

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Diseño de un Circuito Integrado con Tecnología de 180 nm
usando Librerías de Diseño de TSMC: Ejecución de la Síntesis
física, Design Rule Check y Corrección de Errores Obtenidos.**

Protocolo de trabajo de graduación presentado por Antonio Altuna
Hernández, estudiante de Ingeniería Electrónica

Guatemala,

2021

Resumen

El objetivo de esta investigación es mejorar la síntesis física, actual, como parte del flujo de diseño para el desarrollo de circuitos integrados digitales a nanoescala. Esta por ser la tercera iteración del proyecto, para realizar el flujo de diseño, se cuenta con la documentación de las promociones pasadas. El primer paso para realizar este proyecto es el análisis de los resultados anteriores. Para hacer esto se replicarán los trabajos realizados anteriormente por: Luis Nájera y Luis Abadilla, que fueron los encargados de la síntesis física en su respectiva promoción. Luego de esto se migrarán estos procesos de la herramienta de Custom Compiler I a Custom Compiler II. Además se verificarán las librerías de **TSMC** que se están utilizando para corroborar que estén actualizadas y sean compatibles con el proceso de manufactura para el que se está diseñando.

Para etapa de verificación y validación de los resultados se realizarán los siguientes tres procesos de verificación: *Design Rule Check*, *Electrical Rule Check* y *Antenna Rule Check*. Para estos procesos de verificación también se cuenta con la documentación de los trabajos de Matthias Sibrian y Marvin Flores. Con las librerías en orden en conjunto con la síntesis física, se estarán haciendo múltiples pruebas con diferentes circuitos eléctricos digitales como: una compuerta NOT, un Full Adder, un Ripple Carry Adder y un chip de diseño personalizado. Estos cuatro circuitos serán pasados por el proceso de síntesis física y las verificaciones mencionadas anteriormente para validar que los resultados obtenidos por la síntesis física se puedan manufacturar por la empresa **TSMC**.

Antecedentes

La Universidad del Valle de Guatemala (UVG) a lo largo de los años se ha caracterizado por estar a la vanguardia en la investigación de nuevas tecnologías y metodologías en el área de las ciencias aplicadas. El Ing. Carlos Esquit, en el 2009, ingresó como nuevo director del departamento de Ing. Electrónica y Mecatrónica de la universidad y consigo trajo las destrezas y aprendizajes adquiridos en Estados Unidos en las ramas de micro y nano electrónica. Al iniciar, realizó una reforma al mapa curricular de Ing. Electrónica para enseñar dichas disciplinas aprendidas en el primer mundo.

En el 2013, se inicia a impartir el curso de Introducción al diseño de sistemas **VLSI**. Al inicio las herramientas con las que se impartió este curso fueron gratuitas y permitían realizar diseños básicos a micro y nano escala de circuitos. Al año siguiente, el ingeniero formó una alianza con la empresa Synopsys. Esta se encuentra a la vanguardia en lo que se refiere a soluciones que facilitan el diseño en silicio de chips y su verificación. Esta proporcionó herramientas que actualmente utilizan los líderes de la industria de semiconductores. Con estas herramientas se realizó el primer diseño en silicio a nanoescala de 28nm, realizado por el Ing. Jonathan de los Santos [1] en 2014.

En la nueva reforma curricular, que entró en vigencia en 2015, se añadieron los cursos de Nanoelectrónica 1 y 2. El fin de este cambio, que actualmente sigue vigente, es impartir teoría de **VLSI** y desarrollar en los alumnos competencias en el área de investigación del diseño de chips a micro y nano escala. Esto utilizando las herramientas que el software Synopsys le proporciona a la UVG. Con estas herramientas en el 2019 y 2020 se realizó el

diseño para el primer chip con tecnología CMOS a nanoescala en Guatemala.

En el año 2019 los alumnos Luis Nájera y Steven Rubio en conjunto con el apoyo de Interuniversity Microelectronics Centre (**IMEC**) y bajo la supervisión del Ing. Carlos Esquit desarrollaron lo que fue la primera iteración de este proyecto. Los resultados de este equipo fueron la implementación de las librerías de **TSMC** en el desarrollo de un flujo de diseño para la fabricación del primer circuito integrado en la historia de Guatemala, abriendo paso a una nueva área de investigación en la UVG. Los trabajos que documentan dichos resultados se encuentran en [2] y [3].

Los antecesores directos a este proyecto que trabajaron en el ciclo 2020 a 2021 en lo que se refiere a la síntesis física, Verificación **ERC** (Electrical Rule Check), Antenna Rule Check y verificación **DRC** (Design Rule Check) fueron el quipo de: Luis Abadilla [4], Marvin Flores [5] y Matthias Sibrian [6]. Los resultados obtenidos por ellos fueron diferentes simulaciones funcionales, con las librerías de **TSMC** en los programas IC Validator y IC Compiler I. Adicionalmente dejaron documentación para utilizar los programas y scripts para replicar su trabajo.

Justificación

La fabricación de Circuitos Integrados (**CI**) puede ser un proceso largo y riguroso. El elaborar un flujo de diseño para la fabricación de estos **CI** abre nuevas oportunidades para lo que es el diseño y manufactura de estos dispositivos facilitando y automatizando el proceso. En el mercado actualmente hay una gran variedad de **CI** de propósito general. Sin embargo el poder crear **CI** dedicados permite que los dispositivos electrónicos que los utilizan mejoren su desempeño (velocidad, consumo de potencia, etc...). Esto se debe a que están diseñados para cumplir una única funcionalidad. A diferencia de aquellos de propósito general que cuentan con múltiples módulos internos que realizan tareas varias, que si bien lo hacen más versátil, hacen que este no dedique el 100 % de sus recursos a una tarea.

Es por esto que el crear un flujo de diseño funcional, con herramientas de estado del arte, permite impulsar el desarrollo de la industria de semiconductores en Guatemala. Por medio de la creación de plazas laborales que permitan el diseño y manufactura de estos dispositivos en el país así como la comercialización de estos productos.

Los beneficios inmediatos de este proyecto son el facilitar la utilización del flujo de diseño a quienes proceden esta investigación. Para que se pueda invertir tiempo en únicamente a mantener actualizado el flujo de diseño y permitir que aquellos que se quieran beneficiar de este puedan crear proyectos de alto nivel en los departamentos de Electrónica, Macarrónica y Biomédica de la UVG. Además, a largo plazo, que otras carreras puedan colaborar de forma interdisciplinaria para la creación de proyectos más ambiciosos y con un mayor alcance para beneficio de la comunidad UVG y del país.

Objetivos

Objetivo General

Mejorar el proceso actual de síntesis física como parte del flujo de diseño para el desarrollo de un circuito integrado a nanoescala, trabajando en conjunto con los demás grupos, para mejorar la interconexión con los demás módulos, adicionalmente someter el *layout* a las diferentes verificaciones y trabajar en conjunto con los integrantes de mi grupo para solventar violaciones a las normas de diseño de forma eficiente para así obtener un *layout* funcional sin errores de DRC.

Objetivos Específicos

- Migrar la síntesis física de la herramienta IC Compiler I a IC Compiler II.
- Verificar la compatibilidad de las librerías de **TSMC** utilizadas en el desarrollo de la síntesis física.
- Posicionar de forma estratégica los diferentes componentes del circuito integrado para aprovechar el área de $2mm^2$ que nos brinda **TSMC**.
- Posicionar de forma estratégica los diferentes componentes del circuito integrado para que el proceso de ruteo sea eficiente y minimice los problemas en las etapas posteriores del flujo de diseño.
- Validar los resultados de la síntesis física por medio de la verificación DRC, comprendiendo las diferentes violaciones a las normas de diseño y corrigiéndolas hasta que el *layout* este libre de errores.
- Mantener la comunicación con mi grupo de trabajo y los demás para solucionar los errores obtenidos de forma rápida y consolidar el flujo de diseño.
- Obtener y delegar los archivos y *scripts* resultantes del proceso de síntesis física y DRC al grupo encargado de la automatización del flujo de diseño.

Marco teórico

VLSI

Very Large Scale Integration o **VLSI** es el proceso completo por el cual se fabrica un circuito integrado (**CI**), los cuales están compuestos, actualmente, de millones de transistores a nanoescala. El proceso de diseño y simulación se le conoce como Flujo de Diseño. [3] Dicho proceso sigue una estructura top-down la cual implica que el proceso se lleva a cabo desde la jerarquía más alta (macro) hasta el proceso más específico (micro).

Actualmente las computadoras, teléfonos y demás dispositivos electrónicos con capacidad de computo cuentan con un **CI** denominado procesador. El procesador de estos dispositivos

es el encargado de realizar el cómputo de la información y distribuirla a lo largo de las diferentes piezas de hardware que integran a estos dispositivos. Las empresas líder en el diseño **VLSI** para procesadores de dispositivos electrónicos actualmente son Intel, AMD, Samsung y recientemente se incorporó a este campo, de diseño a gran escala, Apple.

Apple actualmente cuenta con su primera generación de procesadores diseñados por ellos especializados para mejorar el actual rendimiento de sus computadoras, tablets y laptops. Este procesador se conoce como *M1* y cuenta con 16 miles de millones de transistores con una tecnología de 5nm[7]. A continuación una imagen:

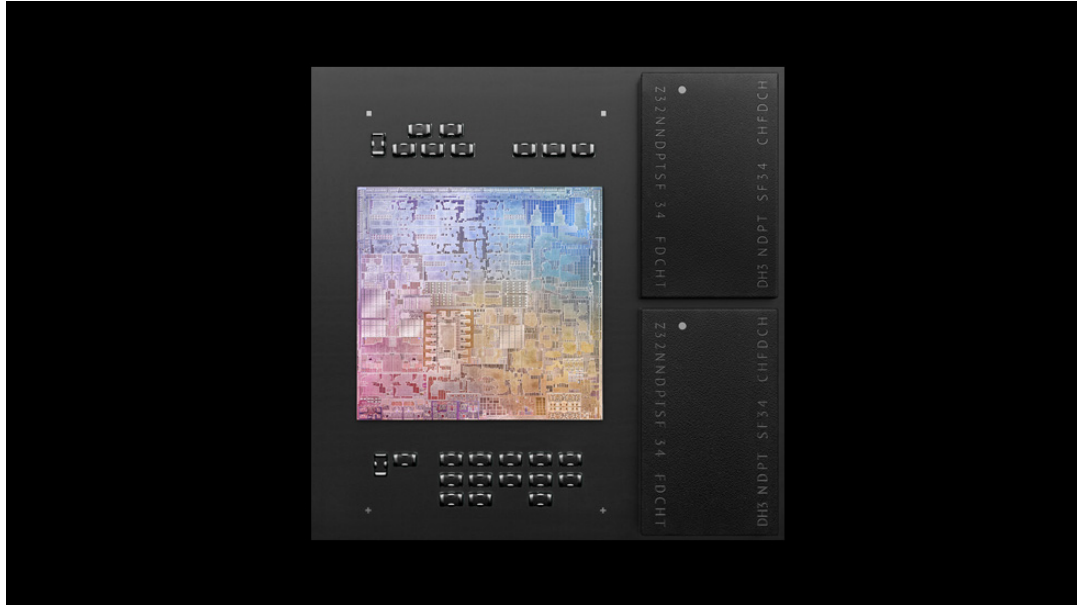


Figura 1: Primer procesador dedicado para computadoras Apple, M1 con 16 miles de millones de transistores con una tecnología de 5nm. [7]

Flujo de Diseño [6]

Es el proceso por el cual se obtiene el diseño de un **CI** fabricable y funcional. Este proceso puede dividirse en múltiples niveles de abstracción que conforman las diferentes capas del diseño de un **CI**. Las seis capas en las que se divide este proceso son: sistema, algoritmo, arquitectura, lógica, física y *signoff*. A continuación se describe a mayor detalle cada uno de ellas. Cabe resaltar que la necesidad de dividir todo el proceso de fabricación en capas surge ya que la complejidad de diseño es tal que requiere de especialización en diferentes disciplinas. El segmentar en capas permite que cada especialista trabaje en un nivel de abstracción específico aportado al flujo de diseño flexibilidad, cooperación y calidad.

Ingenieros de diseño de Intel Corp pertenecientes al equipo de NIKE Design Technology mencionan que en sus procesos de trabajo para el diseño **VLSI** es necesario particionar las diferentes tareas a realizar en el flujo de diseño. Cada fase a cargo de ingenieros especializados. Sin embargo es importante recalcar que puede existir una partición innecesaria

(sobresegmentación del flujo) de tareas que puede resultar en optimización local de cada uno de los procesos. Esto en mayor esquema puede presentar problemas en la optimización general del **CI** ya que en la marcha surgen cambios que afectan diferentes segmentos del flujo de diseño y puede tomar días arreglar estos problemas. [8] Es por esto que es importante la segmentación apropiada del flujo del diseño y que sean equipos de ingenieros especializados que permiten agilizar este proceso. A continuación se sugiere un particionamiento apropiado para el equipo de trabajo con el que se cuenta.

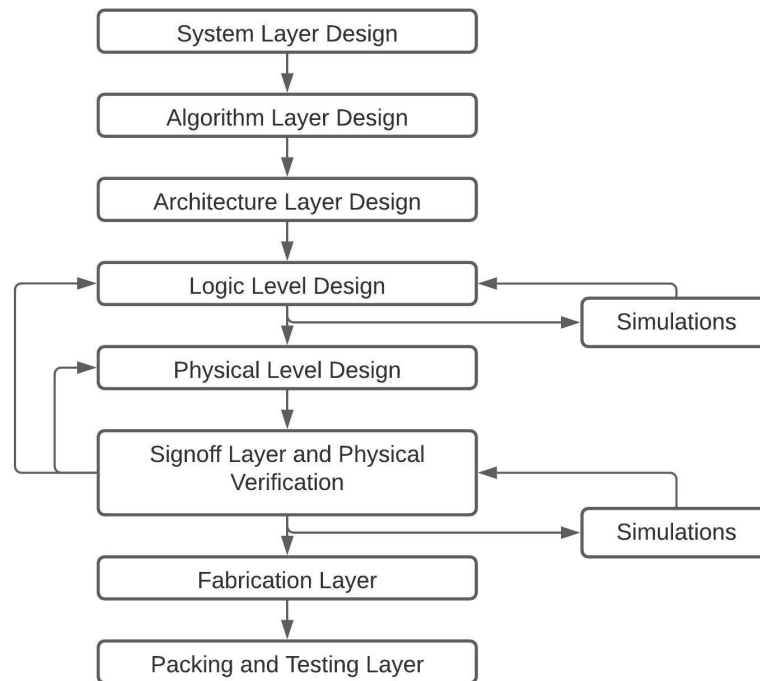


Figura 2: Flujo de diseño simplificado para el diseño de un **CI** digital a nanoescala

Diseño en la capa Sistema

Esta es la jerarquía más alta de diseño, el objetivo que se desea conseguir en esta etapa es diseñar un sistema que cumpla con las necesidades básicas de su aplicación. Las principales características en las que se especializa esta capa son: funcionalidad, desempeño, condiciones de trabajo, dimensiones físicas, empaquetado, pinout, tecnología de fabricación, costos, la potencia consumida y la necesidad de comunicación interna y externa.

Diseño en la capa Algoritmo

Esta capa busca definir una estructura en software que permita una implementación a hardware funcional y sencilla. En esta etapa es importante considerar la capacidad de computo del **CI**, la cantidad de memoria que este requiere, se define la arquitectura del

conjunto de instrucciones (**ISA**) por sus siglas en inglés [9]. En este punto debe evaluarse que tan complejo, que tantos recursos computacionales y que tanta exactitud se requiere según los objetivos establecidos en la capa superior como los protocolos de comunicación y las tareas que debe realizar.

Diseño en la capa Arquitectura

Así como la capa superior busca distribuir los recursos computacionales de forma eficiente, esta capa busca distribuir de forma eficiente los recursos físicos (hardware) para cada una de las tareas que se van a realizar. Aquí se debe de prestar atención a los objetivos del diseño en la capa de sistema como: desempeño, costo, potencia, dimensiones, etc... En este punto la operación de la máquina (**MO**), por sus siglas en inglés, toma lugar y se refiere a como el hardware implementa el **ISA** definido en la capa superior [9]. La definición de *datapaths*, tamaño de los buses, memorias de acceso aleatorio (**RAM**) y su relación con los demás componentes son las tareas de prioridad.

Diseño en la capa Lógica

En esta capa se busca generar una descripción lógica del circuito y hacer una síntesis de la misma para generar lo que se denomina *netlist*. Ambas tareas críticas para el diseño del **CI**

Ya con una arquitectura, que define todo el proceso de interacción del hardware, se requiere definir la conectividad de cada módulo. En el nivel de abstracción más alto de esta capa se traduce en *black boxes* las cuales contienen entradas, salidas y una sincronización temporal definida (reloj). La cual se puede describir utilizando un lenguaje descriptor de hardware o **HDL** por sus siglas en inglés. Los lenguajes más comunes para poder describir el hardware son: Verilog y VHDL. Estos módulos deben ser simulados y verificados para asegurar su funcionalidad.

El siguiente paso es la traducción de estos bloques de **HDL** a elementos de bajo nivel de un circuito y como estos están interconectados entre sí. A el resultado de esta traducción se le conoce como *netlist*. Herramientas de síntesis lógica permiten automatizar estos procesos, lo que facilita esta tarea de traducir de **HDL** a un *netlist*.

Los *netlists* utilizan algo llamado *cell libraries* que contienen los componentes más elementales de un circuito. Los transistores son definidos y conforman subcircuitos que representan compuertas lógicas las cuales en conjunto con los elementos básicos como capacitores, resistencias, entre otros... son descritos tanto sus propiedades como su interconexión. Esto se puede simular y validar utilizando una herramienta de simulación *hspice*.

Diseño en la capa Física

A partir de este punto cada subcircuito descrito en el *netlist* es colocado en un espacio físico en una oblea de material semiconductor (Silicio) la cual está conformada de las

diferentes difusiones, metales, pines y sus interconexiones. Debido a que el fabricante tiene requerimientos de diseño, que se deben cumplir para garantizar la funcionalidad del diseño, es imperativo que se tomen en cuenta y se verifiquen de forma rigurosa. Ya que este proceso toma en cuenta factores como: rendimiento, área, confiabilidad, potencia, entre otros... el proceso se divide en diferentes secciones: *Partitioning*, *FloorPlanning*, *Placement*, *Power and Ground Routing*, *Signal Routing* y *Closure* de las cuales se discutirán a mayor detalle. Al final de todo este proceso un *netlist* asociado es generado el cual contiene todos los cambios y restricciones colocadas en los diferentes pasos de esta capa de síntesis física.

Diseño en la capa *Signoff* y Verificación Física

Una vez se culmina el diseño este debe de ser verificado de forma rigurosa para asegurar que este cumpla con los estándares de fabricación, que concuerde el diseño en silicio con el de el circuito propuesto, etc... Cualquier problema debe ser solucionado en la capa anterior de diseño. Las verificaciones que se realizan se mencionan a continuación:

Design Rule Checking (DRC)

Verifica que el *layout* cumpla con todos los requerimientos de fabricación los cuales están directamente relacionadas a la tecnología que se está trabajando. El realizar un proceso de manufactura a escala nanométrica es complejo debido a que se ve limitado por la tecnología de las herramientas de fabricación con las que cuenta la fábrica. Limitaciones como las distancias entre los metales, difusiones, policilicio, o complicaciones como la pérdida de resolución, las esquinas redondeadas, las conexiones reducidas durante el proceso de manufactura, etc... deben mitigarse y esto se hace cumpliendo con los requerimientos de los fabricantes ya que ellos toman en cuenta estos problemas a la hora de fabricarlo. [10]

Layout vs. Schematic (LVS)

Verifica la funcionalidad del diseño. El *netlist* generado por la capa superior, síntesis física, y el *netlist* generado por la capa de síntesis lógica son comparados para verificar si ambos concuerdan en términos de funcionalidad.

Antenna Rule Checking

Esta fase sirve para prevenir los efectos de antena en el circuito. Los conductores al acumular cargas pueden presentar efectos de antena. Estos pueden acumularse debido a las descargas del rayo laser al momento del grabado, generando descargas no deseadas en las compuertas del transistor. Esto representa un problema ya que puede causar daños a los transistores, en especial a la compuerta del gate, o reducir su tiempo de vida.

Electrical Rule Checking (ERC)

Verifica la calidad de la conexión de tierra y alimentación así como los tiempos de transmisión de señales. Adicionalmente verifica cargas capacitivas y la correcta conexión de los *fanouts*. Una compuerta lógica puede excitar un número finito de entradas a otras compuertas en un mismo nodo (*fanout*). Esta excitación se ve limitada por la corriente que provee el circuito. Por lo tanto debe tomarse en cuenta para la frecuencia de operación y la potencia entregada máxima de una compuerta. Factores que afectan tanto el rendimiento como el tiempo de vida del *CI*.

Parasitic Extraction (LPE)

Este deriva de los componentes geométricos, a nivel nanométrico, sus parámetros eléctricos (Capacitancias y Resistencias) para verificar las características eléctricas del circuito. A esta verificación se ingresa un *netlist* y con la ayuda de la herramienta de Synopsys *starRC* se genera un *hspice* con todas estas capacitancias y resistencias parásitas presentes en el *layout*.

Final Simulation

Luego de haber pasado todas las simulaciones en **CLEAN** o **PASS**, utilizando el *hspice* generado en la verificación de **LPE**, se hace una simulación que sirve para validar que los resultados obtenidos por el circuito descrito por el *layout* que se diseñó cumplan con las mismas funcionalidades que el circuito descrito en la capa de síntesis lógica. Si esto es congruente, entonces se puede concluir la etapa de verificaciones.

Diseño en la capa de Fabricación

Al finalizar todas las simulaciones, se genera un archivo de extensión **.GDS2** el cual contiene toda la información que el fabricante necesita para empezar el proceso de manufactura. Este archivo es enviado a una fábrica que se dedica a manufacturar chips en silicio. En el caso del proyecto, el fabricante es **TSMC**. El proceso en donde se manda el archivo a la fábrica se le conoce como *tapeout*. Aquí se generan las máscaras y por el proceso de fotolitografía, patrones expuestos a los rayos láser se graban en el silicio, así como los diferentes dopajes. El diseño se hace sobre obleas de silicio redondas las cuales varían en tamaño desde los 200mm hasta los 300mm. Por último se clasifican según su estado: funcional o defectuoso, establecido por las pruebas de potencia y velocidad que se establecieron.[10]

Diseño en la capa de Empaquetado y Pruebas

Aquí se cortan los **CI** individualmente de la oblea de silicio donde se fabricaron. A cada **CI** se le conoce como *die* y este es empaquetado según la necesidad del cliente. Los empaquetados pueden ser: **DIP, PGS, BGA, etc...** Se conectan los pines dentro del empaquetado

con el *die* ya sea con *bonding wires* o *solder bumps* y se sella. Como punto final del flujo de diseño, este se prueba en un laboratorio por el cliente para ver que este funcionando de la forma deseada y con los requerimientos propuestos. Concluyendo el flujo de diseño del **CI**

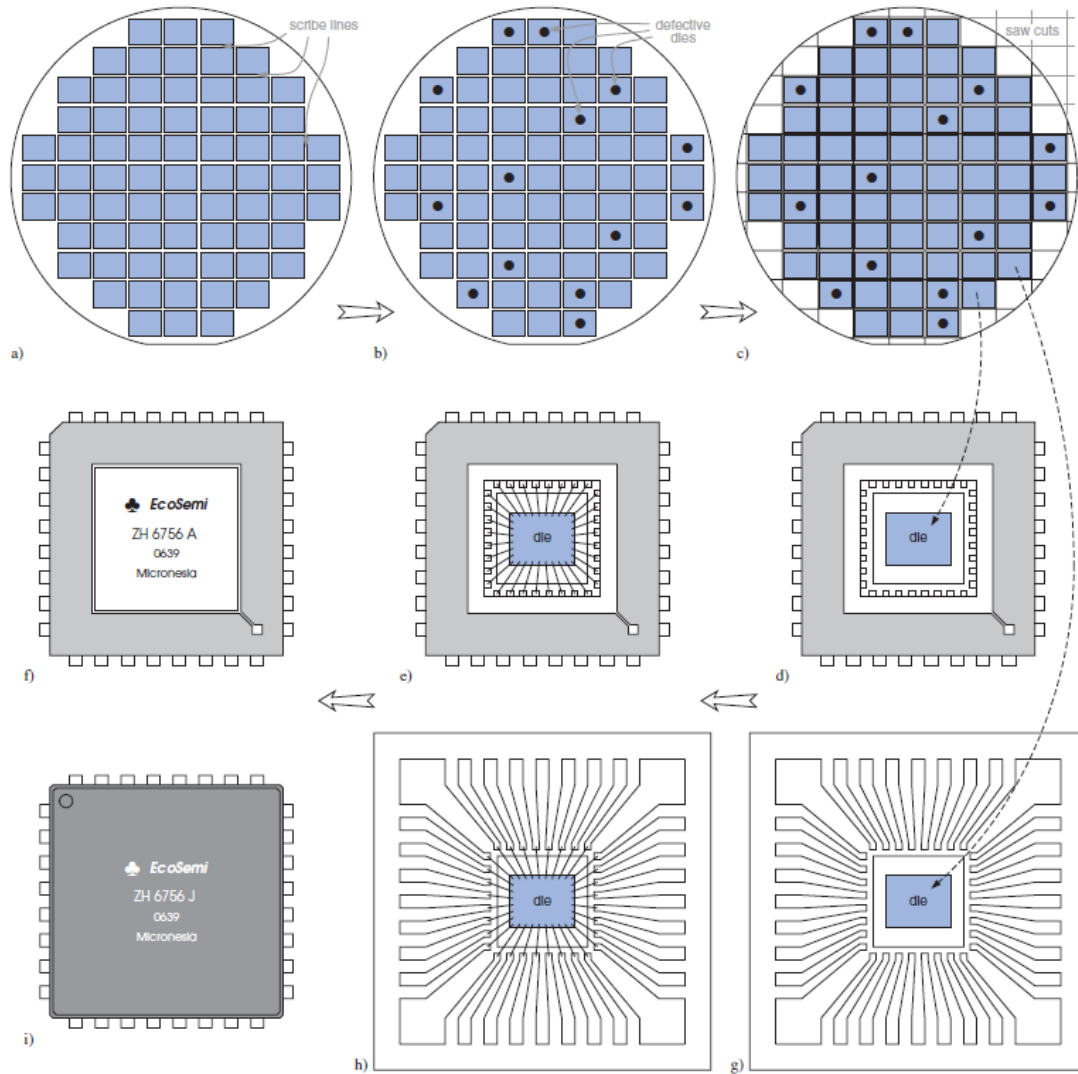


Fig. 11.11 Testing and encapsulation steps (simplified, not drawn to scale). Processed wafer (a), probed wafer with defective circuits inked (b), wafer after sawing (c), a good die attached to package cavity (d), wires bonded to lead frame (e), final IC after sealing, testing, and stamping (f). (g,h,i) correspond to (d,e,f) for a plastic package.

Figura 3: Segmentación de la oblea de silicio resultante en **die**, proceso de empaquetado, colocación de *bonding wires* y **CI** final [11]

Síntesis Física, *Design Rule Check (DRC)*, *Electrical Rule Check (ERC)* y *Antena Rule Check*

Esta es la parte del flujo de diseño que se realizará este trabajo de investigación. Por lo tanto se desarrolla a más detalle la información de los procesos que se realizarán.

Síntesis Física

La síntesis física es una de las partes fundamentales para el desarrollo de este proyecto, ya que aquí se transformará la síntesis lógica a un conjunto de componentes físicos posicionados en silicio y unidos a través de *interconnects*. Esta etapa de diseño está segmentada en diferentes secciones las cuales se describen a continuación:

Partitioning

En esta etapa se secciona el circuito en subcircuitos con el fin de que estos puedan luego ser posicionados y colocados de forma estratégica.

Floor Planning

En esta etapa se hace una relación espacial entre los pines de entradas y salidas del **CI** y los subcircuitos. Esto se hace con la finalidad de establecer prioridades a las secciones del circuito original que deben estar más cerca de estos puertos ya que se desea minimizar retardos y material utilizado.

Placement

En esta etapa se plantean secciones del espacio que se va a utilizar para realizar el **CI**. En el caso de este proyecto se cuentan con $2mm^2$ de espacio para la fabricación del circuito deseado. Las partes seccionadas en la primera etapa se les asigna de forma estratégica una posición física en el silicio con el fin de obtener los mejores resultados en cuanto a la minimización de retardos y distancia entre las entradas y salidas de estos subcircuitos con los demás.

Power and Ground Routing

En esta etapa se hace un ruteo de los planos de alimentación y tierra. Esto se hace de primero ya que todo circuito lógico compuesto de **MOSFETs** tendrá la necesidad de estos, lo que hace crítico que estas señales se hagan primero.

Signal Routing

Se generan *interconnects* entre todos los circuitos lógicos. El software utilizado permite optimizar estas trayectorias a manera de que utilicen menor material. Ya que entre más distancias y más material la cantidad de parásitos incrementará lo que impacta directamente con el rendimiento del **CI**.

Closure

Este es el paso antes de pasar a las verificaciones físicas, en esta fase se realizan las últimas revisiones del *layout*. Se puede realizar un **DRC** básico que proporciona la herramienta de IC Compiler II.

Proceso de Síntesis Física Anterior [12]

A continuación se muestra el proceso de la síntesis física realizada por el estudiante Luis Abadilla en el 2020. Los comandos que utilizó se describen y están colocados en el orden en el que fueron utilizados. Toda la síntesis física se realiza en la herramienta de IC Compiler I, y cuenta con cuatro partes fundamentales.

- **Importación de librerías y creación de *MilkyWay Library*:** En esta etapa se importan las librerías de **TSMC** a utilizar, se crean los archivos donde se documenta el historial de todas las acciones que se realizan en la síntesis física. Además se importan los archivos que provienen de la síntesis lógica. Adicionalmente se crea un directorio de trabajo y se llaman a las librerías *link* y *target* las cuales definen el caso típico.
- **Floor Plan**
 - **Conexiones Lógicas de alimentación y tierra:** Esta sección se dedica a crear las variables y las celdas que le corresponden a los puertos de alimentación.
 - **Creación de Pads y Corners:** Aquí se plantean las esquinas y los *pads*, llamados celdas, los cuales definen y acotan el contorno del **CI**
 - **Configuración de las restricciones:** Una de las partes fundamentales del Floor Plan, ya que aquí se restringe todas las posibles orientaciones, posiciones, conexiones o arreglos que puedan causar conflictos en el ruteo. Indispensable para realizar un *layout* de calidad.
 - **Creación del Floor Plan:** se ejecuta y verifica el Floor Plan propuesto por todas las restricciones y configuraciones previas.
- **Placement:** Aquí se colocan los componentes según las reglas de Floor Plan, además se pueden configurar arreglos que optimicen: *delays*, *fanouts*, distancias y *X-Talk*. También es verificable y reconfigurable.
- **Routing:** Por último se realizan todas las interconexiones, empezando por los puertos de alimentación y luego por las demás interconexiones. Aquí se realiza nuevamente una verificación de la efectividad del proceso y se concluye la síntesis física.

<i>Comando</i>	Funcionalidad
Comandos para la importación de librerías y creación de la <i>MilkyWay</i> y directorios necesarios para el proyecto	
<code>./icc_shell-gui - shared_license</code>	Iniciar el programa y su interfaz gráfica en consola
<code>lappend search_path /home / administrador / Escritorio /FA_ TSMC_ DRC / Libs / tcb018gbwp7t_290a _FE / tcb018gbwp7t /LM</code>	Comando para definir el directorio en donde se encuentran las librerías que se van a utilizar
<code>set link_library "*" tcb018gbwp7ttc.db tcb018gbwp7twc .db tcb018gbwp7tbc.db tpd018nvtc.db "</code>	Hacer mención directa de las librerías que se van a utilizar de todo el catálogo de TSMC (link)
<code>set target_library "tcb018gbwp7ttc.db "</code>	Hacer mención directa de la librería (target) con la que se estará trabajando: caso típico
<code>set tlu_plus_files -max_thuplus /home / administrador / Escritorio / FA_ TSMC_ DRC / Libs / tcb018gbwp7t_290a_FE / thuplus / t018lo_1p6m_typical.thuplus - min_thuplus / home / administrador / Escritorio / FA_ TSMC_ DRC / Libs / tcb018gbwp7t_290a_FE / thuplus / t018lo_1p6m_typical.thuplus - tech2itf_map / home / administrador / Escritorio / FA_ TSMC_ DRC / Libs / tcb018gbwp7t_290a_FE / thuplus / star.map_6M</code>	Comando para agregar la TUPLUS y el archivo .MAP específico dentro de las librerías de TSMC
<code>create_mw_lib -technology /home / administrador / Escritorio / FA_ TSMC_ DRC / Libs / tcb018gbwp7t_290a_FE /tf/ tsmc018_6lm.tf - mw_reference_library {/home / administrador / Escritorio / FA_ TSMC_ DRC / Libs / tcb018gbwp7t_290a_FE / tcb018gbwp7t / home / administrador / Escritorio / FA_ TSMC_ DRC / Libs / iolib / tpd018nw } - bus_naming_style {{%d}} -open /home / administrador / Escritorio / LUIS_ABADIA / Milky_Aba</code>	Crear un directorio y una librería <i>MilkyWay</i> para la documentación de todos los cambios que se harán dentro de la síntesis física.
Hasta este punto se importan los archivos de extensión verilog (.v) y extensión (.sdc) resultantes de la síntesis física.	
Floor Plan Parte 1: Conexiones lógicas de VDD y VSS	
<code>set_app_var mw_logic1_net "VDD"</code>	Definición de una variable para la net de poder del voltaje

<i>set_app_var mw_logic0_net "VSS"</i>	Definición de una variable para la net de poder de tierra
<i>derive_pg_connections -power_net VDD -power_pin VSS -ground_net VSS</i>	Configuración de los pines de poder y las nets de poder según las variables que se definieron en los comandos anteriores.
<i>derive_pg_connections -power_net VDD -ground_net VSS -tie</i>	
<i>report_cell_physical -connections</i>	Verificación de que los comandos anteriores se ejecutaron con éxito. Si el resultado es 1 es exitoso, si es 0 es fallido.
Floor Plan Parte 2: Creación de celdas (Pads y Corners)	
<i>create_cell {C1 C2 C3 C4} PCorner</i>	Creación de las 4 esquinas del chip.
<i>create_cell { VDD} PVDD1CDG</i>	Creación de la conexión de VDD (Pad)
<i>create_cell { VDD} PVSS1CDG</i>	Creación de la conexión de VSS (Pad)
Floor Plan Parte 3: Configuración de las restricciones	
<i>set_pad_physical_constraints -pad_name c1side 1</i>	Configuración del primer pad de esquina en el lado 1
<i>set_pad_physical_constraints -pad_name c2side 2</i>	Configuración del segundo pad de esquina en el lado 2
<i>set_pad_physical_constraints -pad_name c3side 3</i>	Configuración del tercer pad de esquina en el lado 3
<i>set_pad_physical_constraints -pad_name c4side 4</i>	Configuración del cuarto pad de esquina en el lado 4
<i>set_pad_physical_constraints -pad_name "VDDside 3 -order 2</i>	Configuración del pad de VDD en el lado 3 en segunda posición
<i>set_pad_physical_constraints -pad_name "VSSside 4 -order 2</i>	Configuración del pad de VSS en el lado 4 en segunda posición
Floor Plan Parte 4: Creación del Floor Plan	
<i>create_floorplan - control_type width_and_height - core_utilization 0.7 - core_width 5 - core_height 50 - no_double_back - top_io2core 10 - bottom_io2core 10 - left_io2core 5 - right_io2core 5</i>	Creación de la segmentación del espacio a utilizar por los componentes (Floor Plan)
<i>check_physical_design</i>	Verificación del proceso de diseño hasta este punto. Si el resultado es 1 es correcto, si es 0 es incorrecto y se debe corregir.
<i>adjust_fp_floorplan</i>	Este comando termina de hacer los ajustes necesarios para asegurar que este esté bien acotado.
Configuración y creación del placemnet	
<i>set_fp_placement_strategy - adjust_shapes on</i>	Ajuste de las geometrías dentro del Floor Plan

<i>create_fp_placement -effort High -max_fanout 800 - optimize_pins - congestion_driven - timing_driven - consider_scan</i>	Comando para realizar el posicionamiento , con mejor esfuerzo, un gran <i>fanout</i> , optimizando la congestión, el <i>delay</i> y la posición de los pines de entradas y salidas con respecto a los <i>pads</i> .
<i>check_physical_design - post_initial_placement</i>	Este comando permite realizar una verificación del placement si el resultado es 0 esta mal y si es 1 esta correcto.
<i>place_opt -effort high</i>	Optimizar el placement al final para tener mejores resultados maximizando el esfuerzo para que se cumplan las condiciones previamente establecidas.
Creación del Routing	
<i>create_rectangular_rings -nets {VDD VSS} -around core</i>	Creación del anillo de alimentación al rededor de todos los componentes para hacer el routing más eficiente.
<i>create_power_straps -direction vertical -start_at 155 - num_placement_strap 4 - increment_x_or_y 50 -nets {VDD} -layer METAL2 -width 2 - do_not_route_over_macros</i>	Este comando crea también un fácil acceso a los planos de alimentación y tierra. Los <i>power straps</i> dependen de la cantidad de componentes y saturación del espacio.
<i>set_route_mode_options -zroute false</i>	Para cambiar el comando clásico se utiliza este comando.
<i>check_routeability</i>	Este preceso sirve para hacer un análisis preliminar para saber si la herramienta es capaz de rutear el circuito propuesto. De ser cierto el resultado es 1, si es 0 se debe cambiar o la configuración o el placement.
<i>set_route_opt_strategy</i>	Este comando permite analizar si el circuito tiene problemas de delay, de X-talk o errores generales de diseño.
<i>set_preroute_drc_strategy</i>	En este punto se hace un análisis de DRC muy básico, esto es solamente como una pre revisión, antes del análisis DRC por IC Validator.
<i>preroute_standard_cells -mode net -connect both -nets { VDD}</i>	Comando para conectar networks con el anillo de VDD.
<i>190 preroute_standard_cells -mode net -connect both -nets { VSS}</i>	Comando para conectar networks con el anillo de VSS.
<i>preroute_instances</i>	Coando para conectar todas las celdas con los anillos.
<i>route_opt</i>	Conexión entre celdas y fuentes de alimentación.

<i>verify_route</i>	La última verificación de routing se hace aquí, si el valor es 0 debe de verificarse nuevamente y probar con nuevas estrategias de ruteo, si es 1 todo está en orden y se puede proceder a DRC.
Finaliza la Síntesis Física.	

Cuadro 1: Proceso de síntesis física realizado en el 2020, comandos y su descripción.

DRC

Design Rule Check, esta etapa es crítica y juega un papel fundamental en conjunto con la síntesis física. Aquí se comparan los resultados obtenidos de la síntesis física con un *runset*, que en nuestro caso es proporcionado por **TSMC**, el cual contiene todas las restricciones de fabricación. Aquí se hace una exhaustiva revisión de las normas como el espaciado de metales, las interconexiones, el tamaño de los *pads*, etc... se cumplan y concuerden con todas las normas de **TSMC**. Esta etapa no trata acerca de la funcionalidad ni la coherencia entre el diseño lógico y físico que es tarea del **LVS**. Cualquier problema resultante de esta verificación debe ser gestionado por la capa de diseño superior (síntesis física) y de forma iterativa solucionar cualquier problema que se presente hasta obtener un resultado **CLEAN**. [6]

Para esta etapa se pueden presentar una gran cantidad de errores y no existe un orden específico, ni un formato definido para presentarlos. Es por esto que se debe de contar con la documentación del software que se estará utilizando para la realización de este proceso de validación. En el caso de Synopsys, la herramienta que ofrecen es IC Validator. Debido a que existen demasiadas reglas de diseño a continuación se mencionan algunas reglas de diseño comunes. [10]

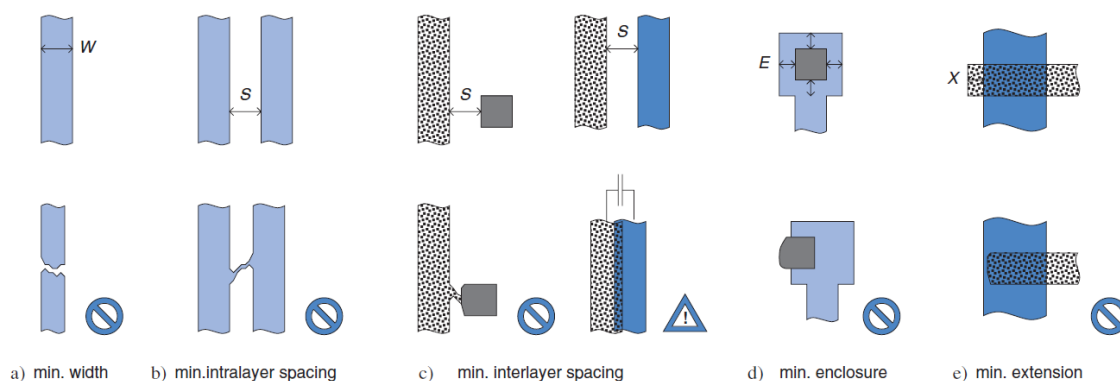


Fig. 11.1 Minimum size rules (top row) and the likely consequences of violating them (bottom row).

Figura 4: Ilustración de las reglas de diseño de condiciones mínimas y como estas se infringen [11]

Minimum Width

Se establece un grosor mínimo de las estructuras para evitar que estas colapsen o se abra el circuito por falta de material. Particularmente las líneas largas y delgadas, sin elementos en sus vecindades, son más susceptibles a sobre-estirarse a diferencia que un conjunto de líneas del mismo grosor. Es por esto que se necesita un mayor grosor.[11]

Minimum Intralayer Spacing

Esta regla limita la distancia entre las diferentes estructuras. Esto se hace para evitar cortos circuitos entre los elementos adyacentes, También puede depender si estas estructuras tienen una conexión eléctrica existente. Por ejemplo, en un proceso de 130nm exige una separación de *Well-to-Well* de 630nm si estos manejan el mismo potencial. De no ser así este es de 1000nm. [11]

Minimum Interlayer Spacing

Esta regla permite verificar la distancia entre un plano y otro, para evitar acoplamientos e interacciones indeseadas entre sí. Problemas como la inflación de capacitancias por contactos entre polisilicio y difusiones o corrientes de fuga por acoplamientos ente uniones PN y una *gate* de policilicio. [11]

Minimum Enclosure

Estas reglas están directamente relacionadas con las estructuras del *layout* en diferentes planos. Estas reglas son uniformes a lo largo de todos los bordes. Por ejemplo para las vias que el contacto sea el adecuado con las superficies que conecta ya que el proceso de manufactura tiene cierta tolerancia en las máscaras y su alineación además de las imperfecciones del proceso de *etching*. Otro ejemplo es que el metal de más alto nivel este debajo de *overglass passivation layer* a lo largo de todas las aperturas para los *pads* así el sellado es hermético. [11]

Minimum Extension

Debido a que existen múltiples capas es importante mantener reglas que delimiten como se sobreponen las estructuras. La forma en la que esto se verifica es por medio de la orientación en la que se sobreponen. El *self-aligned perocess* permite que el policilicio actúe como una máscara para el proceso de implantación de iones [10]. Por lo que si estas capas de policilicio no están correctamente alineadas es posible que ya sea el *drain* o el *source* no se dopen de forma correcta. Esto afectaría considerablemente el desempeño y la funcionalidad del **CI**. Estas reglas ya toman en cuenta las imperfecciones del proceso de *etching* y problemas en la alineación de las máscaras. [11]

Maximum Width

Las vías y en general los contactos se fabrican por medio del proceso de *etching* en la capa de dieléctrico para luego depositar rellenos de tungsteno para llenar las cavidades, antes de depositar la siguiente capa metálica. Para asegurar una calidad uniforme y una superficie plana es necesario delimitar el grosor máximo de estas cavidades. Lo más común es delimitar un único tamaño de vía o contacto. Si en algún caso se desea agregar un contacto

más grande, estas deben segmentares en subgrupos de contactos. A esta técnica se le conoce como *stipple contact/via*. [11]

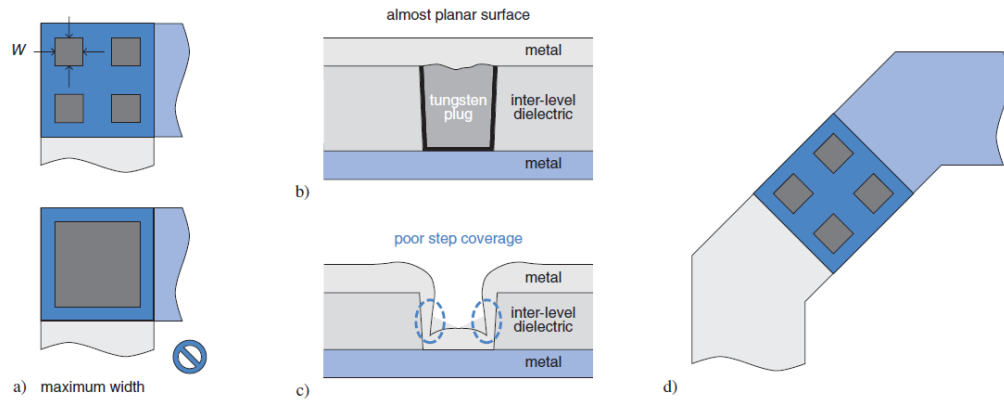


Fig. 11.2 Maximum size rule for contacts and vias. Stipple contact/via versus an oversize single contact/via (a). Cross section of a plugged via (b) and of a historical sink-in via (c). Electromigration-aware stipple contact/via (d), to be explained in section 11.6.1.

Figura 5: Ilustración de la regla de *Maximum Width*, como esta afecta el diseño y la técnica de *stipple contact/via* [11]

Density Rules

La densidad planar de una capa de un layout es definida como el área ocupada por todos los elementos en esa capa, dividido el área de toda la capa. Estas normas permiten limitar la ocupación del plano con cota inferior y superior. Por ejemplo se puede definir que la ocupación esté en promedio entre el 20 % y el 80 % por cada milímetro por milímetro de región. Algunas prácticas como llenar de *dummys*, sin conectar, una región en donde la ocupación no se llegue a cumplir resultan útiles en el proceso de diseño. [11]

Extension Rules

Si bien estas reglas están mejor definidas en el proceso de *Antenna Rule Checking*, el proceso de **DRC** hace una verificación únicamente de la extensión del material conductor. Estas reglas se detallaran con mayor detenimiento en la verificación de reglas de antena. [11]

Verificación de DRC Anterior

En la iteración previa a este proyecto, la cual fue dirigida y ejecutada por Matthias Sibrian y precedida por Luis Nájera el año anterior, cuenta con una metodología específica para realizar el proceso de verificación DRC. El proceso de ejecución de esta verificación esta

segmentado en cuatro pilares fundamentales: Instalar y abrir Custom Compiler, Cargar la celda sintetizada, abrir la vista de *layout* y correr la verificación DRC.

La parte fundamental de este proceso, para el nivel al que se trabajará este proyecto, se centra principalmente en la ejecución de la verificación de las reglas de diseño, comprensión, corrección de errores y revalidación. Es un proceso iterativo que requiere consultar la documentación de TSMC y sus restricciones de diseño. Esta directamente enlazado con el proceso de síntesis física ya que en este se corrigen los errores que esta verificación presenta. A continuación se muestra un diagrama que muestra el proceso de corrección de errores de DRC.



Figura 6: Imagen del proceso de verificación DRC de años anteriores [13]

Debido a problemas con la compatibilidad de IC Validator con IC Compiler I los grupos anteriores se vieron obligados a utilizar Custom Compiler como la herramienta de verificación y su integración con IC Validator. Es por esto que el proceso mostrado debe ser abordado de forma manual, utilizando una interfaz gráfica. Uno de los desafíos que presenta el actual proyecto es utilizar IC Compiler II para poder utilizar IC Validator en consola y abrir paso a una verdadera automatización del flujo de diseño.

ERC [5]

Electrical Rule Check es una etapa de la capa de *singoff* la cual permite verificar que el rendimiento eléctrico del circuito cumpla con las especificaciones del fabricante. Las reglas que verifica esta etapa son las siguientes:

Soft Connect Check

Esta regla establece si la conectividad (unidireccional) de las difusiones que se encuentran en el *N-Well* y la capa superior es la especificada.

Path Check

Esta regla valida lo siguiente: Nodos con un *path* conectado a alimentación pero no tierra, Nodos con un *path* a tierra pero no a alimentación, Nodos sin *path* a tierra o alimentación o Nodos sin un *path* a cualquier otra *net*.

PTAP/NTAP Connectivity Check

Verifica si el *PTAP* se conecta a alimentación y el *NTAP* a tierra.

MOSFET Poer and Ground Check

Verifica que para cualquier **MOSFET**, ya sea tipo N o P, *source MOSFET* y *drain MOSFET* estén conectados correspondientemente a alimentación y a tierra.

Gate Directly Connected to Power or Ground

Esta regla sirve para verificar que exista protección contra descargas electro-estáticas. Si el *gate* de un **MOSFET** tipo P se conecta a alimentación de forma directa podría dañarse el *gate*, así como un **MOSFET** tipo N conectado a tierra directamente.

Floating Gate

Verifica si los contactos están interfiriendo con el *poly gate*

Floating Well

Verifica si existe alguna conexión entre el *N-Well* o *P-Subrate* y alimentación o tierra.

Antenna Rule Checking [5]

Esta verificación busca evitar que el **CI** tenga efectos de antena que pueden dañar las terminales de los transistores (*gate, drain, source o bulk*) durante el proceso de fabricación. Esto puede ocurrir en el proceso de grabado de los caminos, por medio de rayos plasma o en la implantación de iones. Estos rayos pueden hacer que se acumulen cargas no deseadas en

los cables metálicos, los cuales pueden estar conectados a nodos de uniones PN. Algunos de los efectos presentes pueden ser: daño de oxido inducido por plasma (efecto antena), también conocido como *Fowler-Nordheim tunneling*, e inducción de corriente por altas temperaturas.[11] Efectos que pueden quemar o reducir el tiempo de vida de un transistor afectando el rendimiento y tiempo de vida del **IC** completo.

Esta verificación al igual que el **DRC** cuenta con un *runset* específico y utiliza el software de IC Compiler II. Las reglas de antena se centran en especificar el área máxima de metal que se puede conectar a un gate sin el source o drain para actuar como elemento de descarga. Las relaciones entre gate y los demás metales son: 100:1 hasta 5000:1 para que la carga sobre el metal no dañe la compuerta. Algunas formas de solucionar estos problemas pueden ser: seccionar el metal en conexiones de metales más pequeños para reducir la potencia del rayo plasma durante la fabricación o la colocación de un diodo entre el metal y tierra para evitar la conducción de corriente y que esta pueda causar una descarga no deseada por las altas temperaturas.

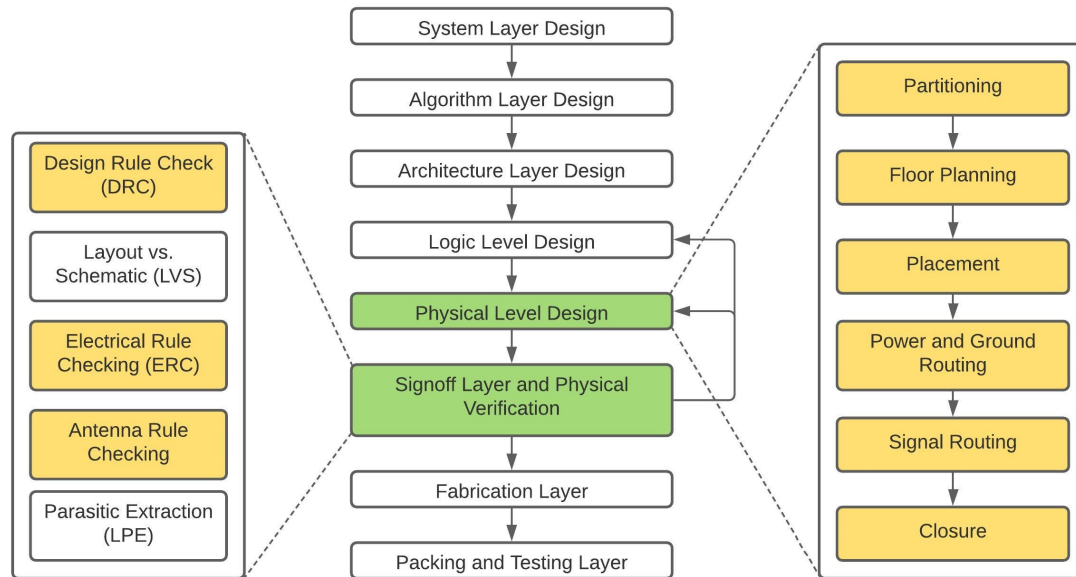


Figura 7: Flujo de diseño que ilustra las partes específicas a realizar de la síntesis física y la etapa de *Signoff* en amarillo

Electronic Design Automation (EDA)

La industria **EDA** existe con el fin de proporcionar herramientas de software para diseñadores electrónicos, para automatizar los procesos de diseño y fabricación de **CI**s. Esto permite que los precios para la fabricación de estos dispositivos disminuya, además de la especialización en cada una de las herramientas de software y separación del proceso de fabricación con el de diseño. Esto genera equipos de trabajo especializados en el área de diseño de **CI**s y por otro lado las compañías que fabrican dichos dispositivos pueden proporcionar

las restricciones y normas de diseño con las que pueden fabricar los dispositivos. Si bien ambas disciplinas se separan estas convergen con las herramientas **EDA**, siendo el puente entre los diseñadores y la industria de semiconductores, reduciendo costos, generando equipos especializados y creando una nueva industria en el mundo.[14]

Actualmente las dos compañías más grandes en la industria **EDA** son Cadence y Synopsys. Para fines de esta investigación se estarán utilizando las herramientas **EDA** de Synopsys. Esta cuenta con todas las herramientas para completar el flujo de diseño propuesto en la Figura 3 hasta la etapa de *Signoff and Physical Verification*. Esto se debe a que Synopsys no es una empresa que se dedica a la fabricación de **CI**s. Además como se mencionó anteriormente este es el puente entre el fabricante y el diseñador, por lo que las restricciones de diseño y fabricación han sido proporcionadas por **TSMC** la empresa a la que se mandará a hacer el **CI**, en donde se podrán concluir las últimas dos etapas del flujo de diseño y obtener el **CI** físico.[15]

Synopsys

Synopsys es una herramienta de diseño y simulación para cada una de las diferentes etapas del flujo de diseño. Estas herramientas son proporcionadas por el departamento de Ing. Electrónica, Mecatrónica y Biomédica de la UVG. Las cuales se aprenden a utilizar en los cursos de Nano Electrónica 1 y 2 son: HSpice, Wave Viewer, IC Compiler, IC Compiler II, Custom Compiler, Verdi3, VCS, Design Vision, Nanotime, IC Validator, StarRC, Formality, Milky Way, entre otras. En los siguientes apartados se desarrolla acerca de las herramientas que son de interés para el desarrollo del proyecto de graduación en lo que serán las etapas de Síntesis Física y en la etapa de *Signoff* y la Verificación Física: **DRC**, **ERC** y *Antenna Rule Checking* [1]

IC Compiler II

La sucesora a la herramienta IC Compiler I, es una de las herramientas líder para lo que es Place and Route. Esta herramienta es esencial para la síntesis física. En esta herramienta permite procesar el *verilog* proveniente de la síntesis lógica para convertirlo en un *layout* que represente el circuito físico en silicio, el cual será luego verificado.[16]

IC Validator

IC Validator es una de las herramientas de Synopsys para la etapa de *Signoff* y Verificación física. Este software permite realizar **DRC**, *Antenna* y **ERC**, verificaciones necesarias para asegurar que el *layout* generado en la etapa anterior pueda ser fabricado.[17]

Metodología

Para poder desarrollar un flujo de diseño capaz de generar un circuito funcional, con la posibilidad de ser manufacturado en silicio, es importante que este este bien definido. Sus diferentes etapas, las entradas, las salidas y su interpretación deben de estar claras. El diagrama que se muestra en la figura 8 define todo el flujo de diseño que se estará manejando a lo largo de esta investigación. De color verde se muestran los diferentes módulos de trabajo los cuales se describen a continuación:

- **Síntesis Física:** Para esta etapa se estarán tomando como entradas las interconexiones moradas. En este caso son un archivo de verilog (.v) el cual proviene de la síntesis física y la realimentación de las diferentes verificaciones, luego de la primera iteración de la síntesis física. El proceso de síntesis física, descrito anteriormente en el marco teórico, es el que se estará trabajando con la herramienta de IC Compiler II de Synopsys. Esta nos permite realizar las diferentes partes que, en orden de realización, se enuncian a continuación: *Partitioning, FloorPlanning, Placement, Power and Ground Routing, Signal Routing y Closure*.

En la primera iteración se realizan todos los pasos desde cero. Al finalizar el proceso, los resultados son almacenados en un archivo GDS (.GDS2) y una *Milky Way Library*. Estos se pueden ver en las interconexiones de color amarillo.

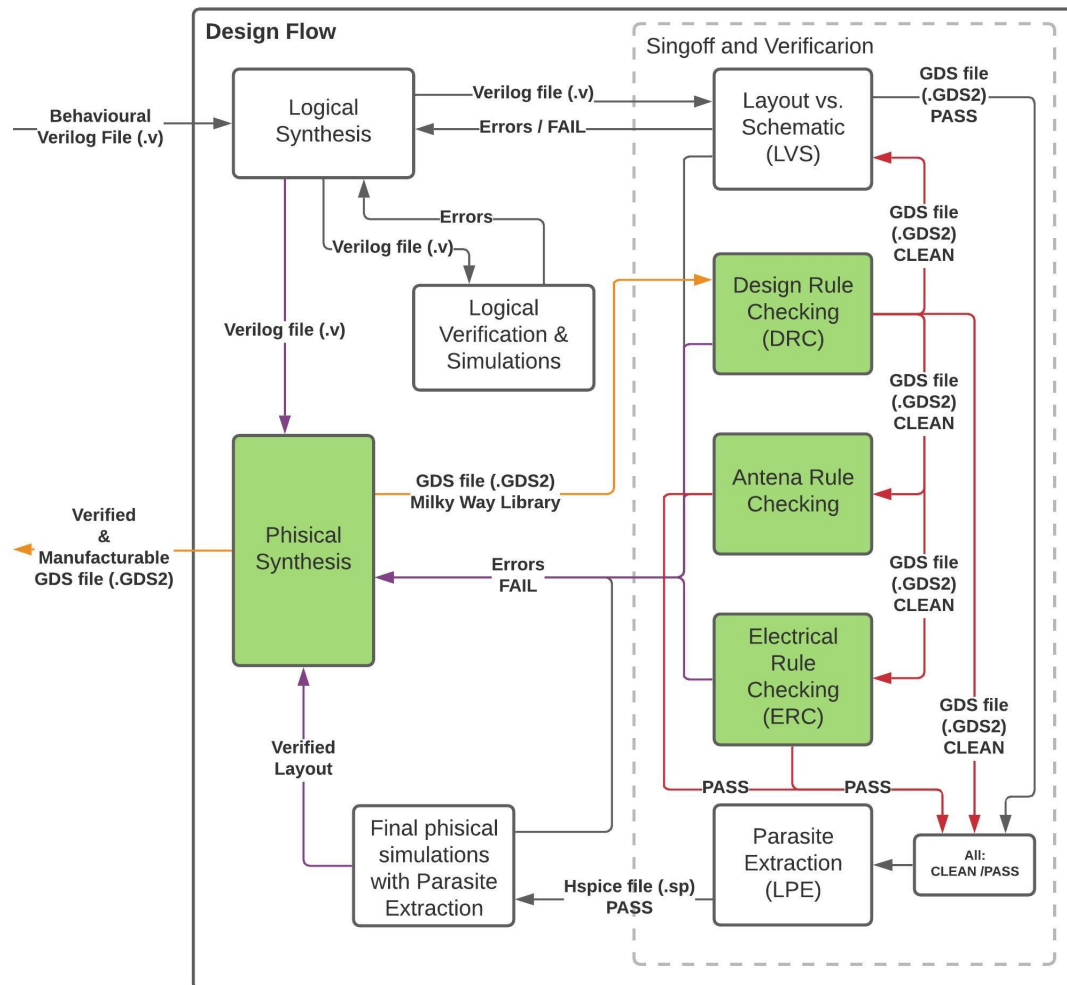
El proceso de síntesis física produce el archivo verificado y listo para manufacturar en su última iteración. Esta última iteración sucede en el momento en donde, luego de haber pasado todo el proceso de *SignOff*, pasa la simulación final con la extracción de parásitos. Esto indica que el *layout* cumple con todos los requerimientos del fabricante y los requerimientos de diseño, aún después de aplicarle todas las no idealidades y restricciones que el software permita.

- **SignOff:** La etapa de *SignOff* consta de cinco etapas importantes. Sin embargo únicamente se estarán verificando tres en esta investigación. Para esta etapa se estará utilizando la herramienta de Synopsys, IC Validator. Las tres verificaciones a realizar son: **DRC**, **ERC** Y *Antenna Rule Checking*. El proceso para realizar estas verificaciones es el siguiente:

De primero se pasa por la verificación **DRC**. A esta etapa le ingresa el segmento amarillo correspondiente a los resultados obtenidos de la síntesis física. Comparándolos con el *runset* de las librerías de **TSMC** se puede obtener un resultado de **CLEAN** o **FAIL**. Si el resultado es **FAIL** se deben revisar los errores y con estos realizar las correcciones correspondientes en la síntesis física. Este proceso se ve representado por el segmento de color morado. Si el resultado es **CLEAN**, entonces los resultados son enviados a todas las verificaciones restantes en un formato GDS (.GDS2), representado por los múltiples segmentos de color rojo.

Para las verificaciones de **ERC** y *Antenna Rule Checking* el proceso es el mismo. Se reciben los resultados **CLEAN** de la prueba **DRC** en formato GDS (.GDS2) los cuales son pasados por las verificaciones. Estas pueden tener dos resultados: **PASS** o **FAIL**. Si el resultado es **FAIL** entonces se revisan los errores y se corrigen en la síntesis física. Para luego ser verificados por el **DRC** nuevamente, para revisar que las correcciones no hayan infringido nuevas normas de diseño de fabricación. Este proceso se realiza

Para comprender el último segmento rojo que conecta los resultados de las verificaciones exitosas, con la verificación **LPE** se espera a que todos los resultados de todas las verificaciones estén en **CLEAN** o **PASS**. Por lo tanto la extracción de parásitos es realizada con el *netlist* del *layout* completamente verificado.



Para realizar este proceso de forma exitosa y crear una síntesis física robusta como la que se desea se realizarán las siguientes tareas:

- 24

- Como segundo punto es importante comprender porque y como funcionan las librerías de **TSMC** para verificar su vigencia y compatibilidad entre ellas. Además, se planea corregir la mayor cantidad de errores posibles de los trabajos anteriores.
- Luego de esto se tiene planeado hacer una migración de IC Compiler I a IC Compiler II. Esto se hará ya que muchas de las herramientas de las otras partes del proceso congenian mejor con esta versión. Además de que el flujo de diseño este más actualizado.
- Teniendo el proyecto en la nueva herramienta y verificando que contenga la mínima cantidad de errores se procede a lo que es la etapa de pruebas, en donde se someterá a la síntesis física a un conjunto de circuitos. Esto permite perfeccionar y observar bajo que circunstancias la síntesis falla y corregirla. Los circuitos que se estarán pasando por la síntesis física son: una compuerta NOT, un Full Adder, un Ripple Carry Adder y un circuito de diseño personal para comprobar que la síntesis funcione.
- Para poder validar todos estos resultados se usará IC Validator, la cual permite hacer las verificaciones de DRC, ERC y Antena. Primero los resultados serán revisados por el DRC y luego por las otras dos verificaciones. Es importante mencionar que esto sucede en paralelo a cada una de las compuertas que se diseñan, para ir perfeccionando la síntesis física.
- En cuanto la síntesis física devuelva resultados sin errores se da por concluida el diseño de cada compuerta. Las compuertas se realizan en el orden en el que se mencionan para subirle la dificultad al proceso de diseño. Por lo tanto los resultados escalan a medida que se concluye cada proceso.

Cronograma de actividades

Cronograma de actividades primer semestre

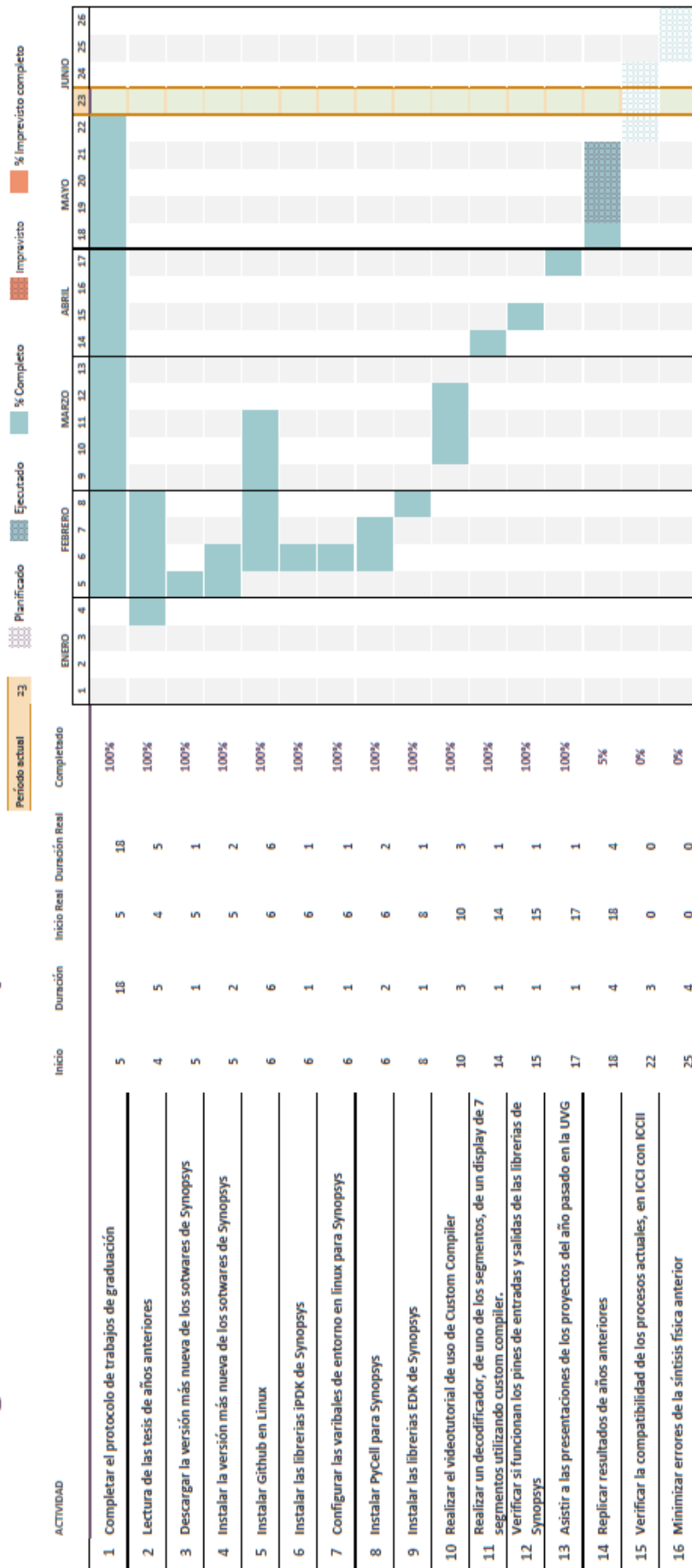


Figura 9: Cronograma para el primer semestre del 2021

Cronograma de actividades segundo semestre

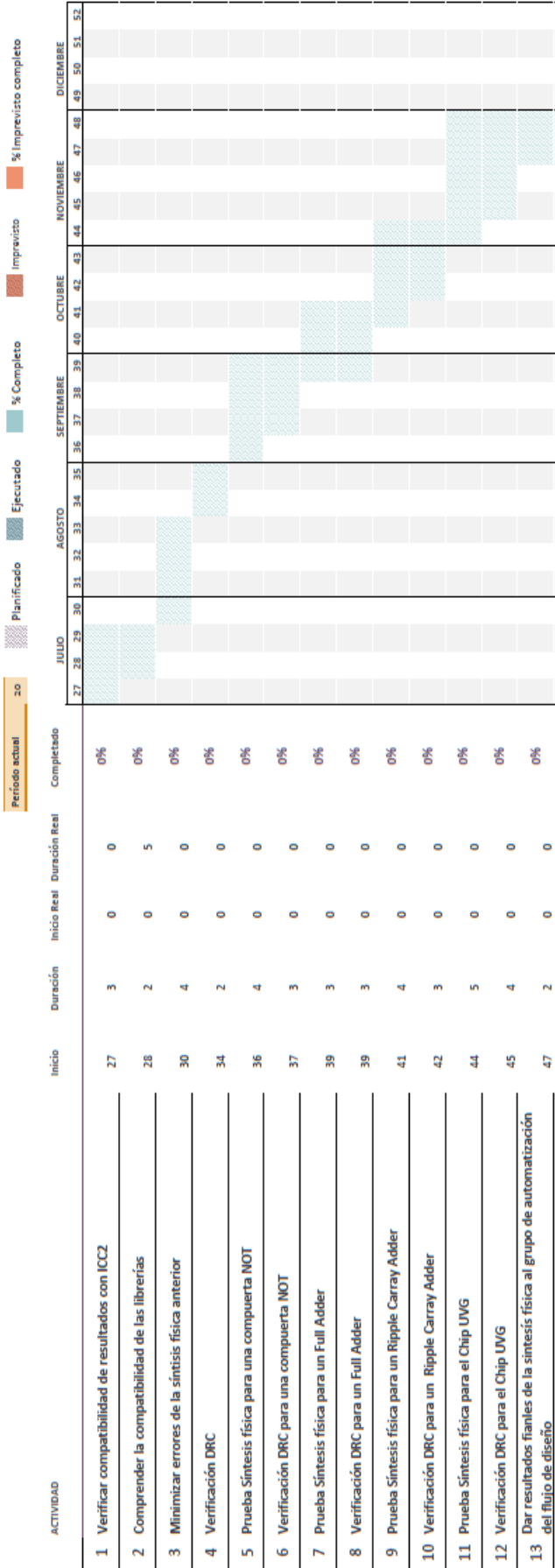


Figura 10: Cronograma para el segundo semestre del 2021

Índice preliminar

Resumen	1
Antecedentes	1
Justificación	2
Objetivos	3
Objetivo General	3
Objetivos Específicos	3
Marco teórico	3
VLSI	3
Flujo de Diseño	4
Diseño en la capa Sistema	5
Diseño en la capa Algoritmo	5
Diseño en la capa Arquitectura	6
Diseño en la capa Lógica	6
Diseño en la capa Física	6
Diseño en la capa <i>Signoff</i> y Verificación Física	7
<i>Design Rule Checking (DRC)</i>	7
<i>Layout vs. Schematic (LVS)</i>	7
<i>Antenna Rule Checking</i>	7
<i>Electrical Rule Checking (ERC)</i>	8
<i>Parasitic Extraction (LPE)</i>	8
Final Simulation	8
Diseño en la capa de Fabricación	8
Diseño en la capa de Empaquetado y Pruebas	8
Síntesis Física, <i>Design Rule Check (DRC)</i> , <i>Electrical Rule Check (ERC)</i> y <i>Antena Rule Check</i>	10

Síntesis Física	10
<i>Partitioning</i>	10
<i>Floor Planning</i>	10
<i>Placement</i>	10
<i>Power and Ground Routing</i>	10
<i>Signal Routing</i>	11
<i>Closure</i>	11
Proceso de Síntesis Física Anterior	11
DRC	16
<i>Minimum Width</i>	16
<i>Minimum Intralayer Spacing</i>	17
<i>Minimum Interlayer Spacing</i>	17
<i>Minimum Enclosure</i>	17
<i>Minimum Extension</i>	17
<i>Maximum Width</i>	17
<i>Density Rules</i>	18
<i>Extension Rules</i>	18
Verificación de DRC Anterior	18
ERC	19
<i>Soft Connect Check</i>	20
<i>Path Check</i>	20
<i>PTAP/NTAP Connectivity Check</i>	20
<i>MOSFET Power and Ground Check</i>	20
<i>Gate Directly Connected to Power or Ground</i>	20
<i>Floating Gate</i>	20
<i>Floating Well</i>	20
<i>Antenna Rule Checking</i>	20
<i>Electronic Design Automation (EDA)</i>	21

Synopsys	22
IC Compiler II	22
IC Validator	22
Metodología	23
Cronograma de actividades	25

Referencias

- [1] J. de los Santos, “Diseño de un sumador/restador completo de 32 bits con tecnología CMOS en un proceso de 28 nanómetros usando aplicaciones de diseño de la empresa Synopsys,” en *Trabajo de Graducacion, Modalidad Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2014.
- [2] S. Rubio, “Definición del Flujo de Diseño para Fabricación de un Chip con Tecnologia VLSI CMOS,” en *Trabajo de Graducacion, Modalidad Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2019.
- [3] L. Nájera, “Implementación de circuitos sintetizados a nivel netlist a partir de un diseño en lenguaje descriptivo de hardware como primer paso en el flujo de diseño de un circuito integrado,” en *Trabajo de Graducacion, Modalidad Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2019.
- [4] L. Abadía, “Posicionamiento e interconexión entre componentes de un circuito sintetizado para el flujo de diseño de un circuito a escala nanométrica utilizando la herramienta de IC Compiler,” en *Trabajo de Graducacion, Modalidad Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2020.
- [5] M. Flores, “Corrección de anillo de entradas/salidas y pruebas de antenna y ERC para la definición del flujo de diseño del primer chip con tecnología nanométrica desarrollado en Guatemala,” en *Trabajo de Graducacion, Modalidad Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2020.
- [6] M. Sibrian, “Verificación de reglas de diseño (DRC) para el desarrollo de un flujo funcional de un circuito integrado con tecnología nanométrica,” en *Trabajo de Graducacion, Modalidad Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2020.
- [7] “Apple unleashes M1,” *Apple Newsroom*, nov. de 2020. dirección: <https://www.apple.com/newsroom/2020/11/apple-unleashes-m1/>.
- [8] K. G. Krishna. B, “Circuit Design Enviroment and Layout Planning,” *Intel Technology Journal*, págs. 13-20, 1999.
- [9] C. Esquit, *Computer Architecture Lecture 1: Introduction and Five Components of a Computer*, jul. de 2020.
- [10] —, *Introduction to VLSI System Design Lecture: Fabrication and Layout*, mayo de 2021.

- [11] K. H, “Digital Integrated Circuit Design From VLSI Architectures to CMOS Fabrication,” *Intel Technology Journal*, págs. 13-20, 1999.
- [12] L. A. López, “Manual Síntesis Física,” en *Manual Síntesis Física para IC Compiler I*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2020.
- [13] M. Sibrian, “Manual de DRC,” en *MANUAL DE USUARIO PARA LA VERIFICACIÓN DE REGLAS DE DISEÑO EN UN CIRCUITO INTEGRADO CON TECNOLOGÍA NANOMÉTRICA*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2020.
- [14] *EDA101 - Introduction to Electronic Design Automation*. Cadence Design Systems, abr. de 2020. dirección: <https://www.youtube.com/watch?v=HkT363NU0BA>.
- [15] Synopsys. dirección: <https://www.synopsys.com/silicon-design.html>.
- [16] —, dirección: <https://www.synopsys.com/implementation-and-signoff/physical-implementation/ic-compiler.html>.
- [17] —, dirección: <https://www.synopsys.com/silicon/mask-synthesis/icv-workbench.html>.