

TRABAJO FIN DE GRADO



Layout de canales de lectura por columna
para sensores de imagen CMOS:
técnicas, extracción y simulación

Autor:

Carlos Molina Ordóñez

Tutores:

Fernando Medeiro Hidalgo

Belén Pérez Verdú

25 de junio de 2018

En agradecimiento

*a Fernando Medeiro Hidalgo,
por lo mucho que aprendí de tí durante el tiempo que compartimos.
Siempre en la memoria.
D.E.P.*

*a mis padres y hermana,
por apoyarme siempre durante estos años.*

Índice general

Índice de figuras	v
Resumen	vi
Objetivos y metodología	1
1. Introducción	3
1.1. Sensores de imagen	3
1.2. Arquitectura de sensores de imagen CMOS	5
1.2.1. Array de píxeles	5
1.2.2. Canal de lectura	7
1.2.3. Otros bloques	7
2. Tecnología CMOS	9
2.1. Introducción	9
2.2. Proceso de fabricación	9
2.3. Transistores CMOS	11
2.4. Otros dispositivos en tecnología CMOS	12
2.4.1. Resistencias	13
2.4.2. Condensadores	13
2.5. Diseño de layout	14
2.5.1. Capas de layout	14
2.5.2. Herramientas de CAD	16
2.5.3. Problemas habituales en el diseño de layout	18
3. Diseño electrónico de un canal de lectura	21
3.1. Estructura general	22
3.2. Operación de lectura	23
3.3. Fuente de corriente	25
3.4. ADC	26

3.4.1. OTA	28
3.5. Generador de rampa analógica	29
3.6. Bloques de polarización	29
3.6.1. PXCS Bias	30
3.6.2. ADC Bias	30
3.7. Rampa digital y serialización	30
4. Diseño del layout de un canal de lectura	32
4.1. Jerarquización	32
4.1.1. Redundancia	33
4.2. Estructura de layout de una columna del canal	35
4.3. Problemas de layout que afectan al diseño del canal de lectura	36
4.3.1. Consumo	36
4.3.2. Acoplos	37
4.3.3. Condensadores del ADC	38
4.3.4. Distribución de señales horizontalmente	38
5. Extracción y simulación	41
5.1. Extracción de parásitos	41
5.2. Simulación	42
6. Conclusiones	46
Bibliografía	48

Índice de figuras

1.1.	Explicación gráfica del efecto blooming en una imagen	5
1.2.	Estructura del array de píxeles	6
1.3.	Diagrama de bloques del chip completo	8
2.1.	Estructura física de transistores NMOS y PMOS	12
2.2.	Estructura física de una resistencia de polisilicio	13
2.3.	Ejemplo de layout con disposición de centroide común	20
3.1.	Esquemático y modo operación de un píxel 5T habitual	23
3.2.	Diagrama de fases para la técnica ping-pong con <i>global-shutter</i>	24
3.3.	Esquemático de la fuente de corriente del píxel	25
3.4.	Esquemático del ADC	26
3.5.	Señales de onda del comparador de rampa	27
3.6.	Esquemático del OTA comparador	28
4.1.	Ejemplo del funcionamiento del sistema de redundancia	34
4.2.	Consecuencias de la curvatura en la alimentación del canal	37
4.3.	Floorplan de una columna de canal de lectura	40
4.4.	Layout del par diferencial del OTA del comparador	40

Resumen

En este trabajo se va a presentar y estudiar el layout de un canal de lectura para sensores de imagen CMOS y los problemas y aspectos importantes a considerar en el proceso de diseño y verificaciones, incluyendo las fases de extracción de parásitos y simulación, tanto previa como posterior al layout.

El canal de lectura de un sensor de imagen es el bloque analógico encargado de transformar la señal de voltaje analógico recibida del píxel, en un valor digital. En la mayoría de los casos, y en concreto, el que el que nos ocupa aquí, el canal de lectura consta de un convertidor de rampa cuyos comparadores se distribuyen uno por cada columna de píxeles, o grupo de ellas, siendo la generación de la rampa, externa al bloque del canal.

El proceso de diseño del canal incluye varias fases desde el estudio inicial y el diseño del esquemático, hasta el diseño del layout y verificaciones físicas, acabando con las simulaciones de su funcionamiento. En este trabajo nos vamos a centrar en la parte del diseño de layout y su posterior extraído y simulación, que, como veremos, es un punto muy importante en el contexto del proyecto.

Objetivos y metodología

Con este trabajo se pretende hacer una recopilación de los problemas y de los puntos claves a la hora de diseñar un canal de lectura. Se trata de una guía que describe los aspectos más críticos cuando se está diseñando el layout de un canal de lectura, haciendo una introducción en los primeros capítulos tanto a la tecnología CMOS usada, como a la arquitectura electrónica de un canal de lectura.

La exposición de dichos problemas y consejos se hace usando un canal de lectura de ejemplo, del cual se describen una serie de características concretas a lo largo del trabajo. Aunque también cabe decir que mucho de lo que se va a tratar bien podría aplicarse a canales ligeramente diferentes, con, por ejemplo, diferente número de columnas, pitch o número de canales apilados.

En cuanto a la tecnología usada, se ha elegido una de 180nm y de 4 metales que define la compañía TowerJazz[®] especialmente para la fabricación de sensores de imagen CMOS[1], que provee de dispositivos para los píxeles como microlentes, array de color CFA (*Color Filter Array*) y *stiching* (tecnología de pegado de dados del chip cuando el tamaño del sensor que queremos construir excede el tamaño máximo de dado).

Tabla 1: Especificaciones usadas en el diseño

Especificaciones	Mín	Valor típico	Máx
Tecnología		TS18	
Número de metales		4	
VDDA	3.0	3.3 V	3.6 V
VDDD	1.6	1.8 V	2.0 V
Temperatura	-20°	27°	90°

El esquemático del bloque se ha diseñado usando *Virtuoso Schematic Editor*[2], el layout con *Virtuoso Layout XL*[3] y las simulaciones con su paquete ADE (*Virtuoso*

Analog Design Environment), todos ellos propiedad de la popular marca CADENCE®. En cuanto a la extracción de parásitos y las verificaciones físicas de LVS (*Layout versus Schematic*), DRC (*Design Rules Check*), ERC (*Electrical Rule Check*) y *Antenna*, se han realizado con el entorno de CALIBRE® [4].

Capítulo 1

Introducción

En este trabajo se va a estudiar el proceso de diseño de un bloque fundamental en cualquier sensor de imagen CMOS, el canal de lectura, que es el encargado de convertir la información física recibida, que en esencia es el número de fotones captados por cada píxel, a una señal electrónica analógica que posteriormente será digitalizada, procesada y, eventualmente, almacenada.

El estudio se va a centrar principalmente en el layout de este canal de lectura y en todos los aspectos a tener en cuenta a la hora de abordar esta tarea. El layout de un sistema microelectrónico consiste en su implementación física sobre una oblea de algún material semiconductor, típicamente silicio cristalino. El diseño de layout está sujeto a una serie de normas y problemas que iremos tratando con mayor detenimiento a lo largo de la exposición.

Para introducir al lector en la materia será necesario describir, aunque sea brevemente, conceptos sobre sensores de imagen, tecnología CMOS y explicar de manera sencilla la arquitectura de un canal de lectura habitual.

Posteriormente se pasará a analizar en detalle los problemas y cuestiones que se plantean a la hora de diseñar el layout de bloques analógicos en general, centrándonos en última instancia en los que afectan directamente a un canal de lectura.

1.1. Sensores de imagen

Un sensor de imagen o cámara fotográfica es, en esencia, un sistema que capta una imagen instantánea de una escena mediante la luz que emiten los objetos que se encuen-

tran en su campo de visión y que llegan a una pantalla donde se almacena la información que proyecta ese rayo de luz, ya sea por un proceso químico o bien electrónico, que es el caso que se va a tratar aquí.

En cuanto a los sensores de imágenes electrónicos se pueden distinguir dos tipos principalmente, los CCD (*Charge-Coupled Device*) y los CMOS (*Complementary Metal-Oxide-Semiconductor*). Las diferencias entre ellos se basan en la tecnología empleada y en la forma de leer el array de píxeles. [5][6]

Sensores de imagen CCD El concepto fundamental de un sensor de imagen CCD se basa en el almacenamiento y propagación de los electrones fotogenerados en cada píxel. Mediante el efecto fotoeléctrico, un fotón que impacte en la zona de silicio fotosensible, si tiene la energía adecuada, arrancará un electrón desde la banda de valencia hasta la banda de conducción, y podrá moverse libremente o arrastrado por campos eléctricos.

La carga almacenada en cada píxel será función lineal de la intensidad lumínica captada. Tras la exposición, las cargas almacenadas en cada uno de ellos se irán transmitiendo al píxel inferior, y, de la misma forma que ocurre en un registro de desplazamiento, la información de todos los píxeles se va transmitiendo de unos a otros, hasta que al final todo el array es leído.[6]

Sensores de imagen CMOS La diferencia fundamental de estos con los sensores CCD es que los CMOS usan un amplificador integrado en el píxel, a lo que nos referimos como APS (*Active Pixel Sensor*) [7]. Esto les da algunas ventajas frente a los CCD, por ejemplo, sufren de menor efecto de “blooming”, que podría traducirse como deslumbramiento o destello, es decir, manchas blancas que pueden producir los puntos muy brillantes a su alrededor por un desbordamiento de los electrones fotogenerados en una zona. Podemos ver un ejemplo en la imagen 1.1. Otra ventaja es que pueden llegar a ser más rápidos y más baratos, ambas debidas al uso de la tecnología CMOS. Gracias a esto, hoy en día cada vez son más comunes que los CCD.

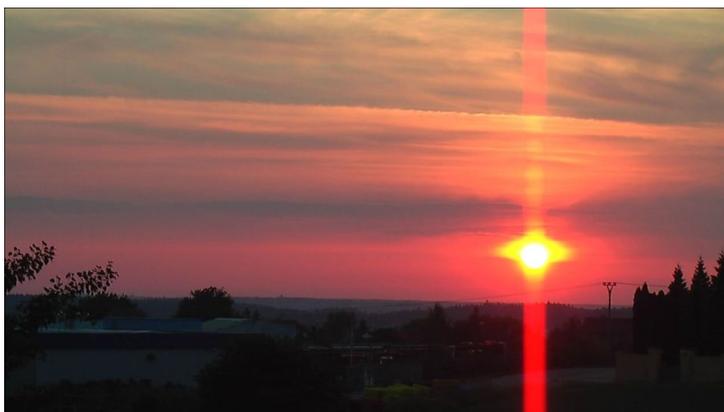


Figura 1.1: Ejemplo de imagen con efecto de *blooming* en los píxeles brillantes ¹

1.2. Arquitectura de sensores de imagen CMOS

1.2.1. Array de píxeles

El dispositivo principal en un sensor de imagen es el píxel, que es el elemento receptor de la luz y el encargado de hacer la primera conversión de fotones a electrones. Estos fotones se traducen en electrones por medio del efecto fotoeléctrico descrito por Albert Einstein: cada fotón que incide en la superficie del silicio es capaz de dar la energía suficiente a un electrón de la banda de valencia para que pase a la banda de conducción y se pueda mover libremente por la red cristalina. Si ahora aplicamos un campo eléctrico podemos guiar a todos los electrones fotogenerados hacia un pozo de potencial donde almacenarlos temporalmente. Este es el llamado proceso de exposición, que tiene una duración definida. En función del tiempo de exposición y de la intensidad lumínica recibida por el píxel, este se cargará con mayor o menor número de electrones.

El array de píxeles es la matriz compuesta de todos los píxeles organizados en filas y columnas donde la imagen formada por el sistema de lentes focalizará la imagen. Habitualmente, los píxeles son cuadrados con un lado de unas 5 a 10 μm , y el array puede tener resoluciones de miles de píxeles en ambas dimensiones, lo que significa que el array puede unos ocupar pocos centímetros, lo que lo convierte habitualmente en el bloque más grande de todo el diseño.

En este estudio vamos a trabajar con un array QSXGA de 2560 columnas por 2048

¹Imágen de Wikimedia Commons[8]

filas, que tiene una relación de aspecto 5:4. Además de los píxeles activos, en un sensor de imagen, habitualmente se incluyen algunos píxeles dummy, esto es, que no van a ser incluidos en la imagen leída. Su función es la de evitar que los píxeles en el borde tengan un entorno diferente al resto de píxeles del interior, lo que puede alterar sus características.

Además de los píxeles dummy, habitualmente se incluyen algunos de los llamados píxeles oscuros, píxeles iguales a los activos salvo que, por construcción no reciben luz. Esto se hace tapando el fotodiodo con los metales superiores, sin modificar el resto de estructuras y transistores. El objetivo de estos píxeles oscuros es ayudar en la calibración de los sensores una vez fabricados. Si leemos píxeles que están tapados obtenemos el verdadero valor de negro y podemos cuantificar el ruido debido exclusivamente al generado por el píxel y el canal, independientemente de las condiciones externas.

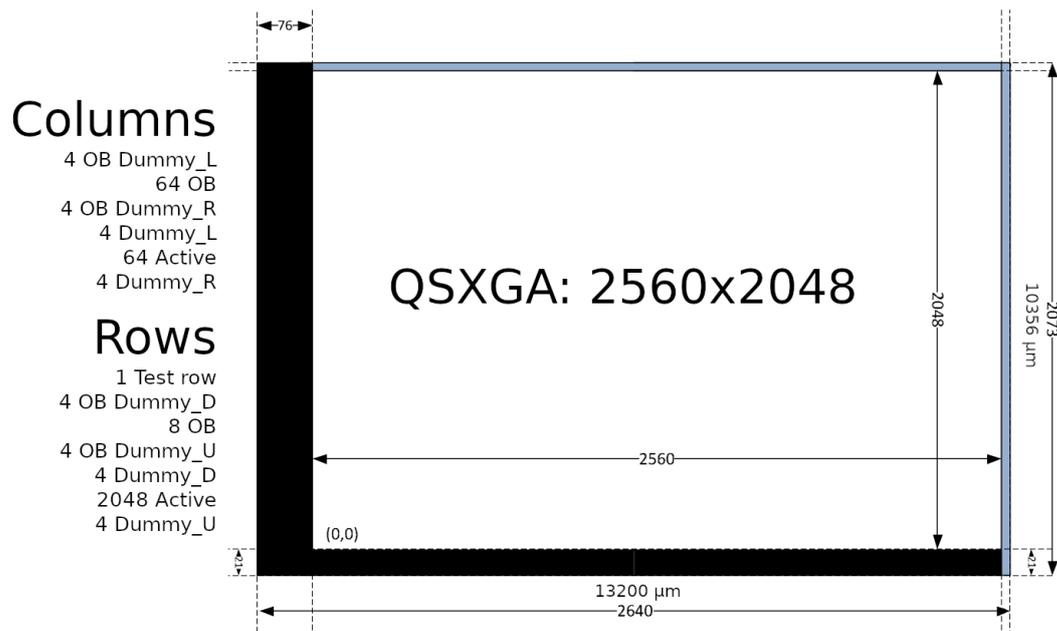


Figura 1.2: Estructura del array de píxeles²

El sensor que estamos estudiando tiene 2560×2048 píxeles activos, 64 columnas oscuras, con 4 píxeles dummy a cada lado de los píxeles activos y oscuros. En cuanto a las filas, tendremos 8 filas oscuras en la parte inferior del array, flanqueadas por 4 filas dummy arriba y abajo, al igual que alrededor de las filas activas. Además se incluye una fila de test que servirá para la calibración simulando que la fila en cuestión ha recibido

²Imágen obtenida del artículo de la bibliografía [9]. Medidas en píxeles, y en μm

una señal concreta. En la figura 1.2 se esquematiza la estructura del array.

1.2.2. Canal de lectura

El canal de lectura de un sensor CMOS, referido habitualmente por sus siglas en inglés **RO** (*Read-Out Channel*), es el bloque que se encarga de traducir el voltaje almacenado en cada píxel durante el proceso de exposición, en un número digital. Esta descripción concuerda con el concepto ampliamente utilizado en electrónica de ADC, siglas en inglés de *Analog-to-Digital Converter*, (Convertidor Analógico-Digital), que, en general toma una señal analógica y la expresa en valores discretizados.

1.2.3. Otros bloques

Ana ctrl row Este bloque se encarga del control analógico de las señales que excitan el array por filas. Se sitúa a la izquierda y/o derecha del array.

vddpix (Pixel supply) Es el bloque analógico que se sitúa en la parte superior del array de píxeles y que provee la alimentación del array y en ocasiones también provee señales que se distribuyan de forma vertical.

Referencias Es un bloque que se encarga de generar tensiones y corrientes de polarización para el resto de bloque de forma que sean independientes de la temperatura, fuente de alimentación y otros ruidos.

SPS o serializador En el bloque digital que se encarga de ordenar en frames y serializar los datos de salida del canal de lectura para que puedan ser enviados al exterior del chip por los puertos LVDS de salida en serie.

SCM o Control Main Es el bloque digital encargado del control de todas las señales y procedimientos que el sensor debe realizar para tomar una imagen, digitalizarla, y enviarla al exterior.

SCB Es el bloque digital de registros donde se guardan las configuraciones que hacen funcionar el sensor de una forma especificada por el usuario.

SSH Es el bloque digital que genera las señales digitales de control del array por filas, que se pasan al *ana ctrl row* para ser bufereadas hacia el array.

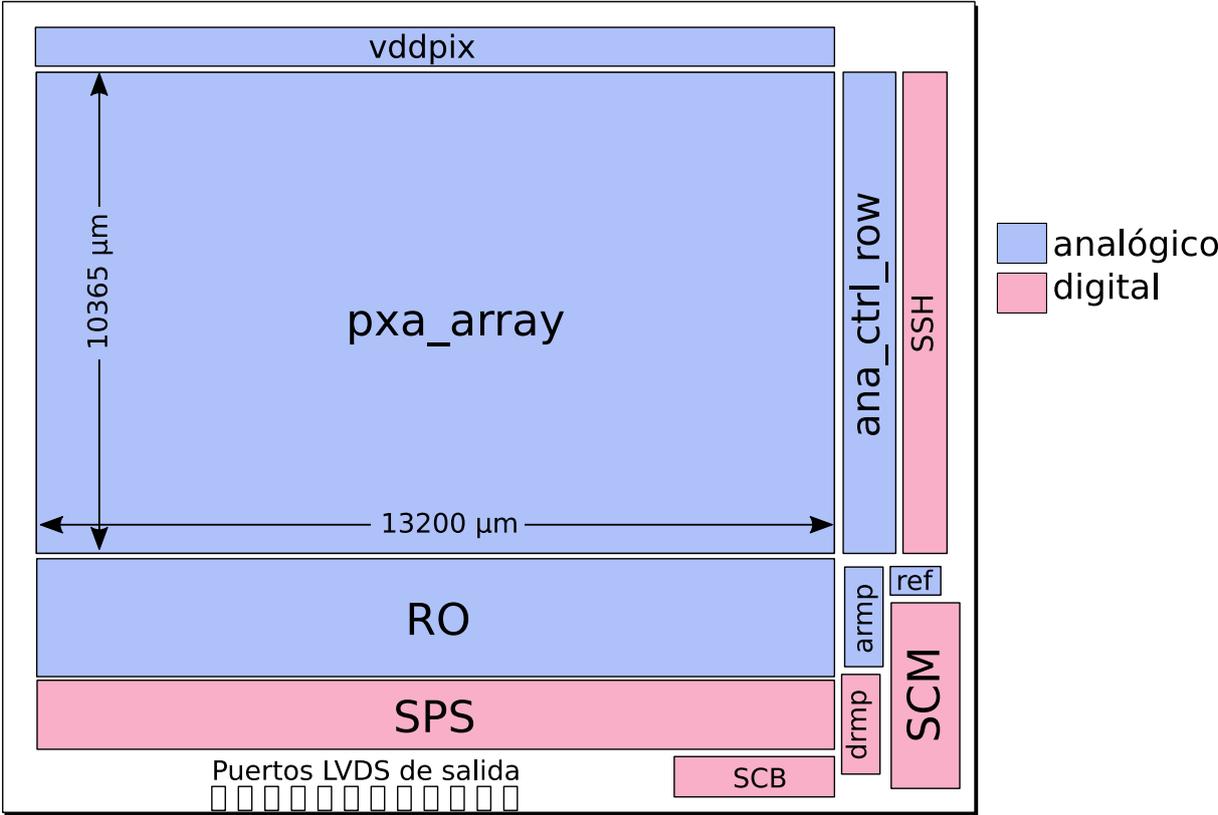


Figura 1.3: Diagrama aproximado del chip completo mostrando la disposición de los bloques más importantes³

³Imagen trabajo propio

Capítulo 2

Tecnología CMOS

2.1. Introducción

La tecnología CMOS, que es ampliamente utilizada en el diseño de circuitos integrados en la actualidad, se basa en la posibilidad de integrar en un mismo sustrato semiconductores con ambos dopados (n y p). Con ello podemos implementar transistores MOSFET tanto PMOS como NMOS en un mismo diseño.

Es una tecnología que cumple ya los 50 años, debido a que empezó a ponerse en práctica a mediados de los años 1960. Inicialmente se usó principalmente en circuitos digitales, debido a que los transistores CMOS solo consumen potencia cuando conmutan, a diferencia de los transistores de unión bipolar. Por otra parte, es más sencillo disminuir su tamaño y tienen un menor coste de fabricación.[10]

Poco a poco se fue introduciendo la tecnología CMOS en el diseño de circuitos analógicos. Los bajos costes y la posibilidad de crear circuitos digitales y analógicos en el mismo chip hacían esta opción muy interesante. Pero aun así, los transistores bipolares eran mucho menos ruidosos y más rápidos que los MOSFET, por lo que la transición fue lenta. Con el desarrollo de la tecnología CMOS, la velocidad y el ruido de estos, se ha visto muy mejorada, y actualmente domina el mercado, aunque en muchos casos se sigue usando tecnología bipolar.

2.2. Proceso de fabricación

La gran mayoría de los circuitos integrados CMOS están contruidos sobre silicio. El silicio (Si), elemento de número atómico 14, es muy abundante en la Tierra, aunque

no se encuentra de forma pura, sino como óxidos de silicio o silicatos. Entre los óxidos de silicio, basados fundamentalmente en la sílice o dióxido de silicio (SiO_2), se encuentran el cuarzo y el sílex, ambos ampliamente extendidos en la corteza terrestre. Los silicatos son sales basadas en el ión silicato (SiO_4), y forman parte de minerales como los feldespatos, micas, berilio.

A pesar de estar en tan alta abundancia en la Tierra, como se dijo antes, el silicio puro no se dá naturalmente, debe ser refinado y cristalizado. El proceso consiste, tratado de manera sencilla, en extraer el oxígeno de los compuestos mencionados arriba a base de añadir carbono y fundir la mezcla en un horno. Tras este y otros procesos obtendríamos silicio relativamente puro, pero en forma policristalina (habitualmente nos referiremos a este como polisilicio o *poly* por su abreviatura inglesa). Esta forma contiene silicio puro, pero cristalizado en pequeños cristales independientes con diferentes planos cristalinos creando efectos de borde en el interior del conglomerado, que anulan las cualidades semiconductoras del silicio.

Para construir un único monocristal de silicio se suele usar el llamado proceso de Czochralski, en el cual, una varilla de silicio usada como semilla se va rotando en un baño de silicio puro fundido a unos 1400°C de manera que va creciendo en diámetro a medida que los átomos de silicio se van depositando en la capa externa. Esta técnica fue introducida por Jan Czochralski en 1915.

Lo que queda es un lingote de aproximadamente un metro de largo y pocas decenas de centímetro de diámetro de silicio monocristalino siguiendo la estructura cristalina del diamante.

Para ser usado en la industria de semiconductores, estos lingotes se deben laminar en obleas de pocos milímetros de espesor sobre las que se implementarán los circuitos integrados, que se suelen distribuir en una matriz ocupando toda la superficie de la oblea que luego será cortada para separar cada “dado”.

Con estos procesos tendríamos tan solo el substrato de silicio sobre el que se debe

construir el circuito, los transistores y otros dispositivos que se necesiten. Esto se hace mediante fotolitografía, una técnica que permite construir cada capa mediante la proyección de luz ultravioleta a través de unas máscaras que permiten o no pasar la luz según el layout diseñado.

Para empezar se recubre toda la oblea con un material fotosensible a la luz UV, que dependiendo de si recibe luz o no, cambia sus propiedades haciendo que, donde ha recibido luz pueda ser eliminado posteriormente y quedar solo donde no se recibió luz, o viceversa. De esta forma obtenemos un recubrimiento selectivo con este material, haciendo que, en un paso posterior, podamos hacer crecer, sobre las zonas sin recubrimiento, una capa de óxido de silicio si queremos un aislante, o de polisilicio, cuyos usos se tratarán más adelante, o de metal como conductor, habitualmente aluminio o cobre, o algún tipo de dopado para crear las difusiones por ejemplo.

Tras la creación de cada capa, en la mayoría de los casos se realiza un pulido fino de la superficie para que la siguiente capa asiente correctamente sobre una superficie plana. Todo el proceso de fabricación se puede consultar con más detalle en la bibliografía, en el libro de Hastings [11].

2.3. Transistores CMOS

El dispositivo fundamental que usa en cualquier circuito integrado CMOS es el transistor CMOS, ya que es la base de funcionamiento de circuitos tanto digitales (puertas lógicas, inversores, flip-flops, buffers), como de circuitos analógicos (amplificadores operacionales y de transconductancia, convertidores analógico-digital, referencias de tensión, bandgap).

El transistor CMOS es un dispositivo electrónico de 3 terminales que se basa en una estructura MOS (Metal-Óxido-Semiconductor)[10][12]. En esta estructura, sobre un sustrato semiconductor se asienta una capa de *óxido* aislante y sobre ella un material metálico, que en las tecnologías actuales suele tratarse del polisilicio que se mencionó

anteriormente. La capa de óxido fino, tiene un espesor muy pequeño, y debe controlarse mucho en la fabricación, ya que variaciones en su grosor, daría lugar a modificaciones de la capacidad de puerta.

Usando la estructura vertical **Metal-Óxido-Semiconductor**, se puede construir un transistor creando a ambos lados de ella, zonas de silicio altamente dopado donde contactaremos dos terminales, que llamaremos *fuente* o **S** (*source*), y *drenador* o **D** (*drain*). En la capa superior del polisilicio colocaremos el terminal llamado *puerta* o **G** (*gate*). En la figura 2.1a podemos ver la estructura física de este dispositivo.

La dimensión de la puerta en la dirección que una drenador y fuente se llama "longitud", notada como **L**. La otra dimensión de la puerta, perpendicular a esta se llama "anchura", abreviada como **W**.

Como mencionamos al principio, también podemos crear transistores PMOS en estas tecnologías. Se hace fabricándolos dentro de un pozo N, esto es, una zona donde el sustrato tipo-p original, se ha dopado de forma que se consigue un sustrato tipo-n. El transistor embebido en este pozo tiene, entonces, la misma estructura, salvo que las difusiones para drenador y fuente son con dopado positivo **P+**. Una representación de este tipo de transistor puede verse en la figura 2.1b.

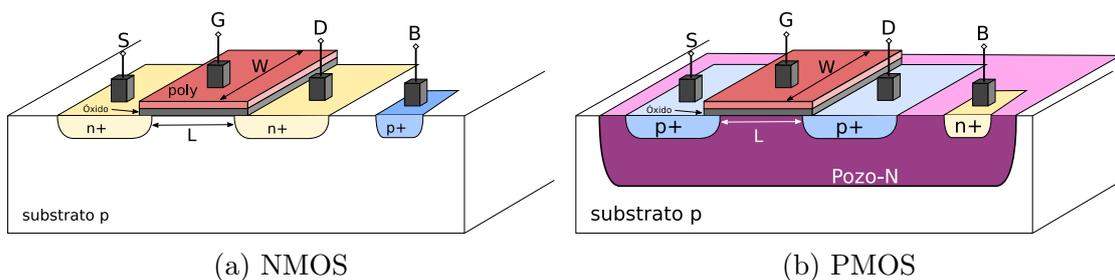


Figura 2.1: Estructura física de transistores MOS en tecnología CMOS

2.4. Otros dispositivos en tecnología CMOS

Usando la misma tecnología es posible disponer en el mismo diseño otros dispositivos además de los mencionados transistores.

2.4.1. Resistencias

En todo diseño microelectrónico resulta prácticamente indispensable en algún momento el uso de algún elemento resistor pasivo. Estos dispositivos podemos crearlos usando la resistividad que ofrecen algunos de los materiales que se emplean en el diseño. En ocasiones puede usarse el sustrato de silicio, o un pozo N⁺WELL, pero una de las técnicas más habituales es usar el silicio policristalino o **polisilicio**.

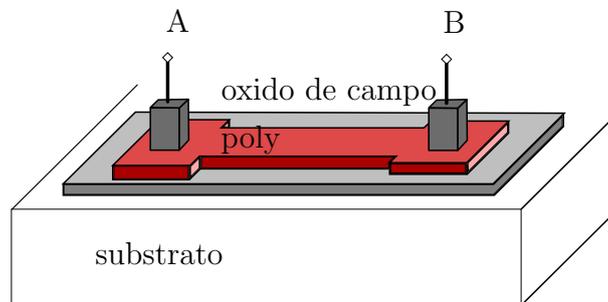


Figura 2.2: Estructura física de una resistencia de polisilicio

Para construir una resistencia de polisilicio (2.2) se crea una tira de polisilicio que se deposita sobre una capa de óxido de campo, conectando sus dos extremos con un contacto a la primera capa de metal. Como vemos, esto se parece a la capa de polisilicio que se usa para construir un transistor CMOS. Pero existe una diferencia fundamental. El óxido sobre el que se deposita el poly, en este caso, es *óxido de campo*, más grueso y menos preciso que el *óxido fino* de los transistores.

2.4.2. Condensadores

Los condensadores son otro elemento fundamental en muchos diseños analógicos. Un condensador, como sabemos, consiste en dos placas conductoras separadas por una pequeña distancia, que almacena carga eléctrica en cada terminal. En el contexto que nos ocupa, tenemos varias opciones para fabricar un condensador:

- **Poly** Llamaremos así al condensador que creamos con una lámina de polisilicio sobre una capa de óxido fino, que en función de la tecnología, puede tener varios espesores definidos por el fabricante. Un terminal sería el poly, y el otro el sustrato.
- **MiM** Entre las diferentes capas de metal tenemos varias posibilidades de crear un condensador también. Normalmente los fabricantes definen unos cuantos tipos. Pueden tratarse de dos láminas rectangulares a diferentes alturas, o, por ejemplo, dos peines que entrelacen sus dedos de forma alternativa.
- **MOS** Por último, podemos usar la capacidad de puerta que siempre tiene un transistor MOS para crear un condensador. Un terminal sería la puerta, y el otro sería el cortocircuito de drenador, fuente y sustrato. La principal ventaja de estos es que su capacidad por unidad de área suele ser elevada debido al pequeño espesor del óxido de puerta. Por contra, tienen la desventaja de presentar un comportamiento no lineal, ya que la capacidad depende del punto de operación del transistor.

2.5. Diseño de layout

El término *layout* hace referencia a la implementación física del circuito electrónico que se quiere fabricar. El layout consiste en un dibujo con toda la información que necesita la empresa fabricante para implementar el circuito sobre la oblea de silicio. Dicha información se representa por medio de “capas” que se distribuyen en un espacio bidimensional. Cada capa tiene un significado y unas normas que cumplir. El trabajo del diseñador de layout es definir, dibujar y verificar el layout de los diferentes bloques que componen el chip siguiendo las normas dadas por el fabricante, y procurando el mejor funcionamiento posible del circuito en el menor área que le sea posible. Como veremos más adelante, el diseño del layout de un circuito afecta o puede afectar mucho al funcionamiento final del mismo.

2.5.1. Capas de layout

Todo diseño de layout está compuesto de una cierta cantidad de capas, de las cuales algunas tienen significado físico directo y otras son capas que usa el software de diseño

para verificaciones o para cambiar propiedades de otras capas.

Naturalmente, los nombres de estas capas están asignados por cada fabricante en concreto, pero los nombres que presento aquí para describirlas son los que usaré para identificarlas durante el resto del documento.

NWELL Esta capa define el pozo N en una tecnología de sustrato P. En ella se deben situar los transistores PMOS y normalmente se polariza mediante difusiones **P+** a la tensión más alta del circuito (VDD o similar).

WB Esta capa define el pozo enterrado (del inglés *Well Buried*), o también llamado pozo profundo (*Deep-N-Well*). Esta estructura puede ser usada para crear zonas de sustrato P aisladas del sustrato P del resto del chip. Dentro de esta zona se pueden disponer, por ejemplo, circuitos CMOS digitales, con objeto de aislar el ruido que puedan generar a circuitos analógicos cercanos.

XN Zona de difusión con dopado negativo N+, estará en las difusiones de drenador/fuente de los transistores NMOS y en los contactos a sustrato de los pozos N.

XP Zona de difusión con dopado positivo P+, de manera complementaria a la XN, se usará para las difusiones de drenador/fuente de los transistores PMOS y para los contactos al sustrato P.

ACTIVE La capa activa indica al fabricante las zonas en las que se debe eliminar el óxido grueso (óxido de campo). Esto lo haremos siempre que haya que contactar una zona de difusión N o P, o queramos crear la puerta de un transistor como se explica en el siguiente párrafo.

GC Zona donde se creará una capa de polisilicio o (*poly*). Si coincide con capa activa, se crearán puertas de transistores sobre una fina capa de óxido de silicio, al cual llamaremos *óxido de puerta*. En los lugares donde no coincida con área activa se construirá sobre óxido grueso, de espesor no tan controlado, y se usará para rutar líneas o para crear resistencias de poly.

CS Contacto. Normalmente son cuadrados que definen contactos entre el Metal 1 y el polisilicio o el sustrato, teniendo que usar en este caso, la capa ACTIVE para poder contactar al sustrato.

M1 Metal 1. Es el primero de los metales y se suele usar para contactar transistores entre sí y con otros dispositivos, o para crear anillos de contactos a sustrato.

M2-M3-TOP_M Otros metales. En función de la tecnología, pueden ser más o menos capas de metal, por ejemplo 4 ó 6. La última capa de metal puede ser diferente, ser más gruesa y menos resistiva, con objeto de usarse como caminos de alimentación, que deben soportar corrientes mayores. Habitualmente, para ayudar al rutado del circuito, se suele definir un criterio de direcciones en función de si la capa de metal es par o impar. En el este diseño se usará el criterio horizontal = impar, vertical = par.

V2-V3-TOP_V Vías. Crean conexiones verticales entre un metal y el inmediato superior. Se suelen usar varios para cada conexión por evitar una posible rotura y para disminuir la resistencia total.

SA Del vocablo inglés **salicide**, que es una contracción de “self-aligned silicide”, esto es, siliciuro auto-alineado. Siliciuro es cualquier compuesto binario de Silicio con otro elemento, generalmente metal; por ejemplo: CoSi_2 ó TiSi_2 . Este siliciuro se dice auto-alineado porque el metal que se deposita sobre el chip durante el proceso de fabricación se adhiere solo sobre donde hay silicio (o poly), y no donde ya se ha hecho crecer la capa de óxido. La reacción del silicio con el metal crea el siliciuro, lo que crea una capa más conductiva que el silicio puro [13].

Por defecto todos el poly y las difusiones van silicidadas, pero usando la capa **SA**, podemos indicar que no se deposite siliciuro sobre algunas zonas, para, por ejemplo, permitir que las resistencias de poly sean más resistivas.

2.5.2. Herramientas de CAD

Las herramientas de CAD (*Computer Assisted Design*) se usan en muchos campos de la ingeniería o la arquitectura, y actualmente, debido a la alta complejidad de los di-

señales son de uso prácticamente obligado.

En el caso de la microelectrónica, estas técnicas tienen dos finalidades fundamentales: el diseño y simulación eléctrica y, por otro lado, la implementación física. Entre ellas hay muchas diferencias, pero también existe una inevitable relación de dependencia. La implementación física viene definida por el diseño eléctrico, pero a su vez, como veremos en muchos casos, este último condiciona el diseño eléctrico.

Por otra parte, en microelectrónica, debemos tener presente que hay una división importante entre dos ámbitos que son bastante diferentes en cuanto al flujo de diseño, simulación e implementación física, aunque ambos se basan en la misma tecnología. Me refiero a la separación entre el ámbito **analógico** y el **digital**. En este trabajo nos centraremos en la parte analógica puesto que el canal de lectura es un bloque fundamentalmente analógico.

Considerando únicamente el layout, estos dos ámbitos, analógico y digital, son también bastante diferentes. En el caso del layout digital, el flujo de trabajo, debido a la gran cantidad de dispositivos que habitualmente son necesarios para diseñar cualquier bloque digital, está altamente automatizado por algoritmos de distribución y rutado automático (*place and route*). Mediante estas herramientas, el diseñador digital de *back-end* implementa el layout de un bloque digital que el diseñador de *front-end* ha diseñado para que tenga un funcionamiento definido.

Para el caso analógico, el diseñador de layout debe dibujar los dispositivos, posicionarlos e interconectándolos de una forma mucho más manual que en el caso digital, debido a una relativa simplicidad de estos circuitos respecto a los digitales. Un bloque analógico es abordable por una persona en unos pocos días o semanas, ayudándose, obviamente, de herramientas de replicación, jerarquizado, celdas prediseñadas y/o parametrizadas. El caso analógico también se realiza de forma más manual debido a la naturaleza de las señales analógicas, al ser estas, en algunos casos, más susceptibles a ruidos, interferencias o acoplos con otras señales.

En los circuitos puramente digitales, las señales cambian de un valor alto (1) a un valor bajo (0), siendo menos importante el valor exácto mientras este se encuentre dentro de los márgenes definidos. Por el contrario, en un circuito analógico, hay que tomar especial cuidado en diseñar un layout que preserve los valores de las señales análogicas y tenga un impacto mínimo en los tiempos de propagación o asegurar que dos dispositivos o bloques que deban comportarse idénticamente así lo hagan.

2.5.3. Problemas habituales en el diseño de layout

A continuación se van a exponer y explicar algunos de los problemas habituales que afectan a cualquier diseño de layout y que el diseñador debe hacer frente para resolver o minimizar mediante su experiencia y la ayuda de las herramientas, de CAD, la simulación post-layout y los consejos del diseñador analógico del bloque en cuestión.

Área

La superficie sobre la que se diseña un chip es limitada, y además es un factor en contra del beneficio económico que tendrá la fabricación y venta del chip. A menor sea el área usada por el chip, más chips caben en cada oblea, que tiene un precio constante, definido por el fabricante en función de la tecnología y otros parámetros. Por lo tanto, como punto de partida, podemos concluir que **el área es siempre un parámetro a minimizar**, a cualquier nivel de jerarquía.

Si bien esto es la idea general, también es verdad que no siempre es la prioridad, o bien el área no supone un problema porque, por ejemplo, debido a la distribución de los bloques, un bloque resulte disponer de área más que suficiente. Puede darse el caso incluso de que se prefiera distribuir un circuito de manera más holgada pero más uniforme, antes que arrinconar todo el circuito en una zona y dejar espacio libre en otras zonas. Esto, por ejemplo, podría ayudar a mejorar la correlación entre dispositivos, o *matching*.

Mismatch

El mismatch es un problema generalizado en muchos campos de la ingeniería donde se requiere la construcción de dispositivos que se comporten igual, y por problemas de fabricación u otros agentes externos se comportan ligeramente diferente.

En el caso del layout, el mismatch puede deberse al proceso de fabricación, que puede crear dispositivos de dimensiones ligeramente mayores o menores, u otorgar al silicio de propiedades ligeramente diferentes, como la concentración de dopado, lo que puede hacer que unas zonas sean más o menos conductivas o que varíen parámetros como la tensión umbral de un transistor.

Otra forma de que se creen diferencias entre unos dispositivos es por ejemplo por la distribución irregular de la temperatura cuando el chip esté funcionando. Debido a la ley de Joule, cualquier corriente circulando por un material, lo calentará en función de la densidad de corriente que lo atraviese. Si en una zona del chip tenemos un bus de alimentación, o un circuito que consuma mucha corriente, calentará la zona cercana, creando gradientes de temperatura. Si tenemos dos circuitos o dispositivos, uno de ellos cerca y otro lejos de esta zona, posiblemente no se comporten igual.

Para minimizar los efectos de los gradientes, se suelen usar estructuras de centro-común, que en teoría, para gradientes lineales, logran el emparejamiento perfecto. Por ejemplo, si tenemos un array lineal de dos tipos de dispositivos diferentes, cada uno con una multiplicidad de 4, deberíamos usar una estructura ABBA|ABBA, en vez de AAAA|BBBB, donde A y B representan instancias del mismo dispositivo. En el primero de los casos, si existe un gradiente lineal en la dirección del array, el efecto de este actuará en positivo en unos y en negativo en otros, de forma que el efecto en A se compensa al efecto en B. En cambio en el segundo caso, como promedio, los A sufrirán el efecto más o menos que los B.

Esto se puede aplicar también para arrays bidimensionales de dispositivos, de forma que se pueden usar estructuras como la que se muestra en la figura 2.3. Algunas de las reglas que se recomienda seguir para un buen matching entre transistores serían [11]:

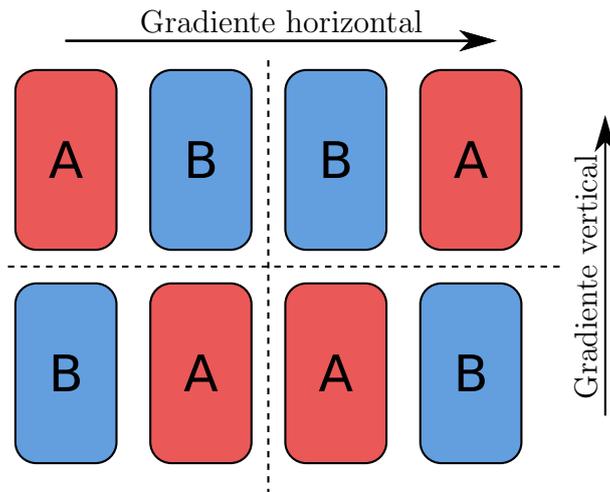


Figura 2.3: Estructura de centroide común para eliminar un mismatch bidimensional lineal

1. Usar dimensiones idénticas para los *fingers* del transistor, esto es, cada uno de los transistores unitarios dispuestos en paralelo, que forman uno mayor.
2. Maximizar el área ($L \times W$) de los transistores, ya que las variaciones relativas en tamaños son menores.
3. Orientar los transistores en la misma dirección, ya que el silicio puede presentar alguna anisotropía inherente a la fabricación.
4. Situar los transistores lo más cerca posible entre ellos. A mayores distancias, mayores pueden ser las diferencias de temperatura, dopado, ruido...
5. Uso de estructuras de centroide común.
6. A ser posible, uso *dummies* en los contornos del array de transistores. Un transistor, o en general, dispositivo *dummy*, es idéntico a uno activo en cuanto a construcción física, pero conectado de forma que no actúa eléctricamente, por ejemplo, un transistor CMOS podría conectarse a tierra todos sus terminales. Con esto evitamos los posibles efectos de borde que afectan un dispositivo que no tiene menos vecinos que los del interior del array.
7. Colocar los dispositivos lejos de dispositivos de potencia o líneas de alimentación que pueden generar fuertes gradientes de temperatura.

Capítulo 3

Diseño electrónico de un canal de lectura

El canal de lectura es el circuito que lee la información almacenada en los píxeles tras el tiempo de exposición. En la arquitectura más habitual, y la que vamos a tratar en este estudio, los píxeles se leen por columnas, entendiéndose que el chip tiene una dirección *vertical* y otra *horizontal*. Siguiendo ese criterio, el canal de lectura se sitúa debajo del array de píxeles de forma que cada columna del canal quede alineada con cada columna del canal.

Idealmente podríamos usar un canal por columna, y meter en ese espacio todo el circuito para la columna. Pero, en algunos casos, debido al pequeño tamaño de los píxeles — suelen tener unos 5, 8, 10 μm ... — resulta complicado o incluso imposible diseñar el layout de dicha circuitería en ese reducido espacio horizontal. Por esta razón, es una práctica común usar el *pitch*¹ de varios píxeles, por ejemplo 2 ó 4, y así tener más espacio para diseñar el layout del circuito. Por contra, debemos apilar estos 2 ó 4 canales en filas, lo que veremos que nos trae algunos problemas a la hora de diseñar el layout.

De ahora en adelante nos centraremos en el caso de un sensor cuyos píxeles tengan un área de 5 μm \times 5 μm , y en el que se apilan 2 canales para leer 2 columnas de píxeles. Por tanto, el ancho de un canal será de 10 μm , encajando exáctamente en la anchura de 2 columnas de píxeles.

¹De ahora en adelante usaremos la palabra inglesa “*pitch*” para referirnos al espaciado con el que se repite una estructura periódica como el canal de lectura o el array

3.1. Estructura general

El canal de lectura es un circuito analógico que va a convertir el voltaje dado por el píxel tras haber sido expuesto a la luz durante un tiempo y convertirlo en un valor analógico bien definido entre unos límites que van a significar *blanco* y *negro*, con una resolución definida.

Por tanto, el canal de lectura es principalmente un ADC (en inglés, **A**nalog to **D**igital **C**onverter, o convertidor analógico-digital). Una arquitectura usada habitualmente es el convertidor de rampa. Este, para la conversión usa un generador de una rampa de voltaje que se usa para comparar contra el valor dado por el píxel. Cuando ambos valores coinciden, la salida del comparador cambia de estado mediante un flanco de subida o bajada. De esta forma la conversión de un valor de tensión se traduce en la detección temporal de un flanco.

De manera simultánea a la rampa analógica se lanza una rampa digital que va contando valores desde 0 hasta un número que viene determinado por la resolución, y que va a evolucionar a la misma velocidad que la analógica. Posteriormente, un circuito digital detectará el flanco y parará el reloj de la cuenta digital, obteniéndose de esta forma un valor digital para el valor analógico leído.

Entre las ventajas de usar un convertidor de rampa frente a otras muchas posibilidades que existen en la literatura para convertir valores analógicos a digitales, podemos destacar las siguientes:

1. Se trata de un circuito muy simple, que puede reducirse a un comparador creado con un OTA simple de 5 transistores, más un par de condensadores y algunas llaves. Cualquier otra arquitectura (convertidor SAR, sigma-delta...) implica muchos más transistores y posiblemente más consumo.
2. Se puede obtener una buena linealidad si se diseña una buena rampa y se hace operar al comparador en un punto de operación similar en todos los casos, consiguiendo ecualizar los tiempos de transición del mismo.

3. Es fácilmente repetible en todas las columnas, ya que podemos separar la generación de la rampa de los comparadores.

3.2. Operación de lectura

Para abordar el diseño de un canal de lectura debemos entender cómo se realiza la operación de lectura de los píxeles. En la figura 3.1a podemos ver el esquemático del píxel 5T que vamos a usar para este estudio. El fotodiodo (PD), es el elemento que más área ocupa del píxel, ya que es el área que recibe los fotones.

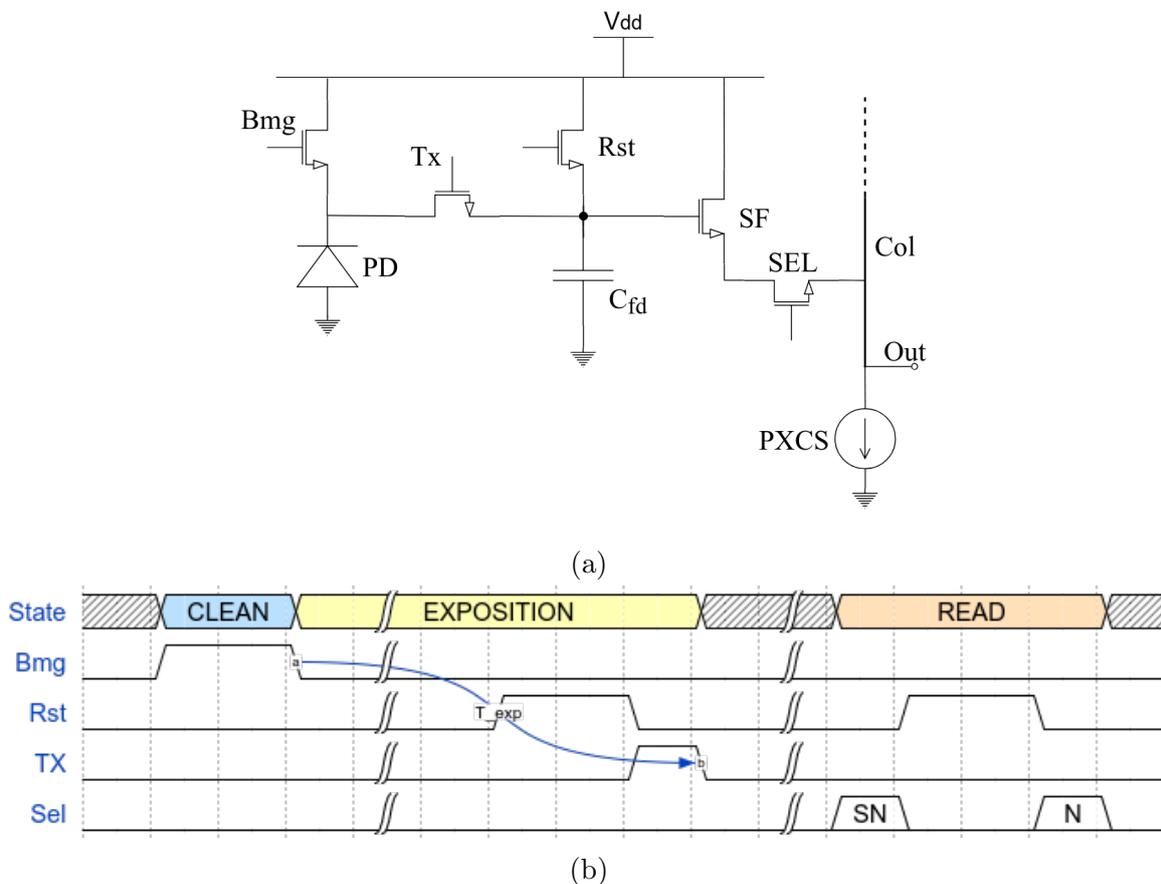


Figura 3.1: Esquemático y modo operación de un píxel 5T habitual

Mirando el modo de operación (3.1b) vemos que antes de la exposición se hace un vaciado del fotodiodo mediante el transistor de BMG, que coloca una tensión alta en el fotodiodo. Durante la exposición, los electrones fotogenerados van bajando la tensión de este nodo. Para terminar la exposición, se hace una transferencia de esta carga hacia el condensador C_{fd} (*Floating Diffusion*), no sin antes dar un pulso en el transistor de RST (Reset), para vaciar la posible carga que tuviera este condensador.

En este punto tenemos la carga con la información de la luz captada por el píxel, almacenada en el condensador C_{fd} , a la espera de que queramos leer este valor mediante la activación del transistor SEL (Selección). Normalmente, para compensar errores debidos a asimetrías en los píxeles, se hace una operación de CDS, siglas de “*Correlated Double Sampling*”. Esto es, medir la señal bruta que tenemos en la “floating diffusion”, e inmediatamente después, limpiar este condensador con un pulso de Reset y volver a leer. De esta forma, simbolizando con S, señal y N, ruido (*noise*), si restamos ambas formas de onda, podemos obtener:

$$S = SN - N \quad (3.1)$$

Podemos hacer esto debido a que ambas medidas están correlacionadas porque se hacen con muy poco tiempo de diferencia y entonces los ruidos son aproximadamente iguales.

Habitualmente existen dos formas de exponer el array: “*global shutter*”, obturación global, y “*rolling shutter*”, obturación consecutiva. En la primera, todas las filas del array se exponen simultáneamente y luego, fila por fila se van leyendo los valores almacenados en las “*floating diffusions*”. En el segundo caso, la exposición se hace secuencialmente, fila por fila, antes de cada lectura.

La operación de *sampling* y comparación de estas señales se realiza en el bloque del ADC, que se describirá más abajo en el apartado 3.4. Existe una interesante técnica para duplicar velocidad de lectura haciendo simultáneamente el muestreo de las señales y su comparación con el valor de la rampa. Esta técnica de lectura en *pipeline* o también llamada *ping-pong*, requiere duplicar el bloque del ADC por cada columna de píxeles que tengamos. El siguiente diagrama 3.2 de fases muestra su funcionamiento.

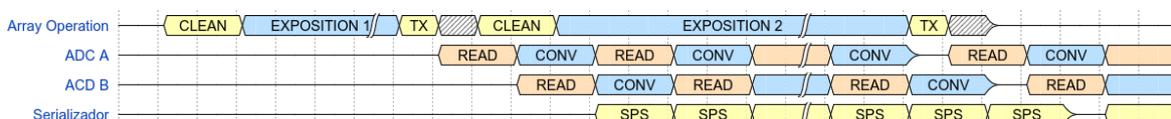


Figura 3.2: Diagrama de fases para la técnica ping-pong con *global-shutter*

3.4. ADC

Como ya hemos comentado, la conversión del valor analógico del píxel, a un número digital se realiza mediante un convertidor de rampa. Para conseguir esto, en cada canal de lectura se implementa un sencillo ADC comparador y unos condensadores donde almacenar temporalmente las señales.

La señal de la rampa analógica se va a distribuir horizontalmente entre todas las columnas del canal de lectura, y está generada por un bloque externo al RO, situado a uno de los lados del mismo, con lo que evitamos incluirlo en área del canal.

El ADC es el bloque que **muestra** y **compara** la señal analógica del píxel con la de la rampa analógica. El esquemático del circuito se muestra en la figura 3.4. La operación del ADC se realiza en 2 fases, lectura (*read*) y comparación (*comp*), y se puede seguir mediante las formas de onda mostradas en 3.5.

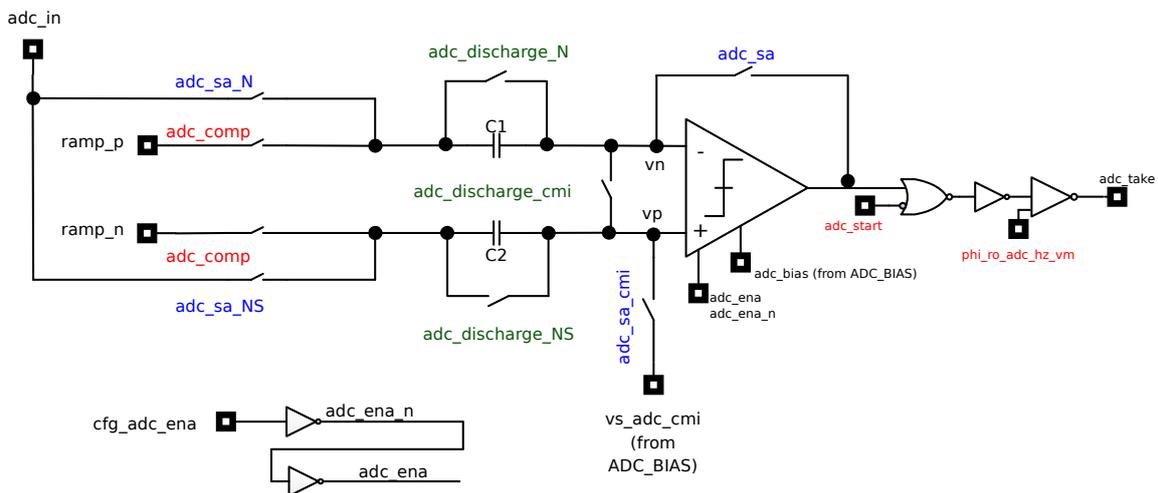


Figura 3.4: Esquemático del ADC

Durante la primera de ellas, el comparador está operando en lazo cerrado, por lo los nodos de entrada *vp* y *vn* se establecen a la tensión de modo común, *vs_adc_cmi*. Al final del periodo de muestreo, las señales “señal + ruido” (NS) y “ruido” (N), quedan

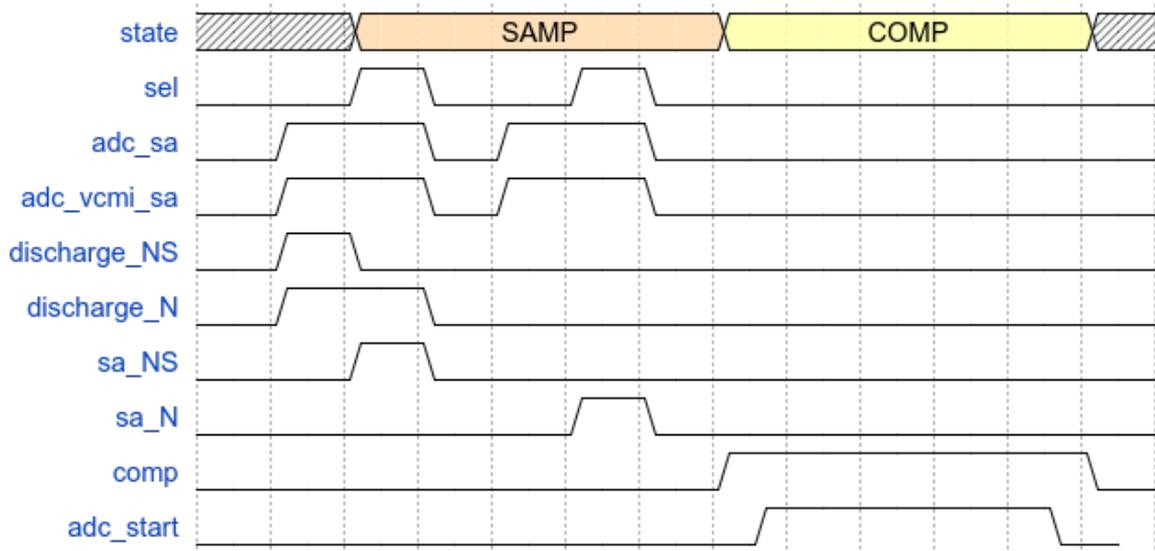


Figura 3.5: Señales de onda del comparador de rampa

almacenadas en los condensadores C_2 y C_1 , respectivamente.

En la segunda fase, el comparador se establece en lazo abierto, por lo que la carga almacenada en los condensadores se mantiene. En ese momento se conecta la rampa diferencial ($ramp_P$ y $ramp_N$), y aplicando la conservación de carga en los nudos v_p y v_n , se puede deducir que el voltaje será:

$$v_n = \frac{v_{cmi} + v_{off}}{1 + \frac{1}{A_0}} + \alpha_1(v_{ramp_p} - v_{pixN}) \quad (3.2)$$

$$v_p = v_{cmi} + \alpha_2(v_{ramp_n} - v_{pixNS}) \quad (3.3)$$

, donde los coeficientes α_1 y α_2 son:

$$\alpha_1 = \frac{C_1}{C_1 + C_p}, \quad \alpha_2 = \frac{C_2}{C_2 + C_p} \quad (3.4)$$

Asumiendo un comparador ideal, el voltaje diferencial a la entrada de este, se puede expresar como:

$$\Delta V = v_p - v_n = (v_{pixN} - v_{pixNS}) - (v_{ramp_p} - v_{ramp_n}) \quad (3.5)$$

, donde la diferencia ($v_{pixN} - v_{pixNS}$) es siempre positiva y se relaciona con la información almacenada en el píxel, y la señal ($v_{rampP} - v_{rampn}$) es linealmente creciente. Por tanto, la salida del comparador debe ser inicialmente un valor alto y cambiar a valor bajo en el instante en que ambas señales sea crucen. Por lo que la información en el píxel puede codificarse con el tiempo registrado por un contador digital, siendo mayor el número digital a mayor la luz recibida por el píxel leído.

La salida del comparador se hace pasar por una puerta NOR para evitar que pueda registrarse una transición sin estar activa la señal de *adc_start*

3.4.1. OTA

El OTA (*Operational Transconductance Amplifier*) es el bloque que actúa como comparador y tiene la estructura de un amplificador diferencial simple, que se muestra en el esquemático de la figura 3.6.

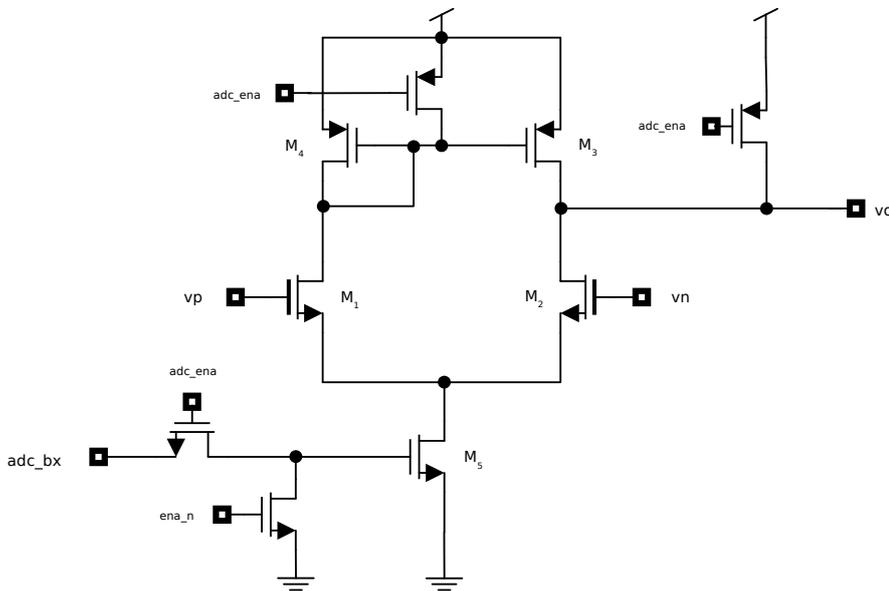


Figura 3.6: Esquemático del OTA comparador

Los transistores M_1 y M_2 forman el par diferencial, con sus correspondiente espejo de corrientes en M_3 y M_4 , y la fuente de corriente M_5 , el resto de transistores sirven para deshabilitar el OTA cuando no se esté usando.

3.5. Generador de rampa analógica

El bloque generador de la rampa analógica se encuentra situado fuera del Canal de Lectura. Se encarga de generar una señal linealmente creciente y su correspondiente decreciente. El rango en el que la rampa varía debe ajustarse en función del rango de entrada del OTA del comparador y teniendo en cuenta el rango en el que varía la salida del píxel, para obtener una resolución deseada y aprovechar al máximo la sensibilidad del sensor.

Es muy importante una precisa generación de esta rampa analógica, con especial atención a la linealidad de la misma. En la bibliografía se describe una arquitectura para el generador de rampa que obtiene una buena linealidad por debajo de un LSB para resoluciones de hasta 15 bits a frecuencias de reloj de 100MHz[14], que está en un rango adecuado para los tiempos de lectura que se suelen tratar en sensores de imagen CMOS.

La señal generada debe propagarse a lo largo de todos los canales de lectura, desde la izquierda hacia la derecha, o al contrario, en su caso. Para conseguir una rápida y efectiva propagación de esta señal sin distorsión y en tiempo, ha de hacerse un cuidadoso estudio de layout, calculando las anchuras y separaciones entre las líneas metálicas que distribuirán la señal y posteriormente, realizando una precisa extracción y simulación del circuito.

3.6. Bloques de polarización

Los bloques del canal descritos anteriormente, la fuente de corriente y el ADC requieren de varias señales de polarización, que habitualmente, además, suelen ser programables, esto es, que permiten diferentes valores de configuración establecidos por el usuario del dispositivo sobre, por ejemplo, la velocidad de lectura, la resolución deseada, el consumo.

Normalmente, para ahorrar espacio, los bloques de polarización, o *BIAS*, se comparten entre varios canales de lectura. De esta forma, el layout de estos bloques puede ser más ancho que alto, con lo que conseguimos un ahorro de área en la dirección vertical.

3.6.1. PXCS Bias

La fuente de corriente necesita de tres voltajes de polarización: vb_{bn} y vb_{cn} , que vienen de un espejo cascodado y la tensión del limitador vb_{ln} . El bloque va a estar compartido por 8 canales ($80\mu\text{m}$ de anchura).

Los valores de la corriente que podrá drenar la fuente de corriente del píxel, va a ser configurable con 5 bits de resolución. Esta configuración será implementada en este bloque de polarización mediante 5 ramas de corrientes que conduzcan, respectivamente, I_0 , $2I_0$, $2I_0$, $4I_0$, $8I_0$, con $I_0 = 125\text{nA}$. De forma que podemos variar la corriente total entre 125nA y $3,875\mu\text{A}$, en pasos de 125nA .

3.6.2. ADC Bias

El ADC Bias tiene que proveer al ADC del voltaje de modo común (vs_{adc_cmi}) y de la tensión de bias de la fuente de corriente del OTA (vb_{adc_bx}), que espejará una corriente generada en el Bias.

La tensión de modo común se puede generar de dos formas, la primera mediante un buffer que replica una tensión generada en el bloque de Referencias, fuera del canal de lectura, y la segunda generando un tensión local por cada grupo de canales. Esta segunda opción también usa una referencia común, pero al ser local, se adapta mejor a la eventual curva en las alimentaciones que podremos tener a lo largo del RO. Este tema se tratará con más detalle en la correspondiente sección de layout.

3.7. Rampa digital y serialización

Una vez la operación de comparación se ha iniciado, la rampa y la señal del píxel, en algún momento se cruzarán, lo que desencadenará una transición en la señal adc_take que es la que finalmente sale de la parte analógica del canal y se va a rutar hacia abajo

para ser registrada por el contador digital.

Dicho contador (o rampa) digital habrá empezado la cuenta desde cero, en un tiempo y a una velocidad adecuados para que el valor más alto de la cuenta digital (la resolución) coincida con el valor final de la rampa analógica y con el valor mínimo de tensión de salida del píxel (máxima intensidad lumínica). En el momento en que se recibe el pulso en la salida *adc_take*, se detiene la cuenta digital y tenemos un valor digitalizado proporcional a la intensidad lumínica captada por el píxel actual.

El bloque que genera la rampa digital DRMP se sitúa a uno de los lados del RO y su señal se distribuye mediante búferes digitales a lo largo del SPS, que es el bloque digital situado justo debajo del Canal de Lectura y que incluye la digitalización y serialización de los datos.

La serialización se efectúa para convertir la operación de digitalización, que se hace de forma paralela, simultánea a todos los canales, a una salida de datos en serie. Estos datos en serie se encaminan posteriormente hacia uno o varios buses de salida LVDS (*Low Voltage Differential Signal*), que harán posible la lectura de las imágenes desde el exterior.

Capítulo 4

Diseño del layout de un canal de lectura

El diseño del layout de un canal de lectura en el contexto de un proyecto suele ser una tarea que ocupa gran parte del periodo de diseño, y además es recomendable abordar en las primeras fases del proyecto, ya que como hemos comentado, el layout va a influir, en mayor medida que otros bloques, en el diseño del mismo. Por tanto, el diseño del canal de lectura suele ser un proceso iterativo: diseño del esquemático, layout, extracción, simulación y eventual rediseño.

4.1. Jerarquización

A la hora de estructurar el diseño de layout de cualquier bloque repetitivo, como es el canal de lectura, es básico idear una estrategia en cuanto a la jerarquización del mismo, ya que usualmente no se trata de un simple array de instancias iguales situadas una al lado de la otra.

Por otra parte, existen estructuras que pueden requerir diferentes periodicidades. Y también hay que adaptarse a la resolución del array, en este caso, al número de columnas. Como ya hemos visto a lo largo de la descripción del esquemático del canal nuestra unidad básica de un canal de lectura, a la que nos referiremos como *ro_chx1*, ocupará una anchura de 2 columnas de píxeles, es decir, una anchura de $10\mu\text{m}$. En esa anchura tiene que ser posible incluir todos los bloques que se repiten por canal, es decir, la fuente de corriente y el ADC de una columna de píxeles.

En un siguiente nivel de jerarquía, incluiremos 8 canales unitarios y sus corres-

pondientes bloques de polarización (PXCS_BIAS y ADC_BIAS), que como vimos, están compartidos por cada 8 canales. De esta forma, el bloque *ro_chx8* tendría una anchura de $10\mu\text{m} \times 8 = 80\mu\text{m}$, y tendríamos canales para leer 16 columnas de píxeles.

Desde 16 columnas hasta las columnas totales del array podemos elegir varios niveles de jerarquías intermedios. Considerando que nuestro array tiene la estructura mostrada en el capítulo 1.2.1, en la figura 1.2, con 2560 columnas activas y 64 columnas oscuras, vamos a hacer una jerarquía que cubra esos 64 píxeles, es decir, 32 canales, por lo que se llamará *ro_chx32* e instanciará 4 unidades de *ro_chx8*.

Para completar las 2560 columnas de píxeles activos, solo hay que instanciar 40 de los bloques de 32 canales. Y para completar el array, tan solo hay que añadir 2 canales (4 píxeles) a izquierda y derecha de la zona activa (píxeles dummy), y un bloque de 32 canales en la zona de las columnas oscuras, con sendos pares de canales a izquierda y derecha para hacer las columnas oscuras dummy.

4.1.1. Redundancia

En todo diseño microelectrónico, debido a defectos en la fabricación, ciertos dispositivos pueden resultar completamente inutilizables o, al menos, con prestaciones por debajo de lo esperado en relación al resto de dispositivos equivalentes. Esto se acentúa en casos como un canal de lectura, en los cuales la integración y la congestión de rutado y dispositivos es muy alta, y además tiene un patrón altamente simétrico y periódico.

Con el paso del tiempo, los fabricantes de sensores de imagen han detectado que el canal de lectura sufre de considerablemente más fallos que el resto de circuitos en el sensor, y por consiguiente es uno de los principales contribuyentes a la disminución del “*yield*” (del inglés: rendimiento), que da cuenta del porcentaje de sensores por oblea que no funcionan o son defectuosos, lo que va en contra del beneficio económico.

Para tratar de disminuir este problema, se suele poner en práctica el uso de canales redundantes, esto es, canales adicionales que podrían ser usados en el caso de que, una

vez fabricado el chip, se detecten fallos en un canal concreto.

Para esto se añade un canal adicional cada cierto número de canales y mediante un sistema de llaves a la entrada de todos los canales podemos redireccionar la salida todos los píxeles en el rango afectado y posteriores al canal del fallo, hacia el siguiente canal. Con esto se puede ver fácilmente que podemos solucionar tantos fallos como canales redundantes sean incluidos, siempre y cuando no se produzca más de un fallo por región.

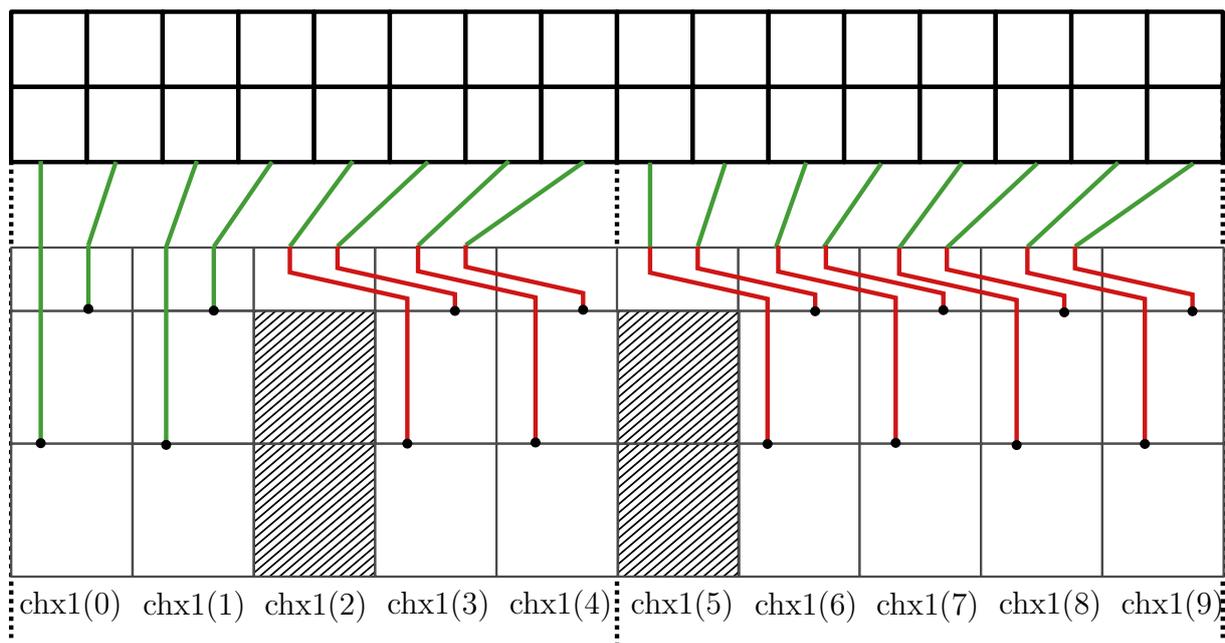


Figura 4.1: Ejemplo del funcionamiento del sistema de redundancia

En la figura 4.1 vemos un ejemplo de aplicación del sistema de redundancia donde tendríamos un canal redundante por cada 4. En el caso de que fallen los canales 2 y 5 mostrados en la imagen, el bloque de redundancia colocado en la parte superior del canal de lectura podría dirigir cada columna a la siguiente, de la forma que muestran las líneas rojas.

Como ya se puede intuir, la anchura del canal de lectura, en el supuesto de usar redundancia se tiene que ver reducida con objeto de incluir un canal adicional cada cierto tiempo. En nuestro caso vamos a incluir un canal redundante por cada grupo de 32 canales funcionando. Con esto, haciendo una sencilla cuenta, podemos establecer la nueva anchura en $9\mu\text{m}$ en vez de las $10\mu\text{m}$ originales. Esto nos permite además tener una anchu-

ra sobrante cada 33 canales que será: $32 \cdot 10\mu\text{m} - 33 \cdot 9\mu\text{m} = 23\mu\text{m}$ de anchura sobrante. Este espacio lo podemos usar para por ejemplo subir las tensiones de polarización desde los BIAS a las líneas horizontales en los correspondientes bloques.

4.2. Estructura de layout de una columna del canal

La estructura completa se muestra al final de capítulo (4.3) en un diagrama simplificado del layout de una columna de canal (*ro_chx1*), mostrando los bloques más importantes, sus tamaños y sus conexiones.

Una vez establecida una jerarquía clara, y suponiendo el uso de la redundancia ya tenemos establecida una anchura para el layout de una columna de nuestro canal. En esta anchura deberemos incluir todos los bloques descritos en el esquemático tratando de hacerlo lo más compacto posible para ahorrar área en la dirección vertical.

Como hemos introducido en el apartado anterior, el bloque de redundancia será el primero que nos encontraremos por la parte superior, más cerca del array de píxeles. Para esto además tiene que existir un rutado que transforme las salidas de los píxeles, espaciadas cada $5\mu\text{m}$ a las entradas de los bloques de redundancia, espaciadas 2 cada $9\mu\text{m}$.

Lo siguiente que tenemos es la fuente de corriente, que es una por cada columna de píxeles, es decir, dos en cada canal unitario, y se van a colocar una encima de otra ocupando ambas la anchura del canal, $9\mu\text{m}$.

Inmediatamente debajo colocaremos el bloque de polarización de las fuentes de corriente, PXCS_BIAS, que como se comparte para 8 canales, en la unidad *ro_chx1*, dejaremos un hueco de la altura necesaria, que se establecerá en función de la altura del BIAS.

Por debajo del PXCS_BIAS, vamos a colocar los ADC, que tiene que haber dos por cada columna de píxeles si queremos usar la técnica de ping-pong. En nuestro caso esto significa tener que implementar 4 ADCs idénticos dispuestos en vertical ocupando las $9\mu\text{m}$ de anchura del canal.

Por último, bajo los 4 convertidores iría el Bias del ADC, que también se comparte por cada 8 canales, por lo que tampoco estaría incluida en columna unitaria.

4.3. Problemas de layout que afectan al diseño del canal de lectura

4.3.1. Consumo

El consumo de un bloque tan grande como es el canal de lectura siempre es algo a tener muy en cuenta a la hora de modelar las alimentaciones que debe tener y cómo afectan estas al funcionamiento del propio bloque y del resto del sensor.

En muchas ocasiones no es tan importante que un bloque tenga un consumo alto, como la dinámica con la que se produzca ese consumo. Si un bloque tiene un alto consumo, pero lo hace de manera más continua que otro que tiene grandes picos de alto, quizás afecte menos al rendimiento total del circuito. En el caso del canal de lectura esto es muy importante porque muchos de los procesos se producen de forma simultánea en todos los canales a la vez. Por esta razón hay que tener especial cuidado a cómo se programan los tiempos de lectura y comparación, y, lo más importante, realizar exhaustivas simulaciones usando extraído de parásitos y modelando muy bien las alimentaciones y los acoplos que podemos tener. Este tema se tratará en más detalle en el apartado [5.2](#) sobre simulaciones.

Otro punto importante a considerar es que el canal de lectura va a ser alimentado, tan solo por ambos extremos. Esto va a generar una curvatura en las alimentaciones que puede traer problemas, manifestándose como un gradiente en la imagen final. Las tensiones de polarización de los bloques de bias, están generadas localmente y referidas a una tierra con curvatura por lo que unos ADC y otros podrían tener tener diferente tiempo de conversión, por ejemplo.

Para tratar de corregir este efecto, se suelen cortocircuitar las tensiones de polariza-

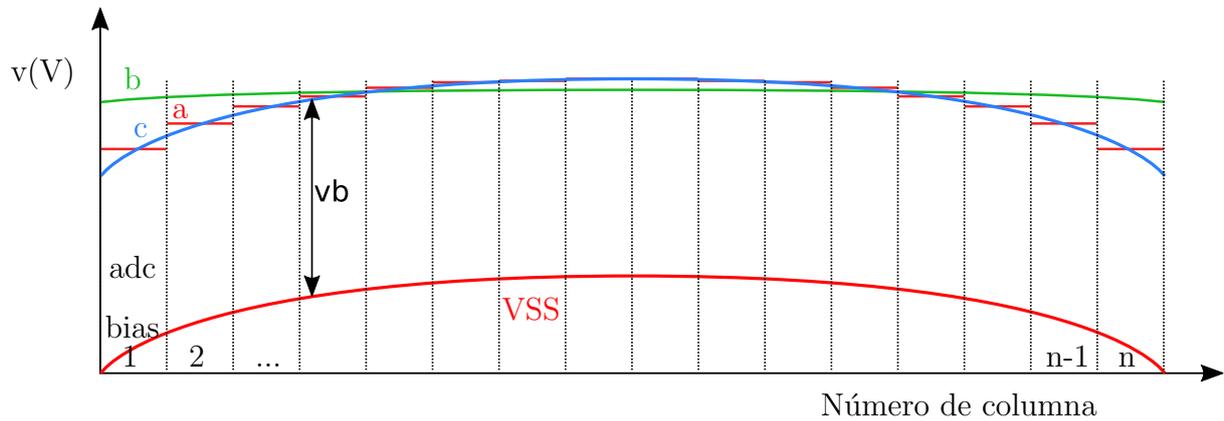


Figura 4.2: Curvatura en la tensión de tierra y una tensión de polarización a lo largo del canal, donde se diferencian 3 posibles estrategias a la hora de cortocircuitar las tensiones de polarización.

ción entre todos los canales del array. En la imagen 4.2 podemos ver 3 casos. El caso (a) lo que ocurriría si no cortocircuitamos los `adc_bias`, el caso (b), si lo hacemos con una línea metálica, con baja resistividad, y el caso (c) si lo hacemos con una línea más resistiva, de poly. La solución elegida es la última porque suaviza mucho mejor los voltajes de polarización entre los diferentes canales, al adaptarse a la irremediable curvatura de la tierra.

4.3.2. Acoplos

Otro tema crítico en el diseño del layout del canal de lectura es los posibles acoplos entre señales debidos principalmente a la apilación de canales requerida por la paralelización de varios píxeles y por el uso de la técnica de ping-pong. Es más, el uso del ping-pong hace incluso más crítico este tema, ya que va a significar que en los mismos periodos de tiempo van a concurrir fases de muestreo y comparación en canales que se encuentran apilados.

Algunas líneas como el `adc_take` que saldría de un canal superior, va a tener que, obligatoriamente, pasar por encima de algún canal inferior que esté haciendo *sampling*. Y, debido a que la transición del *take*, se produce en un momento indeterminado, en función de la luz captada por el píxel leído, puede coincidir que dicha transición se produzca en el periodo final de una fase de muestreo del otro canal. Si esta señal tiene un acoplo con uno de los nodos de entrada del comparador, o con los condensadores, puede ocurrir que

se produzca un pico en la señal que se está muestreando y al ser el final del periodo de muestreo, no dar tiempo a que se reestablezca y haber finalmente modificado el valor leído.

4.3.3. Condensadores del ADC

Los dos condensadores que debe haber por cada ADC son otro punto a tener en consideración, ya que son una parte fundamental del mismo. Lo primero que hay que decir es que suelen ser grandes, más aún si se trata de un sensor de bajo ruido, esto es, que está destinado a captar imágenes con escasa iluminación y se requieren altos rango dinámico y resolución, por lo que se necesitan condensadores mayores.

También son grandes porque habitualmente se implementan usando condensadores metálicos, que usan placas de metal 3 y 4 una sobre la otra. Este tipo de condensador tiene una densidad de capacidad por unidad de área menor que otros como los tipo MOS descritos en [2.4.2](#), pero se usan por la mayor linealidad que estos últimos.

Un punto a tener atención es al apantallado de estos condensadores entre los de las columnas vecinas. Si entre ellos hubiera una capacidad alta, podría haber una transferencia de carga si el valor almacenado en uno de ellos es muy diferente al otro, lo que crearía un efecto de desenfoque de los bordes abruptos en la imagen.

Otro problema que introducen los condensadores, en concreto, que estén fabricados los metales superiores 3 y 4, es que las señales que se distribuyen verticalmente, como el *adc.take* y la salida del píxel, que habitualmente se rutan en los metales más altos para facilitar el apantallado, van a tener que bajar para pasar por debajo de los condensadores. Esto va a posibilitar que se acoplen a otras señales que se ruten en metales inferiores.

4.3.4. Distribución de señales horizontalmente

Como ya se ha comentado en otros apartados, la distribución horizontal de las señales de control, o *fases*, necesita especial atención, sobre todo si el array consta de

muchas columnas.

Una de las principales señales horizontales es la rampa analógica. En el caso de producirse un retraso en la propagación de esta, los valores leídos serían mayores de los verdaderos. Por lo tanto siempre trataremos de disminuir dicho tiempo de propagación, lo que se consigue con baja resistencia, esto es, mayor anchura de líneas, y menor capacidad, mayor espaciado con otras líneas o con el sustrato. Por otra parte, es bueno que no se ensucie con posibles acoplos con señales verticales, por lo que habitualmente se elige rutar la rampa en metal Top de $10\mu\text{m}$ de anchura, sobre una capa de Metal 3 conectado a tierra como apantallamiento.

Siempre y cuando el error producido por el retraso de la rampa quede por debajo de un LSB, no se vería nada en la imagen debido a este retraso. Si no es así, y es inevitable un pequeño retraso, siempre se puede corregir posteriormente, ya que se trata de un error de patrón fijo “*Fixed Pattern Noise*” o FPN, que en la fase de caracterización del sensor, se podría corregir mediante unos parámetros de ajuste por canal.

En cuanto a los retrasos en las fases, debido a que estos afectan a las señales de muestreo, debemos tener más cuidado puesto que podría darse el caso extremo de que las fases de muestreo se descuadren demasiado con la señal de *sel* del píxel y que perdamos periodo de muestreo efectivo. Esto se debe corregir tratando de equalizar la constante de tiempo la señal de *sel* con el resto de fases. Hay que recordar que normalmente la primera tiene mucha menos libertad a la hora del diseño. La anchura y la capacidad del *sel* está muy acotada por el complejo diseño del layout del píxel. Por lo tanto solo podemos actuar sobre el diseño de las líneas de las fases del canal. Debemos comprobar bien anchuras, acoplos a otras líneas y tamaño y cantidad de dispositivos que cuelgan de ellas, tratando de equilibrar las constantes de tiempo de todas ellas.

Para esto, una vez más, es muy importante una cuidada extracción de parásitos y exhaustivas simulaciones post-layout, en diferentes canales a lo largo de todo el array.

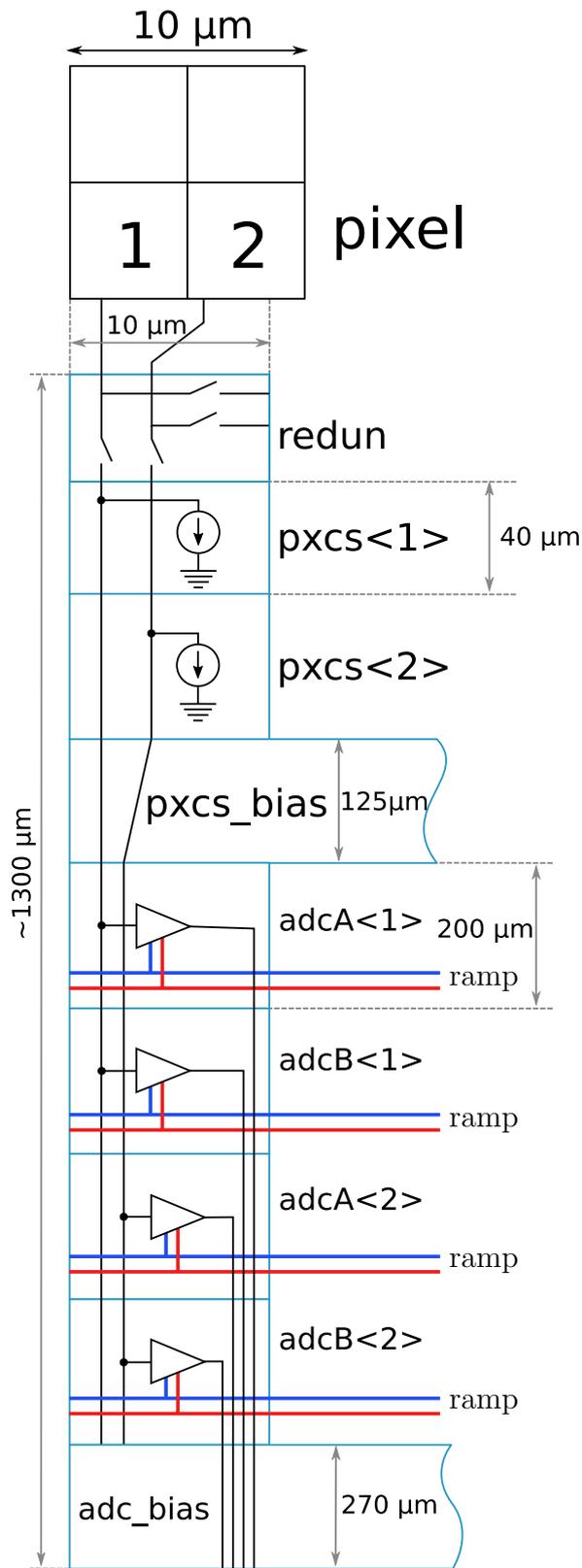


Figura 4.3: Floorplan del layout de una columna del canal de lectura (*ro_chx1*), donde se puede ver la disposición de los bloques principales y sus conexiones más importantes

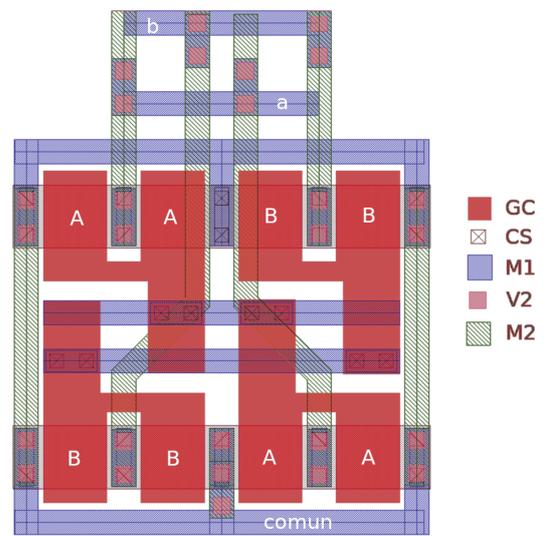


Figura 4.4: Layout del par diferencial del OTA del comparador, donde se ve la estructura de centroide común usada para minimizar el mismatch. A y B son los dos transistores del par diferencial M_1 y M_2 y “común” es el nodo que viene de la fuente de corriente M_5 del OTA. Véase esquemático en 3.6

Capítulo 5

Extracción y simulación

Como se ha ido presentando durante todo el trabajo, todo el trabajo de diseño, tanto esquemático como de layout, no tiene sentido sin hablar de un proceso de verificación de que lo que se ha diseñado funciona como queremos que funcione, y esto queremos comprobarlo, lógicamente, antes de que se fabrique el chip.

Las técnicas de simulación y extraído han evolucionado mucho en los últimos tiempos, con mejores algoritmos y procesadores más potentes ahora podemos simular circuitos mucho mayores que años atrás, llegando a los decenas de miles de transistores o cientos de miles de nodos.

El proceso de diseño es iterativo, se diseña según unos cálculos previos, se hacen simulaciones, primero a nivel de bloques pequeños y luego bloques más grandes y en función de los resultados se pueden ajustar algunos parámetros en el esquemático. Para afinar aún más siempre es necesario realizar también simulaciones post-layout, esto es, habiendo hecho previamente un extraído de parásitos debidos al layout.

5.1. Extracción de parásitos

El esquemático que se diseña en los primeros pasos, habitualmente no tiene ninguna componente debida a aspectos físicos de layout. Las conexiones entre dispositivos son ideales, sin resistencia ni capacidades ni inductancias, pero estas existirán una vez fabricado en el silicio.

Quizás en algunos casos como circuitos estáticos, de bajo consumo, o con amplios márgenes de diseño, el extraído de parásitos no es algo crítico puesto que van a afectar

poco al comportamiento general del circuito, aunque como norma se prefiere realizar siempre una comprobación. En otros casos en cambio, como puede suceder en el caso del canal de lectura, esta extracción es de suma importancia y hay que dedicarle un tiempo considerable a dicho proceso iterativo, en ocasiones alargándose más que el propio diseño inicial.

Los extraídos, por regla general se pueden hacer solo capacitivos, solo resistivos o ambos, y en ocasiones algunos extractores permiten extraer inductancias. El extraído capacitivo además se puede hacer en 2 dimensiones, entendiéndose con esto que solo se consideran las capacidades inmediatas, entre 2 metales a la misma altura o en capas consecutivas en altura. En contraposición podemos afinar más con un extraído 3D que sí considera líneas más alejadas sin acoplo directo. Para nuestro caso es muy recomendable esta alternativa dada la alta concentración de routing y la importancia de los acoplos.

El extraído es un proceso costoso computacionalmente, más aún si lo hacemos con alta resolución, 3D y tratándose de bloques grandes. Lo recomendable es extraer más de una sola columna de canal para ver posibles acoplos entre canales adyacentes y posibles asimetrías entre ellos debido a la asimetría introducida por el hecho de compartir bloques de polarización cada 8 canales. Esto implica que el layout de una columna de ADC, por ejemplo, a pesar de tener idéntico layout, puede llevar algunas señales verticales diferentes.

Lo que una extracción hace básicamente es calcular las capacidades que se ven entre líneas cercanas por layout y resistencias debidas a líneas metálicas o de poly. Posteriormente crea un nuevo circuito añadiendo todas las capacidades y resistencias que sean mayores que un cierto umbral seleccionado.

5.2. Simulación

La simulación, como ya hemos dicho, es un punto importante en el diseño de un circuito microelectrónico. Con una simulación se trata de modelar o imitar el comportamiento de los dispositivos cuando estén fabricados en silicio. Para lo cual, cada dispositivo instanciado consta de un model que define como son las tensiones y corrientes que se es-

tablezerán en sus nodos en función de las tensiones y/o corrientes que reciba y de las cualidades y dimensiones del dispositivo en concreto.

Tras establecer las relaciones de dependencias de unas variables con otras en la malla del circuito, el simulador resolverá el sistema de ecuaciones, y, si converge, dará una solución.

Las simulaciones pueden hacerse de varios tipos, estáticas o *dc*, estas solo calculan el punto de operación de todos los dispositivos y las corrientes y voltajes en cada rama y nodo, las dinámicas o transitorias, que calculan las evoluciones temporales de los voltajes y tensiones de los nodos y ramas partiendo desde una condiciones iniciales. Adicionalmente hay otros tipos de simulaciones, como barridos en frecuencia (*ac*), análisis de ruido en frecuencia, que tiene en cuenta los ruidos que generan los distintos dispositivos, análisis temporal de ruido, que es una simulación transitoria pero con ruidos añadidos.

Al igual que en el diseño de cualquier bloque, el diseñador debe pasar por varias etapas de simulado. Inicialmente, tras el diseño de los bloques pequeños, se harán simulaciones sencillas para comprobar que el bloque diseñado se ajusta a los valores calculados de punto de operación, ganancia, respuesta temporal, ruido...

En un estado más avanzado de diseño se integrarán los diferentes bloques unitarios para comprobar que juntos siguen actuando como se espera, sin fijarnos tanto en el comportamiento específico de una parte del mismo, sino en el rendimiento general del conjunto.

En cuanto el layout de los bloques esté diseñado, al menos una primera versión, es recomendable empezar a simularlo usando el extraído del mismo, ya que en según que casos, puede afectar considerablemente al rendimiento general. En una simulación post-layout lo que esperamos obtener son por ejemplo, formas de ondas más suaves y al mismo tiempo con pequeñas perturbaciones, al contrario de las simulaciones con esquemático, que suelen dar formas de ondas con bordes abruptos y sin perturbaciones.

Si por alguna circunstancia no es posible simular el circuito con el extraído, ya sea

porque todavía no está diseñado, o porque es demasiado grande y es difícil la extracción, siempre es buena práctica realizar alguna simulación en la que el diseñador incluya “a mano” los parásitos que considere más importante, calculándolos del existente o previsto layout.

Esta circunstancia se da muy habitualmente en el caso de las alimentaciones por ejemplo. Normalmente no es viable hacer un extraído de toda la línea que alimenta a un circuito desde los PADs de entrada al chip. Y por otro lado, solo nos interesa la posible caída de tensión que se produzca por el camino, y esta se obtiene con un cálculo simple. Añadiremos una resistencia estimada usando la resistividad, anchura y longitud de cada alimentación. Además se suele incluir un modelo experimental del PAD de entrada. El PAD es el área metálica cuyo aislante superior se eliminará para poder acceder con un hilo de oro y conectar una señal o alimentación interna con el exterior. El PAD consta normalmente de algunos diodos de protección ESD (descarga electrostática). Lo que se suele hacer habitualmente es simular usando el extraído de dicho PAD e incluir además un modelo eléctrico del hilo de oro, mediante una resistencia, capacidad e inductancia determinadas experimentalmente. La inclusión de dicho modelo podrá causar que en nuestra simulación veamos caídas de tensiones o efectos de segundo orden como rebotes o rizados en las señales, que se deberían estudiar y tratar de reducir.

Otro caso en el que no podríamos usar el extraído de todo el bloque sería al tratar de simular todo el canal de lectura, en nuestro caso, las 1320 columnas de canales correspondientes a las 2640 columnas de píxeles totales, activos, OB y dummies. Una simulación de el canal al completo es interesante desde el punto de vista de sistemas para analizar si hay problemas debidos a la propagación como el retraso o la degradación de las señales. Pero lo cierto es que no estamos interesados en obtener datos perfectos y precisos de todas y cada una de las columnas, sino que habitualmente solo queremos conocer el comportamiento en algunas columnas concretas, como por ejemplo la primera, última y alguna de la parte central.

Para esto, lo que se suele hacer es montar una simulación en la que los bloques de interés estén instanciados usando su circuito extraído, y el resto estarían agrupados en

paquetes de varios de forma que la carga total de sus líneas sea equivalente, y estos unidos horizontalmente por resistencias metálicas que emulen la resistencia total de la longitud del grupo de canales elegido. Por ejemplo, podríamos agrupar los canales en grupos de 128 canales, y con 10 grupos como este cubrimos los 1280 canales de la parte activa. De entre ellos podríamos escoger los extremos y uno del centro para separarlos en $127 + 1$ y tomar medidas en ese. También podríamos añadir canales OB y dummies ajustando el tamaño de los grupos.

Con una simulación así podríamos medir el retraso en las diferentes fases y la degradación que sufrirían los flancos en cuanto a *slew rate*, o distorsiones por acoplos durante el camino. Y lo más importante, analizar si dichos efectos son perjudiciales al rendimiento final, o si se observan diferencias apreciables en los valores leídos por las diferentes columnas.

Si en esta simulación añadimos un adecuado modelo de distribución de alimentaciones y emulamos que el array de píxeles esté iluminado con valores blancos y negros en filas alternativas, podríamos ver si existe un acoplo entre canales contiguos y que parte de la carga se transfiera a canales cercanos.

Por último, debemos mencionar las simulaciones de funcionalidad que se llevan a cabo en las fases finales del proyecto, poco antes de ser enviado a fabricar. Estas simulaciones consideran todo el chip, usando modelos Verilog, para las partes digitales y VerilogAMS para las partes analógicas. Dichos modelos simplifican muchísimo el funcionamiento del circuito; en la parte digital con valores digitales ceros y unos, y en la parte analógica se puede tratar con números reales. Con esto podemos simular la funcionalidad general del circuito con menor costo computacional, y así abarcar diseños mucho mayores. Con esto se pretende verificar que todas las señales activan correctamente el funcionamiento de los bloques, y estos envían sus salidas a los siguientes bloques correctamente.

Capítulo 6

Conclusiones

Tras todos los puntos expuestos en el trabajo, podemos concluir que el canal de lectura es un bloque que requiere de una cuidadosa planificación dentro del contexto del proyecto de diseño de un sensor de imagen. Los responsables del proyecto deben dedicar un tiempo amplio al estudio, diseño, layout, simulaciones y verificaciones del bloque.

Los canales de lectura son un tema muy presente en la literatura actual por parte de grupos de investigación y empresas de todo el mundo. El expuesto aquí es un canal estándar donde podemos apreciar bastantes puntos de la dificultad de su diseño, pero existen muchas vías de investigación abiertas a otras arquitecturas, nuevas funcionalidades, mayores resoluciones o menor ruido.

Por ejemplo existen arquitecturas innovadoras para convertidores de mayor resolución ideados para poder alcanzar menores niveles de ruido y poder captar imágenes en condiciones de baja iluminación. Un ejemplo de cómo conseguir esta mayor resolución puede ser dividiendo el ADC en dos etapas, como se expone en el artículo de la bibliografía[15] sobre un sensor de bajo ruido llamado LoNIS (*Low Noise Image Sensor*), desarrollado en AnaFocus[16].

Con lo expuesto en el trabajo, una de las conclusiones que podemos sacar es que el layout del canal de lectura puede condicionar ciertos aspectos del diseño del electrónico del mismo. Por ejemplo, algún acoplo que no podamos reducir porque imposibilidad en el layout, podría hacernos replantearnos el funcionamiento del bloque, o imponer una serie de restricciones a la hora de usarlo.

Todo esto nos lleva a otro punto de especial importancia, que es la necesidad de realizar un exhaustivo y detallado estudio de viabilidad. Este se debe hacer previo al inicio

del proyecto y debe tomar las especificaciones requeridas por el producto final y realizar los estudios pertinentes para saber si será posible la realización del diseño en tiempo, sin grandes costes y tratando de prevenir problemas y/o dificultades durante su puesta en práctica. Y, por supuesto, dicho estudio debe incluir el layout, y los problemas mencionados durante este trabajo.

Por otra parte, en el ámbito de los sensores de imagen, existen innovaciones en cuanto a los píxeles. Este es un tema en el que hay mucho por investigar aún, ya que en busca de nuevas mejoras se están investigando píxeles muy interesantes para aplicaciones 3D, usando la técnica de “ToF” (*Time of Flight*, tiempo de vuelo)[17][18]. Estos píxeles deben ser capaces de captar un pulso de luz emitido desde el sensor y calcular el tiempo que tarda en recibirse tras haber rebotado en algún objeto, para calcular la distancia al mismo. Para esto es necesario, además de píxeles que puedan transferir su carga mucho más rápido, mejoras en el canal de lectura que permitan esa integración de la señal recibida.

Bibliografía

- [1] TowerJazz, “CMOS Image Sensor Technology (CIS).” [Online]. Available: <http://www.jazzsemi.com/cmos-image-sensor.html>
- [2] Cadence Design Systems Inc., “Virtuoso Schematic Editor.” [Online]. Available: https://www.cadence.com/content/cadence-www/global/en_US/home/tools/custom-ic-analog-rf-design/circuit-design/virtuoso-schematic-editor.html
- [3] —, “Virtuoso Layout Suite XL.” [Online]. Available: https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/tools/custom-ic-analog-rf-design/virtuoso-vlsxl-ds.pdf
- [4] Mentor Graphics, “Calibre Physical Verification.” [Online]. Available: https://www.mentor.com/products/ic_nanometer_design/verification-signoff/physical-verification/
- [5] A. Moini, “Image Sensor Architectures,” in *Smart Cameras*. Boston, MA: Springer US, 2009, pp. 81–96. [Online]. Available: http://link.springer.com/10.1007/978-1-4419-0953-4_5
- [6] J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras*, ser. Optical Science and Engineering, J. Nakamura, Ed. CRC Press, aug 2005, vol. 113. [Online]. Available: http://books.google.com/books/about/Image_Sensors_And_Signal_Processing_For.html?id=UY6QzgzgieYC
- [7] E. R. Fossum, “Active pixel sensors: are CCDs dinosaurs?” *Proceedings of SPIE*, vol. 1900, pp. 2–14, 1993. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.408.6558&rep=rep1&type=pdf>
- [8] Kriplozoik, “File:Blooming example.jpg - Wikimedia Commons.” [Online]. Available: https://commons.wikimedia.org/wiki/File:Blooming_example.jpg
- [9] F. Jiménez-Garrido, J. Fernández-Pérez, C. Utrera, J. M. Muñoz, M. D. Pardo, A. Giuliatti, R. Domínguez-Castro, F. Medeiro, and A. Rodríguez-Vázquez,

- “High-Speed Global Shutter CMOS Machine Vision Sensor with High Dynamic Range Image Acquisition 1 and Embedded Intelligence,” *SPIE - The International Society for Optical Engineering*, vol. 8298, p. 10, feb 2012. [Online]. Available: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.912064>
- [10] B. Razavi, *Design of Analog CMOS Integrated Circuits 2nd*, 2016, vol. 6, no. 7. [Online]. Available: <http://www.lavoisier.fr/notice/frLMO62SSCRRJLOO.html%5Cnhttp://doi.wiley.com/10.1111/j.1151-2916.1994.tb07040.x>
- [11] A. Hastings, *The art of Analog Layout*, 2nd ed. Upper saddle River, NJ: Prentice Hall, 2001. [Online]. Available: http://encore.fama.us.es/iii/encore/record/C_Rb1745436?lang=spi
- [12] P. E. Allen and D. R. Holberg, *CMOS analog circuit design*, 2002.
- [13] K. Maex, “Silicides for Integrated-Circuits - TiSi₂ and CoSi₂,” *Materials Science & Engineering R-Reports*, vol. 11, no. 2-3, pp. 53–153, 1993.
- [14] S. Sordo-Ibanez, B. Pinero-Garcia, S. Espejo-Meana, A. Ragel-Morales, J. Ceballos-Caceres, M. Munoz-Diaz, L. Carranza-Gonzalez, A. Arias-Drake, J. Mora-Gutierrez, and M. Lagos-Florido, “An adaptive approach to on-chip CMOS ramp generation for high resolution single-slope ADCs,” *2013 European Conference on Circuit Theory and Design, ECCTD 2013 - Proceedings*, no. 1, 2013.
- [15] A. González-Márquez, A. Charlet, A. Villegas, F. Jiménez-Garrido, F. Medeiro, R. Domínguez-Castro, and Á. Rodríguez-Vázquez, “Design considerations for a low-noise CMOS image sensor,” in *Spie-Ist*, R. Widenhorn and A. Dupret, Eds., vol. 9403, no. March 2015. SPIE/IS&T, mar 2015, p. 94030P. [Online]. Available: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2087381>
- [16] “Teledyne AnaFocus.” [Online]. Available: <https://teledyne-anafocus.com/>
- [17] D. Portaluppi, E. Conca, and F. Villa, “32 x 32 CMOS SPAD Imager for Gated Imaging, Photon Timing, and Photon Coincidence,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 24, no. 2, 2018.
- [18] A. Payne, A. Daniel, A. Mehta, B. Thompson, C. S. Bamji, D. Snow, H. Oshima, L. Prather, M. Fenton, L. Kordus, P. O’Connor, R. McCauley, S. Nayak, S. Acharya,

S. Mehta, T. Elkhatib, T. Meyer, T. O'Dwyer, T. Perry, V. H. Chan, V. Wong, V. Mogallapu, W. Qian, and Z. Xu, "A 512 x 424 CMOS 3D Time-of-Flight image sensor with multi-frequency photo-demodulation up to 130MHz and 2GS/s ADC," *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, vol. 57, pp. 134–135, 2014.