Capítulo 6

Layout del circuito

6.1. Introducción

El capítulo 5 ha presentado la estructura y jerarquía del diseño a nivel de esquemático, presentando cada uno de los bloques desde un punto de vista de su estructura interna hasta nivel de transistor. Este nivel de descripción es suficiente para obtener resultados de simulación que permitan validar el trabajo, al menos de una forma preliminar. Sin embargo, para obtener una descripción completa del diseño es necesario especificar de forma precisa la imagen física en silicio del circuito: el layout. La fabricación del chip se realizará a partir de esta imagen, que deberá ser lo suficientemente compacta como para que sea sencilla de integrar con circuitos de mayor complejidad. De esta forma, el diseñador de sistemas AER podrá seguir diseñando sus sistemas con las mismas pautas que lo hacía hasta ahora, pero se les facilitará la labor de conexionado, pues la interfaz externa que se le proporciona es serie.

El diseño del layout viene siempre fuertemente condicionado por la tecnología a utilizar: las dimensiones mínimas que se permiten, reglas para crear contactos, niveles de metalización disponibles,... Como ya se ha nombrado varias veces, se ha utilizado una tecnología de 90 nm que dispone de siete niveles de metalización distintos para la realización de las conexiones. Las reducidas dimensiones mínimas que se pueden conseguir, unido con el elevado número de metales de los que se dispone, permiten altas densidades de integración y conectividad, lo que facilitará la consecución de los objetivos de diseño. No obstante, se intentará optimizar el conexionado para que no sea necesario el uso de todos los niveles de metalización y dejar los metales superiores para la integración de la interfaz de comunicaciones con los pads de salida o con otros circuitos que puedan alimentar la entrada de datos del serializador-deserializador.

Para llevar a cabo el diseño del layout de una forma efectiva y rápida se han desarrollado en los últimos tiempos herramientas software de ayuda al diseño que permiten sistematizar en gran medida la labor del diseñador. Estas herramientas son muy diversas e incluyen desde programas para el rutado automático hasta potentes programas para la verificación del diseño. En este proyecto se han utilizado básicamente dos de ellas:

- **DRC (*Design Rules Checker*)**: se utiliza para comprobar que el layout generado cumple con todas las reglas que el fabricante utiliza para asegurar que el circuito se puede fabricar con la precisión necesaria. Ejemplos típicos de reglas de layout son: longitud mínima del canal de los transistores, separación mínima entre pistas de metal adyacentes, anchura mínima de las pistas de polisilicio,...
- **LVS (*Layout Versus Schematic*)**: sirve para comprobar que la correspondencia entre la vista layout del circuito y la esquemática es total. Es decir, se encarga de verificar que las conexiones se han hecho correctamente y que verdaderamente se está implementando el circuito que se desea. El software extrae de forma automática una descripción del circuito (un *netlist*) a partir del layout y del esquemático, para luego compararlos y comprobar si coinciden o no.

En este capítulo se tratarán de mostrar las ideas fundamentales que han guiado todo el proceso de diseño, el estilo de layout adoptado y los resultados obtenidos.

6.2. Layout mediante celdas estándar

El diseño mediante celdas estándar tiene como principal ventaja la de permitir la automatización del layout de circuitos digitales, consiguiéndose a partir de ella resultados compactos y robustos. De hecho, muchas herramientas de síntesis de circuitos digitales utilizan este estilo para generar layouts de forma automática a partir de una descripción de la funcionalidad en algún lenguaje de descripción hardware (VHDL, Verilog,...). No obstante, para la realización de este proyecto el rutado no ha sido automático, sino que se ha realizado manualmente. Esto se ha hecho así principalmente por dos razones: por un lado la necesidad de optimizar el diseño en cuanto a su conectividad externa y, por otro lado, conciliar la convivencia en el mismo layout de celdas analógicas y digitales.

Otra de las ventajas esenciales de usar el estilo de layout es que permite una sencilla reutilización de las primitivas de diseño. Esto es especialmente interesante cuando este diseño se hace a través una serie de bloques básicos (biestables, puertas lógicas, transistores, ...) que se repiten múltiples veces: si el diseño se hace de forma inteligente, con sólo implementar una celda y repetirla se puede hacer todo el rutado del circuito sin problemas. Eso simplifica enormemente el diseño y permite ahorrar tiempo de computación, pues sólo se necesita verificar una sóla vez el bloque, pudiéndose repetir luego todas las veces que se desee. Esta forma de proceder permite crear bibliotecas de diseño que ayuden a sistematizar al máximo el proceso de generación de layouts.

En el diseño implementado los bloques digitales se tomaron directamente de las librerías de elementos basados en celdas estándar proporcionados por *STMicroelectronics*, mientras que los elementos analógicos o mixtos han sido elaborado siguiendo una metodología *full-custom* (diseño a nivel de transistor). En este último caso se ha intentado integrar el estilo de layout con el diseño de celdas estándar para que a la hora de unir todos los bloques el diseño fuera lo suficientemente

compacto. El conexionado interno de cada una de las celdas se ha hecho en metal-I (primer nivel de metalización) intentando optimizar al máximo el área ocupada. La interconexión entre bloques se hizo con metal-II o metal-III en función de las restricciones que impusieran en cada momento las propias conexiones.

La figura 6.1 muestra un esquema del rutado utilizado basado en celdas estándar, donde se han indicado todos los elementos que intervienen en el rutado de los bloques. La idea básica que hay detrás de este estilo de layout es fijar la altura de las celdas utilizadas y hacer el diseño variando la anchura de las mismas. Esta altura fija vendrá determinada por la distancia entre las pistas que sirven de alimentación (V_{dd}) y tierra (GND) y que comparten todas las celdas. De esta forma, cuando se quieran conectar dos celdas simplemente hay que colocarlas en el layout lo suficientemente cercanas como para que se produzca la conexión entre las pistas de V_{dd} y GND, quedando una estructura extremadamente compacta.

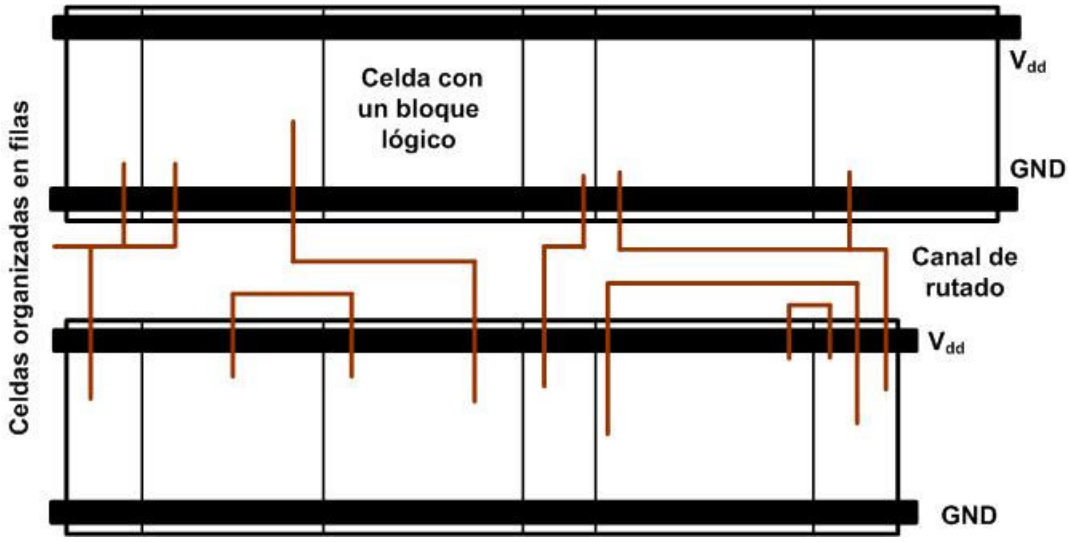

Figura 6.1: Estilo de layout por celdas estándar

Como ya se ha dicho, dentro de cada celda se utiliza metal-I para conectar los transistores entre sí. El siguiente paso es conectar las distintas celdas entre sí para formar unidades mayores de procesado. Una opción sería utilizar el área activa de la propia celda y niveles de metalización superiores para conectar entradas y salidas de las celdas. Como se tienen muchos metales para utilizar, esta opción podría ser viable en cuanto a los recursos de conexionado, pero llevaría a circuitos muy complejos donde los errores de DRC o de conexionado serían muy probables. No obstante, usando esta técnica se conseguirían los resultados óptimos en cuanto a área, pues se eliminaría el canal de rutado que viene representado en la figura 6.1. Sin embargo, hay que recordar que este circuito vendrá integrado con unos pads de salida y unos circuitos LVDS de gran tamaño, de forma que ahorrar unos cuantos micrometros por usar un estilo de layout más complejo no va a suponer que el sistema completo sea mucho más eficiente en cuanto a área. Por tanto, se ha optado por seguir el estilo marcado por la figura 6.1, usándose el canal de rutado para la conexión de los distintos bloques entre sí. Como ventaja de usar este método se consigue

que el rutado sea más sencillo y robusto, pues al tener más espacio se pueden usar contactos muchos más grandes ue son menos propensos a errores.

Una de las particularidades de nuestro diseño es que se tienen que integrar partes digitales con analógicas. Al haberse adoptado un estilo de layout basado en celdas estándar para la parte digital, es necesario adaptar la parte analógica a estos requerimientos. Para hacer esto simplemente habrá que tener en cuenta que la posición de las líneas de alimentación y tierra nos viene fijada a una altura determinada por la librería digital utilizada. Esto introducirá limitaciones en el rutado adicionales, pero a cambio se conseguirá una mejor integración entre las partes digital y analógica. La figura 6.2 muestra un ejemplo de un bloque mixto (el elemento de retraso programable compuesto por una llave de paso y un inversor digital) adaptado para ser integrado en una tecnología de celdas estándar.

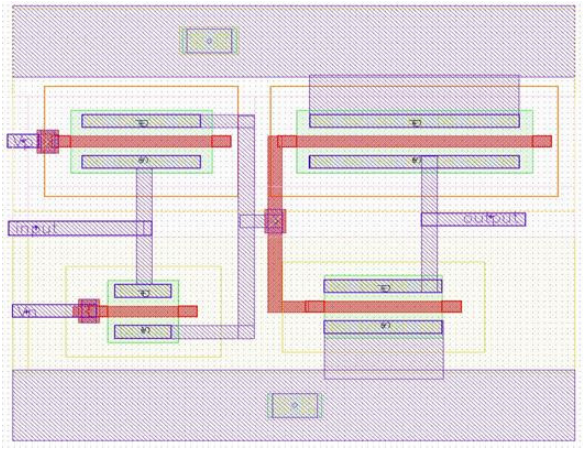

Figura 6.2: Ejemplo de bloque mixto

6.3. Layout de la interfaz

El objetivo de este apartado es el de presentar los resultados obtenidos tras el proceso de diseño del layout a través de un análisis de los bloques generados. Se van a presentar una serie de figuras donde el lector podrá comprobar todas las consideraciones de diseño mencionadas en los apartados anteriores y podrá reconocer cada uno de los bloques de los que se componen los circuitos de la interfaz. Por otro lado, este apartado va a estar centrado en los cicuitos para la serialización y deserialización de los datos AER, que han sido los que se han diseñado completamente. Los bloques que se usan para implementar el driver y el receptor LVDS son de librería, por lo que no tiene mucho sentido detallarlos en exceso.

En la figura 6.3 se muestra una vista general del circuito transmisor (el serializador junto con el codificador Manchester), donde se han señalado cada uno de los bloques que lo compone. En total, la celda tiene un área de 112.35×18.14 μm, habiéndose utilizado unas celdas estándar de altura 4.04μm.

Figura 6.3: Layout del circuito transmisor

La figura 6.4 muestra un detalle de los bloques que constituyen el serializador, así como de la interconexión interna de los mismos siguiendo el estilo de layout mostrado en el apartado 6.2 de este capítulo.

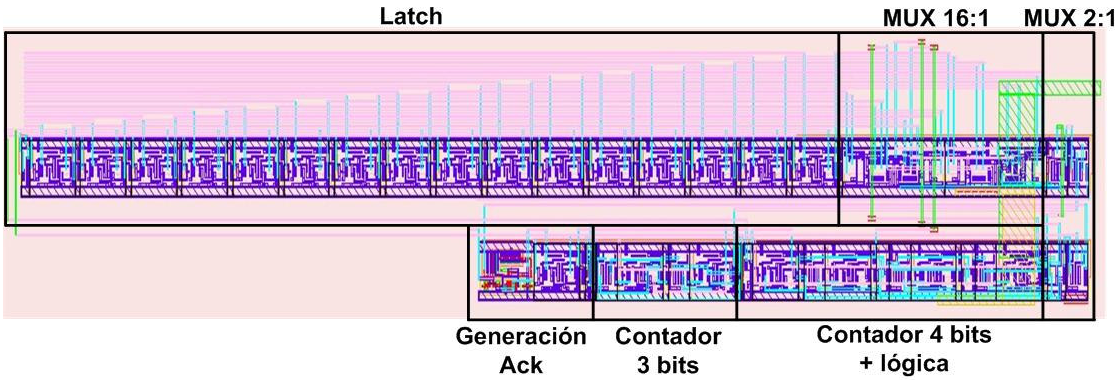

Figura 6.4: Layout del serializador

El codificador Manchester se detalla en la figura 6.5, pudiéndose distinguir tanto los biestables usados para sincronizar los datos con el reloj como la lógica de generación de los datos.

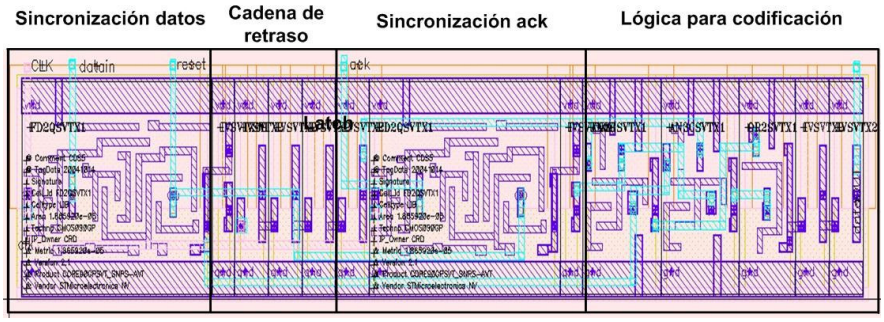

Figura 6.5: Layout del codificador Manchester

Los elementos anteriormente presentados se corresponden con la etapa de transmisión de la interfaz. La figura 6.6 se centra en la etapa de recepción, mostrando el deserializador utilizado en la interfaz, donde se han señalado los componentes más relevantes del mismo. El receptor

se ha diseñado siguiendo el mismo estilo de layout que el transmisor, habiéndose obtenido una celda de dimensiones 117.17 × 25.1 μm. La mayor complejidad del receptor, debida sobre todo a la etapa de salida con un registro de desplazamiento y un latch de 16 bits, hacen que el consumo de área sea mayor.

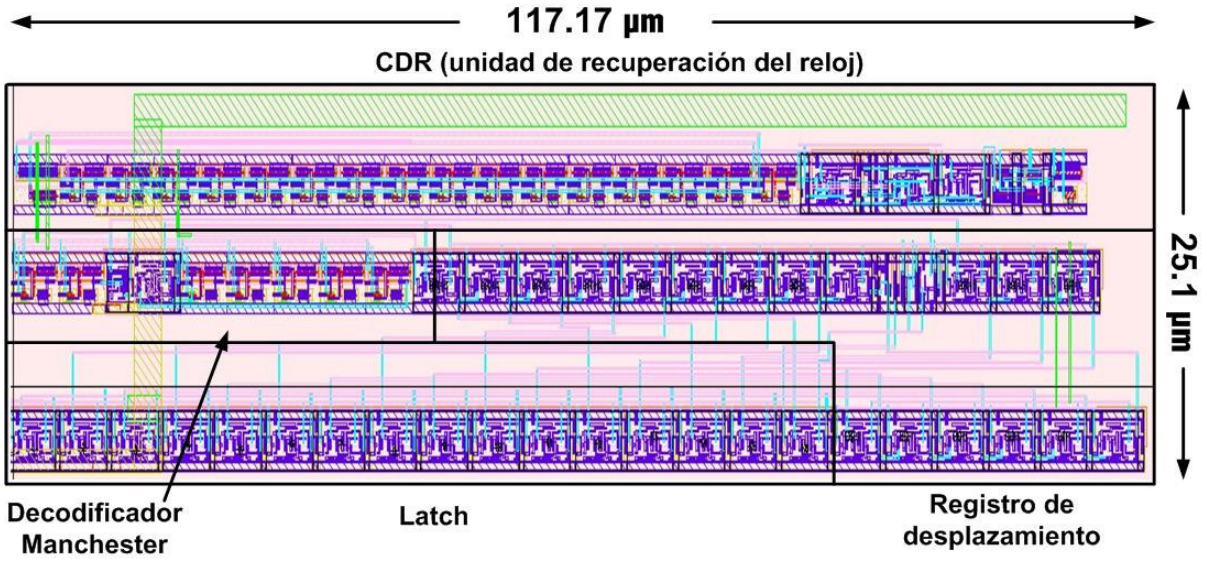

Figura 6.6: Layout del deserializador

La figura 6.7 muestra el layout del circuito utilizado para la recuperación del reloj, así como la decodificación de los mismos a partir de esta señal. Se pueden distinguir cada uno de los bloques que intervienen en el proceso: el retraso de 360º para generar la señal de referencia, el PFD para realizar la comparación de fase y el LPF para generar las tensiones de control para la modificación de las unidades de retraso.

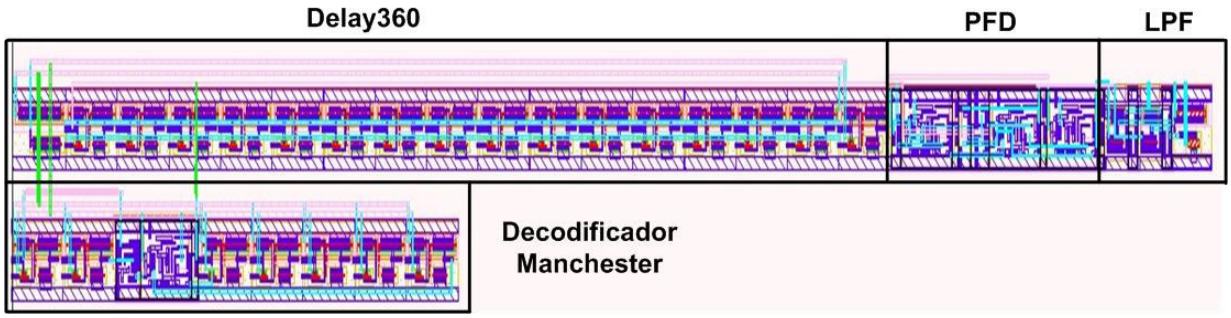

Figura 6.7: Layout del CDR (Clock Data Recovery)

El bloque encargado de desfasar la señal de reloj para generar la señal de referencia simplemente es la conexión en cascada de 16 elementos de retraso, por lo que no tiene mayor relevancia con respecto al layout. Sin embargo, los bloques PFD y LPF sí tienen mayor interés y no se distinguen del todo bien en la figura anterior. Por ello se presenta en la figura 6.8 una ampliación de estos bloques, donde también se ha intentando diferenciar cada uno de los bloques

que componen el circuito.

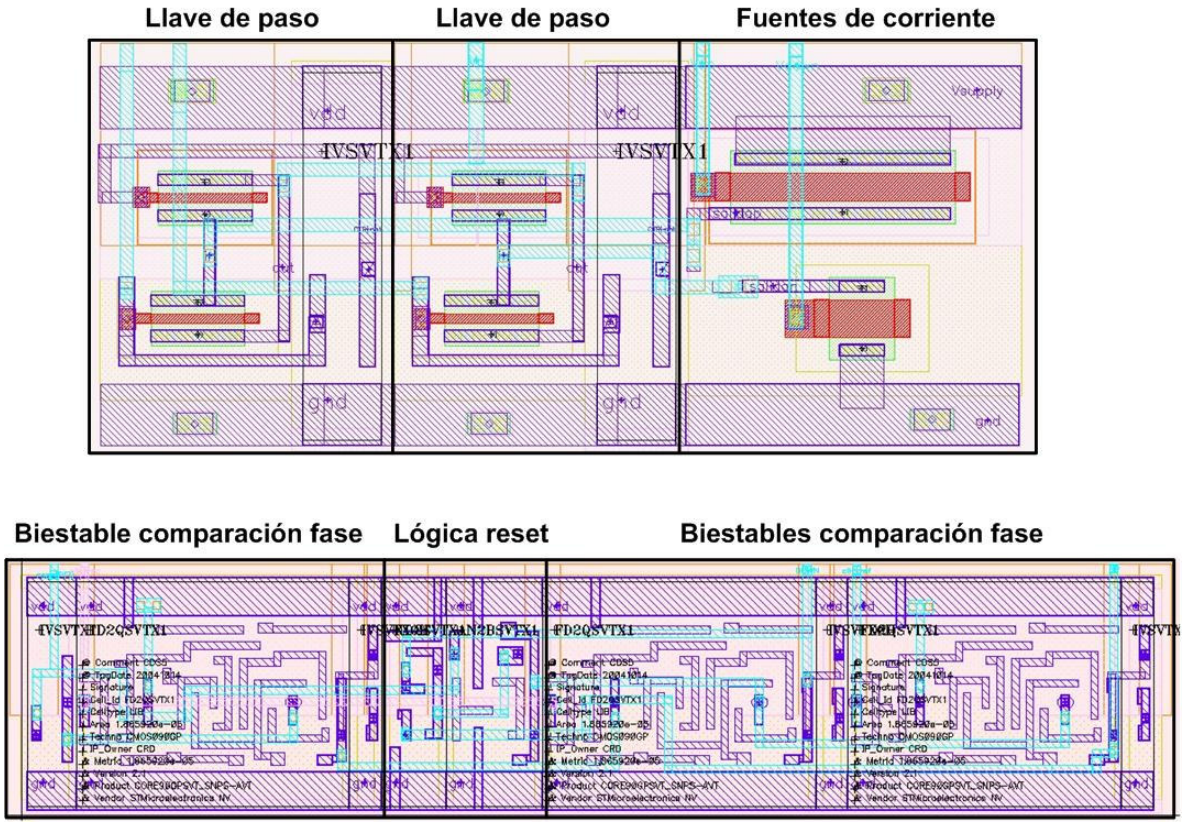

Figura 6.8: Layout de los bloques que componen el DLL

La figura 6.9 presenta el layout del decodificador Manchester utilizando los retrasos programables ya mostrados y la lógica destinada a extraer la información de sincronización de los pulsos recibidos.

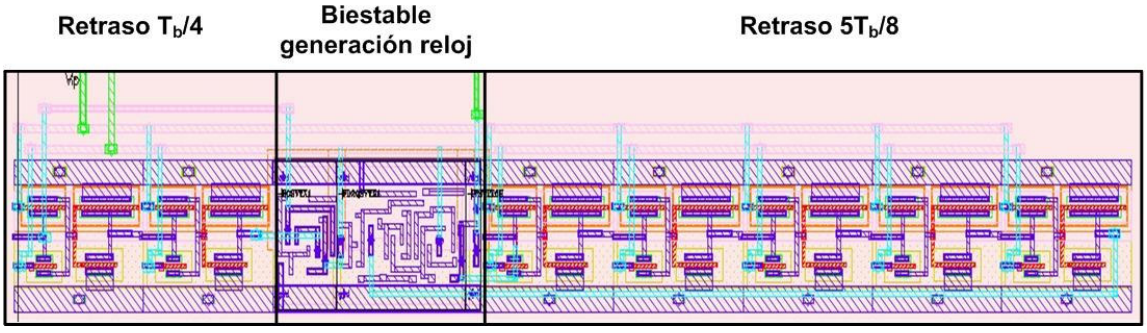

Figura 6.9: Layout del decodificador Manchester

Por último, se presenta una vista general de todo el diseño realizado (figura 6.10), habiendo integrado el serializador-deserializador con los circuitos LVDS de biblioteca usados. Se puede observar en dicha figura que el mayor consumo de área se debe a los circuitos de interfaz, mientras que la parte de serialización-deserialización de los datos supone sólo una pequeña fracción.

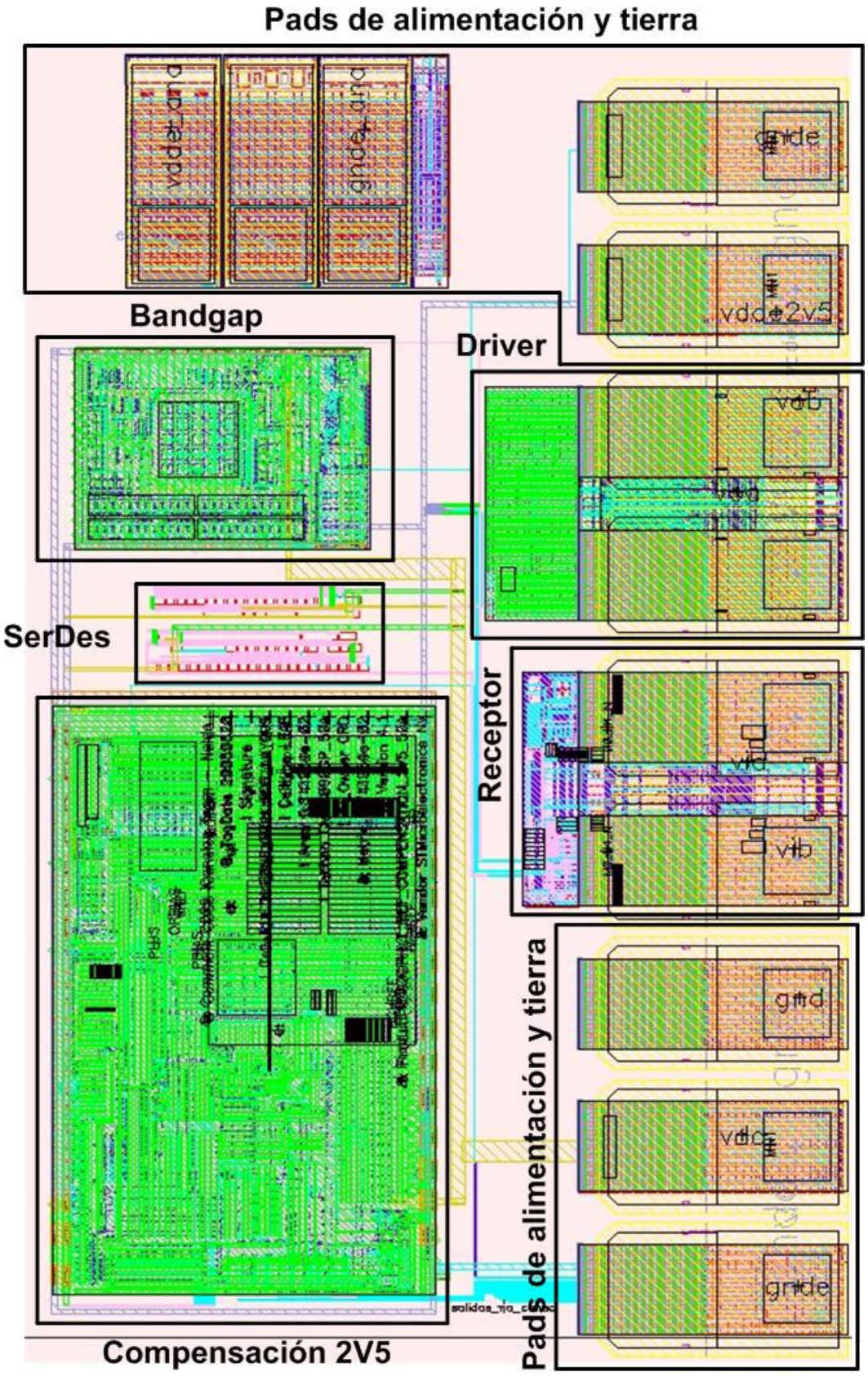

Figura 6.10: Layout completo: vista global