FF. La primera letra se refiere los transistores nmos y la segunda a los pmos, al tener cada transistor características independientes se puede caracterizar su velocidad por separado. En la Fig. 9 se puede ver cómo se comporta el sumador de cada familia ante estas variaciones a 0.5V.

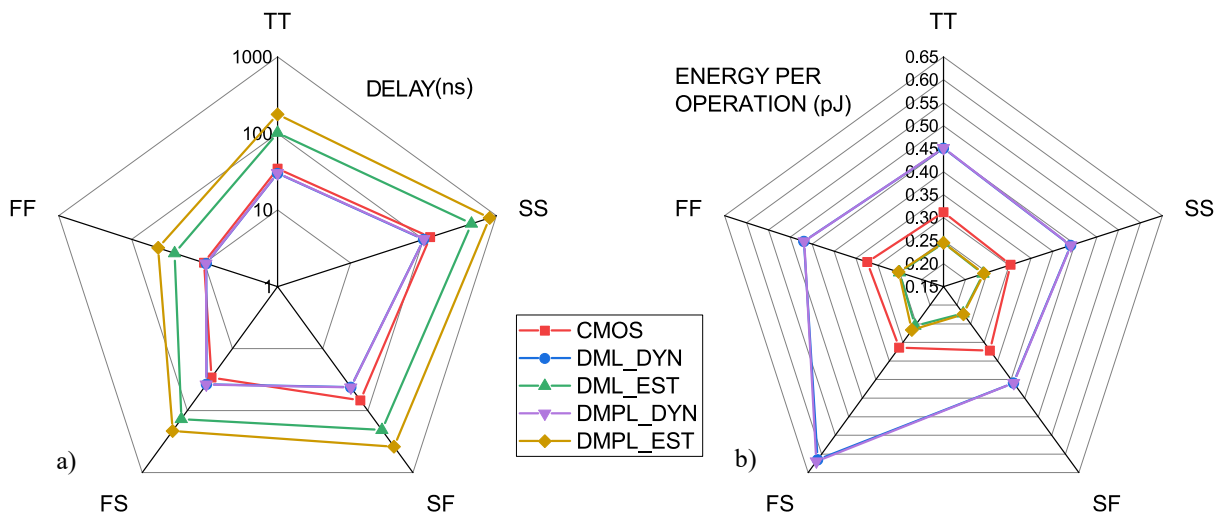


Figura 9. (a) Retardo en función de los *process corners* a 0.5 V. (b) Energía en función de los *process corners* a 0.5V.

En la Fig. 9 (a) se observa como el tiempo de retardo mantiene la misma tendencia en todas las familias, excepto en FS que CMOS se vuelve más rápido y en SF, que la ventaja de velocidad de los modos dinámico aumenta considerablemente siendo hasta un 40% más rápido que CMOS. Esto se debe a que DML y DMPL no tienen el mismo número de transistores tipo n que de tipo p, mientras que CMOS sí. En cuanto al consumo energético mantienen la misma tendencia a excepción de FS donde los modos dinámicos aumentan considerablemente su disipación de energía. Por lo que el sumador es estable en todos sus *corners* exceptuando FS donde empeora su rendimiento y aumenta su gasto energético.

CONCLUSIONES

Se logró realizar un sumador de 32 bits en tres familias lógicas, CMOS, DML y una combinación de DMPL con DML, empleado una metodología de diseño parcialmente a medida en el nodo tecnológico de 65 nm con la librería de TSMC. Se obtuvieron y compararon resultados post-*layout* de los tres sumadores. Los *layouts* permitieron ver como DMPL y DML ocupan menos espacio que CMOS, aunque los dos primeros tienen mayor cantidad de transistores. Estos transistores son los que permiten tener dos modos de operación.

DML y DMPL admiten dos modos de operación según se requiera un modo dinámico para obtener mayor velocidad o estático, para obtener menor consumo energético. Como se pudo ver DMPL en estático consume hasta un 23,7% menos que CMOS en 0.8V, mientras que DML 23.5% menos. Por otra parte DML dinámico es 24.99% mas veloz que CMOS a 0.7V y DMPL logra un 24.9% al mismo voltaje. Por lo que se obtuvieron mejores resultados que CMOS, en general los mejores resultados son desde los 1.2 V hasta los 0.5 V donde ambas compuertas superan notablemente a CMOS.

Para DMPL los mejores resultados van de 0.6 V a 1 V, en resultados más bajos la lógica PTL no tiene un buen rendimiento por lo que se ve reducido el tiempo de retardo. Esto solo se nota en modo estático donde el transistor de precarga no evita ese tiempo de retardo, lo que tiene un punto positivo y otro negativo. Por una parte, en modo estático este tiempo no afecta puesto que no interesa la velocidad, sin embargo, la diferencia es tanta que el consumo es casi idéntico. Por encima de los 0.6 V DMPL si tiene un menor consumo en estático, y en dinámico pierde menos del 0.5% de velocidad respecto a DML.

La diferencia entre DMPL y DML son mínimos, pero se verifica que en voltajes de 0.6 a 1V DMPL consume menos. Estos resultados eran los esperados dado que las limitaciones de DMPL impiden emplear muchas de estas compuertas, por lo que el impacto no es muy alto. De todas formas, si se recomienda su uso a los voltajes mencionados para disminuir el consumo.

Finalmente se puede decir que tanto DML y DMPL son estables ante las variaciones de temperatura, mantienen el mismo comportamiento que CMOS tanto en tiempo de retardo como energía. En cuanto a los *process corners* también se mantienen estables exceptuando en FS donde las características de DML y DMPL no son tan buenas en comparación a CMOS. No obstante, esto se compensa en FS donde los modos dinámicos mejoran notablemente. Esto se debe a la diferencia de transistores tipo n y tipo p en cada sumador. En las demás variaciones los sumadores mantienen la misma tendencia.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Kaizerman, S. Fisher and A. Fish, "Subthreshold Dual Mode Logic," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 5, pp. 979-983, May 2013, doi: 10.1109/TVLSI.2012.2198678.
- [2] V. Yuzhaninov, I. Levi and A. Fish, "Design Flow and Characterization Methodology for Dual Mode Logic," in *IEEE Access*, vol. 3, pp. 3089-3101, 2015, doi: 10.1109/ACCESS.2016.2514398.
- [3] Vicuña, Kevin, Cristhopher Mosquera, Ariana Musello, Sara Benedictis, Mateo Rendón, Esteban Garzón, Luis M. Prócel, Lionel Trojman, and Ramiro Taco. 2021. "Energy Efficient Self-Adaptive Dual Mode Logic Address Decoder" *Electronics* 10, no. 9: 1052. <https://doi.org/10.3390/electronics10091052>
- [4] L. Yavits, R. Taco, N. Shavit, I. Stanger and A. Fish, "Dual Mode Logic Address Decoder," 2020 *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1-5, doi: 10.1109/ISCAS45731.2020.9180587.
- [5] I. Levi, A. Kaizerman, A. Fish, "Low voltage dual mode logic: Model analysis and parameter extraction," *Microelectronics Journal*, Volume 44, Issue 6, 2013, pp 553-560, ISSN 0026-2692, <https://doi.org/10.1016/j.mejo.2013.03.005>.
- [6] N. Lindert, T. Sugii, S. Tang and Chenming Hu, "Dynamic threshold pass-transistor logic for improved delay at lower power supply voltages," in *IEEE Journal of Solid-State Circuits*, vol. 34, no. 1, pp. 85-89, Jan. 1999, doi: 10.1109/4.736659.
- [7] I. Stanger, N. Shavit, R. Taco, L. Yavits, M. Lanuzza and A. Fish, "Robust Dual Mode Pass Logic (DMPL) for Energy Efficiency and High Performance," 2020 *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1-5, doi: 10.1109/ISCAS45731.2020.9181127.
- [8] I. Levi, O. Bass, A. Kaizerman, A. Belenky and A. Fish, "High speed Dual Mode Logic Carry Look Ahead Adder," 2012 *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2012, pp. 3037-3040, doi: 10.1109/ISCAS.2012.6271959.
- [9] F.-C. Cheng, S.H. Unger and M. Theobald, "Self-timed carry-lookahead adders," *IEEE Trans. Comp.*, vol. 49, pp. 659-672, 2000.
- [10] A. De Gloria and M. Olivieri, "Statistical carry lookahead adders," *IEEE Trans. Comp.*, vol. 45, pp. 340-347, 1996.
- [11] N. Weste and D. Harris, *CMOS VLSI design: a circuits and systems perspective*. Pearson, Boston, 2011.
- [12] A. Hastings, *The Art of Analog Layout*. Prentice Hall, United States of America, 2001.