

Universidad Tecnológica Nacional
Facultad Regional Córdoba

Trabajo práctico N°6: Flujo de diseño analógico con CMOS

Dispositivos Electrónicos

3R2

Coordinador: Prieto Angelo 401012
Documentador y operador: Gaston Grasso 401892

Fecha de entrega: 30-11-2025

Índice

1	Introducción	2
2	Actividad 1: especificaciones de diseño	3
2.1	Instalación de herramientas	3
3	Actividad 2: flujo de diseño analógico	4
3.1	Schematic design	4
3.2	Simulation	4
3.3	Layout: placement / routing	6
3.4	DRC	6
3.5	Post-layout	6
4	Conclusión	9

Introducción

En el presente trabajo práctico se aborda el flujo completo de diseño analógico utilizando tecnología CMOS, con el objetivo de comprender y aplicar las etapas necesarias para desarrollar un circuito integrado desde su especificación funcional hasta su implementación física.

Para ello, se emplea el proceso tecnológico SKY130, un Process Design Kit (PDK) de libre acceso que permite realizar todas las etapas del diseño, desde la simulación eléctrica del circuito hasta la verificación del layout respetando reglas de diseño reales. Asimismo, se utilizan herramientas opensource de diseño de circuitos integrados, con el fin de validar esquemáticos, realizar simulaciones y asegurar la correctitud física mediante verificaciones DRC (Design Rule Check) y LVS (Layout Versus Schematic).

El circuito elegido para este trabajo es un inversor analógico CMOS, una de las celdas más básicas y esenciales en el diseño digital y analógico. Su estructura, compuesta por un transistor PMOS y un transistor NMOS en configuración complementaria, permite obtener un comportamiento eficiente y robusto, lo que lo convierte en un punto de partida ideal para comprender el flujo de diseño y las particularidades de la tecnología CMOS.

A lo largo del informe se describen las etapas seguidas en el desarrollo del inversor: el análisis y simulación del esquemático, la implementación del layout siguiendo las reglas específicas del proceso, y las verificaciones necesarias para asegurar la fabricabilidad del diseño. De esta manera, se integra el conocimiento teórico estudiado con herramientas prácticas modernas, brindando una experiencia completa y representativa del flujo de diseño utilizado en la industria de circuitos integrados.

Actividad 1: especificaciones de diseño

El inversor CMOS a diseñar debe cumplir las siguientes especificaciones:

Tecnología: SKY130 (130 nm CMOS).

Voltaje de alimentación (VDD): 1.8 V.

Características del transistor PMOS:

- $L = 0,15$ (Longitud de canal del transistor en micrómetros)
- $W_p = 2,1$ (Ancho del canal del transistor en micrómetros)
- $nf = 1$ (Número de fingers)
- $mult = 1$ (Multiplicidad)
- model: pfet_01v8

Características del transistor NMOS:

- $L = 0,15$
- $W_n = 1,05$
- $nf = 1$
- $mult = 1$
- model = nfet_01v8

Ajustar Capacidad de carga de salida: 2 pF (considerando fan-out).

Tiempo de transición objetivo: menor a 50 ns para subida y bajada.

Instalación de herramientas

Para el diseño y simulación del inversor CMOS se utilizaron herramientas opensource compatibles con el PDK SKY130:

- Xschem para diseñar y simular esquemáticos.
- Ngspice: modelos spice para simulación eléctrica en el entorno de Xschem.
- Magic para ruteo y layout, servirá ademas para verificación del diseño físico (DRC).
- Netgen para realizar la comparación LVS (Layout versus Schematic).

Actividad 2: flujo de diseño analógico

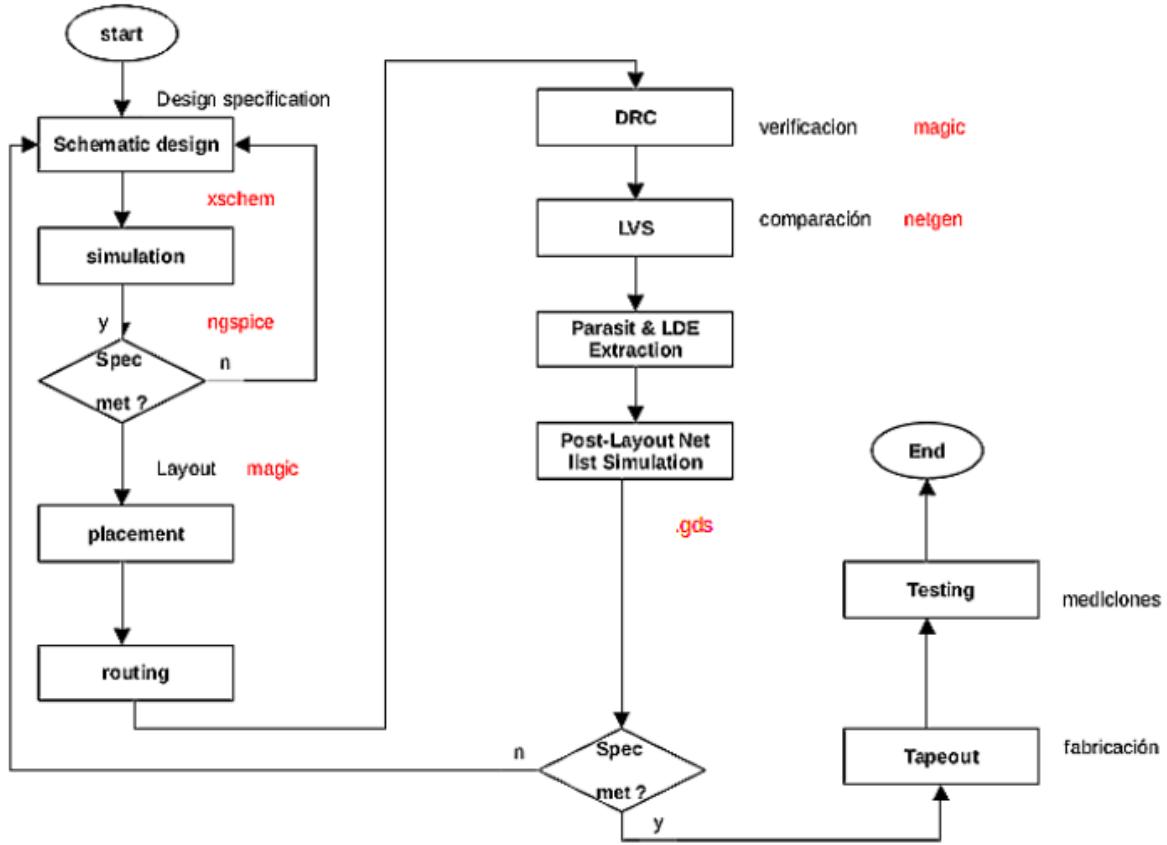


Figura 3.1: flujo de diseño.

Para esta actividad, se sigue cuidadosamente el procedimiento descrito en las consignas del trabajo.

Schematic design

Para este parte del diseño se ha utilizado el programa Xschem. Se puede observar el resultado obtenido en las figuras 3.2 (esquemático del inversor) y 3.3 (símbolo diseñado para el inversor).

Simulation

Para esta parte del diseño también se ha utilizado el programa Xschem. Se ha insertado el símbolo del inversor diseñado en un circuito, y se ha realizado las simulaciones pertinentes,

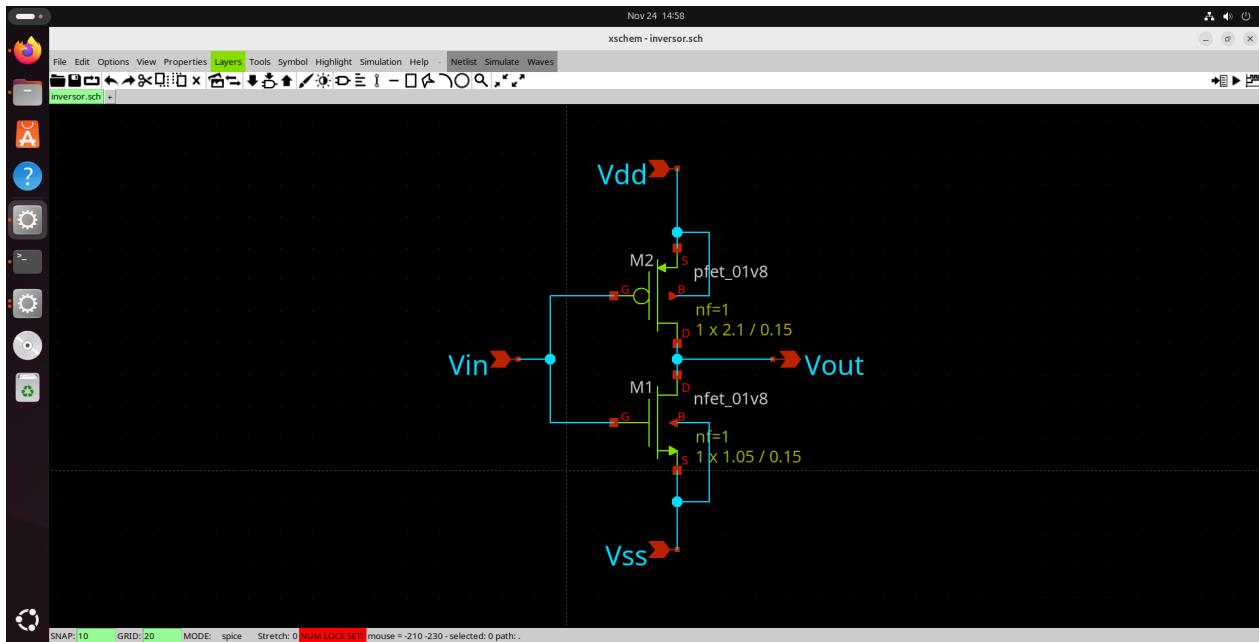


Figura 3.2: captura de pantalla en programa Xschem. Esquemático de inversor.

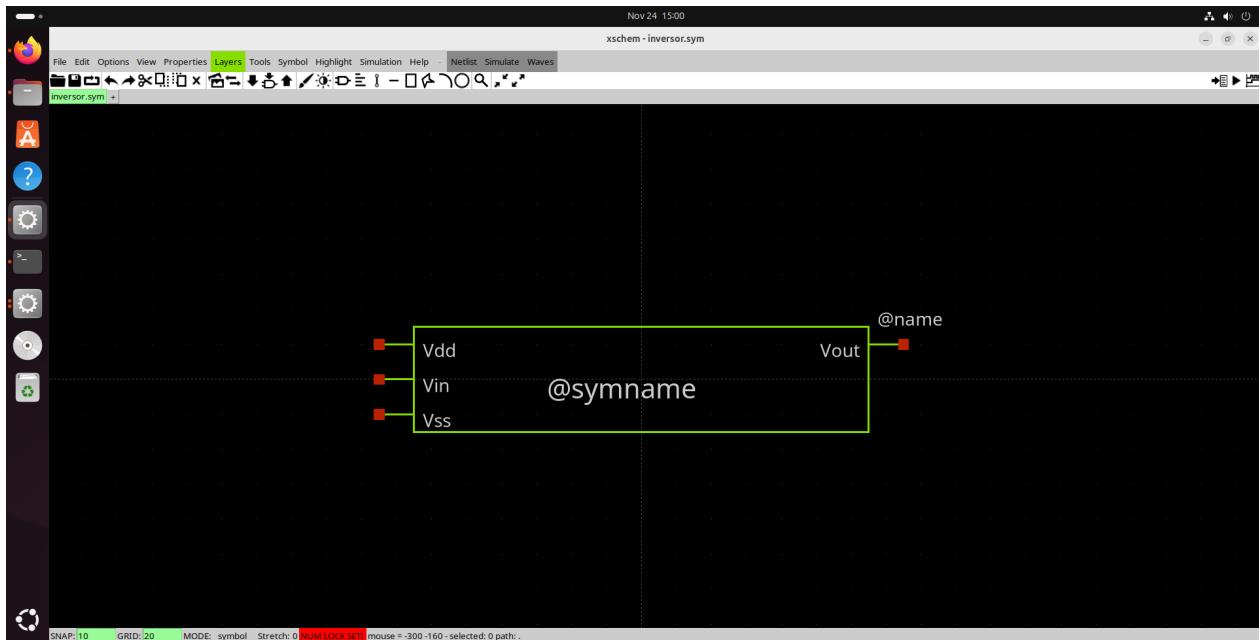


Figura 3.3: símbolo de inversor.

comprobando así el correcto funcionamiento. Se puede observar el resultado en la figura 3.4.

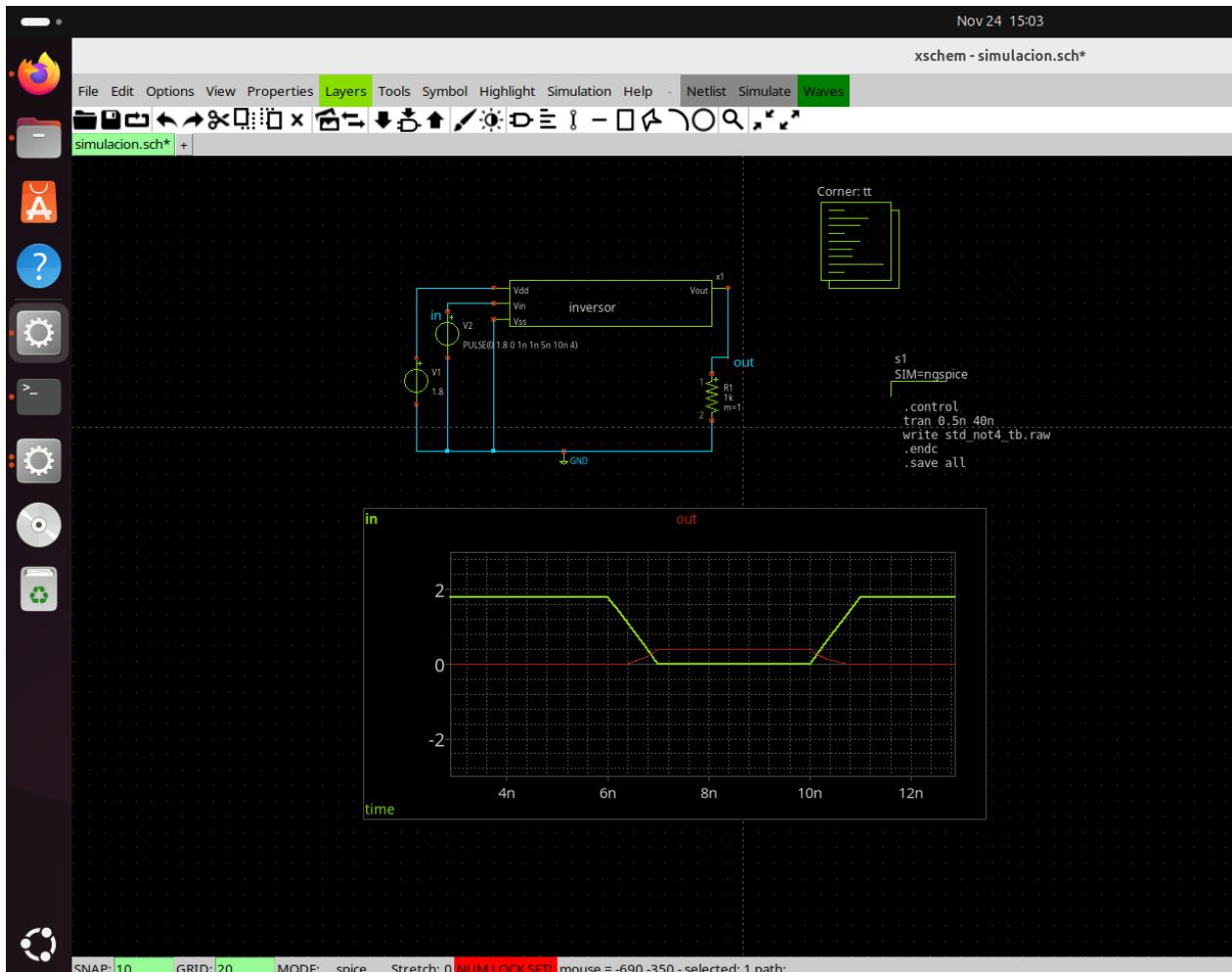


Figura 3.4: simulación de esquemático diseñado.

Layout: placement / routing

Para esta parte del diseño se ha utilizado el programa Magic. Se puede observar el resultado obtenido en la figura 3.5.

DRC

Para esta etapa, se ha utilizado nuevamente el programa Magic, y se ha ejecutado el verificador de reglas de diseño DRC. Se ha obtenido un resultado satisfactorio al no presentarse ningún error. Se puede observar en la figura 3.6.

Post-layout

Una vez finalizado el diseño físico del circuito, se realizaron simulaciones post-layout con el fin de evaluar el impacto de los efectos y parásitos introducidos durante el proceso de

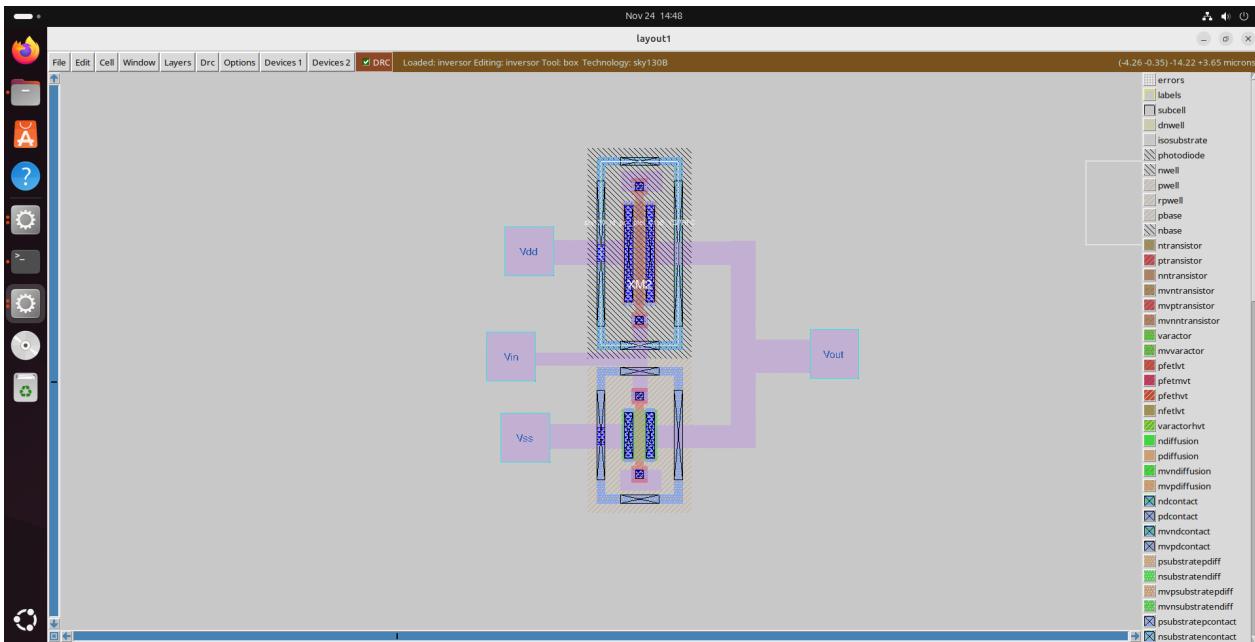


Figura 3.5: enrutado de chip.

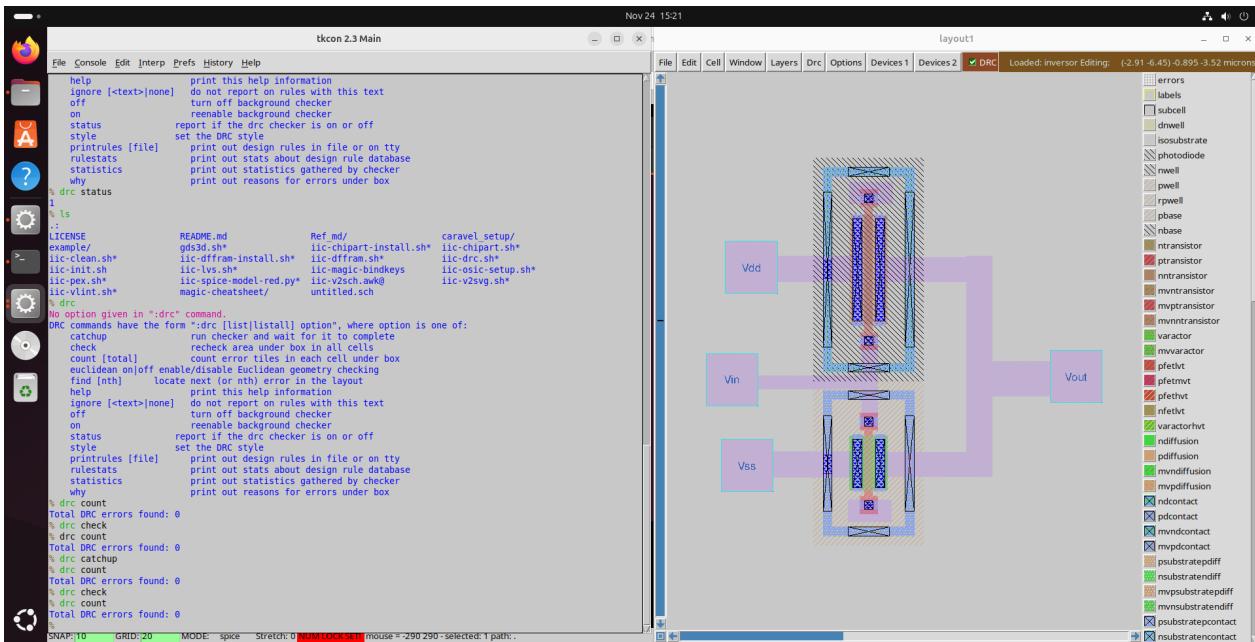


Figura 3.6: captura DRC.

layout. Para ello, primero se efectuó la extracción de los elementos parasitarios, obteniendo un netlist que incluye las resistencias y capacitancias asociadas a las interconexiones y a la geometría final del diseño.

Posteriormente, este netlist fue combinado con los modelos del proceso del PDK Sky130, lo que permitió incorporar las características eléctricas reales de los dispositivos. Con esta

información, se llevaron a cabo simulaciones en un entorno SPICE para analizar parámetros como retardos de propagación, integridad de señal y consumo de potencia.

Luego, se compararon los resultados post-layout con las simulaciones pre-layout basadas únicamente en el esquemático.

Finalmente se generó el archivo .gds, el cual se manda a la foundry para iniciar la producción del chip.

Conclusión

En conclusión, pudimos cumplir los objetivos del trabajo práctico: diseñamos el inversor CMOS en Xschem, lo simulamos, hicimos el layout en Magic y llegamos hasta generar el archivo .gds listo para enviar a la foundry. Eso nos permitió ver el flujo completo, desde el esquemático hasta la implementación en silicio usando el proceso SKY130.

En el camino aparecieron varias trabas con las herramientas opensource (configuración del PDK, integración entre programas, mensajes de error poco claros), pero justamente eso ayudó a entender mejor qué hace cada etapa del flujo de diseño de microelectrónica y a valorar todo el trabajo que hay detrás de un circuito integrado, incluso para una celda tan simple como un inversor.