



Trabajo Fin de Máster
"Máster Universitario en Microelectrónica:
Diseño y Aplicaciones de Sistemas
Micro/Nanométricos"

**Diseño e implementación de lógica de
lectura de un sensor solar para sistemas de
navegación espacial**

Autor: Montserrat Sánchez Gallego
Tutor: Juan Antonio Leñero Bardallo
Fecha: 30 de noviembre de 2020

Agradecimientos

A mis padres, por haberme enseñado que con esfuerzo, trabajo y constancia todo se consigue y por su apoyo incondicional. También por confiar en mí y por haberme proporcionado la mejor educación que ha estado a su alcance.

A mis tutor, D. Juan Antonio Leñero Bardallo por su continua ayuda e interés por el trabajo llevado a cabo, sin los que no hubiera sido posible realizar este proyecto.

En especial, a mis hermanos. De ellos he obtenido todo el cariño, comprensión y apoyo cuando lo necesitaba.

A mi tío David y mi hermana Jénifer, por ser un referente e inculcarme la pasión por la ciencia.

Índice General

Resumen	IX
1. Introducción	1
1.1. Historia de los dispositivos de navegación espacial	1
1.2. La posición del sol en coordenadas esféricas	4
1.3. Sensores solares	6
1.3.1. Sensores solares analógicos	6
1.3.2. Sensores digitales	8
1.3.2.1. Sensores de imagen	9
1.3.2.2. Unión pn, efecto fotoeléctrico	11
1.3.3. Sensores solares asíncronos	12
1.3.3.1. Sensores DVS	13
1.3.3.2. Sensores tipo octopus	14
1.3.3.3. Sensores Time-to-First-Spike	15
1.3.4. Protocolo de comunicación AER	17
1.4. Objetivo del proyecto	18
1.5. Motivaciones para desarrollar el sistema propuesto	19
2. Descripción del sistema	21
2.1. Descripción general del sistema	22
2.1.1. Pixel	23
2.1.2. Arbitrador	26
2.1.2.1. Fair arbiter	26
2.1.2.2. Greedy Arbiter	29
2.1.3. Multiplexor	31
2.1.4. Lógica de selección	34
3. Simulaciones eléctricas	39
3.1. ¿Qué es la simulación?	39
3.2. Cadence	41
3.3. Simulación de los dispositivos	42
3.3.1. Puertas lógicas básicas	43

3.3.1.1.	Inversor	43
3.3.1.2.	NAND	45
3.3.1.3.	NOR	47
3.3.1.4.	SR latch	48
3.3.2.	Arbitrador	50
3.3.3.	Multiplexor	54
3.3.4.	Lógica de selección	58
3.3.5.	Circuito completo	62
3.3.5.1.	Level Shifter	62
3.3.5.1.1.	Level Shifter High-Low	63
3.3.5.1.2.	Level Shifter Low-High	64
3.3.5.2.	Simulación del circuito (píxel y periferia)	66
4.	Layout	73
4.1.	Arbitrador	77
4.2.	Multiplexor	80
4.3.	Lógica de selección	82
4.4.	Periferia	82
	Conclusiones generales y líneas de futuro	85
A.	Esquemáticos	91
		91

Índice de figuras

1.1.	Sensores solares en un microsatélite [34]	3
1.2.	Punto r en el espacio tridimensional en coordenadas esféricas [11]	5
1.3.	Estructura del elemento sensor de un sensor solar analógico [4]	7
1.4.	Indicencia en un sensor solar analógico de cuatro fotodiodos [10]	7
1.5.	Sensor Solar digital fabricado por Solar MEMS [22]	8
1.6.	Definición de un sensor solar digital [29]	9
1.7.	Configuración básica de un sensor de imagen [4]	10
1.8.	Modelo simplificado de un fotodiodo [4]	11
1.9.	Característica I-V de un fotodiodo	12
1.10.	Implementación de un píxel capaz de generar pulsos en función de la iluminación [13].	15
1.11.	Esquema del generador de imágenes TFS en modo de imagen fija [9].	16
1.12.	Señales del canal de comunicación y temporización. (a) bus de datos (d_0, \dots, d_3) y señales de establecimiento de comunicación (r, a). (b) Diagrama temporal [1]	18
2.1.	Diagrama de bloques del sistema. [4]	22
2.2.	Implementación del sistema formado por la matriz de píxeles y la cubierta con el orificio. [4]	24
2.3.	Esquemático del píxel [15]	25
2.4.	Esquema fair arbiter [27]	27
2.5.	Esquema tipo árbol de Fair arbiters [27]	28
2.6.	Diagrama temporal Fair arbiter [27]	28
2.7.	Esquema Greedy arbiter [4]	30
2.8.	Árbol binario de arbitradores de 8 peticiones [12]	31
2.9.	Esquema de funcionamiento de un multiplexor de dos entradas	32
2.10.	Disposición de varias filas del multiplexor en el circuito de periferia	32

2.11. Esquema de los módulos de "0" y "1" que componen el multiplexor [14]	33
2.12. Esquemático del módulo de lógica de selección con tecnología UMC 0.18 (CMOS)	35
2.13. Esquema del circuito de periferia [14]	37
3.1. Herramientas EDA e IP de Cadence Virtuoso [31]	42
3.2. Característica de transferencia de voltaje del inversor CMOS cuando Q_N y Q_P coinciden [25]	43
3.3. Esquemático y símbolo de un inversor con tecnología UMC 0.18 μm (CMOS) A.1	44
3.4. Evolución temporal de señales del inversor CMOS	45
3.5. Esquemático y símbolo de una puerta NAND con tecnología UMC 0.18 μm (CMOS) A.2	46
3.6. Evolución temporal de señales de la puerta NAND	46
3.7. Esquemático de una puerta NOR con tecnología UMC 0.18 μm (CMOS) A.3	47
3.8. Evolución temporal de señales de la puerta NAND	48
3.9. Esquemático y símbolo del latch SR con tecnología UMC 0.18 μm (CMOS) A.4	49
3.10. Banco de pruebas del latch SR	49
3.11. Evolución temporal de señales del latch SR	50
3.12. Esquemático del módulo arbitrador con tecnología UMC 0.18 μm (CMOS) A.9	51
3.13. Banco de pruebas para el arbitrador	52
3.14. Evolución temporal de las señales del arbitrador	52
3.15. Zoom de la evolución temporal de la señales del arbitrador	53
3.16. Evolución temporal de las señales del arbitrador con un análisis paramétrico en función de VDD	54
3.17. Esquemático y símbolo del bit "0" y del bit "1" con tecnología UMC 0.18 μm (CMOS) A.5 A.6	55
3.18. Banco de pruebas para los bits del multiplexor	56
3.19. Evolución temporal de señales del multiplexor	57
3.20. Evolución temporal de las señales del multiplexor con un análisis paramétrico en función de VDD	58
3.21. Esquemático del módulo de lógica de selección con tecnología UMC 0.18 μm (CMOS) A.11	58
3.22. Banco de pruebas para la lógica de selección	60
3.23. Evolución temporal de las señales de la lógica de selección	61
3.24. Evolución temporal de señales de la lógica de selección con un análisis paramétrico en función de VDD	62

3.25. Esquemático y símbolo del level shifter de high a low con tecnología UMC 0.18 μm (CMOS) A.7	63
3.26. Banco de pruebas para el level shifter low-high	63
3.27. Variación temporal de las señales del level shifter de high a low voltage.	64
3.28. Esquemático y símbolo del level shifter de low a high con tecnología UMC 0.18 μm (CMOS) A.8	64
3.29. Banco de pruebas para la lógica de selección	65
3.30. Variación temporal de las señales del level shifter de low a high voltage.	66
3.31. Banco de pruebas para un píxel de 2×2	67
3.32. Evolución temporal de las señales de las filas circuito para $I_{ph1} = 100 \text{ pA}$, dirección del píxel (0,1)	68
3.33. Evolución temporal de las señales de las columnas circuito para $I_{ph1} = 100 \text{ pA}$, dirección del píxel (0,1)	69
3.34. Evolución temporal de las señales del circuito cuando todos los píxeles son iluminados con la misma intensidad, dirección del píxel (1,1)	70
3.35. Evolución temporal de señales de la lógica de selección	71
4.1. Comprobación de las reglas de diseño del arbitrador	74
4.2. Captura de la comprobación de layout vs esquemático en el arbitrador	74
4.3. Efecto antena: (a) esquema de daño por etching de plasma, (b) la carga acumulada en los paths 2-4 se descarga a través del óxido de puerta causando daños sobre este, (c) las carga acumulada en los paths 5-8 se descarga a través de la difusión y no causa daño.	75
4.4. Layout del arbitrador	77
4.5. Capacidades por nodo del bloque arbitrador	78
4.6. Esquemático del módulo arbitrador con capacidades parásitas con tecnología UMC 0.18 μm (CMOS) A.10	79
4.7. Evolución temporal de señales del arbitrador con capacidades parásitas	80
4.8. Layout del módulo bit cero	81
4.9. Layout del módulo bit uno	81
4.10. Layout de la lógica de selección	82
4.11. Layout del bloque periferia y ampliación de la zona central.	83
4.12. Diagrama de la posición de la periferia respecto a la matriz de píxeles	84

A.1. Esquemático y símbolo de un inversor con tecnología UMC 0.18 μm (CMOS)	91
A.2. Esquemático y símbolo de una puerta NAND con tecnología UMC 0.18 μm (CMOS)	92
A.3. Esquemático y símbolo de una puerta NOR con tecnología UMC 0.18 μm (CMOS)	92
A.4. Esquemático y símbolo del latch SR con tecnología UMC 0.18 μm (CMOS)	93
A.5. Esquemático y símbolo del bit "1" con tecnología UMC 0.18 μm (CMOS)	93
A.6. Esquemático y símbolo del bit "0" con tecnología UMC 0.18 μm (CMOS)	94
A.7. Esquemático y símbolo del level shifter de high a low con tecnología UMC 0.18 μm (CMOS)	94
A.8. Esquemático y símbolo del level shifter de low a high con tecnología UMC 0.18 μm (CMOS)	95
A.9. Esquemático y símbolo del módulo arbitrador con tecnología UMC 0.18 μm (CMOS)	96
A.10. Esquemático y símbolo del módulo arbitrador con capacidades parásitas con tecnología UMC 0.18 μm (CMOS)	97
A.11. Esquemático del módulo de la lógica de selección con tecnología UMC 0.18 μm (CMOS)	98

Índice de cuadros

2.1. Tabla de verdad del bloque lógica de selección	36
3.1. Tabla verdad de una puerta NAND de dos entradas	45
3.2. Tabla verdad de una puerta NOR de dos entradas	47
3.3. Tabla verdad de un latch SR con puertas NAND	50
3.4. Valores de las variables asignadas en ADE L para la simulación del circuito completo del sensor solar	66
4.1. Tabla resumen de las capacidades totales	79

Resumen

El objetivo de este proyecto es estudiar, simular y realizar el diseño y layout de la circuitería periférica y de control necesaria para procesar la información recibida por un sensor solar asíncrono en forma de matriz de píxeles a partir de una arquitectura dada.

Partiendo del diseño previo existente realizado de un sensor basado en eventos, diseño implementado con una lógica de 3.3 V, el objetivo de este trabajo es migrar el mismo a transistores de 1.8 V para reducir el consumo de la circuitería de arbitración y aumentar la velocidad. Además, se estudia como la tensión de alimentación puede reducirse por debajo de 1.8 V para disminuir el consumo de energía en diseños de prototipos de sensores solares futuros.

La metodología llevada a cabo ha sido, en primer lugar, el estudio de los tipos de sensores y el protocolo de comunicación AER. Posteriormente, un análisis detallado del funcionamiento de cada bloque que compone la periferia, seguido de un diseño y análisis de resultados mediante la simulación de esquemáticos, bancos de pruebas y simulaciones paramétricas en Cadence Virtuoso. Finalmente, se ha realizado el layout de cada uno de los bloques que componen la circuitería de periferia.

Mediante este trabajo se consigue estudiar la factibilidad y limitaciones de la reducción del consumo en sensores solares futuros. Dicha reducción conllevaría una gran mejora en el mercado de los nanosatélites (dispositivos donde el nicho de mercado para estos sensores es más prometedor) muy demandados en la actualidad.

Abreviaturas y siglas

ACS	Attitude Control System
ADC	Analog Digital Converter
AER	Address Event Representation
APS	Active Pixel Sensor
CMOS	Complementary Metal Oxide Semiconductor
DVS	Dynamic Vision Sensor
DRC	Design Rules Check
DRM	Design Rules Manual
EDA	Electronic Design Automation
FPGA	Field Programmable Gate Array
FOV	Field of Vision
IBM	International Business Machines
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
INTA	Instituto Nacional de Tecnología Aeroespacial
LVS	Layout versus Schematic
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
PEX	Parasitic EXtraction
PDK	Process Design Kit
TFS	Time - to - first spike
VTC	Voltage Transfer Characteristics

Capítulo 1

Introducción

El propósito de este capítulo será introducir la necesidad del uso de sensores solares y realizar una pequeña descripción de los tipos de sensores más utilizados en el mercado y sus características.

1.1. Historia de los dispositivos de navegación espacial

Desde el inicio de los tiempos, el ser humano ha sentido la necesidad de conocer las predicciones meteorológicas, conocer el universo en general. Esto se ha traducido en que uno de los desafíos más importantes cuando un satélite ha sido lanzado al espacio y situado en órbita alrededor de un cuerpo celeste ha sido proporcionarle la orientación y el control de seguimiento adecuados para que no pierda el rumbo y no se salga de la órbita.

Para que se pueda realizar el control de posicionamiento y orientación dinámicos de un satélite es imprescindible tomar como referencia alguna entidad espacial, conocida como Control de Actitud de Satélites (ACS). Este sistema proporciona al satélite conocimiento de la orientación angular del mismo respecto a un sistema de referencia, capacidad de predicción de la altitud o proyección en el tiempo de la orientación del satélite y control preciso de la altitud. En definitiva, permite que se dirija con agilidad el satélite, pero para ello se requieren de actuadores que ejerzan los momentos de fuerza adecuados para su reorientación y estabilización de acuerdo a algoritmos establecidos en tierra, una electrónica asociada entre estos actuadores y unos sensores. Estos últimos son sensores de actitud, encargados de medir la posición relativa del satélite y el cuerpo celeste alrededor del cual orbita. Para la fabricación de estos sensores existen diversas tecnologías sobre las

que han basado su funcionamiento en algún momento: medidores de flujo estabilizado, unidades de referencia vertical y sensores solares (en los que nos centraremos) entre otros muchos.

En el mercado actual existen numerosos fabricantes y centros de investigación o universidades que comercializan o investigan y desarrollan sensores solares con diferentes prestaciones, todos ellos para ser utilizados en diferentes entornos satelitales cumpliendo con las exigencias requeridas. Podemos encontrar un ejemplo en el año 2004, el Instituto Nacional de Tecnología Aeroespacial (INTA) puso en marcha el proyecto NANOSAT-1B que requería un sistema de posicionamiento satelital utilizando como referencia el sol [19]. Los requisitos eran que el dispositivo debía ser de pequeño tamaño, peso y altas prestaciones dadas las reducidas dimensiones del satélite. De estas exigencias surgió un sensor de luz solar llamado *Vector Sol*, encargado por el INTA a la Universidad de Sevilla, quien se ocupó del diseño de su electrónica y que encargó a su vez a la Universitat Politècnica de Catalunya la fabricación del dispositivo foto sensor [22].

En la actualidad los sensores solares se incluyen cada vez más en multitud de ámbitos como la electrónica de consumo, entornos industriales, militares, médicos, inteligencia artificial, etc. Debido a ello, cada vez hay una mayor demanda de los mismos. De las necesidades de lograr altas tasas en las capturas de datos y de facilitar el procesado de la información surgen los sensores asíncronos, los cuales utilizaremos en este trabajo. Poseen ventajas como proporcionar un menor consumo ancho de banda (solo se leen los píxeles iluminados), menor latencia (tiempo de respuesta del orden de milisegundos), menor consumo de energía y mayor facilidad de procesamiento de los datos leídos gracias al preprocesamiento de la información.

Las perspectivas del mercado actual para microsátélites son muy esperanzadoras ya que se está produciendo una expansión significativa. Tanto es así, que aproximadamente unos 7000 microsátélites se lanzarán en la próxima década, un aumento seis veces mayor respecto a la década anterior. Las mejoras de rendimiento y la miniaturización continua remodelan este mercado, ya que los clientes pueden elegir entre satélites más livianos con grandes capacidades o más grandes pero más capaces. Cada uno de los microsátélites que se va a lanzar en la próxima década llevará como mínimo cuatro sensores solares como se muestra en la siguiente imagen [34].

1.1. HISTORIA DE LOS DISPOSITIVOS DE NAVEGACIÓN ESPACIAL3

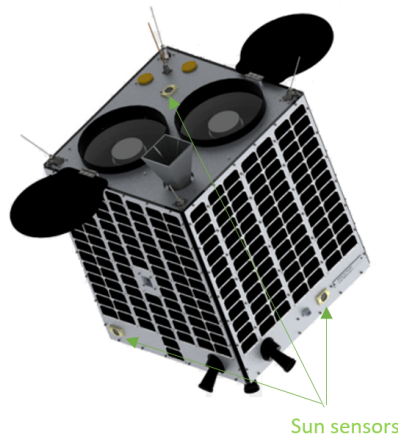


Figura 1.1: Sensores solares en un microsatélite [34]

Los sensores solares se basan en el efecto fotoeléctrico, este efecto fue descubierto por Heinrich Hertz en 1887, mientras realizaba diversos experimentos observó que el arco que salta entre dos electrodos conectados a alta tensión alcanza distancias mayores cuando se ilumina con luz ultravioleta que cuando se deja en la oscuridad. Un Año después, W. Hallwachs, un estudiante de H. Hertz, repitió el experimento. Al incidir una luz ultravioleta sobre una placa de zinc cargada negativamente, ésta se descargaba, perdía su carga, mientras que no afectaba a un cuerpo con carga positiva. Observó también que, si la placa estaba inicialmente sin carga, adquiría carga positiva. En 1889, J. Elster y H. Geitel descubrieron que en algunos metales alcalinos este efecto sí que se producía con luz visible. Además, notaron que el número de partículas emitidas era directamente proporcional a la intensidad que tenía la luz ultravioleta.

En 1900, J. J. Thomson logró identificar las partículas emitidas como electrones gracias a su experimento cuya conclusión era que las partículas que se emitían tenían la misma relación carga-masa que los rayos catódicos, eran electrones. Por este y otros trabajos recibió el premio Nobel en 1906. En 1902, P. von Lenard estudió cómo variaba la energía de salida de los electrones al variar la intensidad. Descubrió que la energía máxima de salida de los electrones era un valor fijo que no dependía de la intensidad de la luz, sino del color de la misma, es decir, de su longitud de onda o frecuencia. Esta energía aumentaba según aumentaba la frecuencia y, por lo tanto, disminuía según aumentaba la longitud de onda. En 1905 recibió el premio Nobel por este trabajo [33].

Posteriormente, Einstein le proporcionó el significado correcto en 1905, en el que decía que un haz de luz se compone de paquetes de energía llamados cuantos de luz o fotones. Cuando el fotón choca contra un electrón en la superficie de un metal, el fotón le puede transmitir energía al electrón, con la cual podría este escapar de la superficie del metal. Además, con este estudio, se planteó la dualidad en el comportamiento de la luz como onda y partícula a la vez.

En 1912, Arthur Lewelyn Hughes, Owes William Richardson y Karl Taylor Compton demostraron experimentalmente el crecimiento lineal con la frecuencia de la energía cinética de los electrones fotoemitidos. Pero fue Robert Andrews Millikan quien midió, en 1916, tanto la frecuencia de la luz como la energía de los electrones emitidos, y obtuvo un valor de la constante de Planck. El valor que conocemos actualmente viene del que obtuvo Robert y del obtenido en experimentos de radiación de cuerpo negro realizados posteriormente [33].

Las restricciones de los nanosatélites en tamaño y peso son cada vez más estrictas, esto se traduce en que los dispositivos que los componen tienen que adaptarse de manera que se requiere una menor generación de potencia para optimizar el consumo. En este mercado este tipo de sensores son muy importantes y tienen un futuro muy prometedor. De la necesidad de la reducción del consumo en este tipo de sensores surge este trabajo.

Una vez se ha introducido la necesidad de incorporar sensores solares así como el efecto en el que se basan, se procede a explicar como identificar la posición del Sol mediante sus coordenadas esféricas.

1.2. La posición del sol en coordenadas esféricas

En el espacio de tres dimensiones, un punto se puede representar por sus coordenadas esféricas (r, θ, ϕ) , donde r es la distancia radial, θ el ángulo polar y ϕ el ángulo azimutal. La siguiente ecuación corresponde a la ecuación de un punto en términos de coordenadas rectangulares:

$$r = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{Bmatrix} r \operatorname{sen} \theta \cos \phi \\ r \operatorname{sen} \theta \operatorname{sen} \phi \\ r \cos \theta \end{Bmatrix} \quad (1.1)$$

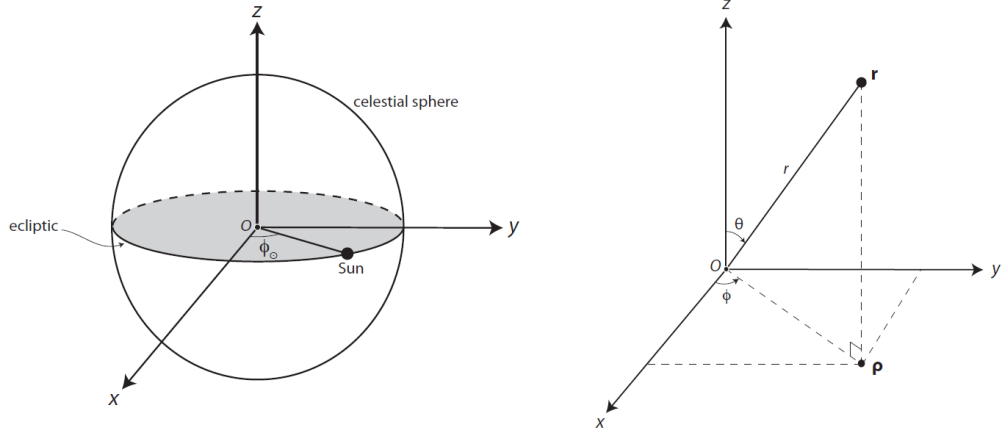


Figura 1.2: Punto r en el espacio tridimensional en coordenadas esféricas [11]

Para la caracterización de la posición de una esfera celeste (planetas, estrellas y cuerpos del espacio) existen posibilidades que simplifican dicha ecuación, si se supone que el radio de la esfera (r) es mucho mayor que el de la Tierra, se puede considerar que dicho cuerpo está centrado en la posición de un observador que se encontrase en la superficie de la Tierra. Es por ello que se puede considerar que el radio de la esfera celeste es igual a 1 (en unidades indeterminadas). Usaremos el símbolo \odot (el símbolo astronómico del sol) para indicar vectores o coordenadas referidas a la posición del sol.

Se toma como referencia el sistema eclíptico en el que el sistema de referencia tiene origen en la Tierra y cuyo plano xy es paralelo al plano eclíptico. En este sistema de referencia, el ángulo azimutal varía con el tiempo y el ángulo polar posee un valor fijo $\theta_{\odot} = \frac{\pi}{2}$.

Si la órbita de la Tierra fuera perfectamente circular, ϕ_{\odot} , aumentaría a una velocidad constante. Esto no se cumple realmente puesto que la órbita de la Tierra es elíptica, pero se considerará así por simplicidad. Por lo tanto se expresa el ángulo azimutal como una función del tiempo. La posición del sol en el sistema de referencia eclíptico se puede expresar del siguiente modo:

$$r_{\odot}(t) = \begin{Bmatrix} \cos\phi_{\odot}(t) \\ \sin\phi_{\odot}(t) \\ 0 \end{Bmatrix} \quad (1.2)$$

Esto es aplicable a un sensor solar que órbita alrededor de la Tierra, sin

olvidar que dicha posición estará referida al centroide del sol.

1.3. Sensores solares

Dado que el objetivo de este trabajo es diseñar e implementar la lógica de lectura de un sensor solar para sistemas de navegación espacial, se va a comenzar realizando un breve descripción de los tipos de sensores solares que existen.

Un sensor solar es un dispositivo utilizado para detectar la posición del sol, para el control de altitud en satélites, sistemas de seguimiento solar en paneles solares o en estaciones meteorológicas, actualización del giróscopo, etc. Es por ello que son ampliamente utilizados en energías renovables y sistemas de navegación espacial (microsatélites). En la actualidad hay varios tipos de sensores solares que se diferencian en sus características tecnológicas y rendimiento, sensores analógicos, de presencia solar y digitales. Los sensores solares analógicos proporcionan señales de salida como una función continua del ángulo solar. Los sensores de presencia solar proporcionan una salida binaria para indicar si el sol está dentro del campo de visión. Los sensores digitales proporcionan una salida discreta codificada que se mide mediante la función ángulo solar.

1.3.1. Sensores solares analógicos

Un sensor solar analógico es un dispositivo compuesto por varios dispositivos electrónicos, el módulo más importante es el sensor (formado normalmente por uno o más fotodiodos) encargado de convertir la información proporcionada por los rayos solares en corrientes eléctricas o tensiones y sobre este se suspende una membrana a una distancia conocida.

Estos sensores funcionan del siguiente modo, poseen una ventana translúcida que proporciona un punto de luz que se desplaza sobre el elemento sensor con el cambio de altitud del Sol. La implementación más común es un fotodiodo de cuatro cuadrantes o dos pares de fotodiodos colocados ortogonalmente entre sí para poder medir el ángulo de incidencia de los rayos solares en ambas direcciones [11]. A continuación, se muestra el esquema de este tipo de sensor [6]:

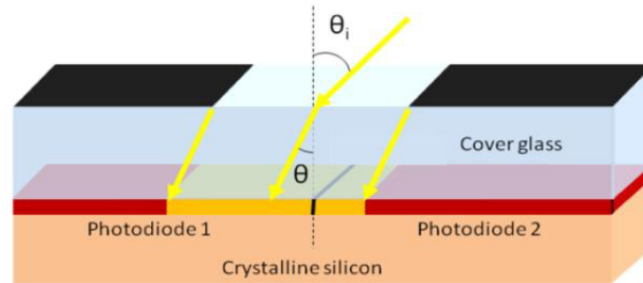


Figura 1.3: Estructura del elemento sensor de un sensor solar analógico [4]

Dos fotodiodos son implementados en un sustrato de silicio con dopado de tipo p. El sustrato sirve como sistema óptico con una apertura en un recubrimiento absorbente negro en la parte superior y un revestimiento antirreflectante en la parte inferior. El principio de funcionamiento es el siguiente: los rayos del sol llegan a los fotodiodos atravesando una cubierta de vidrio situada bajo una ventana y generando corrientes. Es por ello que el agujero define el punto de luz y la fotocorriente se genera en áreas iluminadas, mediante esta se puede obtener el ángulo de los rayos incidentes.

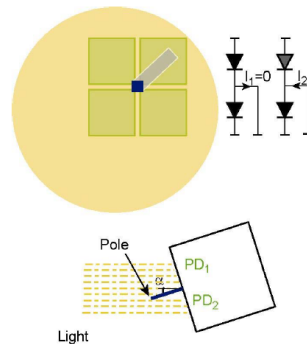


Figura 1.4: Indicencia en un sensor solar analógico de cuatro fotodiodos [10]

Cabe destacar que las señales emitidas por los fotodiodos tienen una potencia muy pequeña, por ello en la mayoría de los casos el uso de estos sensores exige incluir una o incluso varias etapas de amplificación. Este tipo de sensores son simples y tienen una rápida operación pero son muy sensibles al ruido.

1.3.2. Sensores digitales

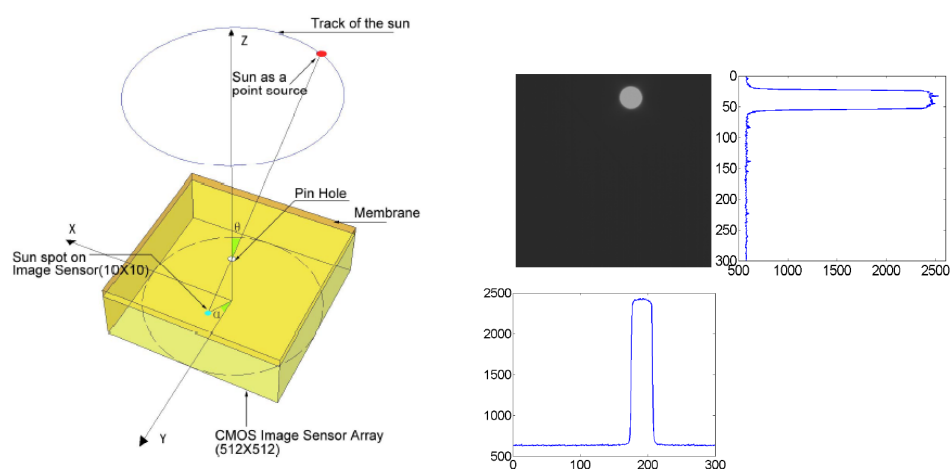
Los sensores digitales son un segmento de mercado actualmente mucho más grande que los sensores analógicos. Hay que tener en cuenta que en el mercado muchas veces estos sensores se anuncian como digitales pero realmente son sensores analógicos con una interfaz digital.

Los sensores solares digitales están compuestos por una matriz de píxeles y circuitería periférica de control y lectura (convertidores analógicos a digital, generadores y controladores de señales, circuitos de procesamiento de señales digitales, etc.) [29]. En general, los sensores solares digitales utilizan un sensor de visión basado en *frames* con tecnología APS, es decir, un sensor de imagen y una óptica dedicada [7]. Este tipo de sensor consiste principalmente en una placa de circuito impreso y la máscara con una ranura. Los *frames* contribuyen a restringir la distancia entre el plano de la máscara y el plano focal, así como la sujeción de las unidades internas del sensor solar; la placa de circuito impresa posee la electrónica necesaria para la conversión de la óptica a la electrónica y realiza el procesamiento de datos y la máscara forma la imagen del rayo incidente.

El funcionamiento es el siguiente, se lee la matriz de píxeles, una vez leída se calcula el centroide de la región iluminada y mediante relaciones trigonométricas se calcula el ángulo de orientación sobre la superficie de la esfera (azimut) y la elevación del sol. A continuación, se muestra la imagen de un sensor solar digital fabricado y comercializado por la empresa sevillana Solar MEMS Technologies [32].



Figura 1.5: Sensor Solar digital fabricado por Solar MEMS [22]



(a) Esquema del sistema del sensor solar (b) Determinación del centroide de la región iluminada del s *tipo p*

Figura 1.6: Definición de un sensor solar digital [29]

Si se analiza el esquema de un sensor solar digital, tal y como se ha descrito anteriormente, se pueden distinguir las siguientes partes:

- **APS:** compuesta por la matriz de píxeles y circuitería analógica.
- **ADC:** convertidor analógico digital.
- **Digital Processing:** circuitería digital encargada del procesado de los datos que proporciona el ADC, cálculo del centroide de acuerdo con un algoritmo específico.

A continuación se explicarán los sensores de imágenes basados en *frames* para entender el comportamiento de los sensores solares digitales.

1.3.2.1. Sensores de imagen

Los sensores de imagen son dispositivos usados para detectar la posición del sol, convertir la luz incidente en una señal eléctrica para su posterior procesamiento. En un sensor de imagen basado en *frames*, la *frame* es una matriz bidimensional que contiene información sobre la escena visual (normalmente intensidad de luz o color). Dichos sensores siempre proporcionan un flujo de datos de salida continuo que puede ser muy redundante si la escena visual no cambia. Esta matriz está compuesta de píxeles, cada píxel tiene un fotodiodo que absorbe la luz que incide y genera una señal eléctrica. Si

se extrae la información de todos los píxeles y formamos la matriz se puede ser capaz de reconstruir la imagen que fue proyectada sobre el sensor.

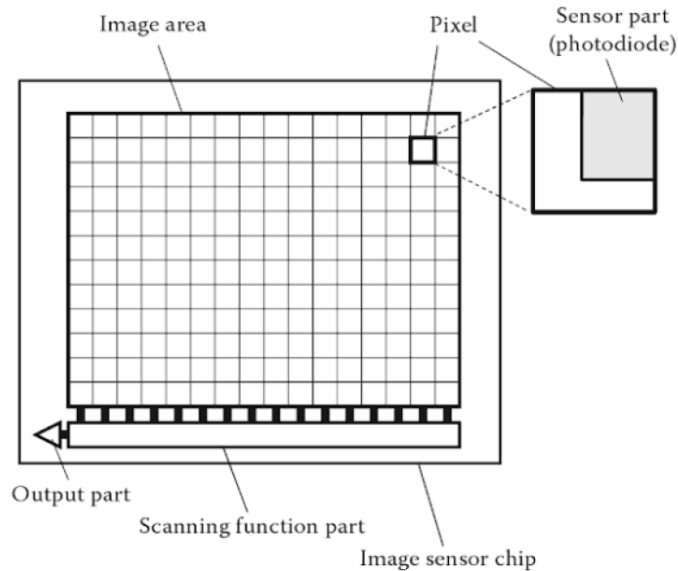


Figura 1.7: Configuración básica de un sensor de imagen [4]

En la imagen 1.7 se muestra un sensor imagen en la que cada rectángulo corresponde a un píxel y tiene un nivel de intensidad dependiente de la intensidad de luz incidente. Finalmente, la imagen óptica es una función continua que tiene un valor determinado para el todo el conjunto de puntos contenidos en el plano.

Los sensores de visión convencionales basados en *frames* funcionan transmitiendo matrices de dos dimensiones en las que todos los píxeles tienen que ser leídos para reconstruir un *frame*. Los *frames* se transmiten de manera síncrona, incluso si no hay cambios en la escena visual, lo que conlleva que el flujo de datos de salida puede ser muy alto si los *frames* tienen una resolución alta. Sin embargo, si la detección es asíncrona el problema anteriormente descrito desaparece. La detección es asíncrona y los píxeles pueden enviar información en cualquier instante. Dicha información es enviada en forma de pulsos que indican que se ha detectado información relevante. Los sensores basados en eventos transmiten las coordenadas (x, y) del píxel que ha emitido el pulso. Los píxeles que no detecten ninguna información relevante no emiten pulso ahorrando de esta manera ancho de banda y consumo de potencia [7].

1.3.2.2. Unión pn, efecto fotoeléctrico

El efecto fotoeléctrico es la formación y liberación de partículas eléctricamente cargadas que se produce en la materia cuando es irradiada con una radiación electromagnética (luz visible o ultravioleta, en general).

El efecto fotoeléctrico es un fenómeno, según el cual, cuando hacemos incidir una luz de determinada frecuencia sobre un metal o un semiconductor, éste liberará electrones de su superficie, dando lugar a una corriente eléctrica.

Se considera un esquema simplificado de un fotodiodo, una unión pn para su estudio. El material tipo P tiene una concentración de huecos mucho mayor que de electrones y al contrario ocurre en el material tipo N. Se crea así una corriente de difusión de electrones de la región N hacia la P. Pero además, la difusión de estos portadores de su región inicial a la otra deja al descubierto algunos iones fijos en la red cristalina, iones aceptores negativos en la región P e iones donadores positivos en la región N. Dicha zona es llamada zona de agotamiento o región espacial de carga y es una región en que no existen ni electrones ni huecos pero contiene átomos donadores ionizados positivamente a un lado y átomos aceptores ionizados negativamente al otro lado. En consecuencia, se forma una zona de con cargas negativas fijas en la cara P de la unión y una zona con cargas positivas fijas en la cara N. La presencia de estas cargas fijas da lugar a la aparición de un fuerte campo eléctrico cuyas líneas de campo se dirigen desde la zona N hacia la zona P. El estado de equilibrio se alcanza cuando la fuerza por difusión igual a la fuerza del campo eléctrico. [33]

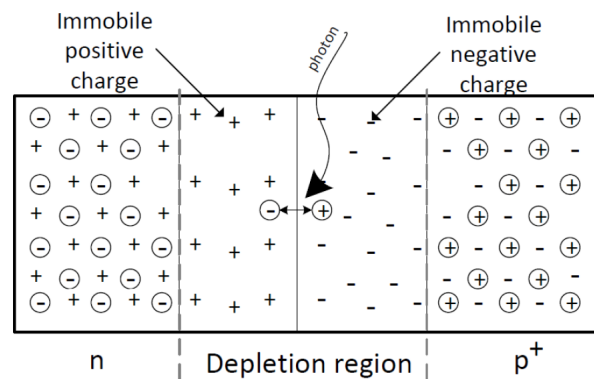


Figura 1.8: Modelo simplificado de un fotodiodo [4]

La curva característica de un fotodiodo se muestra a continuación:

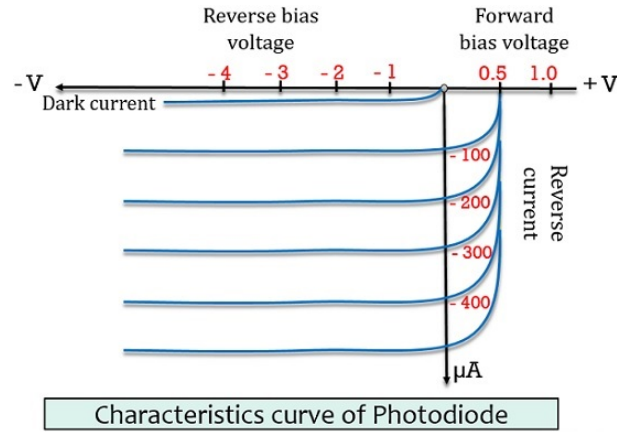


Figura 1.9: Característica I-V de un fotodiodo

En la figura 1.9 se muestra la característica I-V de un fotodiodo, se puede observar que está diseñado para ser usado en polarización inversa (cuadrante III), aún sin luz genera una pequeña corriente conocida como corriente de oscuridad, la cual aumenta con la temperatura. Dicha corriente se considera ruido y limita la mínima corriente que puede detectar, pero cabe destacar que este valor suele ser muy pequeño.

Es importante destacar que para cada material existe una región del espectro electromagnético dentro del cual los fotodiodos de dicho material pueden operar.

1.3.3. Sensores solares asíncronos

Si se atiende a la clasificación de los sensores teniendo en cuenta el modo de lectura podemos encontrar **sensores solares síncronos** que tratan de obtener una representación lo más fiel a la realidad posible y **sensores solares asíncronos**.

Los **sensores solares asíncronos** se nutren de sensores de imágenes asíncronos y surgen de la necesidad de imitar el comportamiento de los sistemas de visión en vertebrados. Los datos de la salida de las retinas son masivos y procesados por el cerebro, que toma las decisiones y envía la información en forma de eventos digitales que ocurren en tiempo continuo. Los cálculos de preprocesamiento de la información, control de ganancia y reducción de la redundancia de datos son la base para intentar emular este

sistema biológico asíncrono. La diferencia principal entre los píxeles síncronos y los asíncronos son que en los primeros el propósito era medir una señal analógica que corresponde al nivel de iluminación, y, en los asíncronos se activan las señales de salida correspondientes a los distintos estímulos de iluminación, de forma que se puede lograr una alta tasa de captura de datos, un alto rango dinámico y un preprocesamiento de la información para facilitar así su posterior procesado. En definitiva, sensor solar asíncrono está formado por una retina de tipo octopus, cuyos píxeles envían eventos con una frecuencia proporcional a la iluminación y una óptica dedica para el procesamiento de la información recibida. [14]

Los primeros sensores creados con esta finalidad no consiguieron ser completamente asíncronos, por ejemplo, en 1994 Mahowald, una alumna de Mead (estudió la implementación integrada de píxeles que emularan el comportamiento de una neurona [20]), consiguió elaborar una tesis que proponía el primer sensor bio-inspirado [18], pero dicho sensor poseía una lectura síncrona. No fue hasta que Silvilotti, Boahen y otros autores presentaron la implementación hardware de los esquemas de representación de dirección de eventos (AER) hasta que se consiguió el procesamiento asíncrono. En el protocolo de comunicación AER un estímulo causa una petición que escribe en el bus para que la periferia se encargue de leer y procesar la información.

A lo largo de la historia se han ido desarrollando diferentes tipos de sensores asíncronos que se explican en las siguientes secciones.

1.3.3.1. Sensores DVS

En 2008, Lichtsteiner [17] propuso el primer sensor DVS (*Dynamic Vision Sensor*), capaz de detectar el contraste temporal de cada píxel. Este píxel contaba con dos salidas que emitían un pulso cuando la iluminación aumentaba o disminuía. Posteriormente, un arbitrador se encargaba de gestionar los pulsos para implementar un protocolo de comunicación AER. El resultado era un sensor que en lugar de leer la matriz completa de píxeles, solo leía los píxeles que variaban su iluminación respecto a la medida anterior, además de prácticamente duplicar el rango dinámico de un píxel APS convencional, a costa de aumentar la complejidad.

Contrariamente a las cámaras estándar, donde los píxeles son adquiridos en intervalos regulares de tiempo, los sensores de visión basados en eventos como los sensores de visión dinámicos (DVS) poseen píxeles asíncronos. Cada píxel del DVS desencadena un evento cuando se produce una acción, es

decir, la salida es un flujo de eventos.

Supongamos que vamos a medir el brillo, los sensores de visión basados en eventos como los DVS tienen píxeles independientes que disparan eventos con cambios de brillo relativos locales en tiempo continuo. Son muy similares a los sensores de imagen tradicionales pero con una ventaja adicional, su alto rango dinámico [21].

Una modificación del píxel de Lichsteiner fue propuesto por Brandli, este cambio consistía en añadir la circuitería necesaria para incorporar la posibilidad de poder utilizarse como píxel APS, denominando esta estructura como sensor de visión dinámico y activo (DAVIS).

1.3.3.2. Sensores tipo octopus

En 2003, Culurciello [3] propuso el primer sensor de tipo octopus, un píxel formado por un fotodiodo con una carga y un elemento comparador que conmuta la salida si la tensión del fotodiodo supera un umbral definido, generando un tren de pulsos proporcional al nivel de iluminación. Los sensores de tipo octopus permiten codificar en frecuencia o ancho de pulso los niveles de iluminación para así realizar una lectura asíncrona de la imagen.

A lo largo de la historia se pueden encontrar distintas implementaciones de este tipo de sensores. Cabe destacar la implementación de Juan Antonio Leñero [13] que propone un píxel con capacidad de autoreseteo y que genere un tren de pulsos cuya frecuencia sea proporcional al nivel de iluminación. A continuación, se muestra un esquema del mismo:

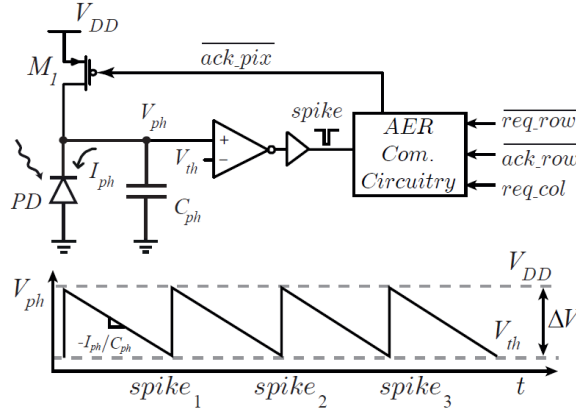


Figura 1.10: Implementación de un píxel capaz de generar pulsos en función de la iluminación [13].

En la figura 1.10 se puede observar que la fotocorriente generada se identifica con I_{ph} , C_{ph} es la capacidad equivalente en el cátodo del fotodiodo y V_{ph} la tensión en el nodo del cátodo. El funcionamiento es el siguiente, en primer lugar se precarga la tensión del fotodiodo a V_{DD} (tensión de alimentación) y posteriormente se descarga con una pendiente proporcional a la iluminación hasta alcanzar un valor umbral V_{th} y se genera un pulso que realice la petición y resetee la tensión del fotodiodo [13]. De este modo la frecuencia del tren de pulsos será la siguiente:

$$f \simeq \frac{I_{ph}}{(V_{DD} - V_{th})C_{ph}} \quad (1.3)$$

1.3.3.3. Sensores Time-to-First-Spike

Un sensor Time-to-first-spike (TFS) es diseñado de manera que cada píxel emite un solo pulso digital cuya sincronización codifica la iluminación del píxel. En lugar de medir la frecuencia de los pulsos, se mide el tiempo desde que se inicia la operación de sensado hasta que se recibe el primer pulso. Con esto se consigue reducir el consumo del sensor así como un aumento del rango dinámico o, dicho de otro modo, un aumento de la relación entre la señal máxima medible y el ruido en condiciones de oscuridad.

Muchos científicos afirman que la información más útil de la retina está contenida en el primer pico después del inicio del estímulo visual, pero los tiempos de reacción son tan cortos que no hay tiempo suficiente para procesar más de un pico de cada neurona por etapa de procesamiento [9]. Es

por ello que surgen los sensores TFS, para superar las deficiencias de diseños anteriores, basados en el tiempo, la iluminancia se transforma en un evento de pulso que solo puede ocurrir una vez por píxel por *frame* en el dominio del tiempo. Los circuitos de lectura asíncronos emiten la dirección de píxel correspondiente después de solicitar acceso al bus de salida mediante el envío de un pulso. Una vez recopilada toda la información temporal, se puede reconstruir la escena dado el tiempo de disparo de cada píxel. Dado que solo se emite un pico de cada píxel por fotograma, el generador de imágenes TFS consume menos energía, menos ancho de banda y requiere una memoria de fotogramas adicional más pequeña que otros lectores de imágenes.

El generador de imágenes TFS utiliza la codificación temporal. En lugar de leer el voltaje analógico en cada fotodiodo en un tiempo de exposición predeterminado, hay un comparador dentro de cada píxel.

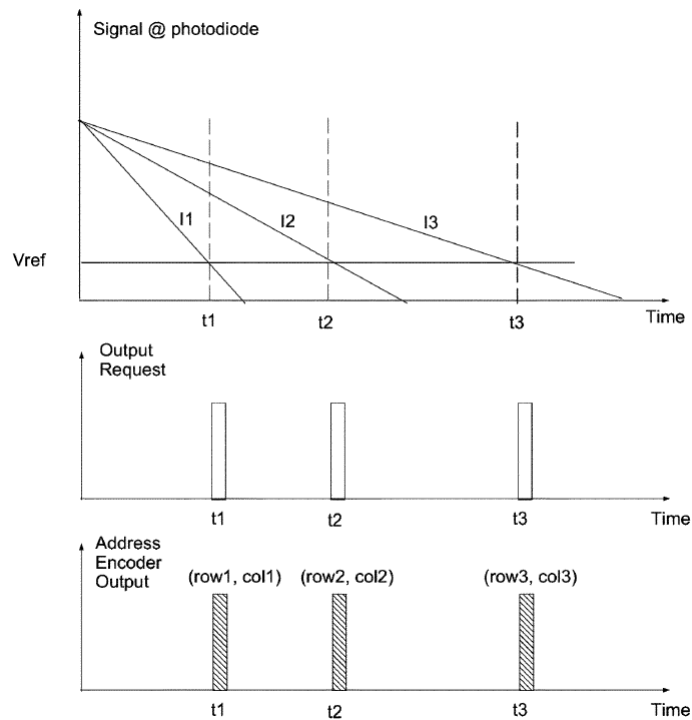


Figura 1.11: Esquema del generador de imágenes TFS en modo de imagen fija [9].

Cuando el voltaje en un fotodiodo cae por debajo de un voltaje de referencia global, el comparador se invierte y el píxel genera un pulso (es decir, se ha disparado). Tal y como se observa en la figura 1.11, se tienen tres pí-

xeles ubicados en las direcciones (fila1, col1), (fila2, col2) y (fila3, col3), con sus correspondientes fotocorriente de I1, I2 e I3. Una vez que se ha disparado un píxel, se deshabilita para el resto del *frame* después de que se emite su dirección. El momento en el que se lee la dirección de un píxel, esta representa la iluminancia del píxel. Para este esquema, en lugar de un ADC, que es necesario para los lectores de imágenes convencionales, se puede utilizar un simple contador digital que registre el tiempo en que se dispara cada píxel.

Dado que la señal se muestrea en el dominio del tiempo, el rango dinámico se puede expresar como la relación entre los tiempos de disparo más largos y más cortos. [9]

Para finalizar, cabe destacar que otra propuesta fue realizar sensores híbridos que aunasen funciones de los sensores DVS y octopus para que fueran capaces de detectar el contraste espacial o el color de una escena visual.

1.3.4. Protocolo de comunicación AER

El protocolo de comunicación AER surge de la necesidad de comunicar la información de forma asíncrona, forma que se asemeja a la del sistema nervioso. A medida que los ingenieros intentaban implementar sistemas que estuvieran organizados como el sistema nervioso, se llegaba a la conclusión que la forma en que el cerebro comunica las señales internas era un gran obstáculo para la electrónica, pero a pesar de ello poseía numerosas ventajas (nombradas en la sección anterior) y es por ello que valía la pena la investigación.

En 1991 cuando Silvilotti [26], Boahen [1] [2] y otros autores presentaron la implementación hardware de los esquemas de representación de dirección de eventos o *Address Event Representation (AER)*, estableciendo un protocolo de comunicación en el que un estímulo causa una petición que escribe en un bus compartido. Normalmente la información es la dirección horizontal y vertical del píxel, así la periferia lee y procesa la información compartida. Este protocolo utiliza principios mixtos analógicos y digitales y aprovecha la modulación de densidad de pulsos para codificar la información [23].

Tal y como se ha descrito en secciones anteriores y teniendo en cuenta que el píxel utilizado será un píxel de tipo octopus se debe considerar que cuando dos o más píxeles independientes tratan de transmitir información de forma simultánea es necesario establecer un mecanismo que garantice que

no se producirán colisiones entre dichos píxeles. Para ello se va a hacer uso del protocolo de comunicación AER. Cada vez que el píxel se dispara de forma asíncrona, el sensor solicita acceso al bus AER compartido. Cuando el bus está listo, las coordenadas de píxeles se envían a través del bus. En caso de que haya varios píxeles tratando de enviar información simultáneamente, dichos píxeles esperarán hasta que el acceso al bus sea otorgado para enviar sus direcciones fuera del chip. En este caso la lógica de comunicación AER maneja la misma para transmitir un evento con las coordenadas (x, y) del píxel que se ha disparado [12].

Se muestra a continuación un diagrama con las señales del canal de comunicación y de establecimiento de comunicación:

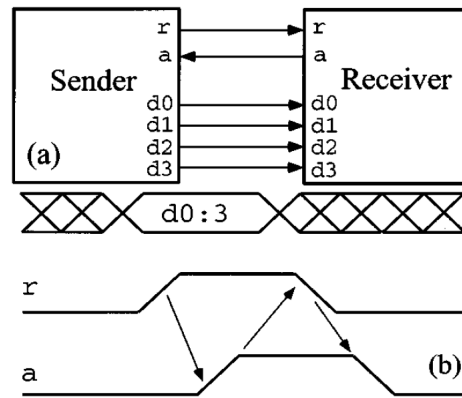


Figura 1.12: Señales del canal de comunicación y temporización. (a) bus de datos (d_0, \dots, d_3) y señales de establecimiento de comunicación (r, a). (b) Diagrama temporal [1]

En la Figura 1.12 aparecen dos elementos, el emisor (*Sender* o sensor solar) y el receptor (*Receiver* o circuito que procesa los datos), el bus de datos (d_0, \dots, d_3) y las señales de establecimiento de la comunicación (solicitud o *request* y admisión o *acknowledge*). En la parte de abajo de la figura se muestra la siguiente secuencia: el emisor inicia la carga de datos en el bus, cuando la señal r sube el receptor los admite y cuando termina de leer los datos sube la señal a [1].

1.4. Objetivo del proyecto

El objetivo de este proyecto, como ya se ha introducido, es estudiar, simular y realizar el diseño y layout de la circuitería periférica y de control

necesaria para procesar la información recibida por un sensor solar asíncrono en forma de matriz de píxeles a partir de una arquitectura dada.

En el mercado actual, el diseño de un sensor solar asíncrono es conocido, pero la demanda de una reducción de consumo en el campo de los nanosatélites favorece a que sea necesario el estudio y mejora continua en este campo. Por ello, el objetivo de este trabajo es migrar el diseño existente de transistores de 3.3 V a 1.8 V para reducir el consumo de la circuitería de arbitración y aumentar la velocidad. Con esta reducción de la alimentación se consigue que las corrientes bajen, los tiempos de carga y descarga de las capacidades parásitas aumenten y la velocidad de conmutación de las puertas lógicas también, lográndose así el propósito inicial.

Mediante este estudio se consigue observar como puede lograrse la mejora deseada en el consumo. Sin embargo, ¿hasta que valor de tensión de alimentación el funcionamiento es correcto? Pues bien, para ello se ha analizado como afecta dicha disminución al funcionamiento del circuito. Con este análisis se pueden obtener las limitaciones y rangos de funcionamiento del circuito.

Después de la verificación del funcionamiento del diseño y como último objetivo se ha realizado el layout de la circuitería de periferia (nos permite hacernos una primera idea del tamaño y del área necesaria) y el posterior post-layout con parásitos capacitivos.

1.5. Motivaciones para desarrollar el sistema propuesto

Cuando estaba estudiando el Grado en Ingeniería Electrónica Industrial, mis intereses se enfocaron al diseño y estudio de circuitos electrónicos mediante simuladores. Estudiar la variabilidad de un circuito es un tema de gran interés para poder conocer cómo funciona, las ventajas y las limitaciones del mismo.

Descubrir la cantidad de posibilidades que posee Cadence Virtuoso para la simulación y análisis de circuitos, donde los retrasos son muy importantes, es un reto. Por ejemplo, estudiar con qué herramienta de Cadence se puede realizar un análisis que nos permita obtener las distintas señales de salida para distintos niveles de la tensión de alimentación, sin tener que realizarlos uno a uno.

Otra de las motivaciones de este trabajo será aprender a realizar un layout óptimo teniendo en cuenta las restricciones de diseño, las cuales favorecen a mejorar la técnica de realización del mismo. Además lograr un diseño acorde a las DRC y con un LVS correcto supone una motivación añadida dada la complejidad del circuito.

Capítulo 2

Descripción del sistema

Una vez se ha realizado una pequeña introducción a los sensores solares y la necesidad de investigación en este campo, se procede a describir en detalle el sistema bajo estudio. Partiendo del diseño del píxel realizado por Antonio de la Calle [4], el cuál se describirá brevemente en esta sección, se continua con una explicación detallada del funcionamiento de cada uno de los bloques que componen la periferia, tema principal sobre el que se desarrolla este trabajo.

2.1. Descripción general del sistema

En primer lugar se muestra un diagrama de bloques del sistema bajo estudio:

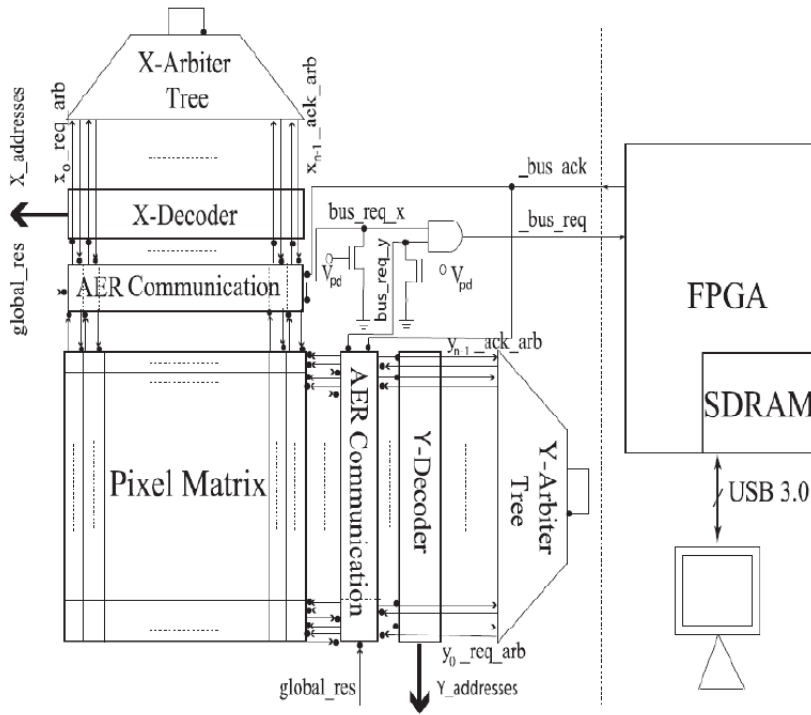


Figura 2.1: Diagrama de bloques del sistema. [4]

En la Figura 2.1 se puede observar en la parte central una matriz de píxeles, tras esta matriz, la periferia o lógica asíncrona y en la parte exterior una FPGA. La matriz de píxeles actúa como emisor, la periferia es la lógica asíncrona utilizada para establecer la comunicación entre la matriz de píxeles y la FPGA, que es el receptor.

El bloque principal de la periferia son los arbitrades, estos bloques están diseñados específicamente para resolver las colisiones que se producen cuando dos o más píxeles solicitan enviar su localización simultáneamente, deben esperar a que se les otorgue acceso al bus para transmitir su información fuera del chip. El arbitrador que se utilizará en este diseño otorga siempre la misma prioridad a sus entradas cuando todas están activas, solo una de las peticiones tendrá acceso al bus o lo que es lo mismo, sólo un píxel podrá enviar su dirección a la FPGA. En el diseño que se mostrará

posteriormente el arbitraje se realizará en dos pasos: primero se arbitran las peticiones de filas y después las columnas. Es así como un píxel podrá comunicar sus coordenadas a la FPGA [14].

El funcionamiento general del sistema es el siguiente: cada vez que el píxel emite un pulso, el sensor solicita acceso para compartir la información y las coordenadas del píxel son enviadas a través del bus. El receptor almacena la dirección del píxel que ha emitido su pulso e indica que ha recibido la información, después las coordenadas del píxel se eliminan del bus compartido para dejarlo disponible para el próximo evento.

A continuación se describirá el funcionamiento de cada uno de los bloques que componen el sistema.

2.1.1. Pixel

Para este trabajo se utilizará el píxel creado por Antonio de Calle Martos en su Trabajo Fin de Máster [4]. El píxel utilizado es un píxel de tipo octopus desarrollado en la tecnología UMC 0.18 μm (CMOS) para un sensor solar basado en eventos y que implementa el protocolo AER de comunicación. El funcionamiento es el descrito en el capítulo anterior, cada vez que un píxel emite un pulso, el sensor solicita acceso para poder enviar la información de las coordenadas, posteriormente, el receptor almacena dichas coordenadas e indica que ha recibido la información y vuelve a dejar el bus libre para poder leer el siguiente evento.

Para una mejor optimización, sobre el sensor se coloca normalmente una cubierta con un orificio, de tal forma que cuando se tenga un sensor debidamente orientado, el disco solar se proyecta sobre un número reducido de píxeles que enviaran eventos fuera del chip. Cada píxel puede dispararse una vez como máximo. Normalmente el píxel más iluminado (el primero que se dispara) está muy cerca del centroide de la región iluminada. En función de la posición del sol, la proyección del disco solar sobre la matriz de píxeles estará localizada en un lugar diferente. Con ello se consigue que los píxeles operen de manera asíncrona y muy rápida, este sensor reúne las ventajas tanto de los sensores analógicos como de los digitales omitiendo la redundancia de datos, la baja velocidad o el alto consumo de potencia.

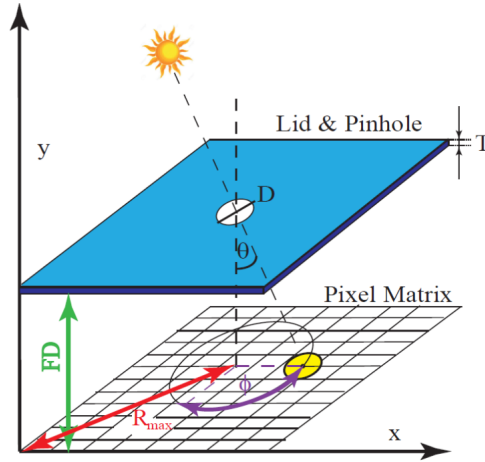


Figura 2.2: Implementación del sistema formado por la matriz de píxeles y la cubierta con el orificio. [4]

El centroide del grupo de píxeles iluminado, (x, y) , será utilizado para calcular la posición del sol (θ, ϕ) , donde θ es la latitud solar y ϕ el ángulo azimutal. La latitud viene dada por:

$$\theta = \tan^{-1} \frac{\sqrt{W(x - x_c)^2 + L(y - y_c)^2}}{FD} \quad (2.1)$$

Donde (x_c, y_c) es el centroide de la ROI cuando $\theta = 0^\circ$. W y L son la anchura y longitud del píxel, respectivamente y FD es la distancia focal, es decir, la distancia que hay entre el sensor y la cubierta perforada. El ángulo azimutal ϕ viene dado por:

$$\phi = \tan^{-1} \left[\frac{W(x - x_c)}{L(y - y_c)} \right] \quad (2.2)$$

Conociendo el valor de estos parámetros y el número de píxeles es posible determinar el campo de visión (FOV, Field of Vision) del sensor. Se puede consultar dicho cálculo en la siguiente referencia [4].

El píxel que diseñó el compañero Antonio forma parte de un sensor solar con operación TFS (*Time-to-first spike*). La operación del mismo comienza con una señal de reseteo global después de la cual los píxeles podrán emitir como mucho un pulso en caso de que detecten información relevante. De esta

manera se consigue que los píxeles que tengan la información más relevante emitan primero el pulso. En el caso de los sensores solares, emitirá primero el píxel que genere más fotocorriente. La principal ventaja es que el modo de operación TFS proporciona un flujo de datos reducido en comparación con los sensores convencionales.

A continuación se muestra un esquema del píxel en cuestión que está diseñado para trabajar a alta velocidad, con muy poco consumo de potencia y ancho de banda.

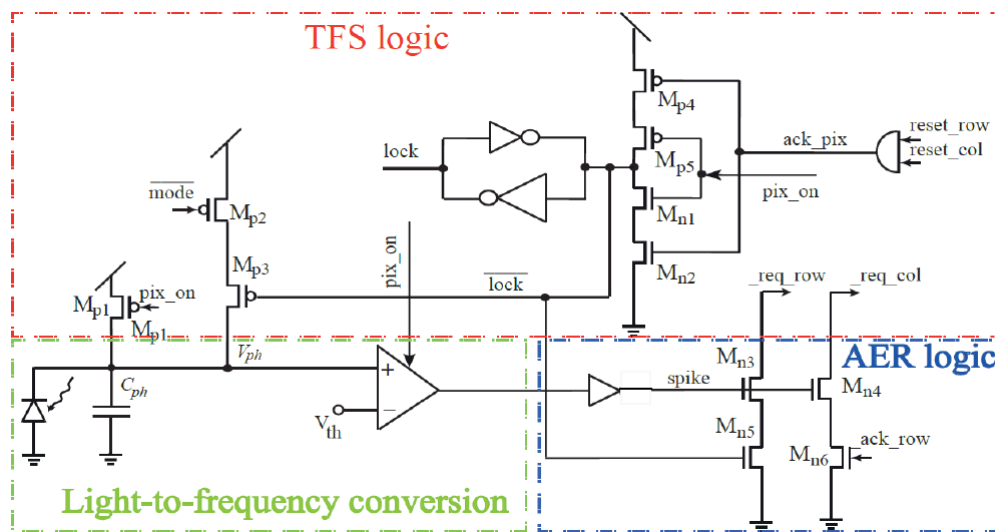


Figura 2.3: Esquemático del píxel [15]

En la figura 2.3 se observa que el circuito del píxel se ha dividido en tres bloques. El primero de ellos (*Light-to-frequency conversion*), como su nombre indica, realiza la conversión de luz en frecuencia, se genera un pulso cuando la carga almacenada en la capacidad es tal que activa la salida del comparador. El segundo bloque (*TFS logic*) es el encargado de devolver a su estado original, al de conversión, una vez que las coordenadas del píxel se enviaron fuera del chip. El tercer bloque (*AER logic*) está compuesto por lógica específica que implementa el protocolo de comunicación AER.

Para obtener información más detallada sobre los distintos modos de operación del píxel puede consultar la referencia [4]

2.1.2. Arbitrador

Un *arbitrador* es un dispositivo que se usa en un sistema para decidir, en cada ciclo, que dispositivo o señal puede ser transmitida. En definitiva, se puede decir que, como su propio nombre indica actúa como arbitrador. Se definen a continuación dos tipos de arbitradores atendiendo a la sincronía, síncronos o asíncronos:

- **Arbitrador asíncrono:** se utiliza en circuitos asíncronos para seleccionar el orden de acceso a un recurso compartido entre las solicitudes asincrónicas. Su función es evitar que se produzcan dos operaciones a la vez cuando no deberían, cuando reciba dos solicitudes de inmediato, debe decidir qué solicitud atender primero [28].
- **Arbitrador síncrono:** se utilizan en contextos síncronos para asignar acceso a un recurso compartido.

En este caso, en nuestro trabajo utilizaremos un arbitrador asíncrono por la naturaleza del circuito y las ventajas que supone. Dentro de este tipo vamos a distinguir entre dos principales: *Fair or non greedy arbiters* y *greedy arbiters*.

2.1.2.1. Fair arbiter

El primer *fair arbiter* asíncrono fue propuesto por Mead y Conway y posteriormente mejorado por Clowes y Corbett, este fue diseñado con puertas lógicas y transistores. Este arbitrador no tiene la posibilidad de atender filas o columnas de forma simultánea. Tal y como su nombre indica un *fair arbiter* es justo porque cuando dos píxeles estén compitiendo por transmitir simultáneamente recuerda la fila o columna a la que dio prioridad, de forma que, si se vuelve a producir otra colisión da prioridad a la fila o columna que no la había obtenido en la colisión anterior.

En la actualidad algunos dispositivos de medición modernos manejan flujos de datos continuos e irregulares. El procesamiento de la información de todo el sistema es esencial para muchas aplicaciones actuales que necesitan transmitir datos en tiempo real como en domótica, control de procesos, etc. Normalmente esta información medida por los dispositivos tiene que ser transmitida a otro dispositivo mediante una comunicación lo más simple posible. Es por ello, que para estos casos, surge la necesidad de tener un arbitrador justo o no codicioso (*Fair arbiter*), que sea capaz de garantizar

que se transmite toda la información sin pérdida de datos. [27]

Este tipo de arbitradores tiene desventajas como que puede transmitir información redundante, mayor tiempo de procesamiento y puede ser propenso a errores.

A continuación, se muestra un circuito de un *fair arbiter* y se procede a analizar el funcionamiento del mismo:

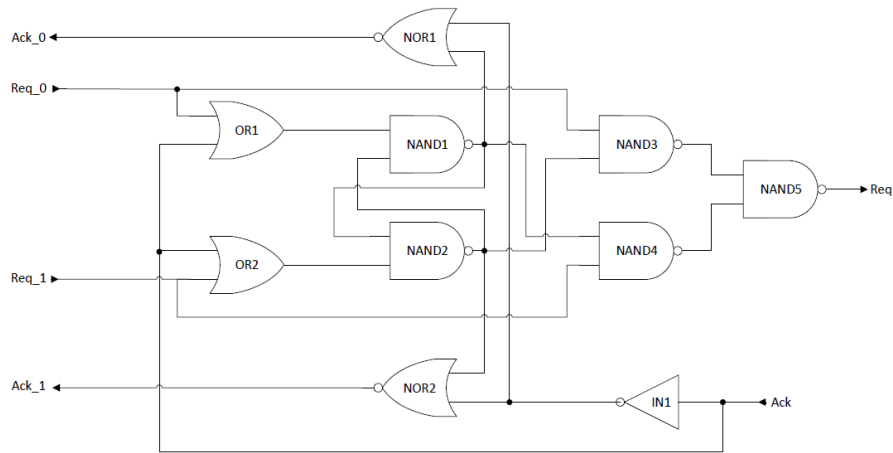


Figura 2.4: Esquema fair arbiter [27]

Como se puede observar en la figura 2.4, el arbitrador posee dos entradas y otorga la transmisión de alguna de ellas enviando un *acknowledge*. En el esquema las puertas NAND3, NAND4 y NAND5 generan la señal de solicitud (*request*) mientras que las puertas NOR1 y NOR2 sintetizan las señales de reconocimiento Ack_0 o Ack_1 del *acknowledge*. Req_0 o Req_1 son las solicitudes de dos canales diferentes. La línea de solicitud se eleva siempre que se requiera el ancho de banda. El canal que gana el arbitraje recibe reconocimiento, es decir, Ack_0 o Ack_1 . El canal mantiene la solicitud activa mientras la solicitud este activa.

Cuando la señal Req de salida se pone en alto, esto indica que el arbitrador ha recibido al menos una solicitud y ha seleccionado uno si ambos están activos. El arbitrador requiere una señal de confirmación (Ack) a alto nivel para activar el reconocimiento. Cuando una de las solicitudes ha sido concedida, otra solicitud no puede activar su reconocimiento correspondiente. Solo puede hacerlo después de que el primer *request* haya cambiado a estado

bajo y el correspondiente ha afirmado bajo y su correspondiente *acknowledge* también. [27]

A continuación, se muestra un esquema tipo árbol de arbitradores justos y se explica su funcionamiento:

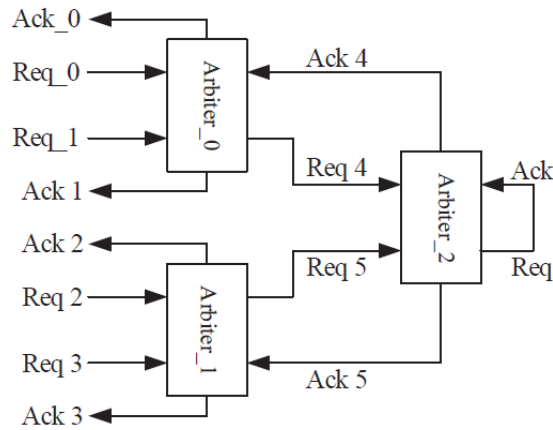


Figura 2.5: Esquema tipo árbol de Fair arbiters [27]

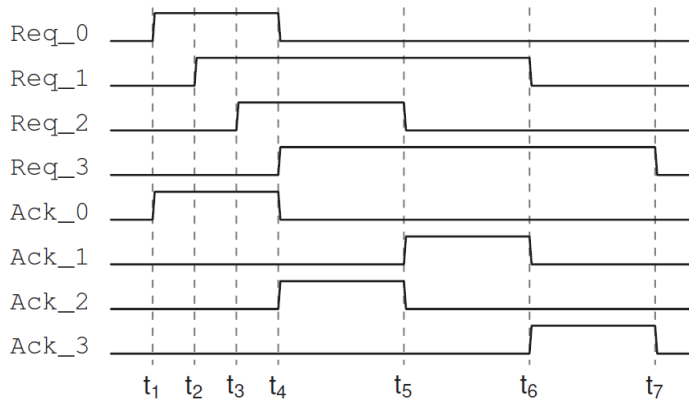


Figura 2.6: Diagrama temporal Fair arbiter [27]

En la figura 2.6 se puede comprobar el funcionamiento del arbitrador. En primer lugar se observa que la señal Req_0 cambia de estado a bajo a alto en t_1 , esto hace que la señal Req_4 tenga el mismo comportamiento, entonces el arbitrador reconoce esta solicitud, la señal Ack_4 pasará de un estado bajo a

alto y con ella la señal Ack_0 que seguirá el comportamiento de Ack_4 . Cuando Req_0 vuelva a cambiar de alto a bajo, Req_4 hará lo mismo y el arbitrador cambiará las señales Ack_4 y Ack_0 de alto a bajo en t_4 . Es en este momento, debido a que Req_2 está en el alto, cuando el arbitrador ve en alto Req_5 y reconoce Ack_5 o lo que es lo mismo se reconoce Ack_2 . Cuando Req_2 baja en t_5 , se repite el mismo proceso y se reconoce Req_1 a través de Ack_1 . Después de completar Req_1 en t_6 , Req_4 pasa a estado bajo y Req_3 obtiene el reconocimiento Ack_3 .

Mientras tanto, si Req_0 es solicitado nuevamente, será concedida sólo después de que todas las demás solicitudes que fueron afirmadas antes de él se conceden. Esto significa que, si se ha una solicitud ha sido hecha por uno o varios otros instrumentos antes de que el resto haya terminado, no será concedida hasta que el resto de solicitudes hayan sido procesadas, por lo tanto es un comportamiento “justo”, se le da la misma prioridad a todos los pxeles. Un punto a tener en cuenta es que si dos solicitudes llegan casi al mismo tiempo, el arbitrador puede entrar en estado metaestable.

2.1.2.2. Greedy Arbiter

El principio de funcionamiento de un *Greedy arbiter* o arbitrador codicioso es atender todas las solicitudes locales primero antes de devolver el control al nivel superior en la jerarquía. Si tenemos varios estímulos simultáneos, el arbitrador dará acceso al bus de forma aleatoria, pero siempre concediendo el mismo a solo un estímulo. En definitiva, el arbitrador egoísta siempre favorece a la misma fila/columna cuando hay una colisión. Este tipo de arbitrador posee ventajas respecto a los *Fair arbiter*, como por ejemplo, son más rápidos y funcionan en condiciones con actividad media de los buses y algunas desventajas como que si la carga de la solicitud comienza a exceder el rendimiento del bus, algunas no idealidades podrían comenzar a aparecer y los eventos se podrían perder. Se muestra un esquema de un arbitrador *Greedy*:

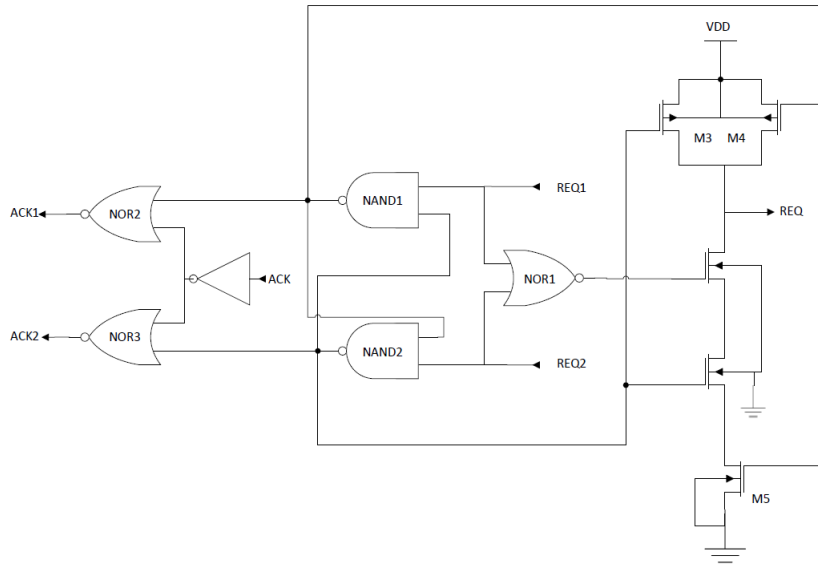


Figura 2.7: Esquema Greedy arbiter [4]

La idea principal es que el flip-flop realizado con las dos puertas NAND (situado en el centro del esquema) puede considerarse un flip-flop con entradas bajas activas. Si no se solicita acceso al bus o lo que es lo mismo $Req_1 = 0$ y $Req_2 = 0$, las dos salidas del flip-flop están a nivel alto. Sin embargo si la señal Req_1 pasa de estado bajo a alto antes que la señal Req_2 , la señal alta de la NAND2 baja y bloquea la señal Req_2 y con ello provocará que suba la Req_1 y viaje hacia el siguiente nivel. [12]

Los arbitadores pueden implementarse como un árbol binario de elementos que manejan varias señales de entrada y emiten una solicitud propia.

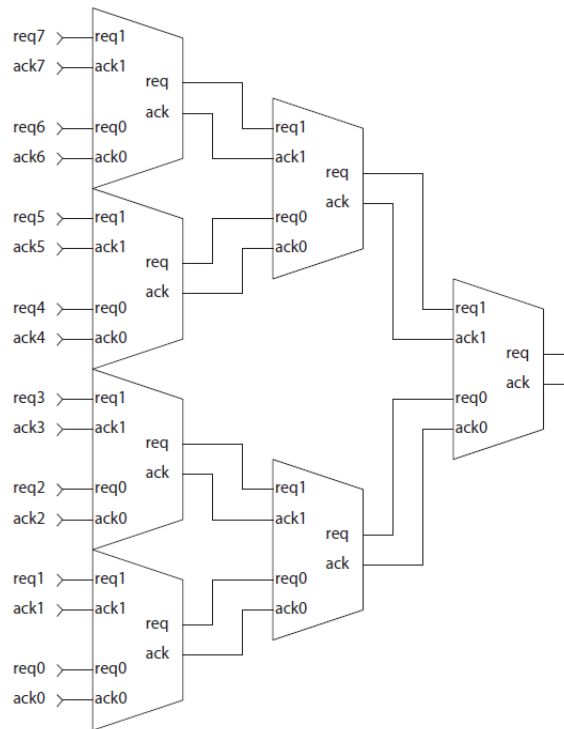


Figura 2.8: Árbol binario de arbitraje de 8 peticiones [12]

En funcionamiento de este árbol de arbitraje es el siguiente, imaginemos que Req_6 gana a Req_7 y con lo cual obtiene acceso al bus en el primer nivel y Req_5 gana a Req_4 , en el siguiente nivel el acceso sería concedido a la señal que ganase entre Req_6 y Req_5 . Lo mismo ocurriría para el resto de arbitraje, de forma que en el último nivel tendríamos solamente una señal, la ganadora en el cómputo global.

Dado que en el ámbito de los nanosatélites se busca el menor consumo posible, este tipo de arbitraje son los elegidos para utilizar en el diseño por su rapidez y bajo consumo.

2.1.3. Multiplexor

Un multiplexor es un circuito combinatorial que tiene varias entradas y una salida de datos. Este circuito está dotado de entradas de control que son capaces de seleccionar solamente una de las entradas de datos para permitir su transición a la salida. La cantidad de líneas de control que debe tener un multiplexor depende del número de canales de entrada. El número de canales

de entrada será 2^n donde n es el número de líneas de selección. Se muestra un esquema de un multiplexor simple con dos entradas y una salida.

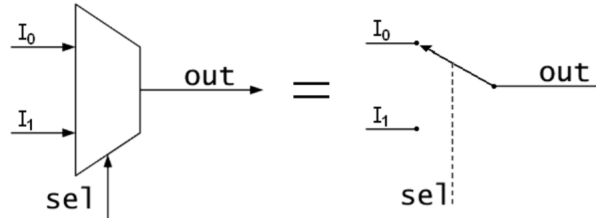


Figura 2.9: Esquema de funcionamiento de un multiplexor de dos entradas

De acuerdo a nuestro circuito de periferia, en la parte central del sistema tenemos un multiplexor que se utiliza para enrutar los datos del canal seleccionado a la salida, o lo que es lo mismo, asignar una palabra digital a cada fila o columna. En nuestro caso tenemos una matriz de 128 filas y, por lo tanto, se necesita un multiplexor de 7 bits mediante el cual podemos identificar $2^7 = 128$ elementos. En total hay 2 bloques de multiplexores en la periferia, uno para las filas y otro para las columnas. A continuación, se muestra un extracto de la disposición de las celdas '1' y '0' en la periferia, la imagen mostrada corresponde a una parte del multiplexor completo. Para ver la disposición específica de cómo se conecta cada fila del multiplexor al arbitrador se puede consultar la figura 2.13.

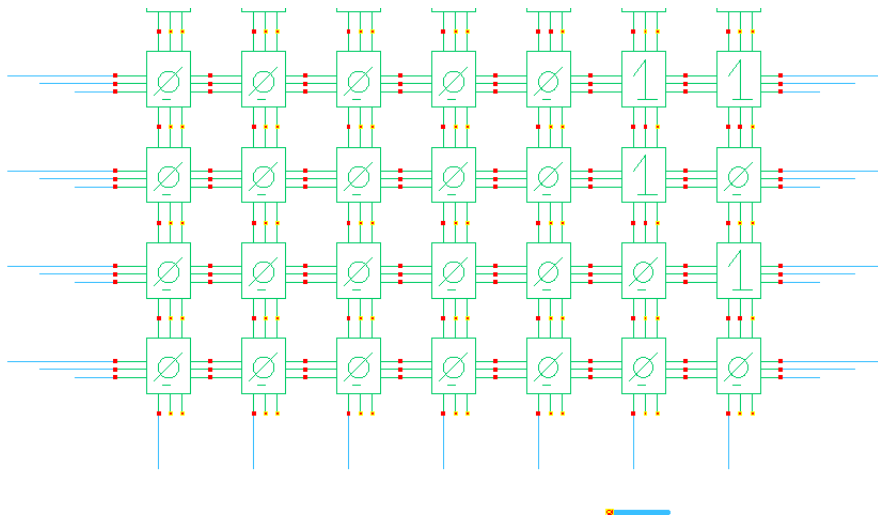


Figura 2.10: Disposición de varias filas del multiplexor en el circuito de periferia

Tal y como se observa, el funcionamiento de dicho multiplexor depende de las señales *ack*, cuando alguna señal de reconocimiento se eleva, selecciona la entrada del multiplexor y se habilita la correspondiente para la lectura, en definitiva, se podría decir que la lectura se realiza a través del multiplexor.

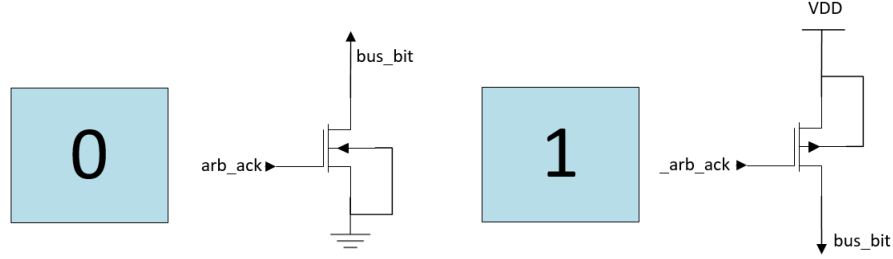


Figura 2.11: Esquema de los módulos de "0" y "1" que componen el multiplexor [14]

Se explica detalladamente el funcionamiento del multiplexor utilizado en nuestro circuito. Antes de ello, se realiza un breve repaso a las ecuaciones que modelan el comportamiento de los transistores MOS, ya que servirán de apoyo para una mejor comprensión de dicho funcionamiento. En un transistor MOS se pueden distinguir tres regiones de funcionamiento: corte, lineal y saturación. Para los transistores, sus expresiones [25] para la corriente de drenador y sus tensiones en los terminales son:

- **Corte**

1. NMOS

$$V_{GS} \leq V_T; V_{DS} > 0 \rightarrow I_D = 0 \quad (2.3)$$

2. PMOS

$$V_{GS} \geq V_T; V_{DS} < 0 \rightarrow I_D = 0 \quad (2.4)$$

- **Lineal u óhmica**

1. NMOS

$$V_{GS} > V_T; 0 < V_{DS} \leq V_{GS} - V_T \quad (2.5)$$

$$I_D = \frac{K'_n W}{L} \left[(V_{GS} - V_T)V_{DS} - \frac{V_{DS}^2}{2} \right] \quad (2.6)$$

2. PMOS

$$V_{GS} < V_T; 0 > V_{DS} \geq V_{GS} - V_T \quad (2.7)$$

$$I_D = \frac{K'_p W}{L} \left[(V_{GS} - V_T) V_{DS} - \frac{V_{DS}^2}{2} \right] \quad (2.8)$$

■ Saturación

1. NMOS

$$V_{GS} > V_T; V_{DS} > V_{GS} - V_T \rightarrow I_D = \frac{K'_n W}{2L} (V_{GS} - V_T)^2 \quad (2.9)$$

2. PMOS

$$V_{GS} < V_T; V_{DS} < V_{GS} - V_T \rightarrow I_D = \frac{K'_p W}{2L} (V_{GS} - V_T)^2 \quad (2.10)$$

Como se puede ver en la figura 2.11 la activación o desactivación de los transistores depende de la señal arb_ack y $_arb_ack$. En el circuito los transistores funcionan como interruptores (saturación o corte). En definitiva, cuando la señal $arb_ack = 1$ o lo que es lo mismo $_arb_ack = 0$, el transistor N se cierra y el P también, se encuentran en saturación y por lo tanto conducen, de forma que se reconoce la dirección de la fila o columna que ha transmitido su información. Sin embargo, cuando la señal $arb_ack = 0$ o $_arb_ack = 1$, el transistor N permanece abierto y el P también (transistores en corte), de forma que la dirección de la fila o columna no puede ser transmitida.

Por otra parte, la cantidad de bloques de ceros y unos que se necesitan dependerá del circuito, tal y como se ha comentado anteriormente. En este caso se necesitará un multiplexor para conocer la dirección de la fila y otro para la de la columna. Solo una dirección del multiplexor será transmitida por fila y otra por columna de forma que se podrá identificar que píxel de la matriz se ha iluminado.

2.1.4. Lógica de selección

El bloque intermedio entre el píxel y el multiplexor es la lógica de selección. Este bloque se utiliza para gestionar las señales de petición y de acceso al bus compartido de cada píxel. Además se encarga y gestiona también el reinicio de cada píxel. Este bloque está formado por cuatro inversores, dos

puertas lógicas NOR, un PMOS y un flip-flop SR y es necesario que se coloque uno por cada fila y por cada columna. Se muestra un esquema del circuito:

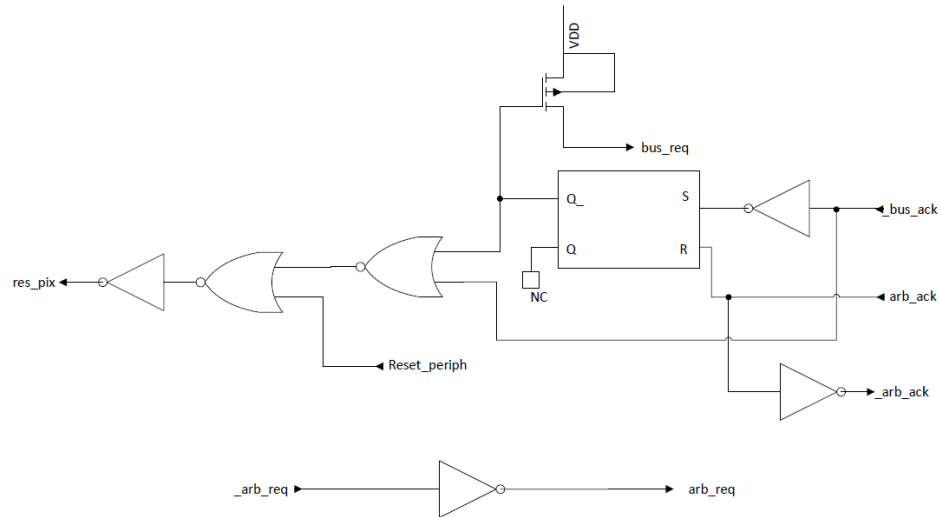


Figura 2.12: Esquemático del módulo de lógica de selección con tecnología UMC 0.18 (CMOS)

A continuación se describe el funcionamiento del circuito de la figura 2.12. En primer lugar podemos observar que el circuito posee dos partes aisladas una de la otra. La primera parte es un inversor aislado que únicamente invierte la señal que recibe del píxel, que será la que posteriormente recibirá el arbitrador. Cuando el píxel solicita acceso al bus para transmitir los datos de la fila que provienen del mismo ($_req_row = _arb_req$) este bloque invierte la señal obteniendo así arb_req . La segunda parte está compuesta de varios inversores, un latch SR, dos puertas NOR y un transistor de pull up. Para entender el comportamiento de este bloque es necesario tener en cuenta que en el circuito completo del sensor se tienen dos señales bus_req conectadas a una puerta NAND como puede verse en la figura 3.31. Se muestra una tabla de verdad de esta parte considerando que la señal $Reset_periph = 0$:

$_bus_ack$	arb_ack	S	R	Q	$Q_$	res_pix
0	0	1	0	1	0	0
0	1	1	1	Q_{n-1}	Q_{n-1}	out ¹
1	0	0	0	1	1	0
1	1	0	1	0	1	0

Cuadro 2.1: Tabla de verdad del bloque lógica de selección

El funcionamiento es el siguiente: cuando la información ha sido procesada se activa la señal $arb_row = arb_ack = 1$, mediante un inversor se obtiene la señal $_arb_ack = 0$. El pull-up se utiliza para tener la señal bus_req (supongamos de la fila) a VDD , bus_req entra a una NAND de dos entradas (ver 3.31) y en la entrada disponible se tiene la misma señal pero para la columna, de forma que hasta que una fila y una columna no hayan transmitido su posición, la salida de NAND ($_bus_ack$) se mantendrá a "1". Cuando ambas hayan transmitido su posición la salida cambiará a "0". De esta manera, y teniendo en cuenta que $Reset_periph = 0$ siempre, el valor de la salida será $res_pix = 1$ y se realizará un reset de píxel que ha transmitido la información preparándose así para recibir el siguiente. De esta forma se consigue que no se produzca un reset del píxel hasta que la posición del mismo no se haya transmitido a la periferia.

Por último se muestra un esquema del circuito de periferia en la que se puede observar la conexión de los bloques que lo componen.

¹Esta salida queda ligada al estado anterior que hubiera. Si $_bus_ack = 1$ y $_arb_ack = 1$ y cambia al estado $_bus_ack = 1$ y $_arb_ack = 0$ entonces la salida $res_pix = 1$ y se producirá un reset del sistema, de cualquier otro modo $res_pix = 0$.

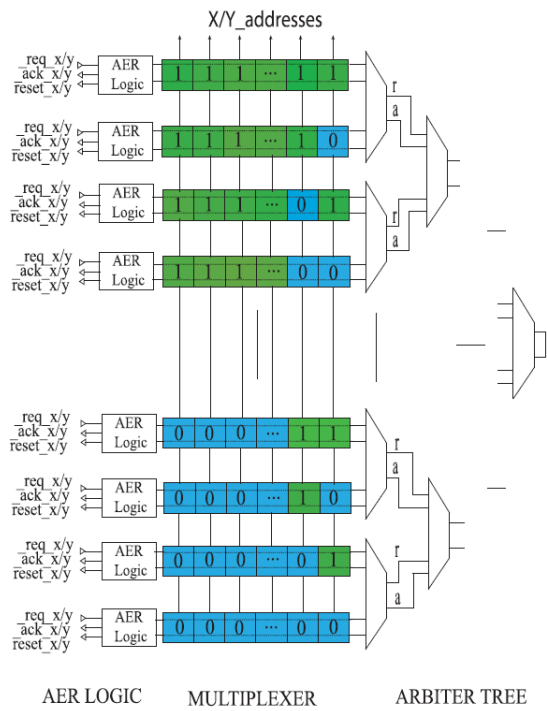


Figura 2.13: Esquema del circuito de periferia [14]

Capítulo 3

Simulaciones eléctricas

3.1. ¿Qué es la simulación?

Entre todas las definiciones de simulación que han aparecido desde 1949 destacamos la dada por Shannon en 1975, ésta dice que *la simulación es el proceso de diseñar un modelo de un sistema real y llevar a cabo experiencias con él, con la finalidad de aprender el comportamiento del sistema o evaluar diversas estrategias para el funcionamiento del sistema* [30].

Teniendo en cuenta esta definición, el objetivo de la simulación es realizar un modelo de los sistemas reales y, experimentando con el modelo, estudiar el comportamiento del sistema real. Un modelo es una forma simplificada de un sistema que describe mediante una serie de ecuaciones y relaciones el comportamiento básico del mismo. En el estudio de un sistema real, principalmente, se utilizan dos procedimientos para obtener sus modelos matemáticos correspondientes, el análisis teórico y el análisis experimental.

En el análisis teórico, el sistema real se modeliza utilizando las leyes fundamentales de la física u otras ciencias, obteniendo relaciones matemáticas concretas para extraer conclusiones sobre el sistema bajo estudio.

El análisis experimental consiste en construir el modelo a partir de mediciones realizadas directamente sobre el sistema. Notemos que de acuerdo con la definición dada de simulación, este proceso de realizar experimentos con el modelo matemático construido es en lo que consiste la simulación del modelo.

En aquellas situaciones reales donde el análisis teórico es complicado, una buena alternativa para el estudio de los sistemas es la simulación. Sin

embargo, el uso de la simulación, como dijo Shannon, presenta ventajas e inconvenientes. Concretamente, el uso de la simulación es de gran utilidad en situaciones como:

- Cuando la resolución analítica de un modelo no puede llevarse a cabo por métodos analíticos.
- Cuando existen los medios necesarios para de resolver analíticamente el problema, pero, dicha resolución es complicada y costosa.
- Si se desea experimentar antes de que exista el sistema.
- Cuando no es posible experimentar sobre el sistema real por ser dicha experimentación destructiva.
- En ocasiones en las que la experimentación sobre el sistema es posible, pero no ética.
- Es de gran utilidad en sistemas que evolucionan muy lentamente en el tiempo.

Y también lleva consigo inconvenientes tales como:

- La construcción de un modelo puede ser muy laboriosa.
- Frecuentemente el modelo omite variables o relaciones importantes entre ellas.
- Resulta difícil conocer la precisión de la simulación, especialmente en lo relativo a la precisión del modelo.

Las herramientas de simulación son ampliamente utilizadas en estudios comportamientos de circuitos antes de su fabricación, de escalado de dispositivos, y optimización de tecnología, por lo tanto deben de tener la capacidad de hacer frente a las prestaciones de los dispositivos actuales y de predecir las de futuras tecnologías y sus limitaciones. Estas herramientas permiten a las compañías predecir el comportamiento de los circuitos o ahorrar grandes cantidades de dinero en los procesos de desarrollo, antes de la fabricación en masa de los distintos componentes y, a los centros de investigación comprobar la viabilidad teórica de dispositivos basados en efectos físicos novedosos con geometrías diferentes a las configuraciones estándar.

En concreto, en los circuitos digitales asíncronos, que son mucho menos robustos frente a *glitches* que los síncronos surge la necesidad de poder obtener los retrasos que se pueden producir de forma precisa. Es en este ámbito

donde la simulación juega un papel esencial, ya que permite obtener dichos retrasos antes de fabricar el circuito.

La simulación de los diferentes dispositivos será llevada a cabo mediante el software especializado de *Cadence*, llamado *Virtuoso*.

3.2. Cadence

Tal y como se indicaba en capítulos anteriores, dado que el propósito de este trabajo es estudiar el circuito de periferia de un sensor solar, se procederá a comprobar el funcionamiento de dicho circuito mediante simulación. Para ello se decide utilizar un software especializado de *Cadence* por su multitud de ventajas como gestionar y administrar múltiples *testbenches* para crear un diseño bien probado y de alta calidad.

Cadence es un entorno de automatización de diseño electrónico (EDA) que permite integrar en un solo contexto diferentes aplicaciones y herramientas, permitiendo así soportar todas las etapas de diseño y verificación de IC desde un único entorno [31]. Estas herramientas son generales y admiten diferentes tecnologías de fabricación.

Cadence permite realizar varios tipos de simulaciones:

- **Simulación a nivel de bloque:** ofrece todas las herramientas necesarias para diseñar y verificar bloques de señal analógica/mixta. Para este tipo de diseño utiliza un análisis de estado estable proporcionado por *Spectre*[®] para evaluar ruido, funciones de transferencia, etc.
- **Simulación a nivel de chip:** Cadence analiza todos los bloques abstraídos en una variedad de lenguajes combinados con bloques a nivel de transistor que convergen en un diseño completo. Proporcionan la gran capacidad y el alto rendimiento necesarios para garantizar que un chip completo funcione según lo previsto, independientemente del rendimiento de los bloques en conjunto.
- **Simulación de señal mixta:** estos diseños integran bloques analógicos y digitales complejos, lo que requiere pruebas y análisis exhaustivos de cómo interactúan los circuitos analógicos y digitales y la influencia que tienen entre sí. Para ello Cadence combina simuladores de circuitos analógicos y digitales *Xcelium*TM[®] y *Spectre*[®]. En nuestro caso una simulación de este tipo no es viable porque en circuitos asíncronos los retrasos de transmisión en función del layout son críticos.

Cadence posee diferentes aplicaciones y herramientas en función de las necesidades, para el diseño de circuitos incluye el entorno *Virtuoso*[®], para soluciones de simulación de *Spectre*[®] y para caracterización y validación *Liberate*[™], así como otras para el diseño especializado. Usando este entorno avanzado, con detección de parásitos, se puede abstraer y visualizar las muchas interdependencias de un diseño analógico, de RF o de señal mixta para comprender y determinar sus efectos en el rendimiento del circuito.

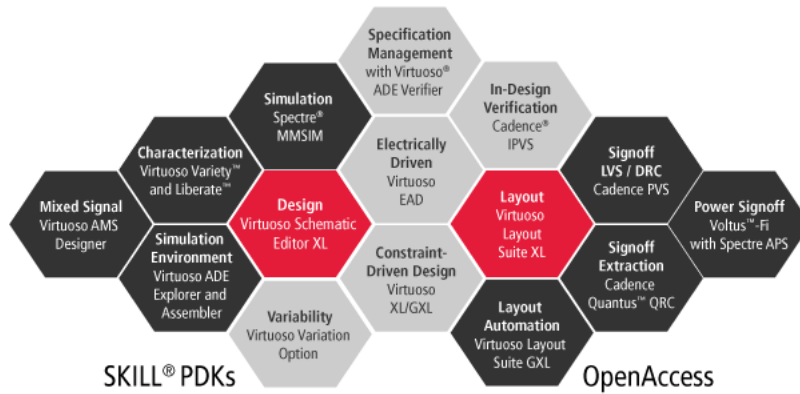


Figura 3.1: Herramientas EDA e IP de Cadence Virtuoso [31]

Tal y como vemos en la imagen anterior *Cadence Virtuoso* posee una gran familia de herramientas que permite a los diseñadores explorar, analizar y verificar completamente un diseño en función de los objetivos de diseño para que se pueda mantener la intención del diseño durante todo el ciclo. Incluye una entrada esquemática, modelado de comportamiento mediante Verilog-AMS, simulación de circuito, diseño personalizado, verificación física, etc. y se puede utilizar para diseños analógicos, de señal mixta, RF y celdas estándar e incluso para diseños de memorias y FPGA.

3.3. Simulación de los dispositivos

A continuación, se muestran los esquemáticos creados en *Cadence Virtuoso* y los resultados de los análisis realizados en las diferentes puertas lógicas y dispositivos necesarios para la realización de la simulación de la circuitería periférica de control y comunicaciones. Es importante comentar antes de empezar que para el diseño del mismo se contaba con una implementación inicial con lógica de 3.3 V y se ha realizado una migración completa del mismo a 1.8 V para reducir el consumo y aumentar la velocidad.

3.3.1. Puertas lógicas básicas

En esta sección se describirán y mostrarán brevemente los resultados de simulación de las puertas lógicas básicas que componen los distintos bloques de la periferia.

3.3.1.1. Inversor

En lógica digital, un inversor es una de las puertas más sencillas y a la vez más utilizadas, implementa una negación o inversión de la señal digital, su función principal es invertir la señal de entrada. Un inversor CMOS es un dispositivo integrado formado por un NMOS y un PMOS. Utilizando esta tecnología es posible diseñar un circuito inversor cuya disipación de potencia en corriente continua sea prácticamente nula. Es decir, sólo consume potencia en los transitorios que representan cambios de estado a la salida. Por esta razón, la tecnología CMOS se utiliza ampliamente en circuitos digitales, y en especial es ventajosa para equipos de bajo consumo de potencia.

La funcionalidad eléctrica se caracteriza por la función de transferencia de voltaje (VTC), voltaje de la salida en función del voltaje de entrada. La VTC se representa mediante un gráfico en el que se pueden obtener parámetros como el margen de ruido, la ganancia y los niveles lógicos entre otros. A continuación, se muestra la VTC de un inversor y se procede a explicar los diferentes parámetros de diseño que intervienen en la misma:

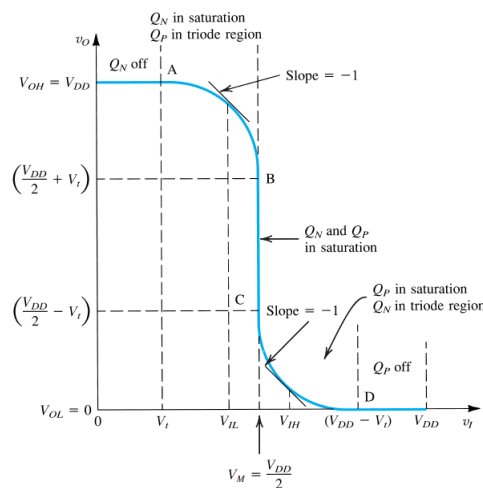


Figura 3.2: Característica de transferencia de voltaje del inversor CMOS cuando Q_N y Q_P coinciden [25]

La VTC representa la variación que experimenta la tensión de salida del inversor en función de las variaciones de la tensión de entrada del mismo. En un inversor ideal, la transición se presenta cuando la tensión de entrada alcanza el 50 % del valor de V_{DD} . Sin embargo, en un inversor CMOS el perfil dependerá de las características de los transistores que lo forman.

Para que el punto medio (V_M) se encuentre en $\frac{V_{DD}}{2}$ el factor de ganancia de ambos transistores debe ser el mismo, es por ello que se debe cumplir lo siguiente:

$$\beta_n = \beta_p \Rightarrow k'_n \left(\frac{W}{L}\right)_n = k'_p \left(\frac{W}{L}\right)_p \Rightarrow \left(\frac{W}{L}\right)_p = \frac{k'_n}{k'_p} \left(\frac{W}{L}\right)_n \quad (3.1)$$

La movilidad de los portadores de un transistor n y otro p son distintas y por ello k'_n y k'_p también lo son. Por tanto, tal y como se ha descrito antes que el cambio de $\frac{\beta_n}{\beta_p}$ se traduce en VTC asimétricas y habrá que diseñar los tamaños de los transistores en función de la necesidad.

Se diseña el circuito de un inversor en Cadence y se estudian las señales de entrada y salida. Asimismo, tenemos el siguiente esquemático y símbolo para esta puerta:

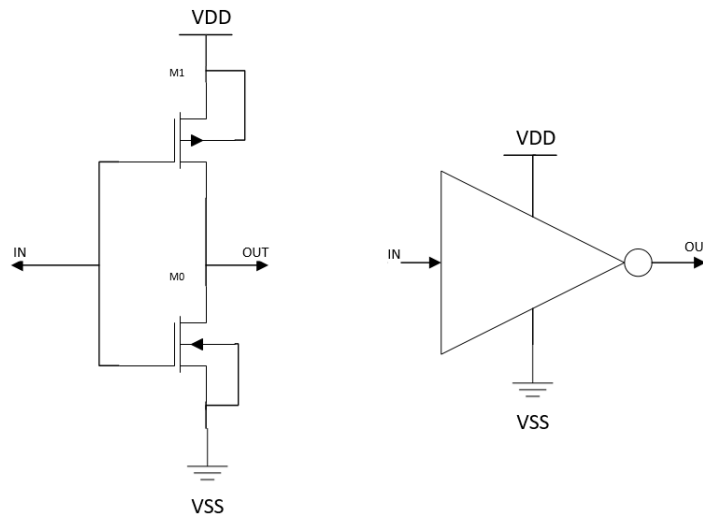


Figura 3.3: Esquemático y símbolo de un inversor con tecnología UMC 0.18 μm (CMOS) A.1

A continuación, se genera un banco de pruebas para estudiar si el com-

portamiento es correcto, obteniendo así la siguiente figura:

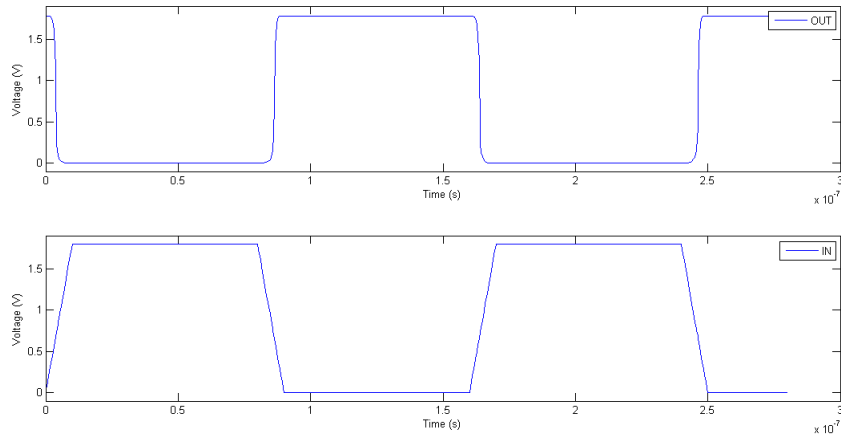


Figura 3.4: Evolución temporal de señales del inversor CMOS

En la figura 3.4 se observa que la señal de salida es exactamente la señal de entrada invertida, el inicio de las rampas de subida y bajada de la señal se producen en el mismo punto. Cabe destacar que la pendiente no exactamente la misma, pero para el caso de estudio de este trabajo se puede considerar que la diferencia en la pendiente de las rampas de subida y bajada es asumible.

3.3.1.2. NAND

En lógica digital, una puerta NAND, es una puerta que produce una salida falsa solamente si sus entradas son verdaderas, o lo que es lo mismo cuando todas sus entradas están a "1" o en estado alto, su salida es "0" o estado bajo, mientras que cuando al menos una sola de sus entradas o ambas están a "0" o en baja, la salida será "1" o en alto. Este comportamiento se puede describir mediante su tabla de verdad:

A	B	OUT
0	0	1
0	1	1
1	0	1
1	1	0

Cuadro 3.1: Tabla verdad de una puerta NAND de dos entradas

Una puerta NAND CMOS es un dispositivo integrado formado por dos NMOS y dos PMOS tal y como se muestra en el siguiente circuito:

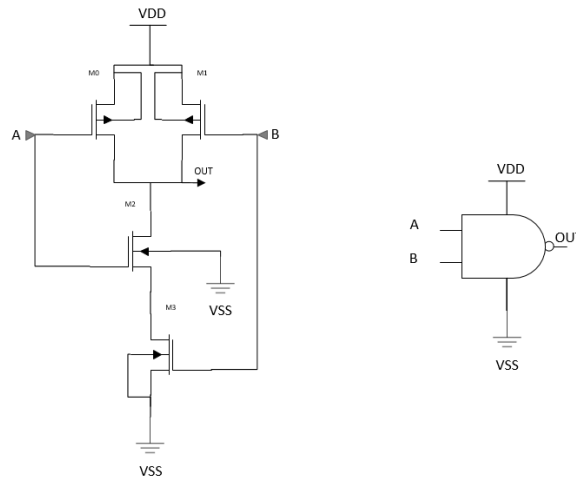


Figura 3.5: Esquemático y símbolo de una puerta NAND con tecnología UMC 0.18 μm (CMOS) A.2

Se crea el correspondiente banco de pruebas para estudiar si la variación temporal de las señales para todos los casos de estudio es correcta. Se introducen dos señales de entrada cuadradas A y B de periodos diferentes. Los resultados obtenidos son:

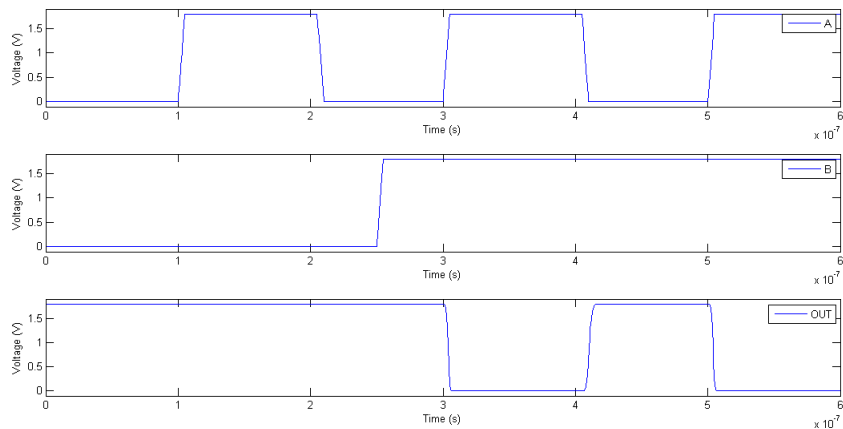


Figura 3.6: Evolución temporal de señales de la puerta NAND

En la figura 3.6 se puede observar que las dos entradas elegidas contemplan los casos de la tabla de verdad de una puerta NAND. Se comprueba que el comportamiento descrito anteriormente se cumple en el diseño realizado y por lo tanto, la puerta NAND funciona correctamente.

3.3.1.3. NOR

En lógica digital, una puerta NOR, se comporta de modo que cuando todas sus entradas están a "0" o valor bajo, su salida está a "1" o valor alto, mientras que cuando una sola de sus entradas o ambas están en "1" o alto, su salida será "0" o valor bajo. Las puertas NOR se pueden combinar para generar cualquier otra función lógica. Este comportamiento se puede describir mediante su tabla de verdad:

A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	1

Cuadro 3.2: Tabla verdad de una puerta NOR de dos entradas

Una puerta NOR CMOS es un dispositivo integrado formado por dos NMOS y dos PMOS cuya posición en el circuito es la siguiente:

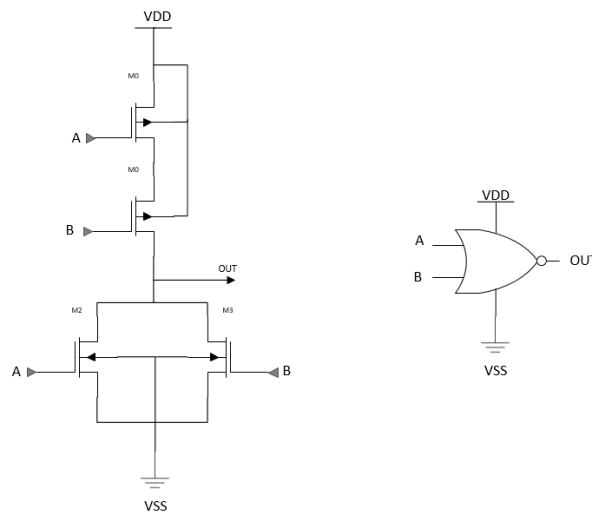


Figura 3.7: Esquemático de una puerta NOR con tecnología UMC 0.18 μm (CMOS) A.3

Por otro lado, se diseña un banco de pruebas para estudiar si la variación temporal de las señales es correcta. Los resultados obtenidos son:

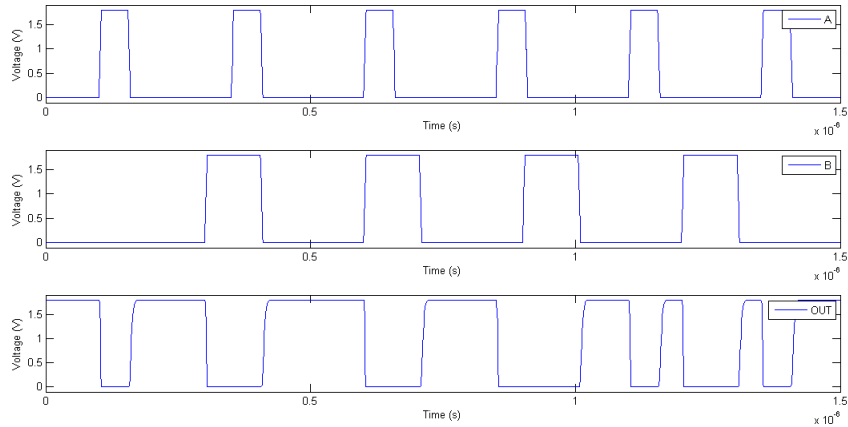


Figura 3.8: Evolución temporal de señales de la puerta NAND

En la figura 3.7 se puede observar que las entradas elegidas estudian todos los casos de la tabla de verdad de una puerta NOR de dos entradas. Se comprueba que el comportamiento descrito anteriormente se cumple en el diseño realizado y por lo tanto, la puerta NOR funciona correctamente.

3.3.1.4. SR latch

Un latch es un circuito electrónico biestable asíncrono que se utiliza normalmente para almacenar información en sistemas lógicos digitales. El cambio de los estados de salida se produce en función del estado presente en las entradas y de los estados previos en las salidas (retroalimentación). Los latches a diferencia de los flip-flops no necesitan una señal de reloj para su funcionamiento. El funcionamiento en cada una de las señales es el siguiente:

- **SET = RESET = 0.** El latch se encuentra en su estado de reposo, y la salida Q y Q' se mantendrán en el estado que tenían antes de que se produjera esta condición de entrada.
- **SET = 0, RESET = 1.** Esta condición siempre borrará el valor presente en Q. Es decir, restablece el latch.
- **SET = 1, RESET = 0.** Esta condición siempre establecerá el valor de salida Q en 1. Es decir, inicializa el latch

- **SET = RESET = 1.** Esta condición trata de restablecer e inicializar el latch al mismo tiempo, provocando resultados inesperados. Esta es la condición que usa en el arbitrador para asentir aleatoriamente una de sus entradas cuando hay colisiones.

A continuación, se muestra el circuito para el latch SR creado en *Microsoft Visio*:

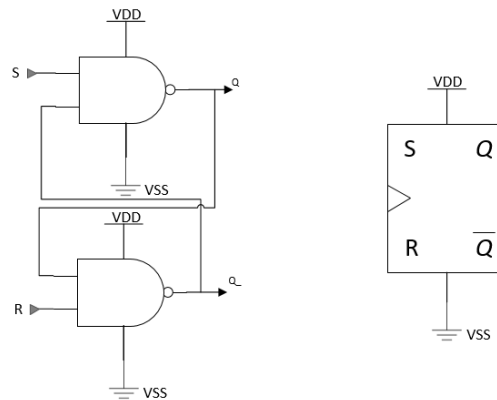


Figura 3.9: Esquemático y símbolo del latch SR con tecnología UMC 0.18 μm (CMOS) A.4

Se diseña el siguiente banco de pruebas para obtener el estudio de las señales para todos los casos posibles de señales de entrada:

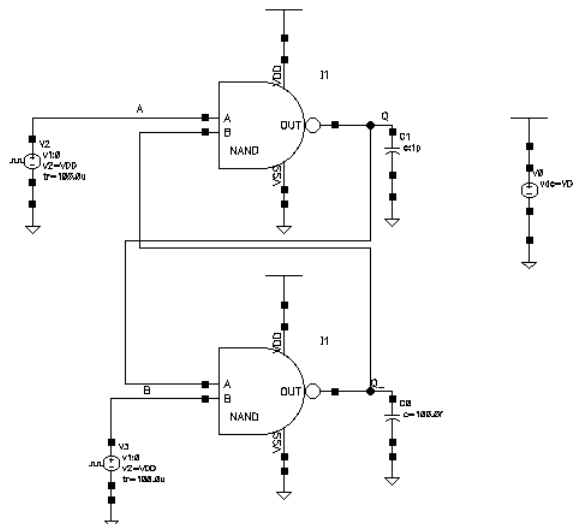


Figura 3.10: Banco de pruebas del latch SR

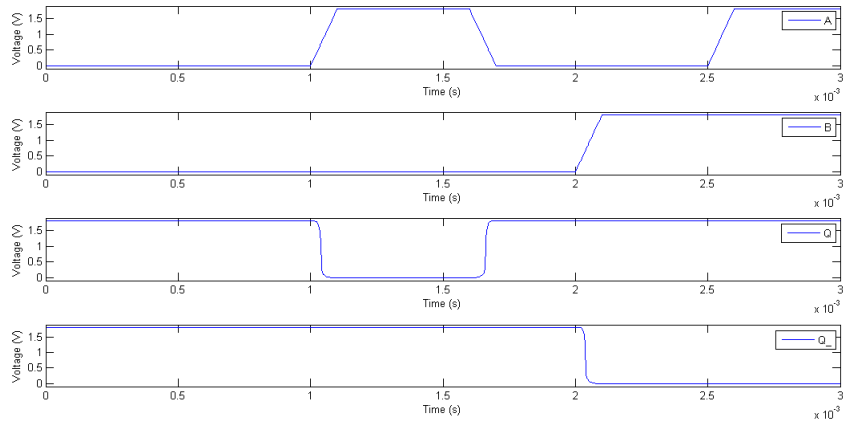


Figura 3.11: Evolución temporal de señales del latch SR

En la figura 3.11 se puede observar como las señales de salida obtenidas (Q y Q_{-}) se comportan de acuerdo al comportamiento esperado del latch SR, es decir, de acuerdo con su tabla de verdad.

S	R	Q	Q ₋
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q_{n-1}	Q_{-n-1}

Cuadro 3.3: Tabla verdad de un latch SR con puertas NAND

3.3.2. Arbitrador

Dado que el funcionamiento del circuito arbitrador se explicó en el capítulo 2 en detalle, en esta sección se procede a mostrar el esquemático del arbitrador *Greedy* creado en *Microsoft Visio*, su posterior banco de pruebas y los resultados obtenidos en el orden indicado.

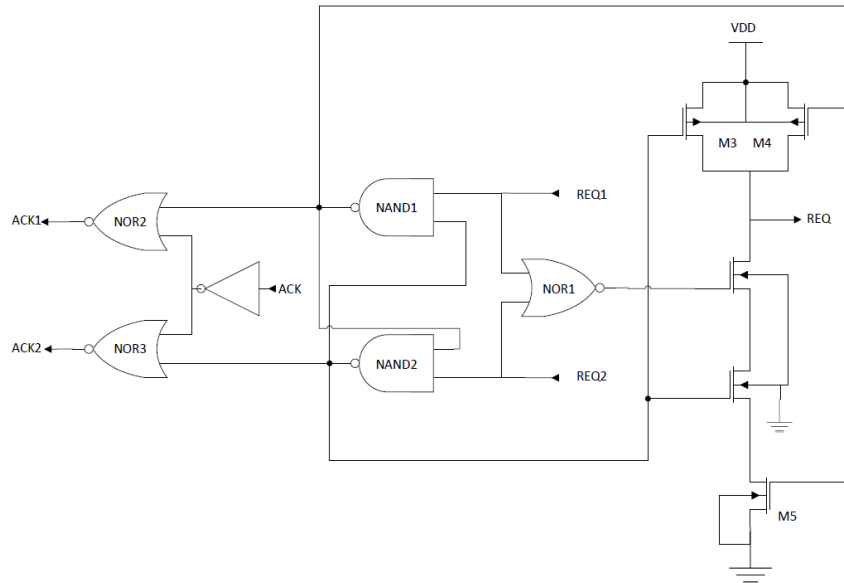


Figura 3.12: Esquemático del módulo arbitrador con tecnología UMC 0.18 μm (CMOS) A.9

Como se ha visto en capítulos anteriores y se observa en la figura 3.12, este bloque posee latch SR con entradas bajas activas (NANDs). El caso de funcionamiento a destacar se produce cuando no hay ninguna solicitud activa, o lo que es lo mismo, set y reset están configurados, entonces nos encontraríamos en su estado 'ilegal'. Ambas salidas se forzarán a 1. El estado en el que incurrirá el latch SR vendrá determinado por qué solicitud alta activa (*req0* o *req1*) llega en primer lugar. Si *req0* es la ganadora, entonces la salida de la puerta NAND inferior se convertirá en 0 y la señal *req* se propagará a la siguiente etapa. Si ese *req* es admitido por un *ack* entrante, se establece *ack0* y a instancia solicitante retirará su *req0*. Si mientras se ha establecido *req1*, esa solicitud también se concede sin liberar *req*.

Para realizar el banco de pruebas y comprobar el funcionamiento del arbitrador se estimulan las entradas *req0* y *req1* con un tren de pulsos con periodos diferentes de forma que las señales a veces coincidan en estado alto o en bajo y otras veces no. Con este test se puede observar como varían las señales *ack0* y *ack1* en función de las señales *req0* y *req1*. Hay que destacar que se han conectado condensadores de 1 *pF* para modelar la carga en las salidas *ack0* y *ack1*.

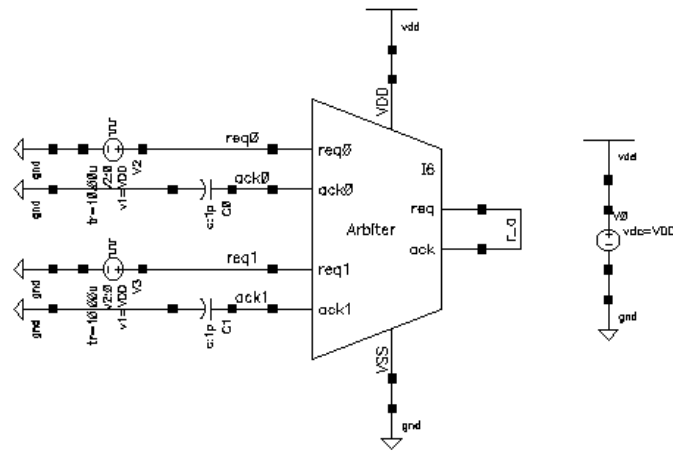


Figura 3.13: Banco de pruebas para el arbitrador

Los resultados de la simulación del banco de pruebas son los siguientes:

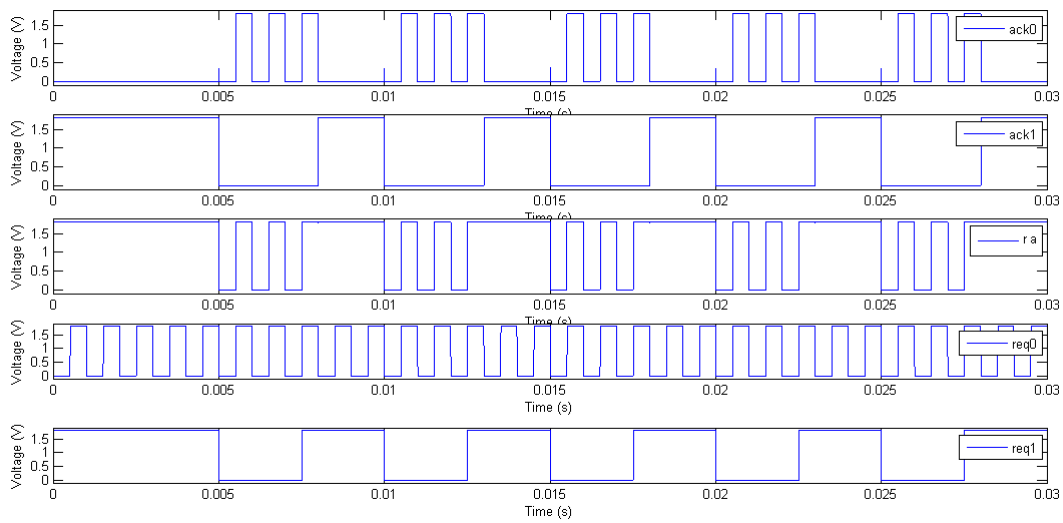


Figura 3.14: Evolución temporal de las señales del arbitrador

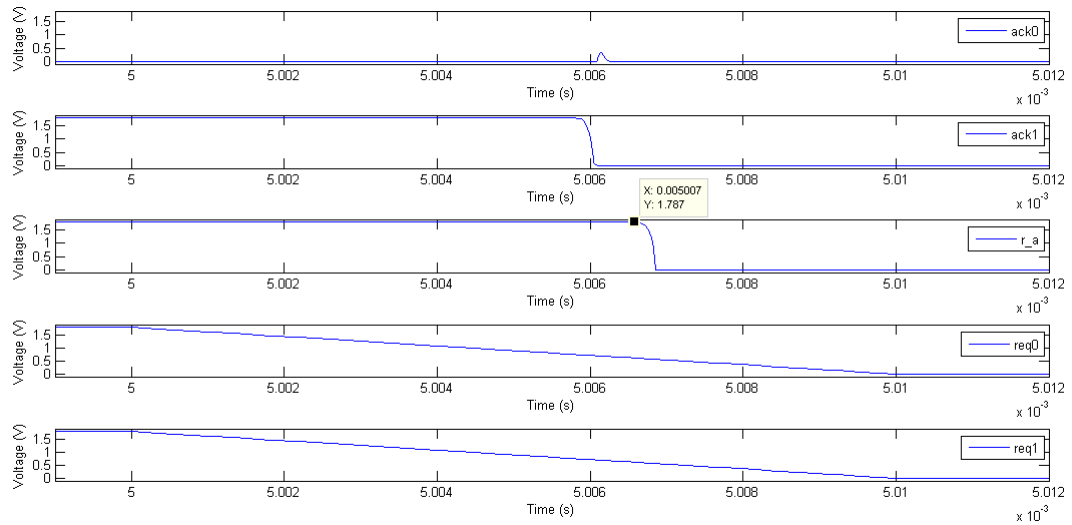


Figura 3.15: Zoom de la evolución temporal de las señales del arbitrador

En la figura 3.14 se puede observar que inicialmente la señal $req1 = 1$ o lo que es lo mismo, se activa, es decir, suponiendo que hablamos de píxeles, el píxel de la fila 1 solicita acceso al bus y la señal $ack1 = 1$ indica que se concede dicho acceso. En el siguiente ciclo se observa que la señal $req0$ esta a "1", es decir, el píxel de la fila 0 solicita acceso al bus y la señal $ack0$ pasa a valer "1", es decir, se concede dicho acceso. Pero llega un momento en el que "dos píxeles" solicitan acceso al bus simultáneamente $req0 = req1 = 1$, en este caso el arbitrador solo puede conceder el acceso a una de ellas. Si seguimos observando los siguientes ciclos siempre que este caso se produce favorece la misma solicitud (arbitrador egoísta). Además de una simulación transitoria se ha realizado un análisis de Corner mediante ADE XL para corroborar que efectivamente los resultados son los mismos.

En la figura 3.15 tenemos un zoom de la evolución temporal de las señales del arbitrador. Con este zoom se puede observar que una vez una petición ha sido realizada, el arbitrador tarda aproximadamente unos 1 ns en admitir la petición o lo que es lo mismo poner ack en estado alto, una respuesta bastante rápida.

Así se observa que el funcionamiento es correcto, pero el caso de estudio se ha realizado para una tensión de alimentación de 1.8 V . Es importante considerar en el caso de tener una tensión de alimentación más pequeña qué ocurriría. Cuando se reduce la tensión de alimentación de los transistores

las corrientes bajan, los tiempo de carga y descarga de las capacidades parásitas aumentan y la velocidad de conmutación en puertas lógicas también, entonces los retrasos pueden llegar a ser significativos. Por ello, se realiza un análisis paramétrico variando la tensión de alimentación desde 0.2 V a 1.8 V con un salto de 0.05 V , los resultados son los siguientes:

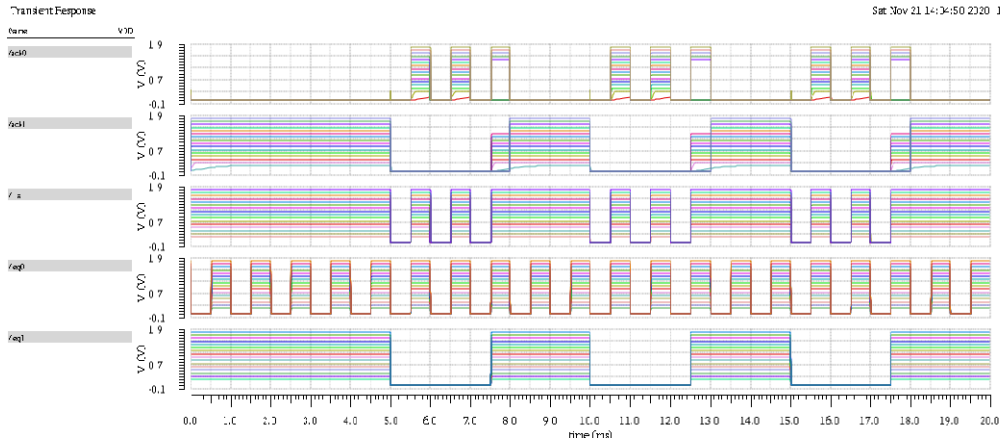


Figura 3.16: Evolución temporal de las señales del arbitrador con un análisis paramétrico en función de VDD

De la figura 3.16 se puede observar que para tensiones bajas de alimentación se produce un retraso en las señales de salida *ack*. En concreto, a partir de $V_{DD} = 0.4\text{ V}$, el retraso que se produce es significativo, en concreto, de alrededor de 0.1 ms . Para valores inferiores como $V_{DD} = 0.2\text{ V}$ el retraso es mucho mayor (aproximadamente 1 ms) pero importante si se tiene en cuenta que la velocidad de respuesta era del orden de 1 ns . Para valores superiores a $V_{DD} = 0.4\text{ V}$ se observa como dicho retraso no se produce. Por lo tanto podemos concluir que el arbitrador funciona bien en un rango de alimentación de $[0.45\text{ V}, 1.8\text{ V}]$.

3.3.3. Multiplexor

Del mismo modo que en el caso del arbitrador, dado que el comportamiento del multiplexor ha sido descrito en detalle en el capítulo anterior se muestra directamente el esquemático diseñado en *Microsoft Visio* de los bloques del bit "1" y del bit "0":

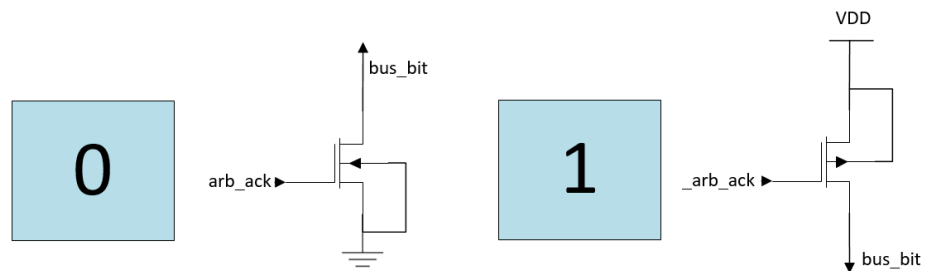


Figura 3.17: Esquemático y símbolo del bit "0" y del bit "1" con tecnología UMC 0.18 μm (CMOS) A.5 A.6

Para comprobar que cuando una fila o columna del píxel transmite bien la dirección del mismo se ha creado el siguiente circuito y su correspondiente banco de pruebas. En este circuito se estimulan las entradas *req0* y *req1* con un tren de pulsos con periodos diferentes y se comprueba que la salida $y < 0 >$ indica la fila del "píxel" que ha transmitido la información.

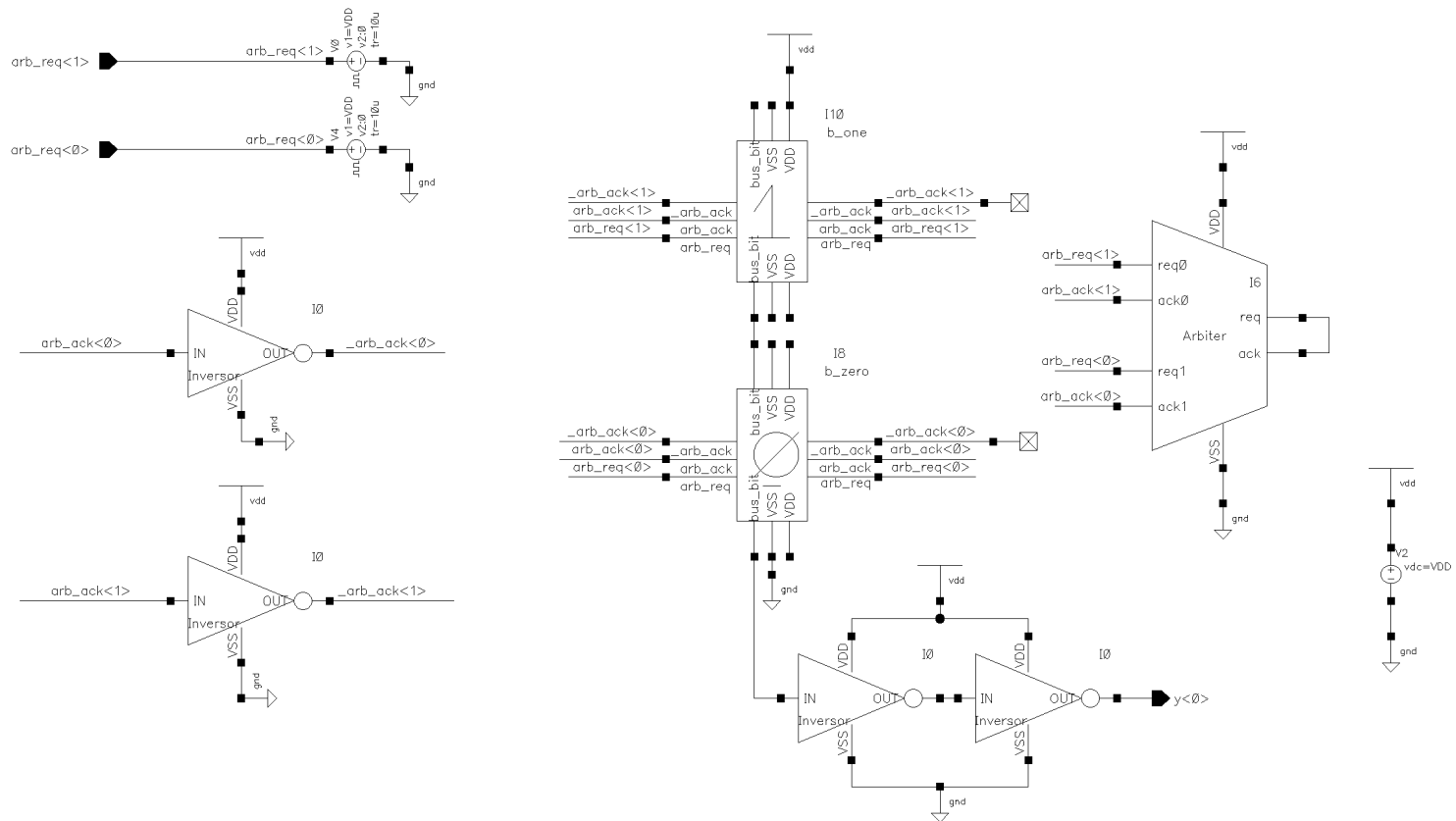


Figura 3.18: Banco de pruebas para los bits del multiplexor

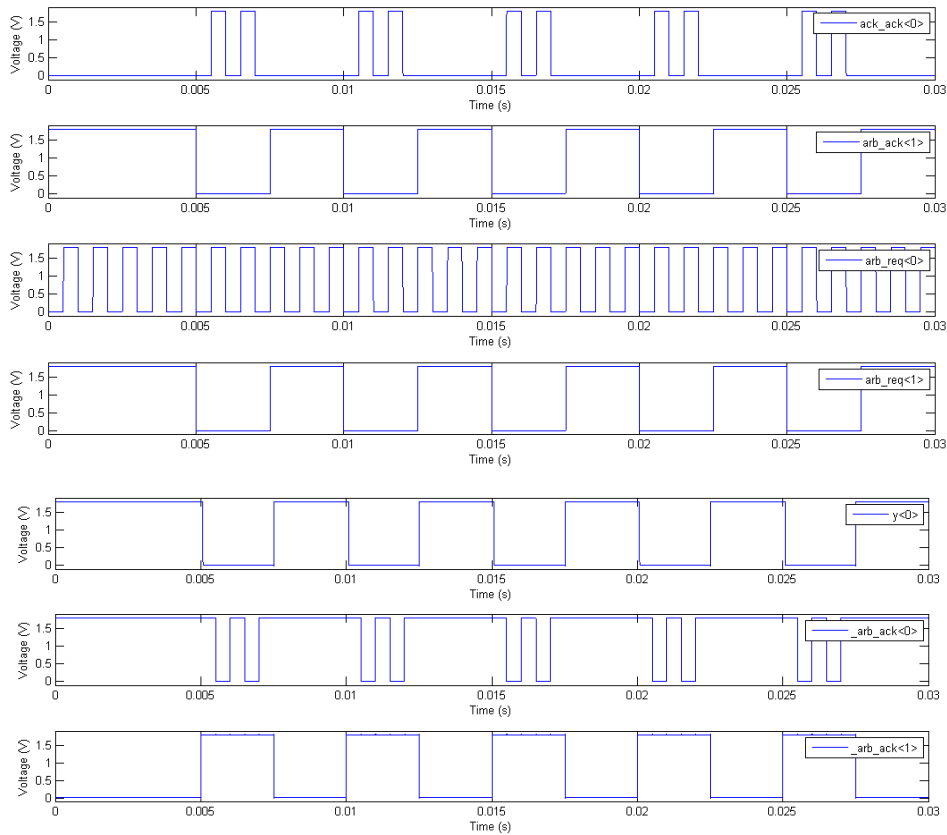


Figura 3.19: Evolución temporal de señales del multiplexor

En la figura 3.19 se puede observar como la señal $y < 0 >$ cambia su valor a "1" mientras la señal $arb_ack < 1 > = 1$, esto quiere decir que el "píxel" que ha enviado la información es el de la fila 1. Cuando la fila "0" envía información el valor de $y < 0 >$ será el valor de la corriente residual o de fuga, que será del orden de pA , en esta imagen dado la idealidad de las señales y que el píxel no está conectado no se puede distinguir este comportamiento, pero en secciones posteriores se podrá diferenciar claramente.

Se caracteriza el rango de funcionamiento de este bloque y para ello se realiza un análisis paramétrico. Se varía la tensión de alimentación entre 0.2 V y 1.8 V con un incremento de 0.05 V y se observa cuando el circuito deja de funcionar correctamente.

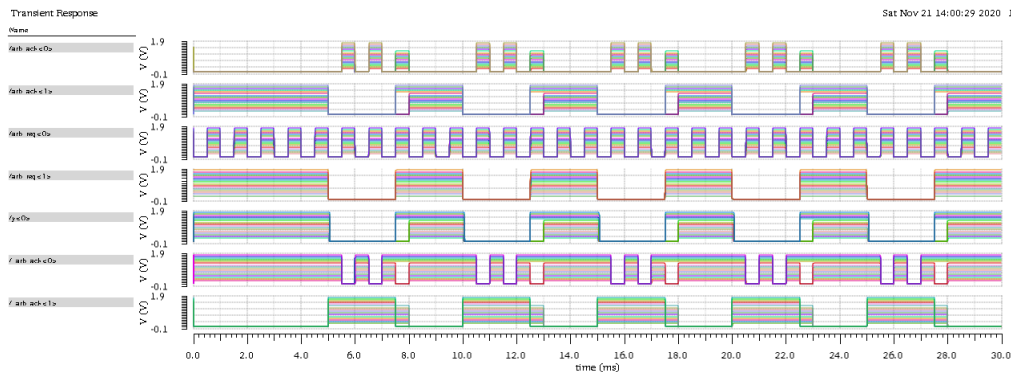


Figura 3.20: Evolución temporal de las señales del multiplexor con un análisis paramétrico en función de VDD

Tal y como se muestra en la figura 3.20 el circuito deja de funcionar a una tensión de 0.2 V , teniendo en cuenta que es un caso en el que la señal de entrada, al igual que las secciones anteriores, es ideal (no posee ruido, ni offset, ni retrasos, etc.).

3.3.4. Lógica de selección

Del mismo modo que en casos anteriores se muestra el esquemático diseñado en *Microsoft Visio* del bloque encargado de la lógica de selección:

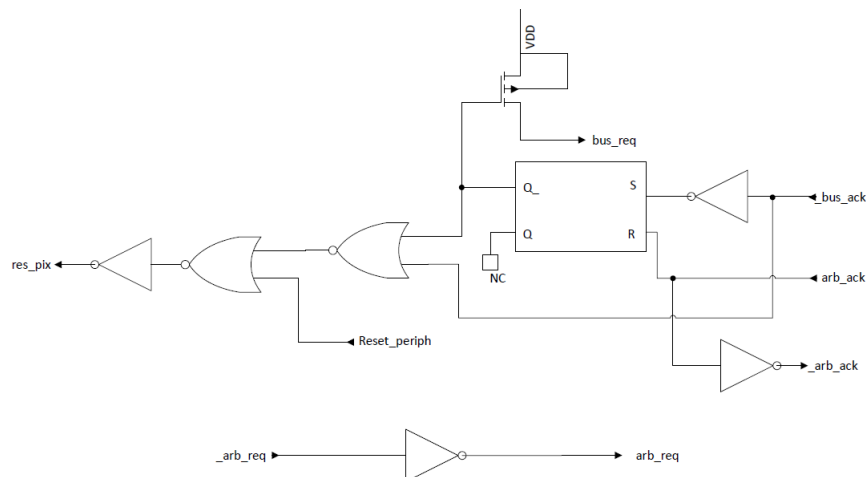


Figura 3.21: Esquemático del módulo de lógica de selección con tecnología UMC $0.18\ \mu\text{m}$ (CMOS) A.11

Se crea el siguiente banco de pruebas, en este, de igual modo que para los bancos de pruebas del arbitrador y multiplexor se estimulan las entradas `_req0` y `_req1` con un tren de pulsos con periodos diferentes y se comprueba que el sistema funciona correctamente (se realiza la inversión de las señales, que se produce un reset en caso de solicitarse, etc).

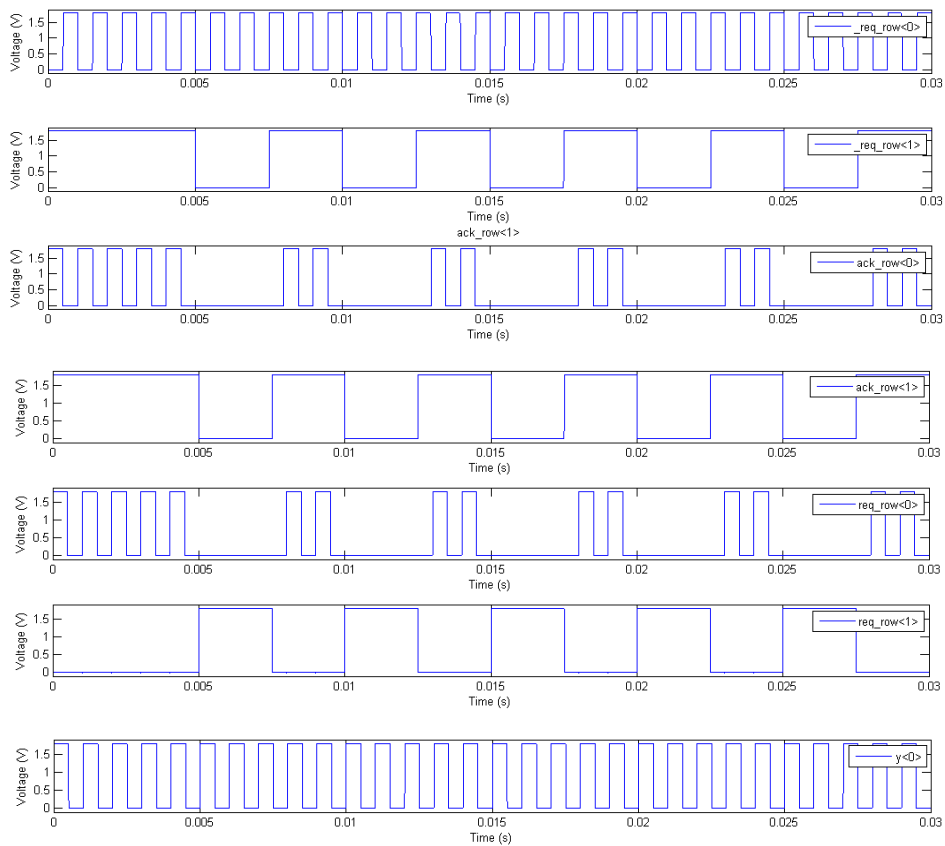


Figura 3.23: Evolución temporal de las señales de la lógica de selección

En la figura 3.23 se muestran solamente algunas de las señales del circuito pero son suficientes para comprobar el funcionamiento del mismo, por ejemplo, se puede observar que la inversión de la señal `_req_row` se realiza correctamente, así como que la señal `y < 0 >` sigue funcionando de forma adecuada.

De igual modo que para el caso del arbitrador y multiplexor, se realiza un análisis paramétrico en que se observan los mismos resultados que se obtuvieron en el caso del multiplexor y con ello, las mismas conclusiones (poseen el mismo de rango de funcionamiento).

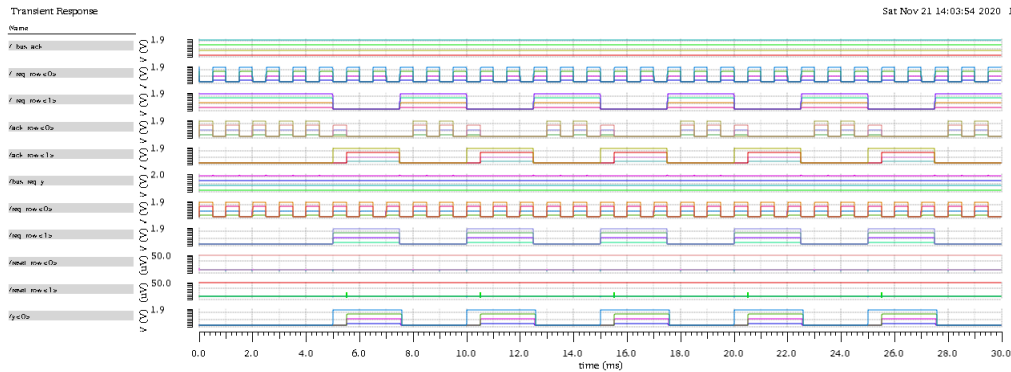


Figura 3.24: Evolución temporal de señales de la lógica de selección con un análisis paramétrico en función de VDD

En la figura 3.24 se observa como la inicialmente se realiza una petición para transmitir la fila 0 ($req_row < 0 > = 1.8 V$) y el arbitrador procesa dicha petición y la admite poniendo a estado alto $ack_row < 0 >$, mientras la señal $y < 0 >$ indica que la fila 0 ha enviado la información. En este banco de pruebas la parte referente al reset no se ha comprobado porque va ligada al píxel, se realizará dicha comprobación cuando se pruebe el circuito con el píxel.

3.3.5. Circuito completo

Antes de simular el circuito completo hay que tener en cuenta que las señales que salen del píxel diseñado por Antonio de la Calle [4] son de 3.3 V y nuestra periferia se ha diseñado para funcionar a 1.8 V. Por ello, para adaptar la tensión de las señales de un módulo a otro va a ser necesario la utilización de *level shifter* que mantengan la forma de las señales pero adapten el valor de tensión de las mismas en función de la necesidad. Dado que tenemos señales en ambas direcciones vamos a necesitar uno que adapte la señal de 3.3 V a 1.8 V y otro que lo haga de 1.8 V a 3.3 V. Mostramos, a continuación, los circuitos de los mismos y las simulaciones realizadas.

3.3.5.1. Level Shifter

La restricción de este circuito será que funcione para el rango de alimentación más bajo posible. Así, para elegir los tamaños de los transistores, se ha tenido en cuenta que sean mínimos y lo más parecidos posibles entre ellos, consiguiendo así un menor consumo de potencia. Además, los tamaños también dependen de la tensión a convertir, es por ello que se han ajustado

para que funcionen en el mayor rango de tensión de alimentación posible.

3.3.5.1.1. Level Shifter High-Low

Se diseña un level shifter que permite adaptar las señales que salen del píxel de valor 3.3 V a señales que entran en la periferia de valor 1.8V. Es circuito elegido para ello es el siguiente:

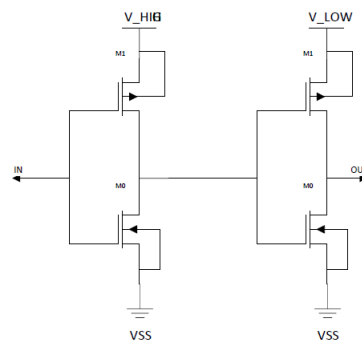


Figura 3.25: Esquemático y símbolo del level shifter de high a low con tecnología UMC 0.18 μm (CMOS) A.7

En este caso tenemos dos inversores, el primero que invierte la señal y el segundo cuya señal de salida es una señal de tensión de valor V_{low} y forma idéntica a la señal de entrada. Se diseña el siguiente banco de pruebas para comprobar que ante una señal de entrada cuadrada de valor 3.3 V la salida es la señal de entrada pero de 1.8 V.

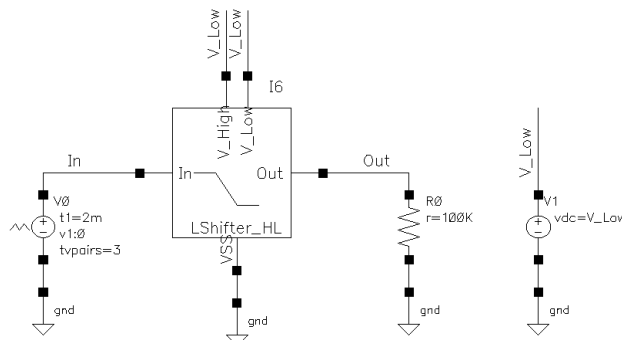


Figura 3.26: Banco de pruebas para el level shifter low-high

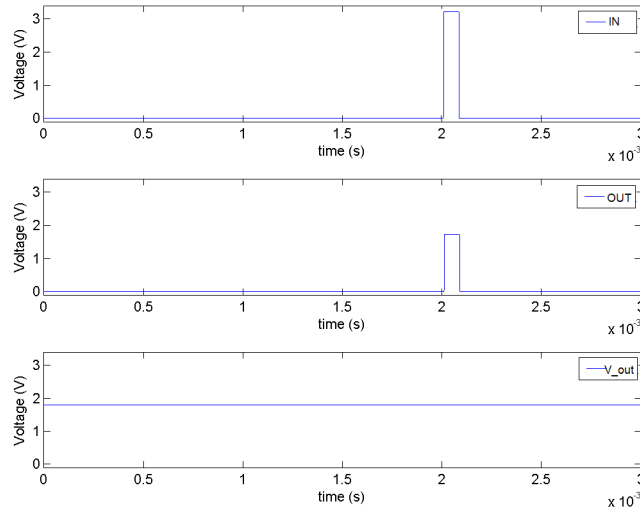


Figura 3.27: Variación temporal de las señales del level shifter de high a low voltage.

Se obtiene el comportamiento esperado, para una señal de entrada de 3.3 V la señal de salida es idéntica a la señal de entrada pero de 1.8 V.

3.3.5.1.2. Level Shifter Low-High

Se diseña un level shifter que permita adaptar las señales que salen de la periferia de valor 1.8 V a señales que entren en el píxel de 3.3 V.

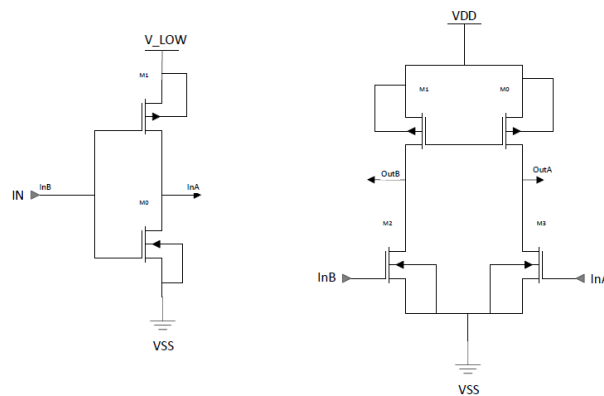


Figura 3.28: Esquemático y símbolo del level shifter de low a high con tecnología UMC 0.18 μm (CMOS) A.8

En este caso, al igual que en el caso anterior tenemos dos etapas, la primera en que solamente se realiza una inversión de la señal y la segunda en la que se amplifica la misma a la tensión V_High y se invierte. Se diseña el siguiente banco de pruebas para comprobar que ante una señal de entrada cuadrada o con cualquier otra forma de valor 1.8 V la salida sea la señal de entrada pero amplificada a 3.3 V .

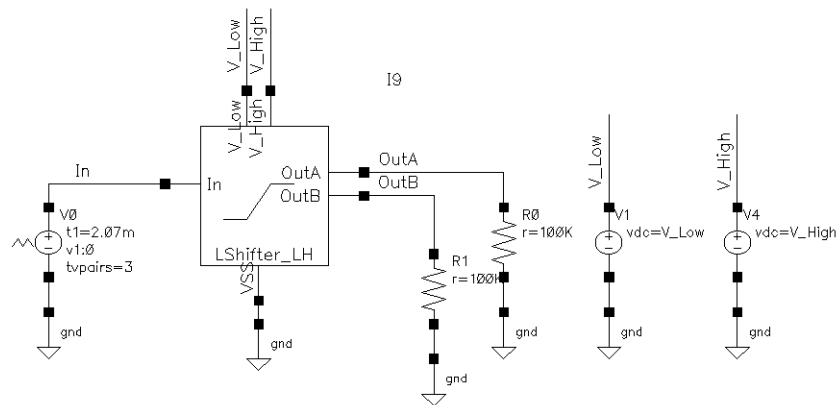


Figura 3.29: Banco de pruebas para la lógica de selección

Para este estudio se realizaron varias pruebas, la primera fue comprobar que funcionase correctamente con señales de 1.8 V de tensión de entrada. Posteriormente se realizó un análisis paramétrico variando dicha tensión para comprobar el rango de funcionamiento de este bloque $[0.6\text{ V}, 1.8\text{ V}]$. Se muestra, a continuación, la evolución de las señales para la tensión más baja de funcionamiento de dicho bloque de 0.6 V como se demostró en el correspondiente análisis paramétrico realizado.

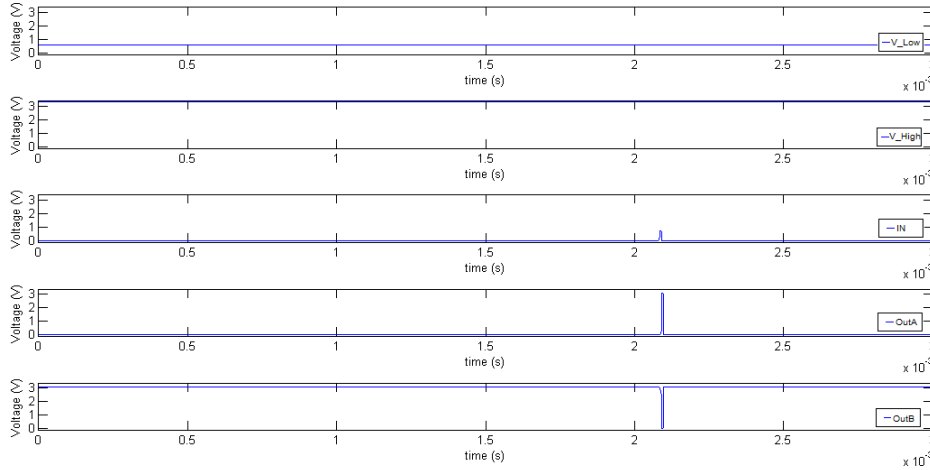


Figura 3.30: Variación temporal de las señales del level shifter de low a high voltage.

3.3.5.2. Simulación del circuito (píxel y periferia)

Para finalizar este capítulo, se realiza un esquemático que incluya la periferia y el píxel permitiéndonos así comprobar el correcto funcionamiento de la misma. Se decide utilizar un circuito simplificado, considerando una matriz de píxeles de 2×2 . Este caso de estudio será suficiente para poder comprobar el correcto funcionamiento del circuito que se muestra en la figura 3.31. Los valores de las variables utilizadas para la simulación de este circuito son los que se muestran en el siguiente cuadro:

Nombre	Valor
VDD_LS	1.8 V
Iph0	1 pA
Iph1	100 pA
Iph2	1 pA
Iph3	1 pA
Vpd	700 mV
VDD	1.8 V
vpull_up	1.4 V
Vth	1 V
Vbias_comp	1 V
t_pix_on	1ms
VDD_pixel	3.3 V

Cuadro 3.4: Valores de las variables asignadas en ADE L para la simulación del circuito completo del sensor solar

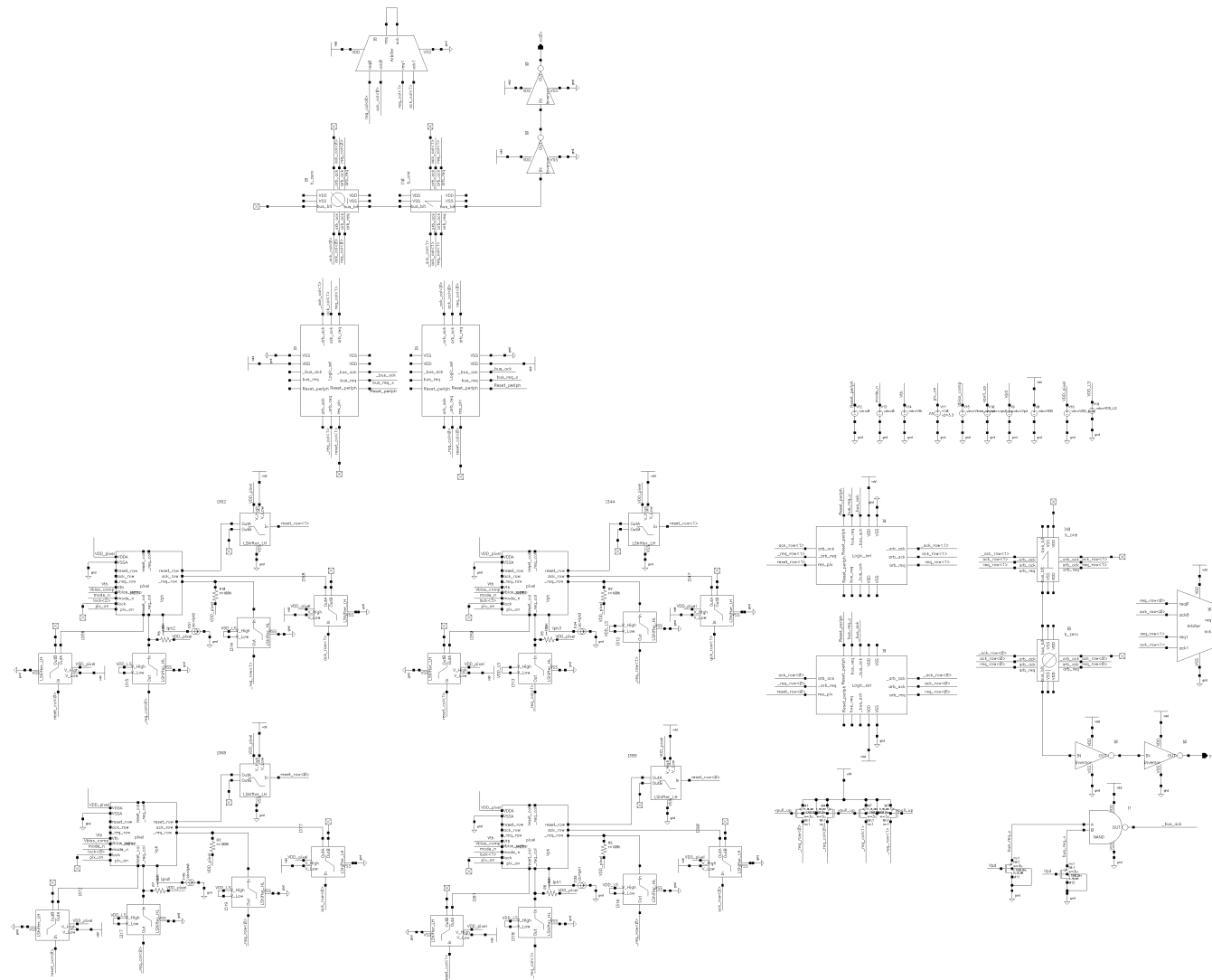


Figura 3.31: Banco de pruebas para un píxel de 2 x 2

Tal y como se observa en el circuito de la figura 3.31 cada píxel posee una fuente de corriente para poder estimularlos. La finalidad de la misma es suministrar a alguno de ellos más corriente que al resto y comprobar que efectivamente ese píxel (el de mayor corriente) es el que transmite su posición. Si fijamos la corriente del **píxel 1** con un valor $I_{ph1} = 100 \text{ pA}$ y el resto de píxeles a $I_{ph0} = I_{ph2} = I_{ph3} = 1 \text{ pA}$, el píxel que debería mandar su petición y su posterior reconocimiento es el **píxel 1**, cuya fila es la 0 y la columna la número 1. En la siguiente figura podemos observar este caso de estudio.

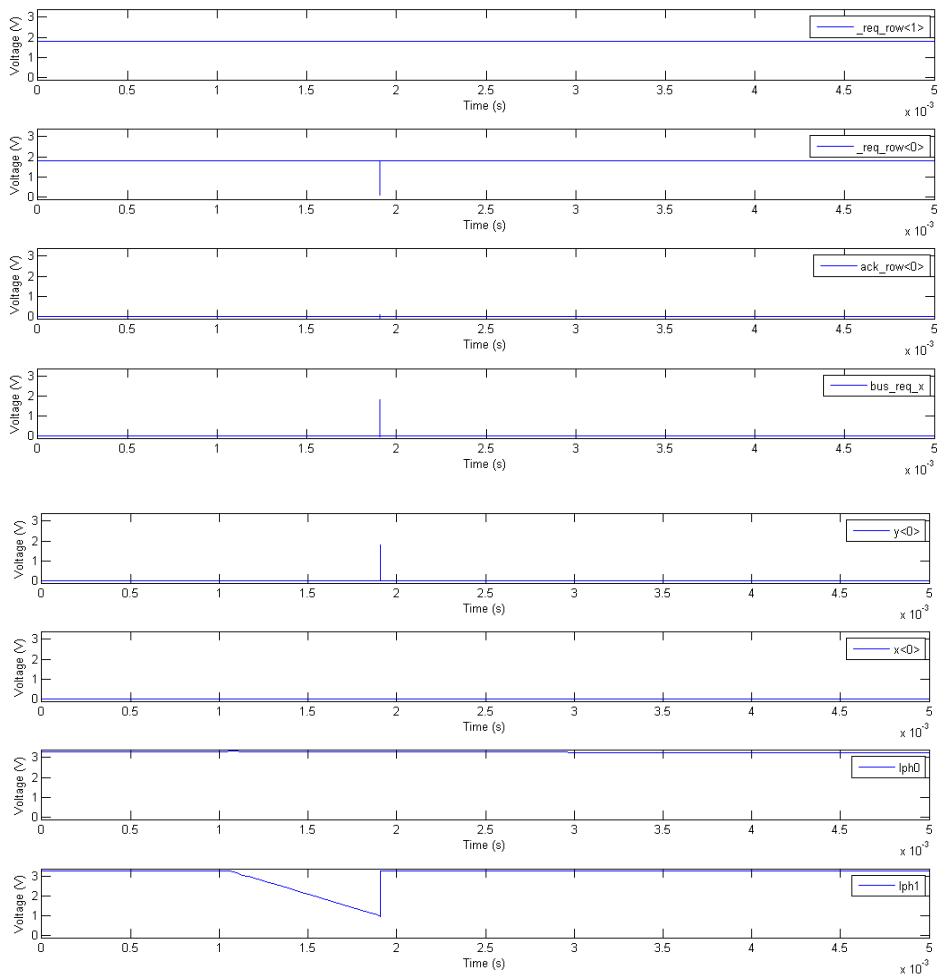


Figura 3.32: Evolución temporal de las señales de las filas circuito para $I_{ph1} = 100 \text{ pA}$, dirección del píxel (0, 1)

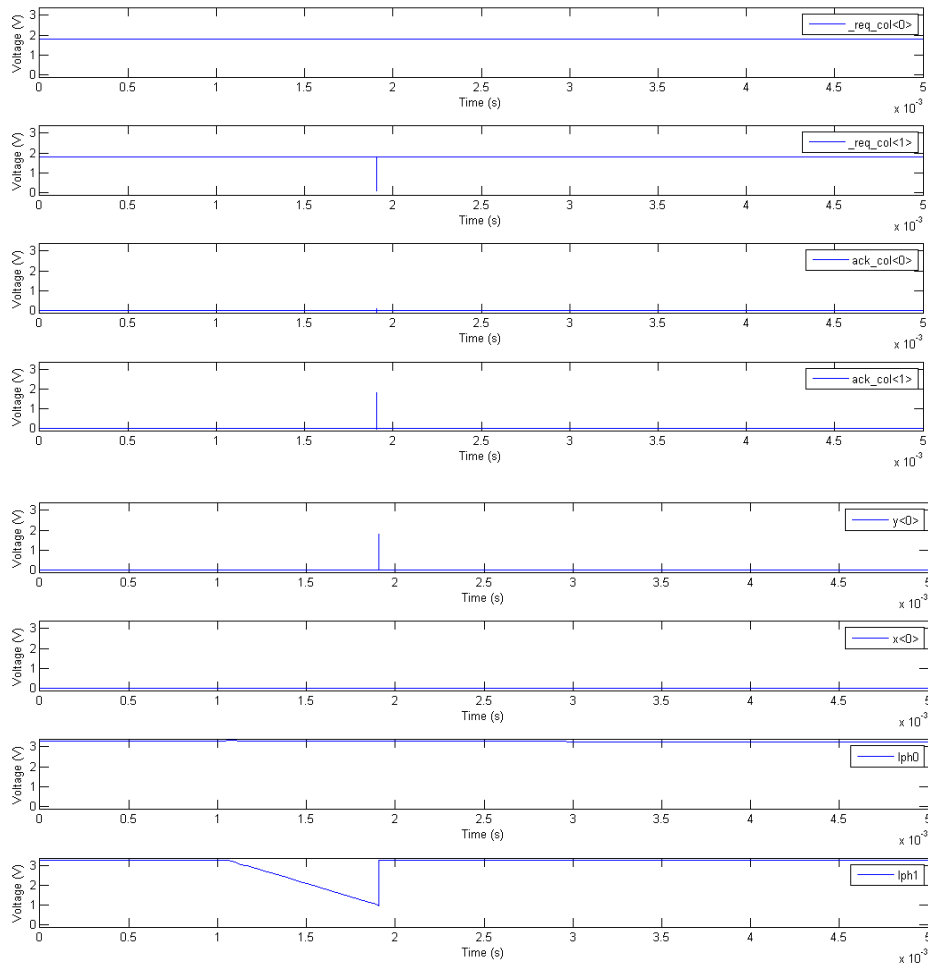


Figura 3.33: Evolución temporal de las señales de las columnas circuito para $I_{ph1} = 100 \text{ pA}$, dirección del píxel (0,1)

En la figura 3.32 y 3.33 se puede observar que la dirección del píxel (x, y) coincide con la del píxel que se estimuló con más corriente, así como que la petición y el posterior reconocimiento que se activan son los pertenecientes a la fila 0 y columna 1 del píxel.

Una vez comprobado esto, se fijan todos los valores de corriente de los píxeles a 100 pA de forma que el arbitrador tenga que resolver las posibles colisiones que se puedan producir. Se muestra el comportamiento en detalle del circuito cuando todos los píxeles son iluminados con la misma intensidad.

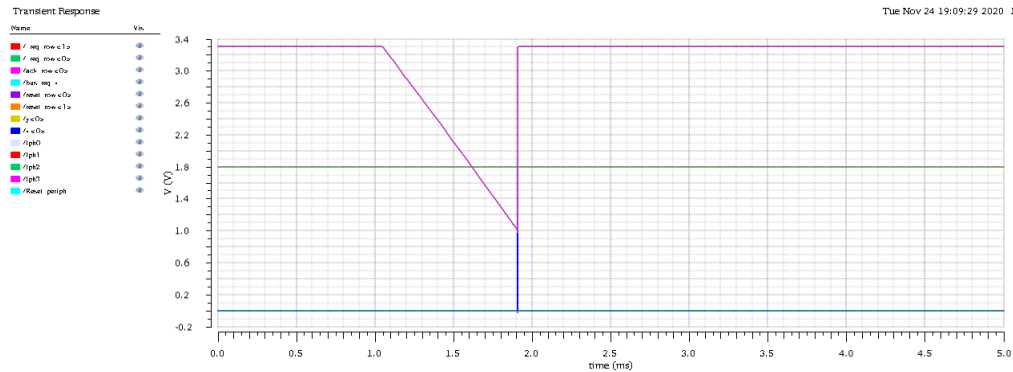


Figura 3.34: Evolución temporal de las señales del circuito cuando todos los píxeles son iluminados con la misma intensidad, dirección del píxel (1,1)

El funcionamiento del circuito es el siguiente, en el estado inicial, la señal pix_on está a nivel bajo, la tensión en el nudo $V_{ph} = 3.3 V$ y el condensador tiene almacenada carga. Cuando la tensión V_{ph} alcanza la tensión umbral (V_{th}) el comparador decide y su salida cambia de 0 a 1 (out_comp). A continuación, el píxel solicita acceso al bus de comunicación para transmitir sus coordenadas. Esto se lleva a cabo en dos pasos: primero baja la señal $_req_row$, que se transmite por la fila correspondiente de la matriz hasta la circuitería de la periferia que responde concediendo el acceso al bus y subiendo la tensión en ack_row , lo que permite que el píxel baje la señal $_req_col$. Se puede consultar mediante las señales x e y que píxel ha enviado la información. Con $_req_row$ y $_req_col$ a cero el píxel tiene acceso al bus compartido y sus coordenadas se transmiten al exterior, en este caso comparte la dirección (1,1). Por último, se suben las señales $reset_row$ y $reset_col$, que son las entradas de la puerta NAND, esta devuelve un 0 bajando a cero la tensión del nudo $lock_n$, la salida del comparador cae a cero y suben las señales $_req_row$ y $_req_col$, entonces bajan las entradas de la NAND y las tensiones $lock$ y $lock_n$ permanecen a sus valores anteriores. El píxel permanecerá en este estado hasta que la señal pix_on baje y vuelva subir (pulso). Con la bajada de pix_on sube $lock_n$ y se abre la llave y el píxel vuelve a su estado inicial listo para detectar fotones.

De igual modo que para los casos anteriores, finalmente se realiza un análisis paramétrico para ver las limitaciones de funcionamiento del circuito cuando la tensión de alimentación es baja. Se pueden obtener conclusiones anticipadas teniendo en cuenta los precedentes de los módulos anteriores. Se puede tener casi la certeza que para tensiones menores a $0.6 V$, o incluso mayores si tenemos en cuenta posibles retrasos o errores que introduce el

módulo del píxel, es muy posible que no funcione correctamente. Esto se comprueba en la siguiente figura:

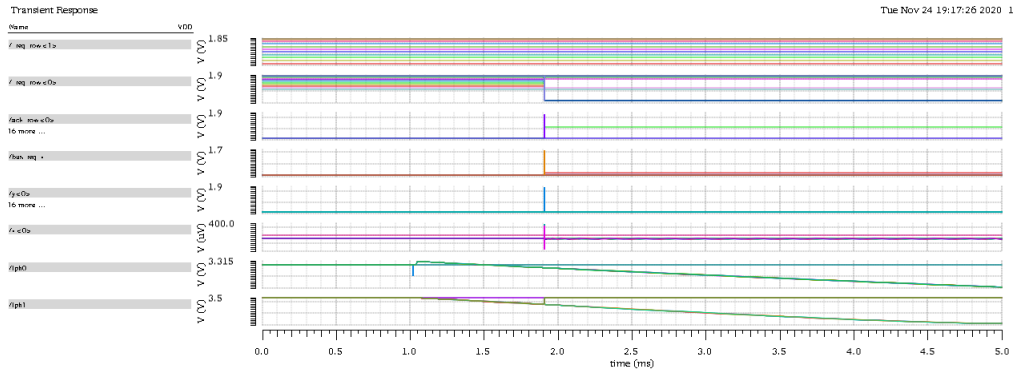


Figura 3.35: Evolución temporal de señales de la lógica de selección

En este caso, se puede analizar el comportamiento observando la señal $ack_row < 0 >$, en esta tenemos dos líneas de colores diferentes, una verde y otra morada. La señal de color verde es la señal ack cuando la tensión de alimentación $VDD = 0.84 V$ y la señal de color morada cuando la alimentación $VDD = 0.94 V$. Tal y como se observa, la señal morada representa un correcto funcionamiento del circuito mientras la verde no. Por ello, se puede concluir que el rango de funcionamiento de este circuito (píxel + arbitración) en relación con la tensión de alimentación es $[0.94 V, 1.8 V]$. Dadas las conclusiones de los puntos anteriores, se puede observar como los level shifter limitan el rango de funcionamiento de circuito completo.

Capítulo 4

Layout

En este capítulo se ha realizado el layout para saber el área que ocupa y los retrasos que introduce la periferia así como las simulaciones post-layout con parásitos capacitivos. Antes de enviar el diseño a fabricar es importante asegurar que se cumplen los siguientes puntos:

- Que el layout cumple con las reglas de diseño (DRC).
- Que el circuito representado en el layout es completamente idéntico al circuito del esquemático.
- Que no hay errores de antena que puedan dañar las puertas de los transistores.

Para realizar el layout y verificar los puntos anteriores se utiliza la herramienta de extracción y verificación de *Cadence* llamada *Calibre*. Esta herramienta, integrada en el entorno de diseño de *Cadence Virtuoso*, se utiliza para la verificación de las reglas de diseño (DRC), la comprobación layout vs esquemático (LVS) y la extracción parásita. Gracias a la extracción de parásitos y la extracción de parámetros reales del circuito se puede obtener un modelo eléctrico que refleje fielmente el diseño del circuito y asegure la precisión de la simulación y los análisis posteriores al diseño. Los beneficios obtenidos con el uso de *Calibre* son el ahorro de tiempo, esfuerzo y dinero.

Para verificar que el layout cumple con las reglas de diseño se necesita pasar el DRC (Design Rule Check). Para realizar la comprobación de DRC es necesario proporcionarle a *Calibre* las reglas de DRC de la tecnología usada. Normalmente este fichero sería proporcionado por la fábrica que se encarga de la fabricación del chip o circuito, este documento recibe el nombre DRM (Design Rule Manual). En este documento se encuentran entre otras las distancias mínimas entre capas o pistas de metal, el número mínimo de

Por último, se comprueba que el diseño no va a sufrir daños por el efecto antena. Hay que tener en cuenta que cuando se construyen las pistas de metal para conectar los dispositivos, estas se conectan a las puertas de poli silicio de los transistores que pueden permanecer flotantes hasta que sea depositada la capa de metal inmediatamente superior. Si las pistas son lo suficientemente largas y no están cubiertas por una capa de óxido, pueden permanecer como un condensador absorbiendo la carga durante las etapas de fabricación. Esta energía acumulada se puede descargar en forma de corriente y ocasionar daños permanentes como la rotura del óxido de puerta. En el efecto antena la cantidad de carga acumulada es proporcional al área de metal expuesto. A continuación, se muestra una imagen descriptiva de lo anterior:

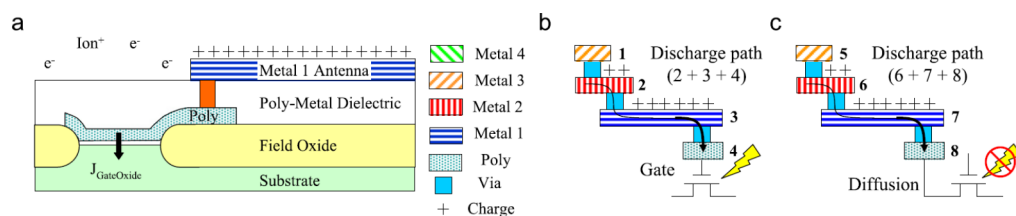


Figura 4.3: Efecto antena: (a) esquema de daño por etching de plasma, (b) la carga acumulada en los paths 2-4 se descarga a través del óxido de puerta causando daños sobre este, (c) las carga acumulada en los paths 5-8 se descarga a través de la difusión y no causa daño.

En la figura 4.3 se puede observar como las pistas se fabrican en el siguiente orden: poly, metal 1, metal 2, metal 3 y metal 4. Los *paths* del 2 al 4 pueden acumular la carga suficiente para crear la corriente de descarga y dañar el óxido. En la figura 4.3 (c) las pistas de material conductor se conectan a una difusión, los *paths* del 6 al 8 se descargan a través de la difusión y no causa daños.

Para reducir el efecto antena existen tres soluciones muy populares y utilizadas actualmente:

- **Salto de metal:** se abre el camino del metal problemático y se completa la conexión con el metal superior.
- **Niveles altos de metal:** se usan niveles altos de metal para el camino problemático.
- **Conectar un diodo:** se conectan diodos a la puerta de los transistores y se colocan cerca, de tal forma que estos drenan la carga manteniendo el nivel de tensión.

A la hora de hacer el layout, hay que tener en cuenta que *Cadence* dispone normalmente de 5 ó 6 tipos de capas de metales para las tecnologías modernas. No obstante, este número de metales disponibles dependerá de la tecnología utilizada. Si se utilizan aleatoriamente sin definir un criterio, en función de la complejidad del diseño, puede ser muy problemático para conseguir un layout óptimo. Es por ello que antes de comenzar con los diseños vamos a definir el criterio utilizado para los mismos:

- **Primer nivel de metal (ME1)**: se ha definido el uso de este metal para las conexiones entre terminales de los transistores y contactos a sustratos.
- **Segundo nivel de metal (ME2)**: este nivel de metal se fabrica encima del ME1. Se va a utilizar para algunas conexiones entre transistores y para algunas señales como *Reset_periph*, *_bus_ack* y *bus_req*. Pero en definitiva, su uso se restringe al plano y, es decir, será posicionado verticalmente.
- **Tercer nivel de metal (ME3)**: su utilización se restringe a conexiones horizontales (plano x) y a señales globales como *req0*, *req1*, *ack0*, *ack1*, *req*, *ack*, *res_pix_arb_req* y *arb_ack*.
- **Cuarto nivel de metal (ME4)**: se usa para hacer una malla del nudo de referencia (VSS).
- **Quinto nivel de metal (ME5)**: se usa para crear una malla de alimentación (VDD).

El quinto de nivel de metal no se ha utilizado ya que no ha sido necesario debido a la estructura del diseño de la periferia. Por tanto, queda reservado para posibles mejoras u optimización de la misma cómo crear una malla de alimentación. El sexto nivel de metal (ME6) tampoco ha sido utilizado, se reserva para apantallar los diseños y evitar que se vean afectados por la luz.

La idea inicial de este trabajo fue crear el layout de cada una de las puertas por separado de manera que se pudieran integrar en el posterior diseño de los bloques de la periferia. Con ello se conseguiría tener módulos estándar y se reduciría el tiempo de realización de layout. Por este motivo, inicialmente se realizaron dichos módulos básicos (inversor, NAND, NOR y latch SR). Posteriormente, dado que la altura del layout estaba limitada a $10\ \mu\text{m}$ para poder arbitrar píxeles de este ancho, se llegó a la conclusión de que con los módulos creados era imposible cumplir dicha condición y se insertaron los transistores directamente sobre el diseño con el objetivo

de cumplir esta restricción de diseño. Se procede a mostrar y explicar los diseños de layout de cada uno de los módulos de la periferia.

4.1. Arbitrador

Al igual que en secciones anteriores, el arbitrador se conecta por la izquierda en paralelo con el multiplexor y este a su vez con la lógica de selección y por la derecha con otro arbitrador o con el mismo si se trata del último arbitrador. La restricción de diseño a tener en cuenta en este módulo es que la altura del bloque tiene que estar limitada a $10\ \mu\text{m}$ para poder arbitrar píxeles de un ancho mínimo de $10\ \mu\text{m}$, sin embargo, la anchura no tiene limitación de tamaño alguno. En el diseño actual, que se muestra a continuación, se ha conseguido lograr un tamaño del rectángulo de $9.8\ \mu\text{m} \times 14.5\ \mu\text{m}$ (alto x ancho). En el diseño se han seguido las reglas descritas anteriormente, las señales *req0*, *req1*, *ack0*, *ack1*, *req*, *ack* se han ruteado con el metal 3.

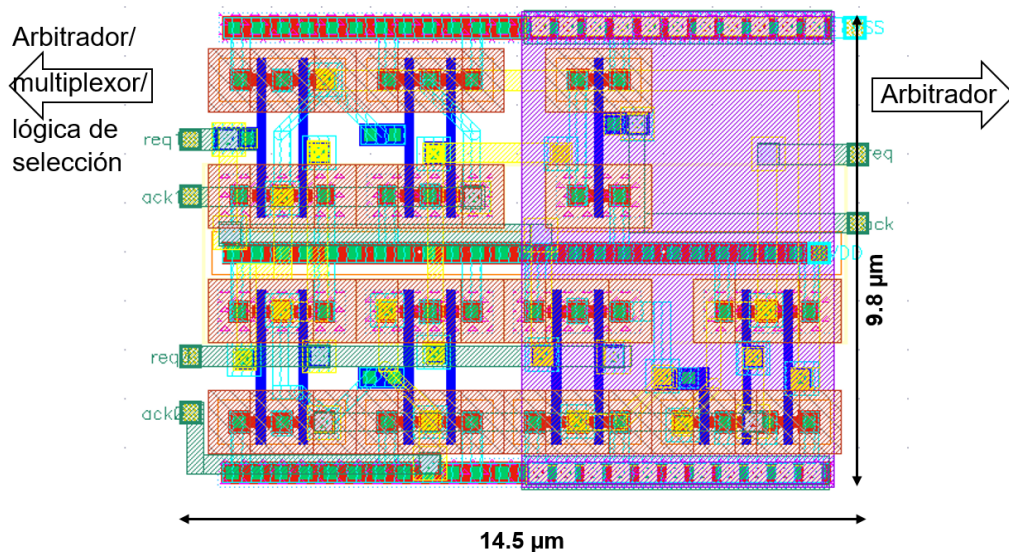


Figura 4.4: Layout del arbitrador

En esta celda, teniendo en cuenta las simulaciones y conclusiones obtenidas en el capítulo 3 que mostraban que para bajas tensiones de alimentación se puede llegar a tener retrasos significativos en las señales, se procede a obtener las capacidades de dicho módulo. Seguidamente se muestra el valor de las capacidades parásitas obtenidas en el bloque del arbitrador y después se obtendrá el retraso del mismo para tensiones bajas de alimentación. Estas

capacidades se obtienen con la opción de Calibre RVE que abre el fichero de datos *.svdb* que se genera mediante un PEX (Parasitic EXtration). Para que PEX funcione el análisis LVS debe haber sido correcto, así los parásitos extraídos pueden correlacionarse con las redes en el esquemático. Con esta opción se obtienen las capacidades parásitas de cada nodo del circuito tal y como se muestra a continuación.

No.	Layout Net	Source Net	R Count	C Total (F)	CC Total (F)	C+CC Total (F)
1	req1	NET40	49	1.27699E-15	4.41242E-15	5.78941E-15
2	req1	REQ1	30	8.35703E-16	2.28112E-15	3.09682E-15
3	req0	REQ0	21	7.03174E-16	1.96033E-15	2.66350E-15
4	4	4	45	1.09823E-15	4.53426E-15	5.63249E-15
5	5	5	31	7.78362E-16	2.13507E-15	2.91403E-15
6	ack	ACK	13	3.66777E-16	1.31906E-15	1.67584E-15
7	7	7	22	3.56725E-16	1.26270E-15	1.61942E-15
8	vss	VSS	98	1.40373E-15	5.94262E-15	7.34634E-15
9	9	NET037	2	9.34153E-18	2.25862E-16	2.35204E-16
10	10	10	2	1.09060E-17	1.75821E-16	1.86727E-16
11	ack0	ACK0	21	1.51885E-16	1.45077E-15	1.60265E-15
12	12	12	2	7.20966E-18	2.45899E-16	2.53085E-16
13	13	NET042	2	7.19027E-18	2.49594E-16	2.56791E-16
14	ack1	ACK1	14	1.54007E-16	1.73453E-15	1.88858E-15
15	15	15	2	1.25004E-17	1.69598E-16	1.82098E-16
16	req	REQ	20	1.86347E-16	2.03257E-15	2.21892E-15
17	17	17	2	5.37781E-18	2.36875E-16	2.42253E-16
18	18	18	2	5.21598E-18	2.33286E-16	2.37504E-16
19	VDD	VDD	69	2.70032E-16	1.62099E-15	1.89101E-15

Figura 4.5: Capacidades por nodo del bloque arbitrador

Este tipo de análisis extrae las resistencias y capacidades entre los nodos. Como se puede observar en la imagen 4.5 tenemos varios tipos de capacidades, pero en definitiva "C" significa que una capacidad parásita se extrae a tierra o lo que es lo mismo, que todas las capacidades parásitas en ese nodo se suman y se conectan a tierra. "CC" es el tipo de capacidad más complejo de extraer y su funcionamiento es que se extrae un condensador de acoplamiento para cada par de nodos. Hay muchas más opciones que pueden explorar en PEX, por ejemplo, solo para la extracción de parásitos se puede modificar que en lugar de obtener la capacidad por nodo se obtenga para múltiples no o se puede cambiar el nodo de referencia.

Para crear el banco se ha generado uno nuevo, pero con las mismas entradas y características que el mostrado en la figura 3.13 con el objetivo de poder comparar los resultados con los obtenidos sin las capacidades. Se ha realizado un análisis paramétrico (alimentación desde 0.2 V a 1.8 V con un salto de 0.05 V) para estudiar como varían las salidas para los diferentes valores de alimentación.

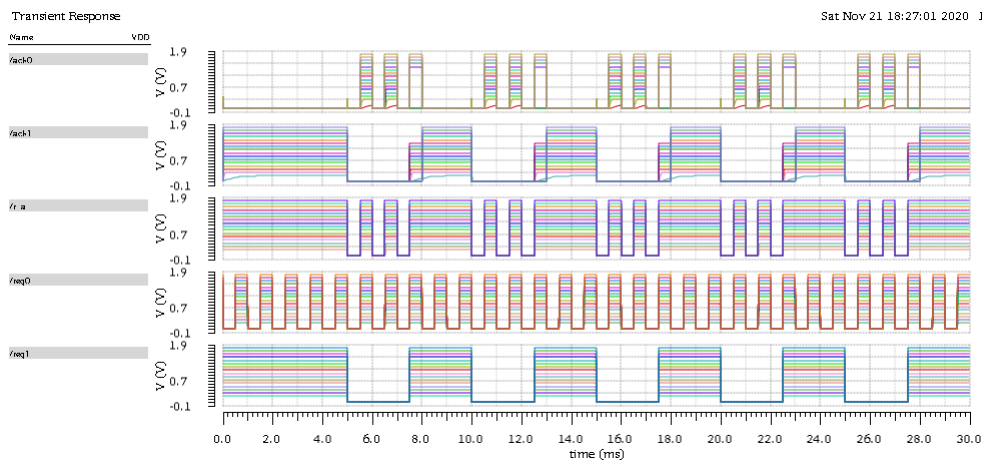


Figura 4.7: Evolución temporal de señales del arbitrador con capacidades parásitas

De la figura 4.7 se puede observar que para tensiones bajas de alimentación el retraso introducido es mayor que en el caso del arbitrador sin capacidades parásitas. A partir de $V_{DD} = 0.4\text{ V}$, el retraso que se produce es significativo, en concreto, de alrededor de 0.2 ms . Para valores inferiores como $V_{DD} = 0.2\text{ V}$ el retraso es mucho mayor (aproximadamente 2 ms). Para valores superiores a $V_{DD} = 0.4\text{ V}$ ese retraso no se produce. Por lo tanto, se puede concluir que el arbitrador funciona bien en un rango de alimentación de $[0.45\text{ V}, 1.8\text{ V}]$ y que las capacidades parásitas para bajos valores de alimentación hacen que el retraso que se produzca sea el doble que sin ellas. Se puede obtener el retraso total que introduce el árbol completo sumando los retrasos de cada etapa.

4.2. Multiplexor

Este bloque se conecta por la izquierda, en paralelo con la lógica de selección y por la derecha con el arbitrador. Para el diseño del layout del multiplexor se han creado los layout de cada uno de los bits por separado. Se tiene en cuenta que para crear el multiplexor los módulos de cada bit irán

unidos los unos a los otros, por ello, ha sido necesario realizar el layout de forma que ambos tengan el mismo tamaño y exactamente la misma posición de los metales para facilitar la posterior conexión de los mismos. El tamaño del bloque es $10\ \mu\text{m} \times 7\ \mu\text{m}$ (alto x ancho).

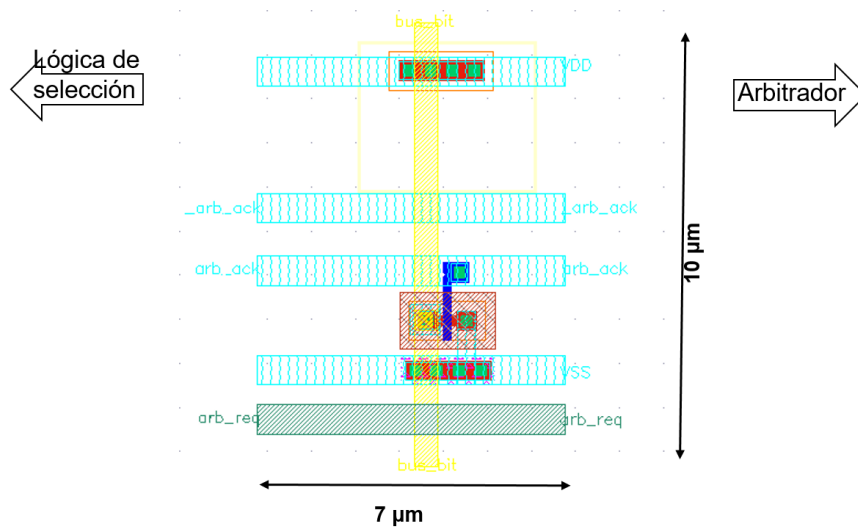


Figura 4.8: Layout del módulo bit cero

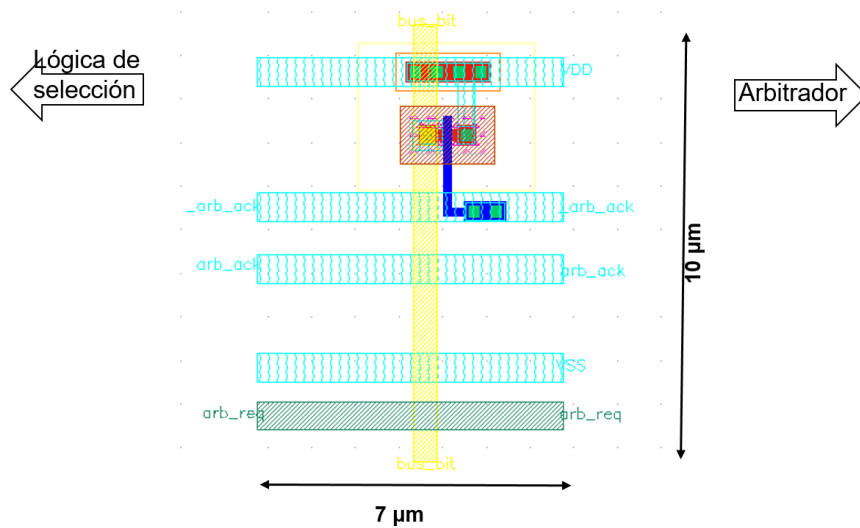


Figura 4.9: Layout del módulo bit uno

4.3. Lógica de selección

Este circuito se conecta por la izquierda con un píxel y por la derecha con el multiplexor. Las restricciones de diseño son que la altura del bloque tiene que estar limitada a $10\ \mu\text{m}$, sin embargo, su anchura no tiene limitación de tamaño al igual que en el arbitrador. En el diseño actual, que se muestra a continuación, se ha conseguido lograr un tamaño del rectángulo de $9.98\ \mu\text{m}$ \times $15.7\ \mu\text{m}$ (alto \times ancho).

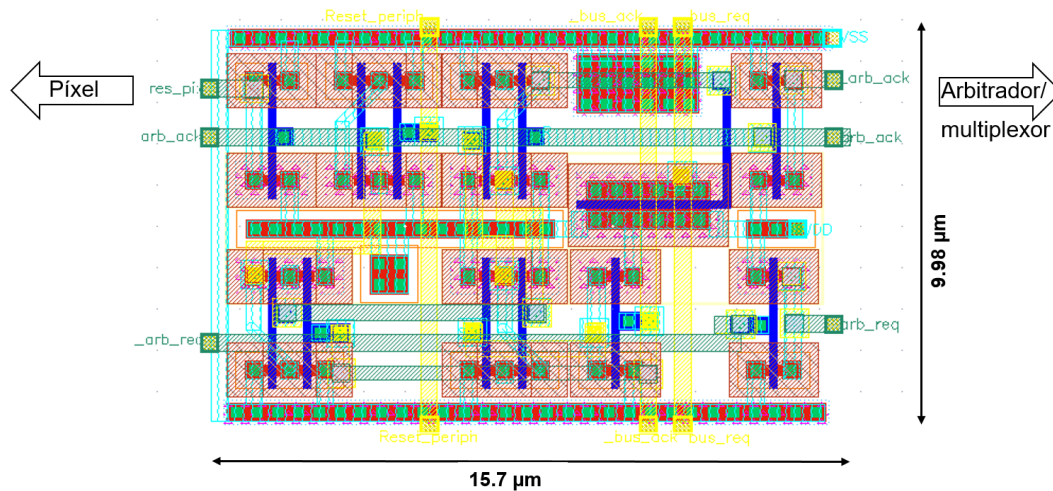


Figura 4.10: Layout de la lógica de selección

4.4. Periferia

En el layout del bloque periferia, se tiene la siguiente jerarquía, 128 bloques de lógica de selección, el multiplexor de 7 bits y el árbol de arbitradores compuesto por 127 arbitradores. El bloque tiene una altura de $1720\ \mu\text{m}$ y una anchura máxima de $258\ \mu\text{m}$ en el centro y algo menor en los extremos. En la siguiente figura se muestra el layout y una ampliación del mismo en la que se puede distinguir lo descrito.

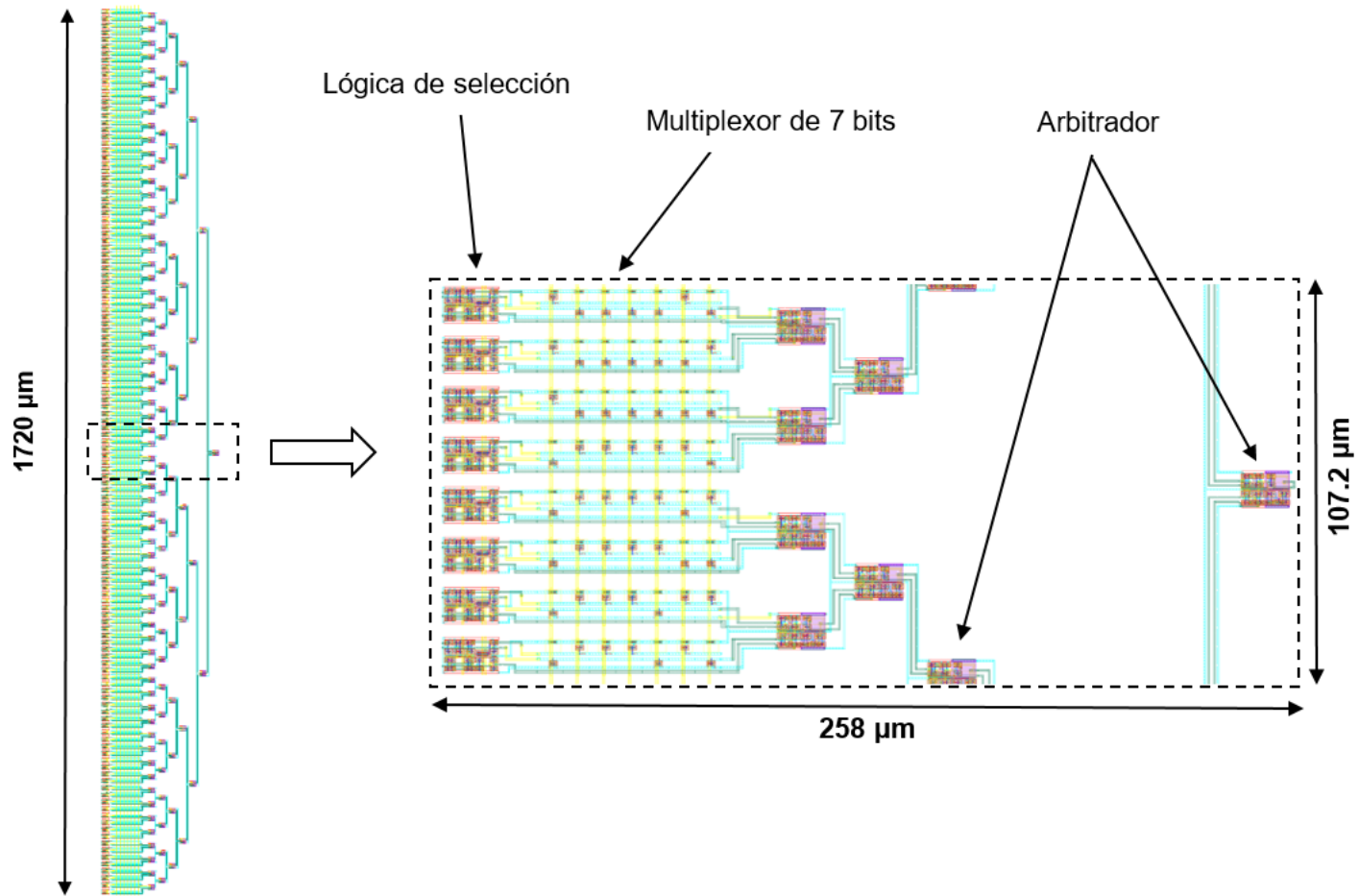


Figura 4.11: Layout del bloque periferia y ampliación de la zona central.

Por último, se muestra un esquema de la estructura de conexión de la periferia a la matriz de píxeles para comprender mejor la estructura realizada del layout:

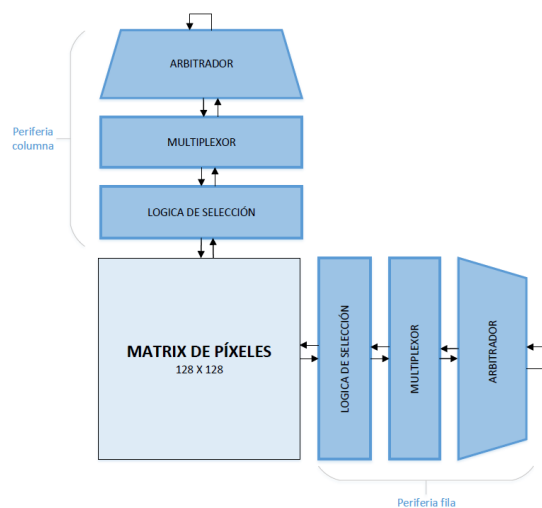


Figura 4.12: Diagrama de la posición de la periferia respecto a la matriz de píxeles

Conclusiones generales y líneas de futuro

Conclusiones del proyecto

En este proyecto se ha diseñado la circuitería de periferia con tecnología UMC 0.18 (CMOS) para un procesar la información de un sensor solar, se ha comenzado con el diseño de cada uno de los bloques más básicos hasta llegar al diseño de la periferia completa. Se ha realizado una simulación transitoria mediante un Cadence Virtuoso de cada circuito y en los bloques de la periferia se ha realizado un análisis paramétrico en función de la tensión de alimentación con el fin de estudiar como afecta dicha tensión al comportamiento del circuito. Los aspectos más importantes se describen a continuación:

- Estudio de los tipos de sensores solares que existen en el mercado.
- Estudio del protocolo de comunicación AER.
- Análisis del funcionamiento de cada uno de los bloques que componen la periferia, así como de las ventajas e inconvenientes que poseen respecto otros tipos.
- Desarrollo de varios dispositivos electrónicos, en concreto de arbitradores, multiplexores, lógica de selección: se han realizado desde la primera fase (esquemático) hasta la última (layout).
- Preparación de bancos de pruebas para estudiar todos los posibles casos en cada módulo.
- Estudio de la variación temporal de las señales de cada módulo.
- Verificación de DRC, Layout vs Schematic de cada uno de los módulos y extracción de parásitos del bloque arbitrador.

- Generación de un circuito arbitrador introduciendo las capacidades parásitas para obtener el retraso que puede introducir una alimentación baja en el comportamiento de dicho dispositivo.

Líneas de futuro

Teniendo en cuenta los resultados obtenidos en este trabajo, son varios los aspectos que podrían abordarse de cara a estudios posteriores.

- Fabricar la circuitería periférica, testarla y comprobar si los resultados coinciden con los de simulación.
- Realizar un estudio más exhaustivo para conseguir mejorar el rango de funcionamiento de los bloques level shifter con el fin de conseguir que el circuito funcione para tensiones de alimentación menores a 0.6 V. Estos bloques introducen una limitación real de funcionamiento del circuito, no la periferia en sí.
- Estudiar como se puede reducir el retraso introducido por el módulo arbitrador para tensiones de alimentación menores a 0.4V.
- Realizar una optimización del layout del circuito.
- Comprobar que la circuitería es tolerante a radiación espacial.

Bibliografía

- [1] Boahen, K. A. Point - to - point connectivity between neuromorph chips using address events. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47 (5), 416 - 434, 2000.
- [2] Boahen, K. A. Retinomorph Vision Systems: Reverse Engineering the Vertebrate. *Cal. Inst. of Tech*, 1996.
- [3] Culurciello, E., Etienne - Cummings, R. and Boahen, K. A. A Biomorph Digital Image Sensor. *IEEE J. Solid-State Circuits*, 38 (2), 281 - 294, 2003.
- [4] De la Calle Martos, A. Diseño de un sensor solar basado en eventos *Trabajo Fin de Máster*, 2019.
- [5] De la Calle Martos, A. Gomez - Mechán, R., Leñero - Bardallo, J. A., Rodríguez - Vázquez, A. Sun tracker sensor for attitude control of space navigation systems. *Society for Imaging Science and Technology*, 2020.
- [6] Delgado, F. J., García, J., Tarrida, C. L., Quero, J. M., Ortega, P., Térés, L., Serra - Graells, F. and Cabruja, E. Integrated design of a smart analog sun sensor with CMOS technology. *2012 IEEE International Conference on Industrial Technology*, 184 - 188, 2012.
- [7] Fossum, E. R. Active Pixel Sensors: Are CCD's Dinosaurs?. *Jet Propulsion Laboratory, California Institute of Technology*, 1992.
- [8] Gomez - Mechán, R. Diseño de un sensor de imagen asíncrono autoalimentado mediante captación de energía solar. *Trabajo Fin de Máster*, 2020.
- [9] Guo, X., Qi, X. and Harris, J. G. A Time - to - First - Spike CMOS Image Sensor. *IEEE Sensor Journal*, 7 (8), 165 - 1175, 2007.
- [10] Wu, H. W., Emadi, A., de Graaf, G., Leijtens J. and Wolffenbuttel, R. F. Design and fabrication of an albedo insensitive analog sun sensor, 2011.

- [11] Jenkins, A. The Sun's position in the sky. *University of Costa Rica*, 2013.
- [12] Leñero - Bardallo, J. A. Bioinspired Processing, Algorithms and Circuits. *Second Thematic Block*, 2019.
- [13] Leñero - Bardallo, J. A., Carmona - Galán, R., Rodríguez - Vázquez, A. A high dynamic range image sensor with linear response based on asynchronous event detection. *2015 European Conference on Circuit Theory and Design (ECCTD)*, 1 - 4, 2015.
- [14] Leñero - Bardallo, J. A., Farian, L., Guerrero - Rodríguez, J. M., Carmona - Galán, R., Rodríguez - Vázquez, A. Sun sensor based on a luminance spiking pixel array. *IEEE Sensors Journal*, 17 (20), 6578 - 6588, 2017.
- [15] Leñero - Bardallo, J. A., Farian, L., Guerrero - Rodríguez, J. M., Carmona - Galán, R., Rodríguez - Vázquez, A. A sun sensor implemented with an asynchronous luminance vision sensor. *ESSCIRC 2017 - 43rd IEEE European Solid State Circuits Conference*, 67 - 70, 2017.
- [16] Leñero - Bardallo, Pérez - Peña, R., Carmona - Galán, R., Rodríguez - Vázquez, A. Pipeline AER arbitration with event aging. *2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD*, 1 - 4, 2017.
- [17] Lichtsteiner, P., Posch, C. and Delbruck, T. A 128 x 128 120 dB 15 μ s Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits*, 43 (2), 566 - 576, 2008.
- [18] Mahowald, M. An analog VLSI system for estereospic vision. *Springer Science and Business Media*, vol. 256, 1994.
- [19] Martínez Álvarez, A. and Lombardi Gómez, D. J. Caracterización de dispositivos fotovoltaicos espaciales. *Trabajo de grado*, 2009.
- [20] Mead, C. and Mahowald, M. A Silicon Model of Early Visual Processing. *New York: Pergamon*, 1998.
- [21] Mueggler, E., Gallego, G. and Scaramuzza, D. Continuous - Time Trajectory Estimation for Event - based Vision Sensors. *University of Zurich*, 2015.

- [22] Ortega, J. G., Tarrida, C. L., Quero, J. M., Delgado, F. J., Ortega, P., Castaner, L., Reina, M., Angulo, M., Morilla, Y. and Lopez, J. G. MEMS solar sensor testing for satellite applications. *2009 Spanish Conference on Electron Devices*, 345 - 348, 2019.
- [23] Rivas Pérez, M., Gómez Rodríguez, F., Paz Vicente, R., Linares Barranco, A., Vicente Díaz, S. and Cascado Caballero, D. Tools for Address - Event - Representation Communication Systems and Debugging. *International Conference on Artificial Neural Networks. Universidad de Sevilla*, 289 - 296, 2005.
- [24] Shoushun, C. and Bermak, A. A low power CMOS imager based on time - to - first - spike encoding and fair AER. *2005 IEEE International Symposium on Circuits and Systems*, 5306 - 5309, vol. 5, 2005.
- [25] Sedra Smith, A. and Carless Smith, K. *Microelectronic Circuits. 6th Edition*, 2009.
- [26] Silvilotti, M. *Wiring Considerations in Analog VLSI Systems with Application to Field - programmable Networks. Cal. Inst. of Tech*, 1991.
- [27] Subbarao, G. A., and Häfliger, P. D. Fair Arbitration for Multi - Instrument Space Probe. *2017 15th IEEE International New Circuits and Systems Conference (NEWCAS)*, 185 - 188, 2017.
- [28] Thu, A. M., Tuan, D. A., Shoushun, C. and Seng, Y. K. Adaptive priority toggle asynchronous tree arbiter for AER - based image sensor. *2011 IEEE/IFIP 19th International Conference on VLSI and System - on - Chip, Hong Kong*, 66 - 71, 2011.
- [29] Xie, N., Theuwissen, A. J. P., Büttgen, B., Hakkesteegt, H., Jasen, H., and Leijten, J. Micro - Digital Sun Sensor: an Imaging Sensor for Space Applications. *2010 IEEE International Symposium on Industrial Electronics*, 3362 - 3365, 2010.
- [30] La simulación computarizada como herramienta didáctica de amplias posibilidades. http://www.rcim.sld.cu/revista_18/articulos_htm/simulacioncomputarizada.htm Last accessed: August 2020.
- [31] Cadence. https://www.cadence.com/en_US/home.html Last accessed: Sep 2020.
- [32] Solar MEMS. <https://www.solar-mems.com> Last accessed: Sep 2020.

- [33] El efecto fotoeléctrico. <https://www.redalyc.org/pdf/104/10413309.pdf> Last accessed: Oct 2020.
- [34] 7,000 Small Satellites to be Launched over Coming Decade. https://euroconsult-ec.com/6_August_2018 Last accessed: Nov 2020.

Apéndice A

Esquemáticos

En este apéndice se mostrarán los esquemáticos realizados en *Cadence Virtuoso* ampliados

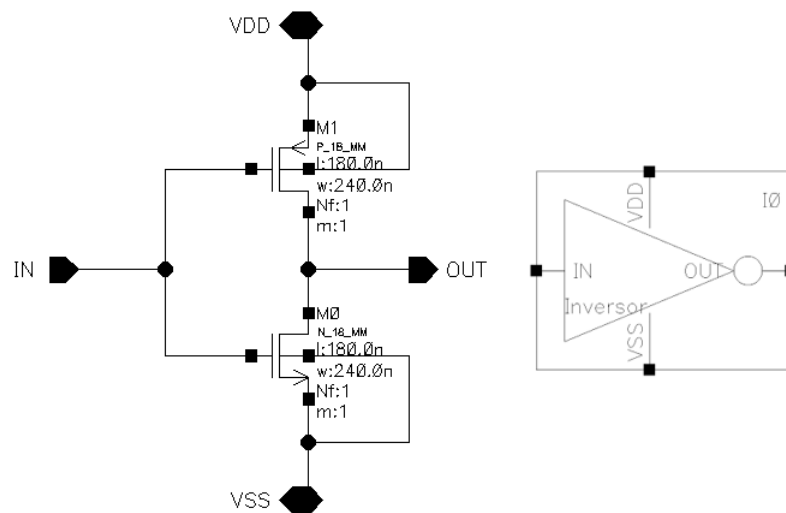


Figura A.1: Esquemático y símbolo de un inversor con tecnología UMC 0.18 μm (CMOS)

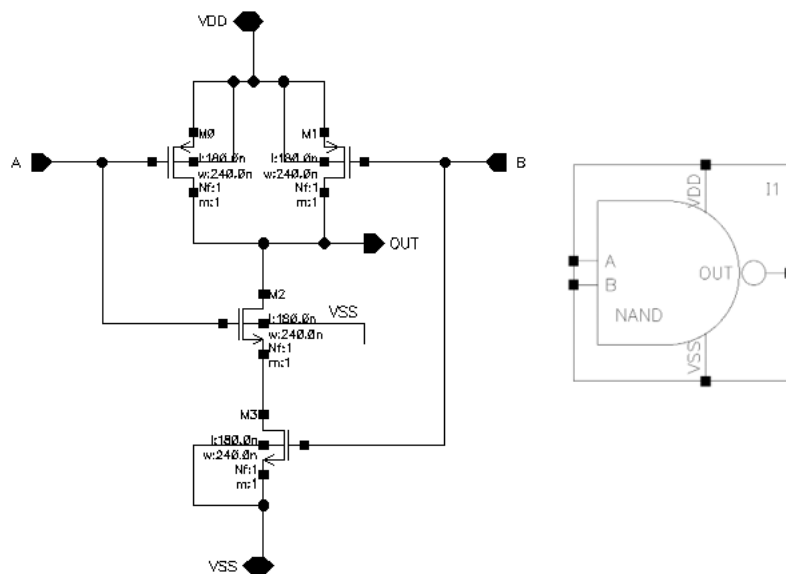


Figura A.2: Esquemático y símbolo de una puerta NAND con tecnología UMC 0.18 μm (CMOS)

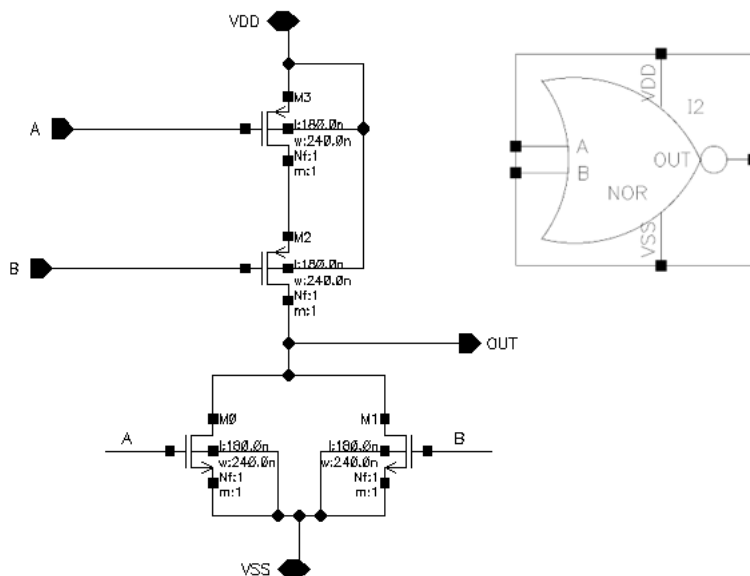


Figura A.3: Esquemático y símbolo de una puerta NOR con tecnología UMC 0.18 μm (CMOS)

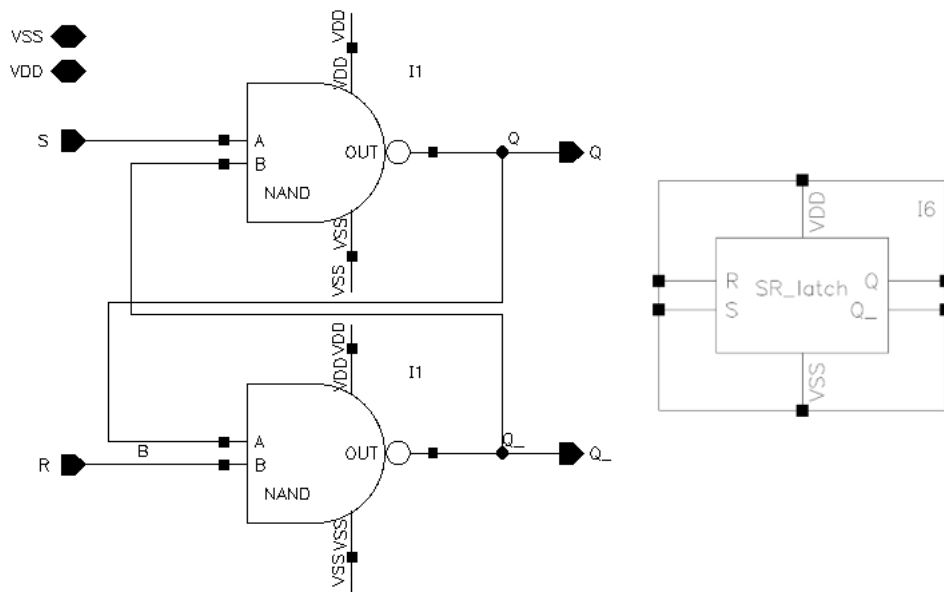


Figura A.4: Esquemático y símbolo del latch SR con tecnología UMC 0.18 μm (CMOS)

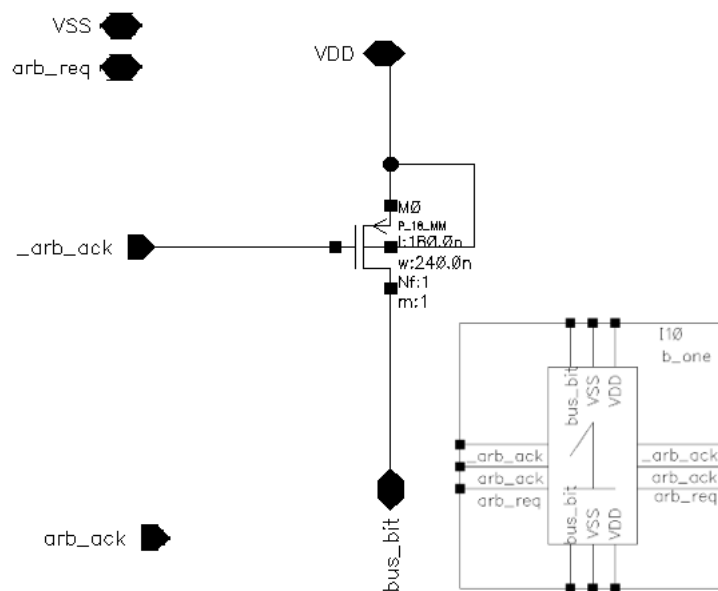


Figura A.5: Esquemático y símbolo del bit "1" con tecnología UMC 0.18 μm (CMOS)

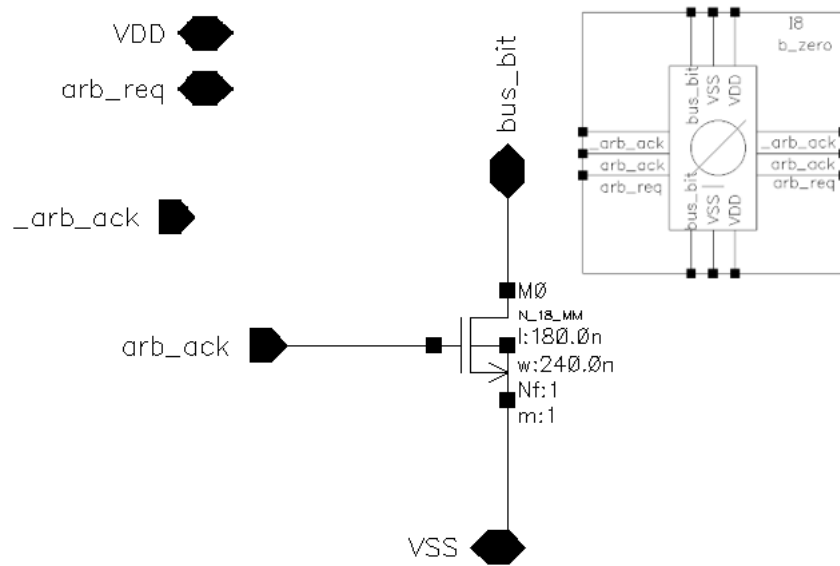


Figura A.6: Esquemático y símbolo del bit "0" con tecnología UMC 0.18 μm (CMOS)

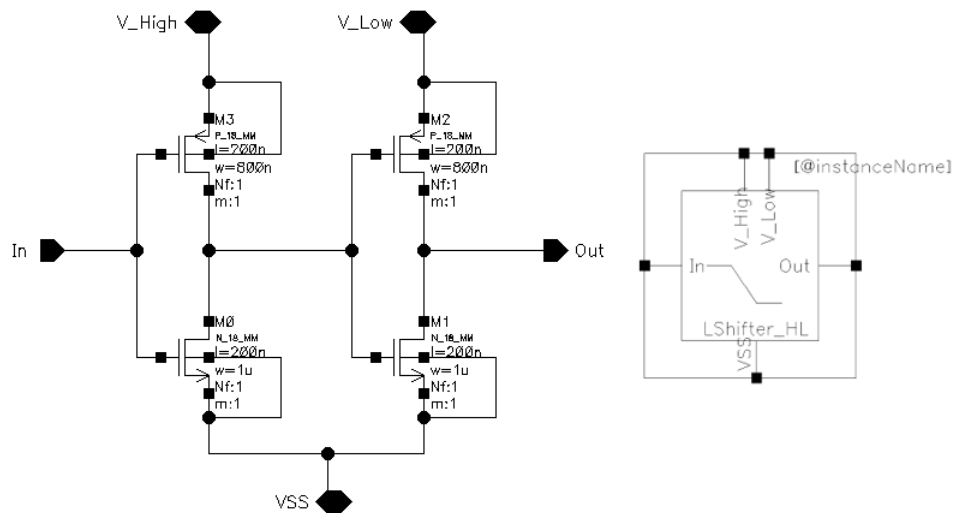


Figura A.7: Esquemático y símbolo del level shifter de high a low con tecnología UMC 0.18 μm (CMOS)

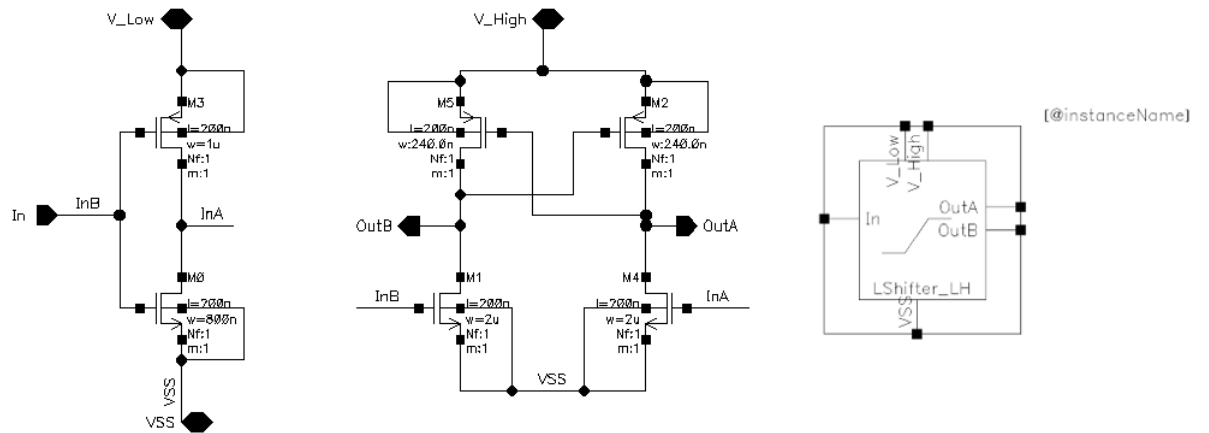


Figura A.8: Esquemático y símbolo del level shifter de low a high con tecnología UMC 0.18 μm (CMOS)

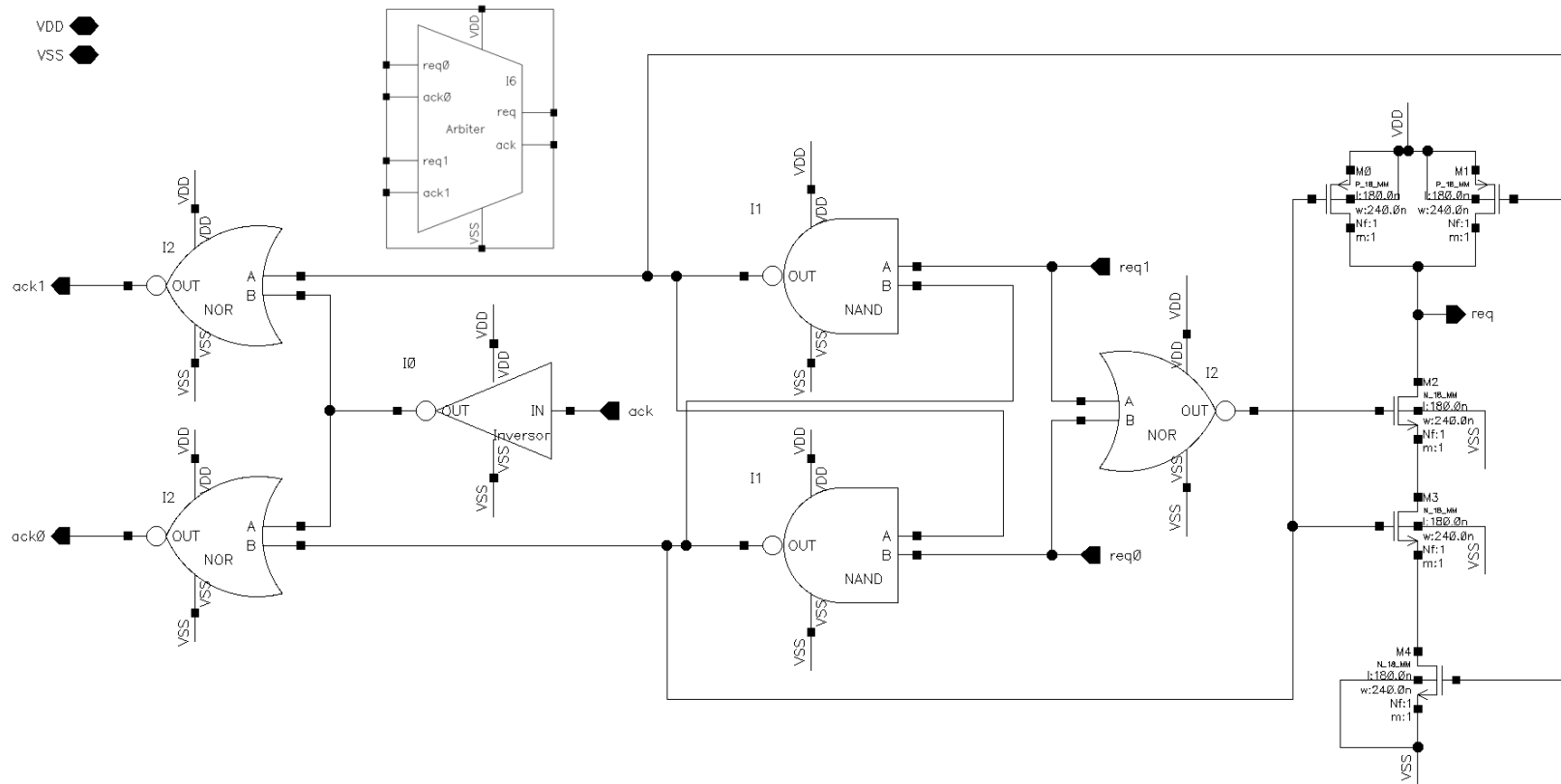


Figura A.9: Esquemático y símbolo del módulo arbitrador con tecnología UMC 0.18 μm (CMOS)

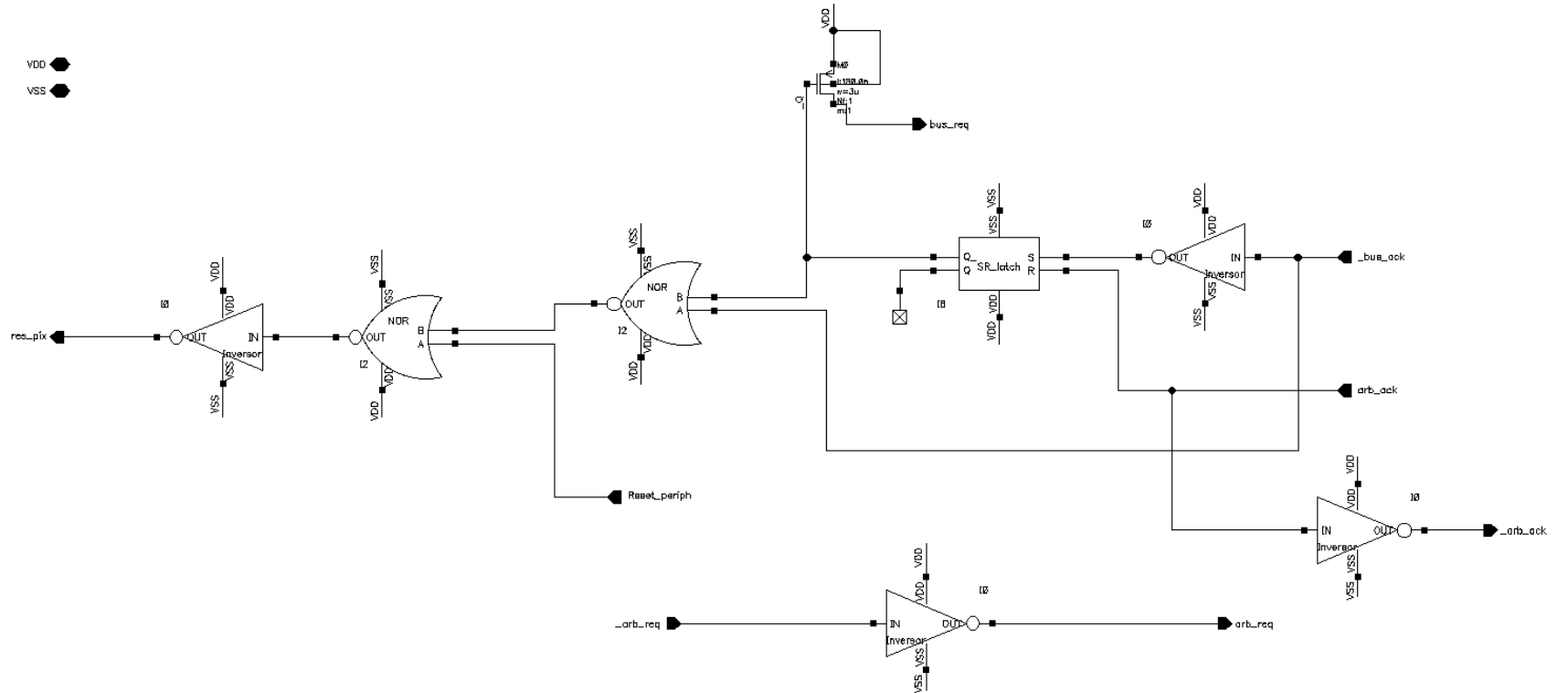


Figura A.11: Esquemático del módulo de la lógica de selección con tecnología UMC 0.18 μm (CMOS)