

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018,
publicado en el Diario Oficial de la Federación el 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

MAESTRÍA EN DISEÑO ELECTRÓNICO



REPORTE DE FORMACIÓN COMPLEMENTARIA EN ÁREA DE CONCENTRACIÓN EN SISTEMAS EMBEBIDOS Y TELECOMUNICACIONES

Reporte técnico para obtener el grado de

MAESTRO EN DISEÑO ELECTRÓNICO

Presenta: Guillermo Rentería Zamudio

Asesor: Zabdiel Brito Brito

Asesor: Jorge Arturo Pardiñas

Tlaquepaque, Jalisco. Agosto de 2016.

Contenido

1. Introducción:.....	3
2. Resumen de los proyectos realizados:.....	3
2.1 Diseño de un regulador de voltaje tipo buck.	3
2.2.1 Introducción:.....	3
2.1.2 Antecedentes:.....	3
2.1.3 Solución desarrollada:.....	4
2.1.4 Análisis de resultados:	5
2.1.5 Conclusiones:	6
2.2 Codificador de canal en un sistema de comunicación digital.	6
2.2.1 Introducción:.....	6
2.2.2 Antecedentes:.....	7
2.2.3 Solución desarrollada:.....	7
2.2.4 Análisis de resultados:	8
2.2.5 Conclusiones:	9
2.3 Diseño de un PCB.	9
2.3.1 Introducción:.....	9
2.3.2 Antecedentes:.....	9
2.3.3 Solución desarrollada:.....	10
2.3.4 Análisis de resultados	11
2.3.5 Conclusiones	12
2.4 Conclusiones generales	12
Apéndices	13
Diseño de un regulador de voltaje tipo buck.....	14
Codificador de canal en un sistema de comunicación digital.....	29
Diseño de un PCB.....	35

1. Introducción:

Decidí elegir como área de concentración sistemas embebidos y telecomunicaciones, dado por mi interés personal, así como el apego más cercano a mi desempeño laboral. Dentro de esta concentración, las materias cursadas, así como sus proyectos finales, fueron las siguientes:

- Taller de diseño de tarjetas de circuito impreso.
 - o Diseño de un PCB, desde esquemáticos hasta layout.
- Sistemas embebidos avanzados.
 - o Diseño de un planificador de tareas cooperativo.
- Diseño de sistemas analógicos basados en dispositivos comerciales.
 - o Diseño de un regulador de voltaje tipo buck.
- Procesamiento digital de señales.
 - o Filtro ecualizador pasa altas.
- Introducción a las comunicaciones digitales y analógicas.
 - o Codificador de canal en un sistema de comunicación digital.

Los proyectos a mencionar en este reporte serán de las materias de:

- Diseño de sistemas analógicos basados en dispositivos comerciales.
- Taller de diseño de tarjetas de circuito impreso.
- Introducción a las comunicaciones digitales y analógicas.

2. Resumen de los proyectos realizados:

2.1 Diseño de un regulador de voltaje tipo buck.

2.2.1 Introducción:

Los reguladores de voltaje tipo buck son convertidores de potencia, DC/DC sin un aislamiento galvánico, el cual obtiene a su salida un voltaje continuo menor que a su entrada. Una forma de reducir tensión continua es mediante un divisor de tensión; sin embargo, esto implica mucho gasto de energía en forma de calor. En cambio, si se utiliza un regulador tipo buck, puede llegarse a tener una alta eficiencia y autorregulación.

2.1.2 Antecedentes:

Como ya se mencionó, los reguladores tipo buck de DC-DC proveen de una eficiencia de potencia mucho mayor que la de los reguladores lineales, los cuales son circuitos que bajan el voltaje disipando la potencia en forma de calor, lo cual no eleva la corriente de salida. Los reguladores tipo buck pueden llegar a alcanzar una eficiencia comúnmente mayor al 90%, haciéndolos útiles para tareas tales como el convertir la fuente principal de alimentación de una computadora (usualmente 12V) a voltajes menores que necesiten dispositivos tales como USB, RAM, etcétera.

2.1.3 Solución desarrollada:

Para este proyecto se contempló una entrada variable de 9V a 15V, teniendo a la salida un voltaje de 1.5V, con una variación máxima del 15%. El regulador cuenta con las etapas de modulador, filtro de salida y compensación. La sección del modulador fue diseñada de tal forma que su entrada es la salida del amplificador de error, la cual es usada para comparar la salida con la referencia. Su salida es el nodo de fase; la ganancia del modulador es el voltaje de entrada dividido por el voltaje de pico a pico del oscilador. Esta sección se muestra en la figura 1.

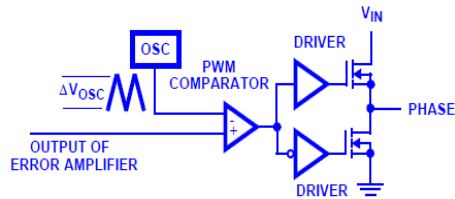


Figura 1: modulador.

Posterior a esto, se desarrolló el filtro de salida, el cual consiste en un inductor y un capacitor en serie con una resistencia. La entrada del filtro de salida es el nodo de fase de la etapa del modulador, mientras que la salida de esta etapa es la salida del regulador. El diagrama de dicho filtro se muestra en la figura 2.

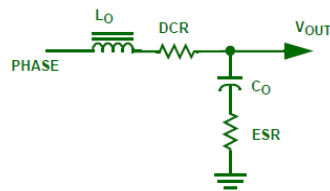


Figura 2: filtro de salida.

Finalmente, la red de compensación es la que permite al regulador ajustar perturbaciones de la carga o cambios en el voltaje de entrada (figura 3). Utilizando una compensación tipo 3 se da forma al perfil de la ganancia con respecto a la frecuencia, dando un boost de 180 grados a la fase usando dos ceros. Dicho boost es necesario para contrarrestar efectos de resonancia poco amortiguada ubicada en el filtro de salida en el polo doble.

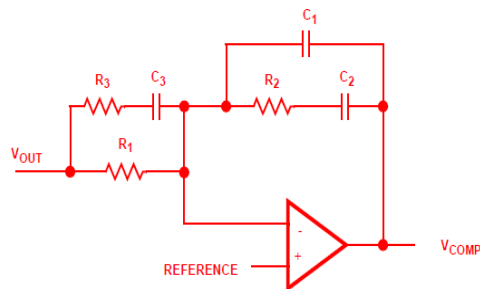


Figura 3: red de compensación.

2.1.4 Análisis de resultados:

Una vez integrando las tres partes ya mencionadas, se cuenta con el regulador de voltaje completo, mostrado en la figura 4:

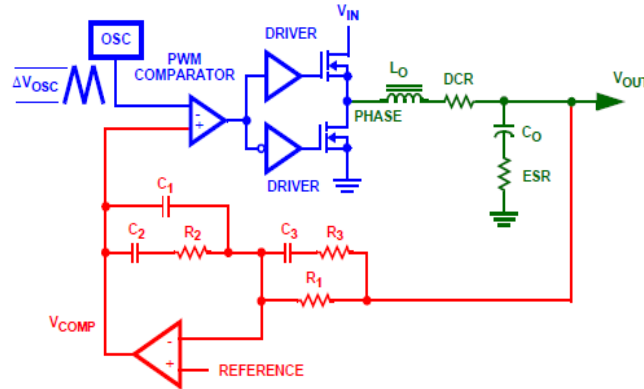


Figura 4: Regulador de voltaje.

Posteriormente se procedió a sacar la ecuación de ganancia del sistema, tomando en cuenta parámetros como el rango de voltaje de entrada, el voltaje de salida, corriente máxima de salida y una frecuencia fija; se tuvo que partir de valores iniciales para los componentes discretos (inductor y capacitor), en donde, una vez tomando en cuenta los parámetros ya mencionados, se hicieron ajustes necesarios para llegar a los valores ideales. El controlador utilizado fue el modelo LTC3775 de Linear Technology, el cual cumple con las características necesarias para este diseño particular. Finalmente, para el compensador, se tuvo que elegir la relación de las resistencias de feedback RA y RB, mostradas en la siguiente figura:

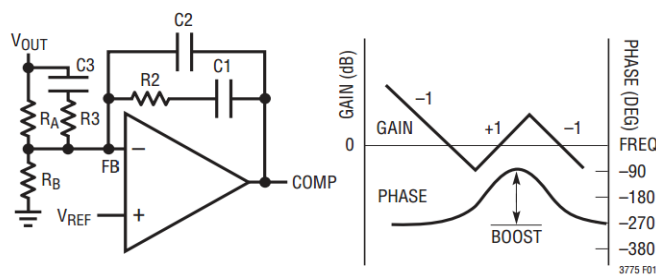


Figura 5: compensador.

Una vez hecho esto, se puede proceder a calcular R2. La relación de R2/RA recorre la ganancia de lazo abierto hasta el ancho de banda deseado. La implementación fue puesta a prueba usando el software de LT Spice, obteniendo los siguientes resultados a la salida del regulador:

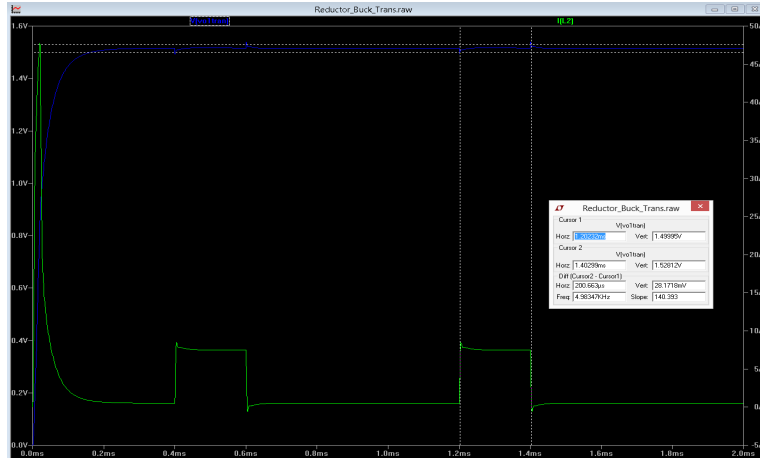


Figura 6: simulación del transitorio.

2.1.5 Conclusiones:

Por medio de este proyecto fue posible poner en práctica el diseño de un regulador tipo buck, con etapas de modulador, filtro de salida y compensador. Dicho diseño puede tener aplicación directamente en el campo de PCBs, área en la que yo me especializo actualmente, y en donde he podido aplicar parte de los conocimientos adquiridos, al mejorar ciertas características del layout. Las simulaciones fueron realizadas en LT Spice, en donde se verificó que la salida del regulador cumpliera con las especificaciones deseadas. Actualmente utilizo la herramienta de Hyperlynx, en donde, de forma similar, realizo simulaciones para conocer valores tales como caídas de voltaje y densidad de corriente a la salida de un regulador.

2.2 Codificador de canal en un sistema de comunicación digital.

2.2.1 Introducción:

En un canal de comunicación, por lo general se cuenta con un codificador y decodificador. La información es codificada con el objetivo de disminuir la probabilidad de errores provocados por los efectos del ruido del canal; se agrega información redundante que permite al receptor detectar errores y corregirlos, tal como se muestra en el diagrama de la figura 7.

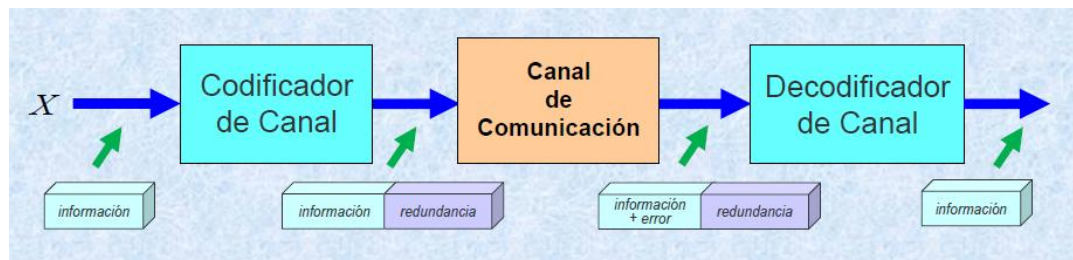


Figura 7: codificador de canal.

2.2.2 Antecedentes:

En cualquier canal de comunicación, la información enviada puede estar afectada por desvanecimientos, evitando alcanzar el nivel de señal adecuado para obtener una relación señal a ruido (S/N) ideal para recuperar la información sin errores. Con el fin de detectar y corregir los bits erróneos, se aplica a la señal transmitida una serie de transformaciones. Como se menciona en la introducción, el objetivo es reducir la probabilidad de error, o garantizar una S/N necesaria para cierta tasa de error, transformando secuencias binarias en secuencias que incluyan una redundancia estructurada; dichos bits redundantes son utilizados para detectar y corregir errores.

2.2.3 Solución desarrollada:

Para este proyecto se realizó un sistema de comunicación que contiene la función de codificación de canal, comparando los resultados respecto al sistema sin codificador, utilizando la herramienta de Matlab. El sistema de comunicación es mostrado en la figura 8, en donde el canal binario simétrico permite configurar directamente la probabilidad de error que se desea sobre los datos binarios a su entrada.

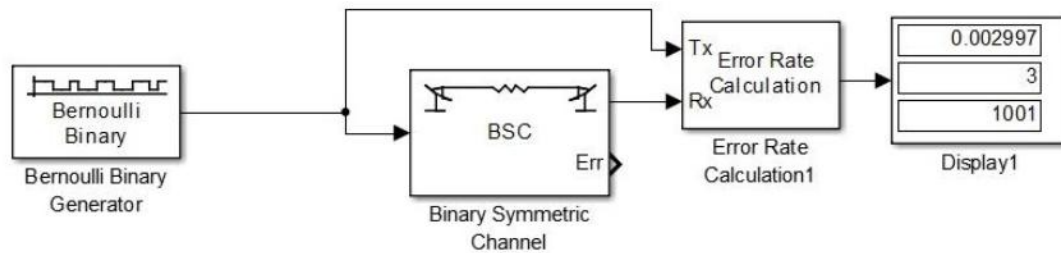


Figura 8: sistema de comunicación digital con canal binario simétrico.

Con el uso de Matlab, se ejecutó la simulación del sistema en varias ocasiones para diversos valores de probabilidad de error del canal, generando por lo menos un total de 1000 bits de información. En este paso se constató que a mayor probabilidad de error, mayor el bit error rate, como se muestra en la siguiente tabla:

Tabla 1: Probabilidad de error y BER.

P = 0.005	P = 0.01	P = 0.015	P = 0.025	P = 0.05	P = 0.1	P = 0.15	P = 0.2	P = 0.25	P = 0.3
0.58	1.66	0.97	2.92	5.17	11.13	14.94	20.89	25.39	31.73

Posteriormente se amplió el sistema de la figura 8, añadiendo un codificador y decodificador de canal del tipo de Bloques de Hamming (figura 9). La configuración por omisión de este bloque es un código (7,4). La entrada al codificador es una trama de la longitud correspondiente a la longitud de la palabra de información (4), de tal manera que la fuente de información tiene configurada su salida de la misma manera.

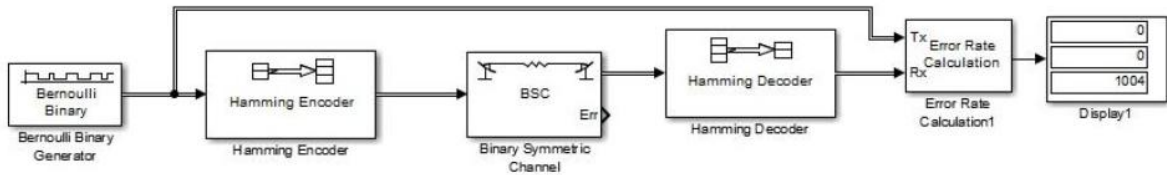


Figura 9: sistema de comunicación digital con canal binario simétrico y codificador de canal de Hamming.

2.2.4 Análisis de resultados:

Utilizando nuevamente Matlab, se ejecutaron varias simulaciones del sistema, y utilizando los mismos valores de probabilidad de error mostrados en la tabla 1, se obtuvieron los Bit Error Rates mostrados en la tabla 2:

Tabla 2: comparación de resultados con y sin codificación Hamming.

	P = 0.005	P = 0.01	P = 0.015	P = 0.025	P = 0.05	P = 0.1	P = 0.15	P = 0.2	P = 0.25	P = 0.3
w/o coding	0.58	1.66	0.97	2.92	5.17	11.13	14.94	20.89	25.39	31.73
Hamming	0	0	0.09	1.17	2.14	6.44	13.08	20.5	26.66	34.17

La siguiente gráfica muestra la comparación de los dos métodos, graficando la probabilidad de error configurada en el canal contra la probabilidad de error obtenida después del decodificador.

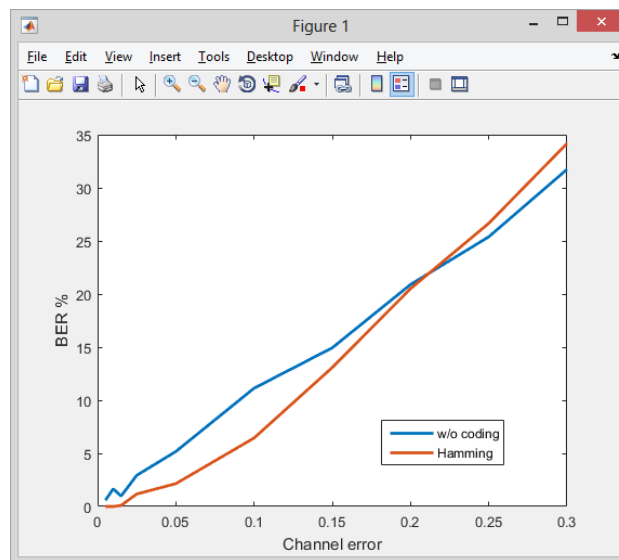


Figura 10: prob. de error en el canal VS prob. de error obtenida después del decodificador.

2.2.5 Conclusiones:

Al observar la tabla 2 y la gráfica de la figura 10 es claramente visible que la codificación Hamming es muy útil para reducir la probabilidad de error obtenida después del decodificador. Sin embargo, cabe destacar que esto ocurre sólo para un rango de las probabilidades de error en el canal, dado que cuando esta probabilidad llega a volverse muy grande, entonces el resultado de la codificación/decodificación Hamming llegará a ser mayormente erróneo a que si no se utilizara ningún tipo de codificación. Esto se debe principalmente a que las distancias mínimas del código Hamming se tornan más cortas por el simple hecho de la adición de ruido, corrompiendo los datos; a menor distancia, menor capacidad de detectar y corregir errores.

2.3 Diseño de un PCB.

2.3.1 Introducción:

Un PCB (siglas en inglés de Printed Circuit Board) es por concepto una tarjeta la cual soporta mecánicamente y conecta eléctricamente componentes electrónicos usando pistas conductoras, pads y otras características a partir de láminas de cobre adjuntas a material de sustrato no conductor. Los componentes usados generalmente son soldados al PCB.

Un PCB se puede caracterizar por ser de una sola capa, de dos capas o multicapa. Las pistas conductoras en diferentes capas pueden ser conectadas mediante el uso de vías. Aquellos PCBs con mayor número de capas permiten una densidad de componentes mayor. Actualmente el FR4 es el material de sustrato mayormente utilizado en la industria; un bloque básico de PCB puede estar conformado por un panel de FR4 pegado con una delgada capa de cobre por uno o ambos lados.

2.3.2 Antecedentes:

Los PCBs son usados ampliamente en la industria electrónica. Para su elaboración, se requiere de un esfuerzo adicional de diseño para definir el layout de los circuitos a utilizar, aunque la manufactura y el ensamble puedan ser automatizados. El manufacturar circuitos con PCBs resulta ser más barato y rápido con respecto a otros métodos de cableado, dado que los componentes son montados y conectados en una única pieza. En 2014, su mercado mundial excedió la cifra de \$60.2 mil millones de dólares.

Actualmente, se diseñan PCBs con software de layout dedicado, generalmente con los siguientes pasos:

- Captura de esquemáticos.
- Dimensiones de la tarjeta.

- Posición de componentes.
- Stack up de la tarjeta.
- Impedancias de las líneas.
- Ruteo de señales.
- Archivos Gerber para manufactura.

2.3.3 Solución desarrollada:

A lo largo del curso se planteó el diseño completo de un PCB, en donde se comprenden los siguientes 4 pasos:

- Creación de un proyecto y librerías.
- Creación de footprints y pads.
- Creación de esquemáticos.
- Diseño de layout.

Partiendo de la creación del proyecto y librerías, el software utilizado fue Allegro Design Entry CIS, el cual cuenta con dichas capacidades, así como para la captura de esquemáticos. Una vez creado un proyecto, se procede a crear una librería, la cual contiene la información lógica de cada símbolo a agregar en un proyecto (resistencias, capacitores, inductores, ICs, etc.). Cada librería puede contener uno o más símbolos; por ejemplo, la librería de resistencias puede incluir varias resistencias de distintos valores. Posteriormente, se debe crear un dibujo para cada símbolo; mediante un lienzo se tiene la capacidad de dibujar al criterio del diseñador la forma que tendrá dicho símbolo (sin embargo, se recomienda seguir con los estándares ya establecidos). El dibujo del símbolo debe contar tanto con el cuerpo como con los pins del componente.

Una vez que ya se tenga definido un proyecto y librerías, el siguiente paso en el diseño consiste en crear footprints y pads. Para esto, se utilizan las herramientas de Pad Designer y PCB Editor de Cadence, así como el LP Calculator. Para cada footprint, se debe contar con la respectiva hoja de datos del fabricante, ya que es ahí donde se encuentra la información referente a dimensiones del componente. Para aquellos casos en los que no se mencione el valor del pitch (distancia entre pins), se debe usar la herramienta de LP Calculator, en la cual, una vez que se ponen ciertos valores de dimensiones tales como largo, ancho y altura del cuerpo, largo, ancho y altura del pin, se podrá obtener un land pattern, el cual define la distancia entre pins, así como nuevas dimensiones sugeridas de largo y ancho para cada pin utilizado en el footprint. El siguiente paso consiste en el uso de la herramienta Pad Designer, la cual tiene como objetivo el crear archivos conteniendo un pin (pad) con dimensiones específicas. Los archivos de pads son referenciados tantas veces como se requiera para uno o más

símbolos. Y por último, para finalizar con la creación de un footprint, se debe utilizar la herramienta de PCB Editor, en donde se pondrán las dimensiones obtenidas previamente de las hojas del fabricante, del LP Calculator, así como referenciar a los archivos de pads creados con Pad Designer.

El tercer paso consiste en la creación de esquemáticos, definiendo un proyecto listo para utilizarse en un board file, mediante la creación de un netlist. Dicho proyecto debe contar con las hojas suficientes para contener todas las conexiones que se deseen realizar, incluyendo también todos los símbolos lógicos de los cuales partirán dichas conexiones. Dentro de una página de esquemático, se pueden realizar tareas tales como: agregar un símbolo lógico, conexiones de un símbolo a otro dentro de la misma página, conexiones a otras páginas, conexiones por medio de buses, nombrar señales, conexiones de PWR o GND, etc. Una vez terminado el diseño de esquemático, es posible exportar el netlist, el cual será utilizado para el diseño físico del PCB.

2.3.4 Análisis de resultados:

Una vez realizados los tres pasos anteriores, es posible finalmente proceder al diseño del layout del PCB. Las tareas primordiales para esto son:

- Colocación de componentes.
- Creación de reglas de diseño.
- Fanout de componentes.
- Ruteo de señales.
- Fine tune del ruteo.
- Limpieza de DRCs.
- Acomodo de silkscreen

Siguiendo las reglas de diseño, aunado con la ayuda de otras partes técnicas, como ingenieros mecánicos y de manufactura, se debe proceder a rutear la totalidad de las señales existentes en la tarjeta, lo cual suele ser el paso más tardado. Algunos puntos a definir son: rutear en capas internas o externas, definir ruteo horizontal o vertical, tener referencia de GND, creación de planos de poder, así como el fine tuning para señales que lo requieran. Todo esto debe ser realizado con el criterio del ingeniero de CAD, cumpliendo con las reglas especificadas para cada interfaz.

Una vez teniendo todas las señales ruteadas, así como cero errores, se debe trabajar en silkscreen, lo cual será la leyenda que tenga cada componente en la tarjeta, con el objetivo de identificarla rápidamente teniendo la tarjeta físicamente. Una vez hechos estos pasos, la tarjeta debe estar lista para fabricarse en términos de manufactura.

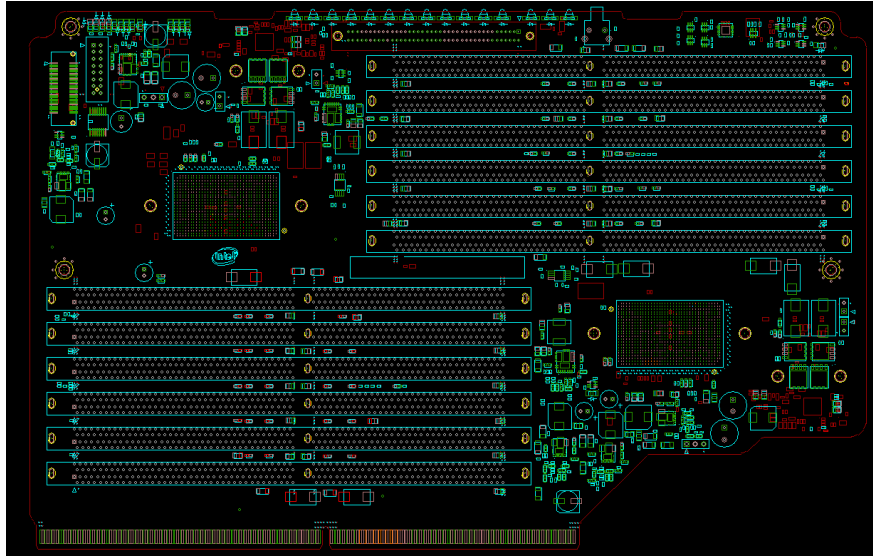


Figura 11: diseño de un PCB.

2.3.5 Conclusiones:

Siendo un ingeniero que labora actualmente como diseñador de PCBs, los temas vistos durante este curso fueron de gran ayuda, dado que se obtiene una perspectiva más amplia, partiendo desde la creación de proyectos, librerías, footprints y esquemáticos hasta llegar al diseño del layout del PCB, consiguiendo una mayor versatilidad en mi trabajo. Usualmente, un diseñador de PCBs solamente está enfocado en la parte física del diseño, es decir, trabajar únicamente en el board file, haciendo tareas de layout, tales como placement y routing. Gracias a este proyecto, poseo ahora las capacidades para realizar el diseño completo de una tarjeta mediante el uso del software de Cadence.

2.4 Conclusiones generales

En este resumen he detallado los proyectos de las tres materias que elegí del área de concentración de sistemas embebidos y telecomunicaciones que considero dejaron un mayor impacto en mí. Dicho impacto va de la mano con el transcurso que he tomado a lo largo de mi carrera laboral, desarrollándome en el diseño de plataformas para servidores durante ocho años, y ahora para amplificadores de audio. Gracias a los conocimientos adquiridos al concluir la maestría, poseo las herramientas necesarias para saber cómo mejorar y ampliar mi visión en un diseño electrónico. Los reguladores de voltaje siempre serán necesarios cuando se utilicen circuitos trabajando a diferentes niveles; la codificación de canal siempre será necesaria si se desea reducir la probabilidad de error en la transmisión de datos; y por último, los PCBs son la base de cualquier diseño electrónico actual. Cada uno de estos tópicos son pilares de la electrónica actual.

Apéndice

Diseño de un regulador de voltaje tipo buck

Contenidos

<i>Descripción general.</i>	15
<i>Modulador.</i>	15
<i>Filtro de salida.</i>	16
<i>Red de compensación.</i>	17
<i>Sistema completo.</i>	19
<i>Especificaciones del regulador.</i>	19
<i>Inductor.</i>	20
<i>Capacitor.</i>	20
<i>Controlador.</i>	21
<i>Compensador.</i>	23
<i>Simulación.</i>	24
<i>Costos y materiales.</i>	27

Descripción general.

Un regulador tipo buck es un convertidor de potencia, produciendo a su salida un voltaje continuo menor que su entrada. En sí, es una fuente conmutada con dos dispositivos semiconductores, siendo ambos MOSFETs, un inductor y un capacitor. El conmutar tiene ventajas sobre un divisor de tensión, tal como el ahorro de energía en forma de calor y eficiencia.

Se contempla una entrada variable de 9V a 15V, teniendo a la salida un voltaje de 1.5V, con una variación máxima del 15%. El regulador debe contar con las siguientes etapas:

- Modulador.
- Filtro de salida.
- Compensación.

Modulador.

La entrada del modulador es la salida del amplificador de error, la cual es usada para comparar la salida con la referencia, mientras que su salida es el nodo de fase. La ganancia del modulador es el voltaje de entrada dividido por el voltaje de pico a pico del oscilador; dicho dato puede obtenerse de la hoja de datos del controlador a usarse.

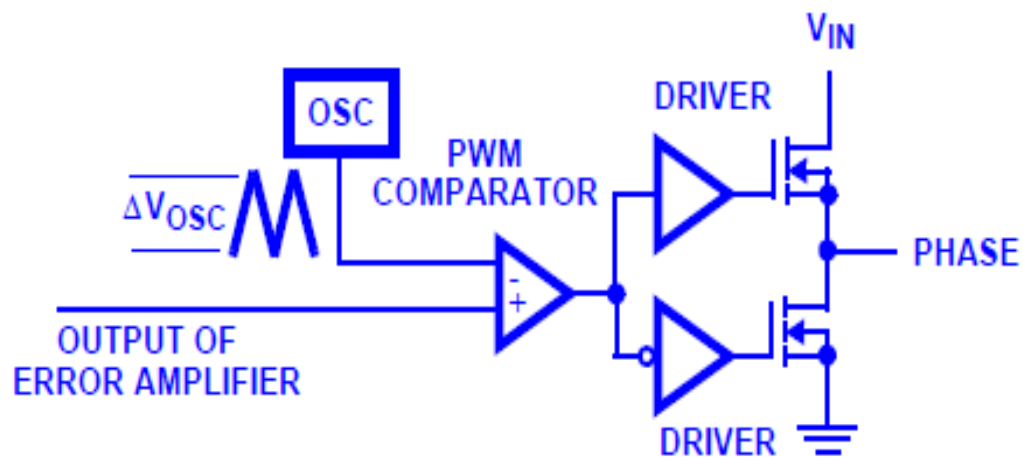


Figura 12: modulador.

La ganancia del modulador está dada por la siguiente ecuación:

$$Gain_{Mod} = \frac{V_{in}}{\Delta V_{osc}}$$

Filtro de salida.

El filtro de salida consiste en el inductor de salida, así como en la capacitancia. Se debe incluir tanto la resistencia DCR en serie con el inductor, así como la ESR del capacitor. La entrada del filtro de salida será el nodo de fase de la etapa del modulador, mientras que la salida de esta etapa es la salida del regulador.

La función de transferencia está dada por la siguiente ecuación:

$$Gain_{Filter} = \frac{1 + s * ESR * C_{out}}{1 + s * (ESR + DCR) * C_{out} + s^2 L_{out} * C_{out}}$$

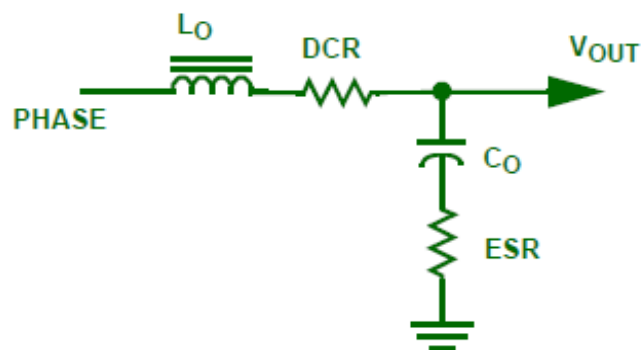


Figura 13: filtro de salida.

Red de compensación.

Cerrar el lazo de control permite al regulador ajustar perturbaciones de la carga o cambios en el voltaje de entrada. Una compensación adecuada permite un ancho de banda predecible y estabilidad incondicional. Utilizando una compensación tipo 3 se le da forma al perfil de la ganancia con respecto a la frecuencia, dando un boost de 180 grados a la fase usando dos ceros. El boost es necesario para contrarrestar efectos de resonancia poco amortiguada ubicada en el filtro de salida en el polo doble.

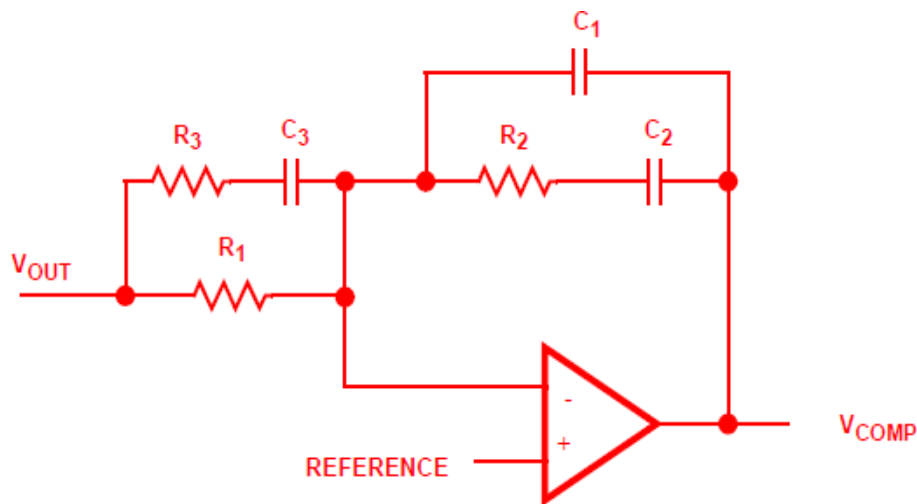


Figura 14: red de compensación.

El procedimiento para cálculo de componentes es el siguiente:

- Escoger un valor para R1 entre 2k y 5KOhms.
- Escoger una ganancia de R2/R1 la cual mueva la ganancia de lazo abierto al ancho de banda deseado, permitiendo que el crossover de 0dB ocurra en el rango de frecuencias donde la compensación de tipo 3 tenga su segunda ganancia plana. R2 está dada por la siguiente ecuación:

$$R_2 = \left(\frac{DBW}{F_{LC}} \right) * \frac{\Delta V_{OSC}}{V_{IN}} * R_1$$

- Cálculo de C2 poniendo el cero al 50% de la frecuencia del doble polo del filtro de salida.
- $$C_2 = \frac{1}{\pi * R_2 * F_{LC}}$$

- Cálculo de C1 poniendo el primer polo en la frecuencia cero del ESR:

$$C_1 = \frac{C_2}{2\pi * R_2 * C_2 * F_{ESR} - 1}$$

- Poner el segundo polo a la mitad de la frecuencia de switching. Poner también el segundo cero en el polo doble del filtro de salida, obteniendo las siguientes ecuaciones:

- $R_3 = \frac{R_1}{\frac{F_{SW}}{2 * F_{LC}} - 1}$

- $C_3 = \frac{1}{\pi * R_3 * F_{SW}}$

- La ganancia de la compensación debe ser comparada con la ganancia de lazo abierto del amplificador de error. La ganancia de la compensación no deberá exceder dicha ganancia del amplificador de error dado que ésta es el factor delimitante de la compensación.

- La ganancia de la red de compensación está dada por la siguiente ecuación:

- $Gain_{Comp} = \frac{R_1 + R_3}{R_1 * R_3 * C_1} * \frac{\left(s + \frac{1}{R_2 * C_2}\right) * \left(s + \frac{1}{(R_1 + R_3) * C_3}\right)}{s * \left(s + \frac{C_1 + C_2}{R_2 * C_1 * C_2}\right) * \left(s + \frac{1}{R_3 * C_3}\right)}$

La ganancia en decibelios, así como la fase se muestra en la siguiente gráfica de Bode:

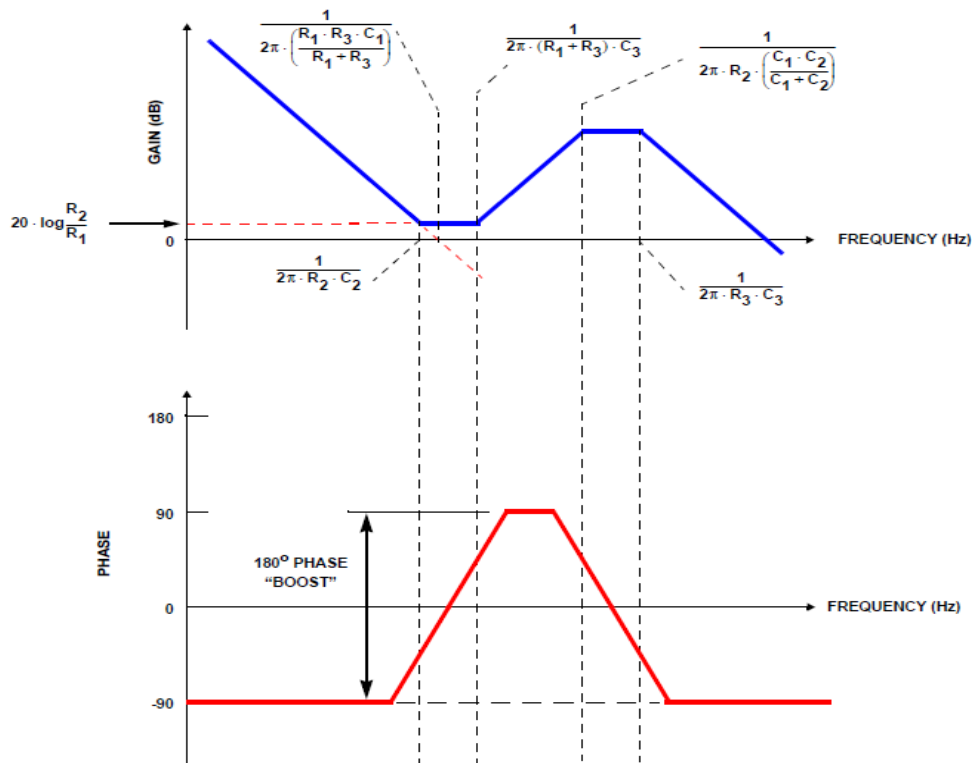


Figura 15: ganancia asintótica de Bode.

Sistema completo.

Integrando las tres partes ya mencionadas, se cuenta con el regulador de voltaje completo:

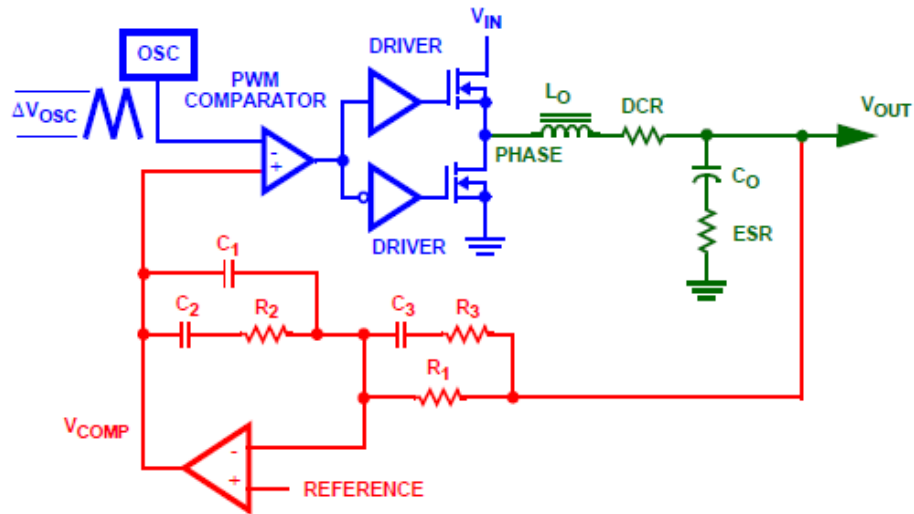


Figura 16: regulador completo.

La ganancia está dada por la siguiente ecuación:

$$Gain_{System} = \frac{R_1 + R_3}{R_1 * R_3 * C_1} * \frac{\left(s + \frac{1}{R_2 * C_2}\right) * \left(s + \frac{1}{(R_1 + R_3) * C_3}\right)}{s * \left(s + \frac{C_1 + C_2}{R_2 * C_1 * C_2}\right) * \left(s + \frac{1}{R_3 * C_3}\right)} * \frac{V_{in}}{\Delta V_{osc}} * \frac{1 + s * ESR * C_{out}}{1 + s * (ESR + DCR) * C_{out} + s^2 * L_{out} * C_{out}}$$

Especificaciones del regulador.

El regulador de voltaje cumplirá con las siguientes especificaciones:

- El rango del voltaje de entrada es de 9V a 15V.
- El voltaje de salida debe de ser menor a 1.5V con una variación máxima de tolerancia, rizado y transitorio de 5%.
- La corriente máxima de salida es de 10A, con un escalón máximo de 7A.
- Frecuencia fija de 500kHz.

Inductor.

Para escoger un valor inicial del inductor, se parte de la siguiente fórmula:

$$\Delta I = \frac{V_{OUT}}{L * f_{SW}} \left(1 - \frac{V_{OUT}}{V_{IN}} \right)$$

Donde ΔI es la variación máxima de corriente en el inductor, que para este diseño es de 7A.

$$L = \frac{1.5V}{7A * 500KHz} \left(1 - \frac{1.5}{15} \right) = 0.386\mu H$$

Puesto que la frecuencia es fija y se necesita una reducción en el rizado de corriente en el inductor, se incrementó su valor a $1.3\mu H$ con un DCR de $2.3m\Omega$.

Capacitor.

El capacitor de salida está determinado por la resistencia en serie equivalente ESR para minimizar el rizado del voltaje de salida durante los cambios de corriente en la carga. La fórmula del voltaje es la siguiente:

$$\Delta V_{OUT} \leq \Delta I_L \left(ESR + \frac{1}{8 * f_{SW} * C_{OUT}} \right)$$

Para ΔV_{OUT} de 75mV y despejando para C_{OUT} :

$$C_{OUT} = \frac{\Delta I_L}{(8 * f_{SW}) * (\Delta V_{OUT} - \Delta I_L * ESR)}$$

Si se considera que la capacitancia equivalente utilizará más de 1 capacitor para reducir el ESR que se estima que es de $3m\Omega$:

$$C_{OUT} = \frac{7A}{(8 * 500KHz) * (75mV - 7A * 3m\Omega)} = 33.94\mu F$$

Para tener un rizado de voltaje pequeño durante el transitorio se incrementó el tamaño de la capacitancia de salida por dos capacitores de $470 \mu F$ con un ESR total de $1.5 m\Omega$. Para el caso del capacitor de entrada, este debe tener un ESR bajo para minimizar la caída de voltaje cuando los MOSFETs se encienden y su capacitancia debe ser lo suficientemente grande para poder mantener el voltaje de entrada hasta que la fuente pueda responder. Se escoge un capacitor de tipo electrolítico de $220 \mu F$ con un voltaje por lo menos 1.4 veces más grande que el voltaje máximo de entrada, es decir a 25V.

Controlador.

El controlador a utilizar es el modelo LTC3775 de Linear Technology, el cual cuenta con las siguientes características:

- High Frequency Synchronous Step-Down Voltage Mode DC/DC Controller
- Amplio rango de voltaje de entrada, de 4.5V hasta 38V.
- Compensación tipo feed-forward con un amplificador de error de gran ancho de banda para una respuesta rápida en modo transitorio.
- Una salida de voltaje de 0.6 a 0.8 veces el voltaje de entrada.
- Un voltaje de referencia 0.6V.
- Limitador de corriente programable.
- Censado de corriente.
- Soft-start programable.
- La frecuencia de oscilación es de 500KHz.

El controlador tiene un voltaje de oscilación de PWM de 1.25V. La ganancia de la etapa de modulación está dada por la siguiente ecuación:

$$GANANCIA_{MOD} = \frac{V_{in}}{\Delta V_{osc}}$$

Además, utiliza un amplificador operacional interno para censar el voltaje de salida V_{OUT} . En estado estable la ganancia de salida del regulador es:

$$V_{OUT} = V_{REF} * \left(1 + \frac{R_A}{R_B} \right)$$

Con $R_A = 2.05K\Omega$ y $R_B = 1.37K\Omega$ y el voltaje de referencia interno 0.6V, se tiene que:

$$V_{OUT} = 0.6V * \left(1 + \frac{2.05K\Omega}{1.37K\Omega} \right) = 1.4978V \approx 1.5V$$

El esquemático del controlador se muestra en la siguiente figura:

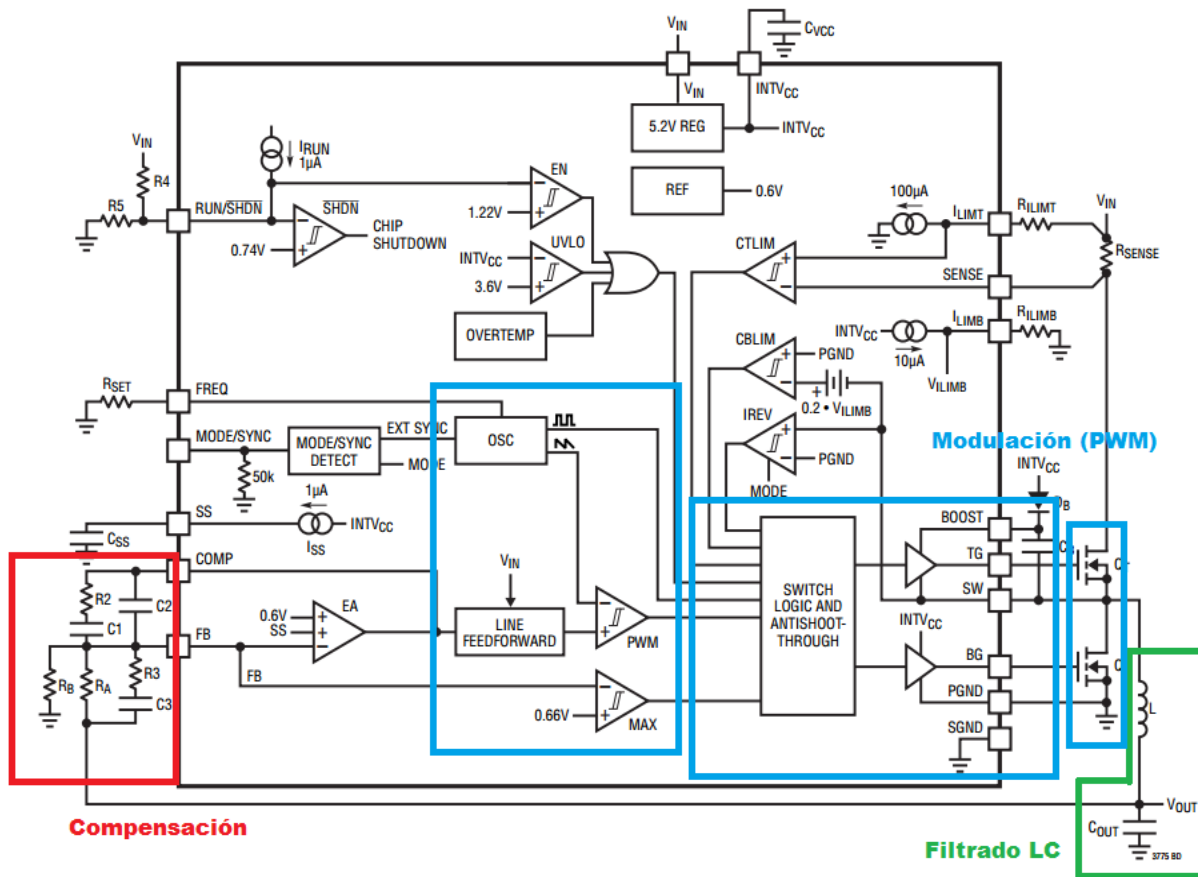


Figura 17: esquemático del controlador.

Compensador.

La red de compensación utiliza dos ceros para dar un boost a la fase del sistema de 180°, con el fin de contrarrestar los efectos de la resonancia ocasionados por el doble polo del filtro LC. Un margen de fase de mayor a 45° representa un sistema estable.

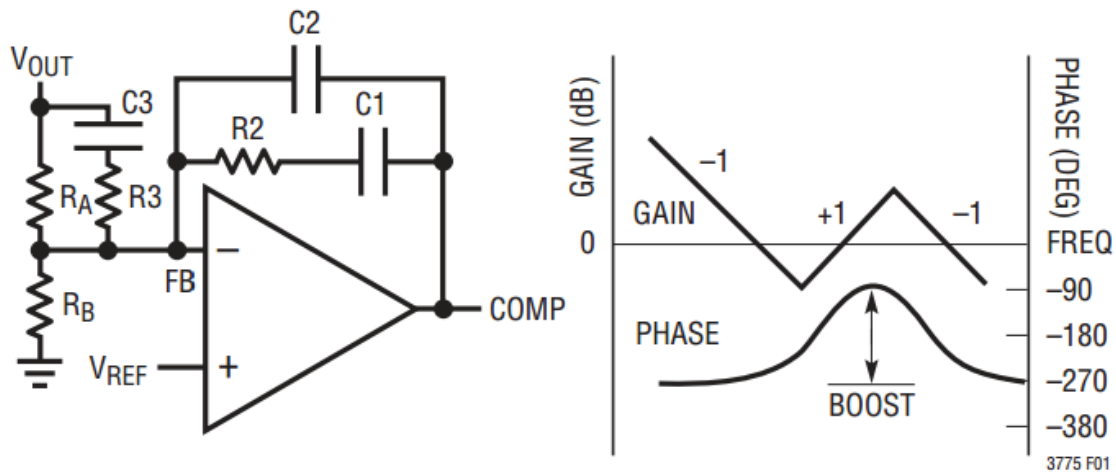


Figura 18: compensador.

Una vez elegida la relación de las resistencias de feedback \$R_A\$ y \$R_B\$, basándose en \$R_A\$ se puede calcular \$R_2\$. La relación \$\frac{R_2}{R_A}\$ recorrerá la ganancia de lazo abierto hasta el ancho de banda deseado.

Para un ancho de banda deseado \$DBW=90Kz\$, \$\Delta V_{OSC}=1.25V\$ y \$R_A=2.05K\Omega\$, se tiene que:

$$R_2 = \left(\frac{DBW}{F_{LC}} \right) * \frac{\Delta V_{OSC}}{V_{IN}} * R_1$$

Donde \$F_{LC}\$ es la frecuencia de resonancia dada por:

$$F_{LC} = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{1.3\mu H * 940\mu F}} = 4.53KHz$$

$$R_2 = \left(\frac{90KHz}{4.53KHz} \right) * \frac{1.25V}{15V} * 2.05K\Omega = 3.394 K\Omega \approx 3.4K\Omega$$

Si se coloca el primer cero en $F_{Z1} = F_{LC}$, el valor de C_1 está dado por la siguiente relación:

$$C_1 = \frac{1}{2\pi * R_2 * F_{LC}} = \frac{1}{2\pi * 3.4K\Omega * 4.53KHz} = 10.3nF \approx \mathbf{0.01\mu F}$$

Colocando el primer polo en la frecuencia f_{ESR} de 0.079MHz, se tiene para C_2 :

$$C_2 = \frac{C_1}{2\pi * R_2 * C_1 * F_{ESR} - 1} = \frac{10.3nF}{2\pi * 3.4K\Omega * 10.3nF * 0.079MHz - 1} = 629pF$$

El segundo polo se coloca a la mitad de la frecuencia de conmutación, obteniendo así los siguientes valores para el resto de los componentes discretos:

$$F_{P2} = F_{sw} = \frac{500KHz}{2} = 250KHz.$$

$$R_3 = \frac{R_A}{\frac{F_{sw}}{2 * F_{LC}} - 1} = \frac{2.05K\Omega}{\frac{250KHz}{4.53KHz} - 1} = 37.8\Omega$$

$$C_3 = \frac{1}{\pi R_3 * 250KHz} = \frac{1}{\pi * 37.8\Omega * 250KHz} = 34.nF$$

Simulación.

La simulación del regulador de voltaje fue realizada mediante el software de LT Spice, cuyo esquemático se muestra en la siguiente figura:

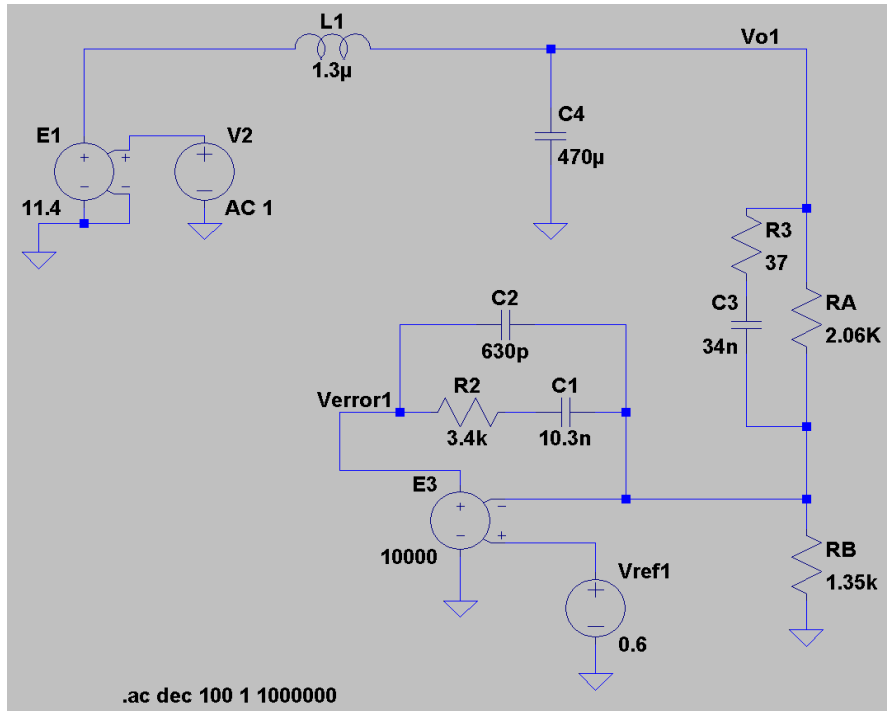


Figura 19: esquemático en LT Spice.

La primera simulación se realizó con los datos calculados de las fórmulas anteriores, lo cual da como resultado un margen de fase bajo ($\sim 28^\circ$).

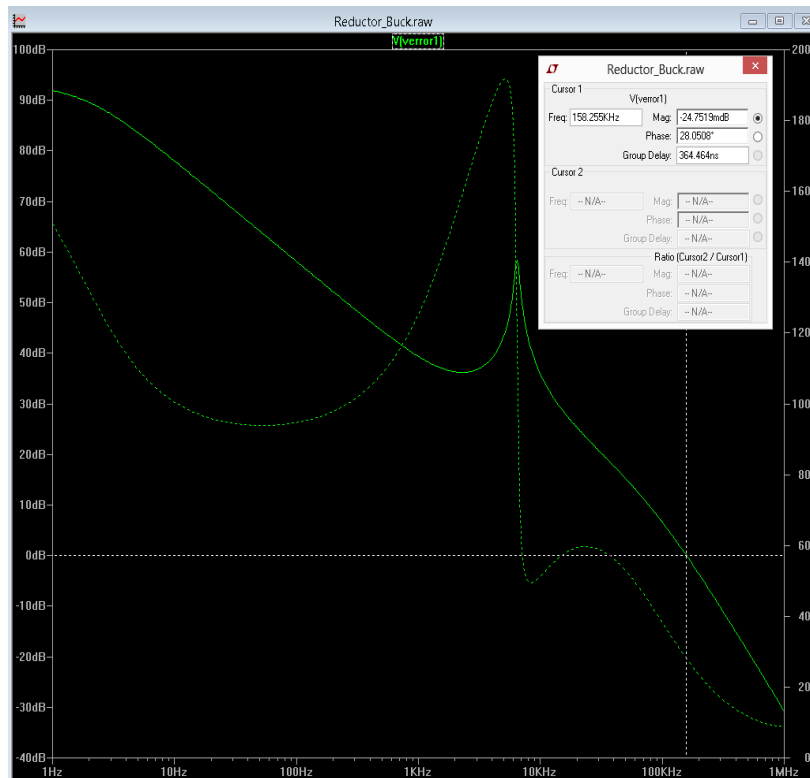


Figura 20: simulación de AC del regulador con datos calculados.

Posteriormente, ajustando los valores de $C3 = 22nF$, $R3 = 45 \text{ Ohms}$, se logra obtener un margen de fase de $\sim 57^\circ$.

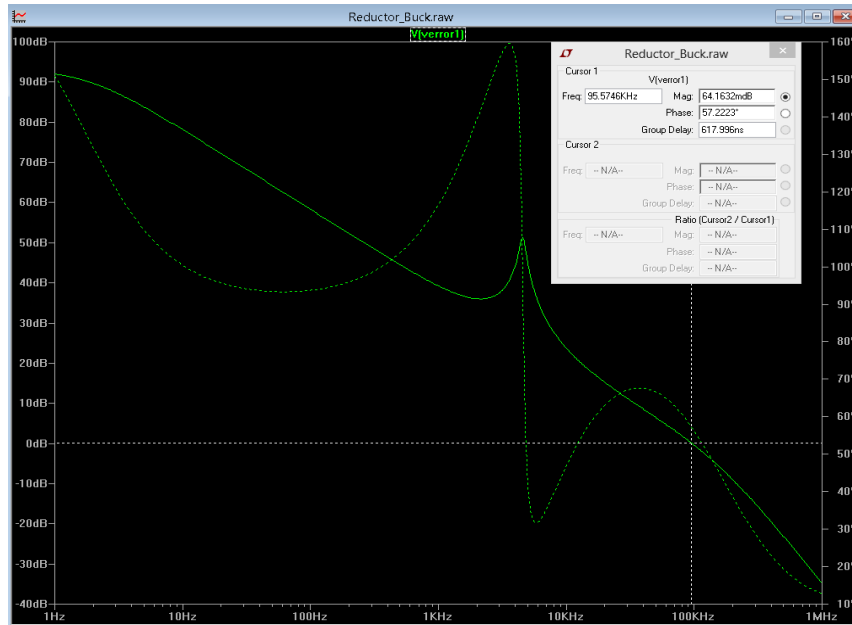


Figura 21: simulación de AC del regulador después del ajuste.

La simulación del transitorio se observa en la siguiente imagen, en donde la salida se muestra con la línea azul, teniendo a la salida 1.5V, con un Istep de 7A y un voltaje Pk-Pk de $\sim 28mV$.

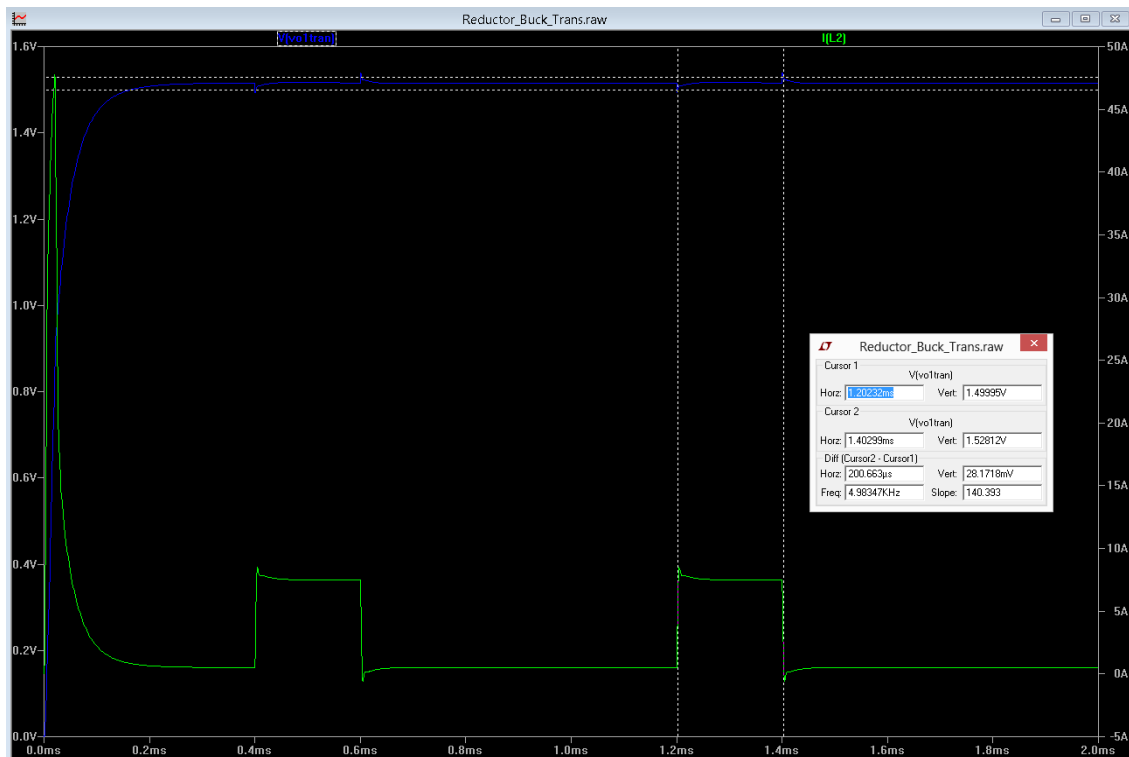


Figura 22: voltaje a la salida del regulador.

Así también, se implementó el esquemático del diseño utilizando la herramienta de Orcad, el cual se muestra en la siguiente figura.

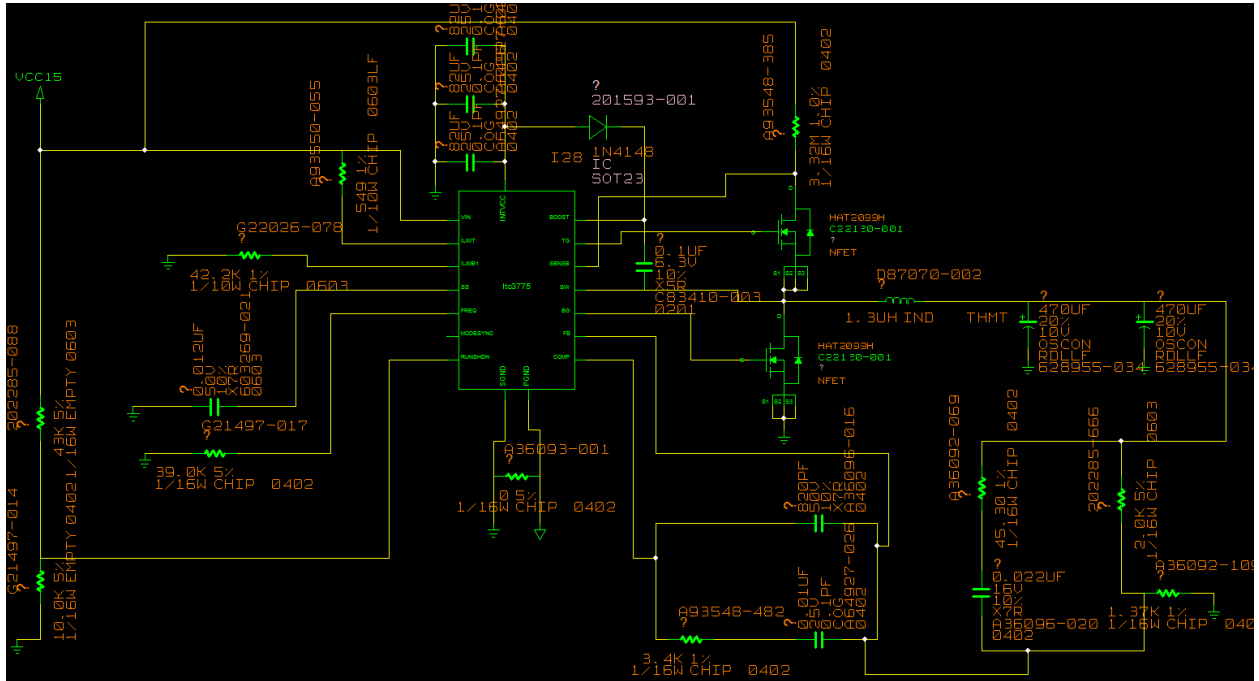


Figura 23: esquemático del regulador de voltaje.

Costos y materiales.

Finalmente, se hizo la búsqueda de componentes que conforman la implementación del regulador. Cabe mencionar que esto se hizo sólo con el propósito de tener una idea sobre el valor aproximado del diseño, sin llegar a hacer el diseño físicamente. La siguiente tabla muestra el fabricante, número de parte, descripción, cantidad, precio y proveedor de cada componente utilizado en el diseño.

Fabricante	Part No.	Descripción	Qty	Precio	Consult	Stock	Proveedor
Nichicon	UPW1V820MED	CAP ALUM 82UF 35V 20% RADIAL	3	\$0.99	23/11/14	6054	Digikey
Nichicon	UMV1H0R1MFD1TP	CAP ALUM 0.1UF 50V 20% RADIAL	1	\$0.12	23/11/14	2000	Digikey
Panasonic	ECA-1EM471B	CAP ALUM 470UF 25V 20% RADIAL	2	\$0.28	23/11/14	27000	Digikey
Murata Electronics	GRM155R71E103KA01D	CAP CER 10000PF 25V 10% X7R 0402	2	\$0.2	23/11/14	3623671	Digikey
Murata Electronics	GRM155R71C223KA01D	CAP CER 0.022UF 16V 10% X7R 0402	1	\$0.1	23/11/14	391096	Digikey
Murata Electronics	GRM033R71E821KA01D	CAP CER 820PF 25V 10% X7R 0201	1	\$0.1	23/11/14	93404	Digikey
Panasonic	ERJ-1TYF271U	RES 270 OHM 1W 1% 2512 SMD	2	\$1.92	23/11/14	26800	Digikey
Panasonic	ERJ-3EKF45R3V	RES 45.3 OHM 1/10W 1% 0603	1	\$0.1	23/11/14	88130	Digikey
Panasonic	ERJ-2GE0R00X	RES 0.0 OHM 1/10W JUMP 0402	1	\$0.1	23/11/14	2795572	Digikey
Samsung	RC1005F2102CS	RES 21K OHM 1/16W 1% 0402	2	\$0.2	23/11/14	108225	Digikey
Rohm	MCR03ERTF3902	RES 39K OHM 1/10W 1% 0603	1	\$0.1	23/11/14	23970	Digikey
Panasonic	ERJ-3EKF4322V	RES 43.2K OHM 1/10W 1% 0603	1	\$0.1	23/11/14	77136	Digikey
Rohm	MCR01MRTF1002	RES 10K OHM 1/16W 1% 0402	1	\$0.1	23/11/14	723759	Digikey
Panasonic	ERJ-3EKF3401V	RES 3.4K OHM 1/10W 1% 0603	1	\$0.1	23/11/14	63010	Digikey
Panasonic	ERJ-3EKF1371V	RES 1.37K OHM 1/10W 1% 0603	1	\$0.1	23/11/14	46866	Digikey
Panasonic	ERJ-2RKF2051X	RES 2.05K OHM 1/10W 1% 0402	1	\$0.1	23/11/14	34883	Digikey
Panasonic	ERJ-3GEYJ305V	RES 3M OHM 1/10W 5% 0603	1	\$0.1	23/11/14	85465	Digikey
Micro Commercial Co	1N4742A-TP	DIODE ZENER 12V 1W DO41	1	\$0.25	23/11/14	14765	Digikey
Murata Power Solutions Inc	38S132C	FIXED IND 1.3UH 10.5A 5 MOHM	1	\$1.77	23/11/14	1250	Digikey
Renesas Electronics	RJK0305DPB-00#J0	MOSFET N-CH 30V 30A LPAK	2	\$3.04	23/11/14	5909	Digikey
Linear Technology	LTC3775EMSE#PBF	IC REG CTRLR BUCK PWM VM 16-MSOP	1	\$3.82	23/11/14	331	Digikey

Costo total del diseño: \$13.69 USD.

Codificación de canal

Contenidos

<i>Objetivo:</i>	30
<i>Realizar un sistema de comunicación básico</i>	30
<i>Simulación del sistema sin codificación</i>	30
<i>Codificador tipo Hamming</i>	31
<i>Simulación del sistema con codificación</i>	32
<i>Resultados</i>	33
<i>Conclusiones</i>	34

Objetivo:

- Comprender el funcionamiento de un codificador de canal en un sistema de comunicación digital.
- Realizar un sistema de comunicación que contenga la función de codificación de canal y comparar los resultados respecto al sistema sin codificador.

Actividades:

Realizar en la herramienta de simulación MATLAB las operaciones siguientes:

Realizar un sistema de comunicación básico

Como el mostrado en la figura 1. El canal binario simétrico permite configurar directamente la probabilidad de error que se desea sobre los datos binarios a su entrada.

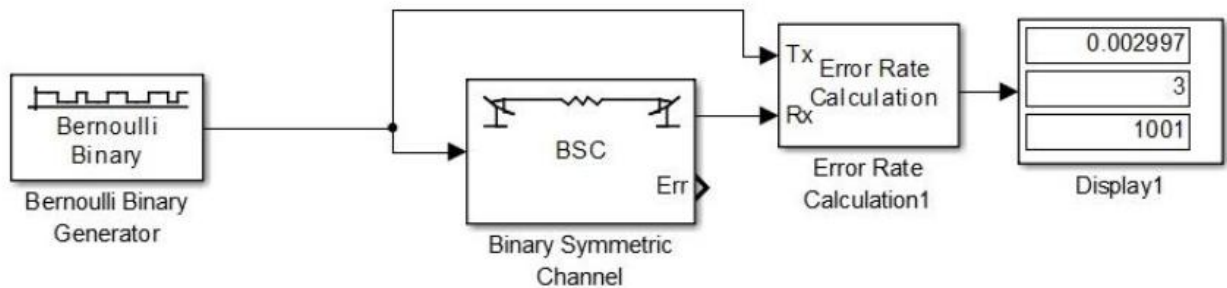


Figura 24: sistema de comunicación digital con canal binario simétrico

Simulación del sistema sin codificación

Ejecutar la simulación del sistema en varias ocasiones para diversos valores de probabilidad de error del canal, generando por lo menos un total de 1000 bits de información. Utilizar al menos los valores siguientes de probabilidad: 0.005, 0.01, 0.015, 0.025, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3. Constatar que la probabilidad de error obtenida es prácticamente la configurada.

Como se expresa en el punto 1, el código consta de tres bloques, los cuales son un generador binario, un canal simétrico y un cálculo de error. La simulación se ejecuta en varias ocasiones con los valores de probabilidad de error ya mencionados mediante un ciclo **for**. El resultado para cada probabilidad es guardado en la matriz **nDataM**. Finalmente para obtener el cálculo de error, para cada resultado guardado en la matriz **nDataM** se tendrá un bit error rate distinto; mediante el uso de dos ciclos **for** se obtienen dichos bit error rates los cuales serán guardados en la matriz **BER**. El código completo del script es el siguiente:

```

nDataM=[];
BER=[];
prob = [0.005 0.01 0.015 0.025 0.05 0.1 0.15 0.2 0.25 0.3];

%%%%%%%%%Binary Generator%%%%%%%%%
data=round(rand(1,1024));
dL=length(data);
%%%%%%%%%

%%%%%%%%%Binary symmetric channel%%%
for c=1:length(prob)
    ndata = bsc(data,prob(c));
    nDataM(c,:) = ndata;
end
%%%%%%%%%

%%%%%%%%%Error Rate Calculation%%%%%%%%%
for c=1:length(prob)
    value=0;
    for d=1:dL;
        if (nDataM(c,d)~=data(d))
            value=value+1;
        end
    end
    BER = [BER (value/dL)*100];
end
%%%%%%%%%

```

Los bit error rates obtenidos en una simulación para cada probabilidad de error fueron los siguientes:

P = 0.005	P = 0.01	P = 0.015	P = 0.025	P = 0.05	P = 0.1	P = 0.15	P = 0.2	P = 0.25	P = 0.3
0.58	1.66	0.97	2.92	5.17	11.13	14.94	20.89	25.39	31.73

Codificador tipo Hamming

Ampliar el sistema de la figura 1 añadiendo un codificador y un decodificador de canal del tipo Codificador de Bloques de Hamming. La configuración por omisión de este bloque es de un código (7,4), figura 2. Es necesario considerar que la entrada al codificador debe ser una trama de la longitud correspondiente a la longitud de la palabra de información (4), por lo que la fuente de información debe de tener configurada su salida de la misma manera (“frame”).

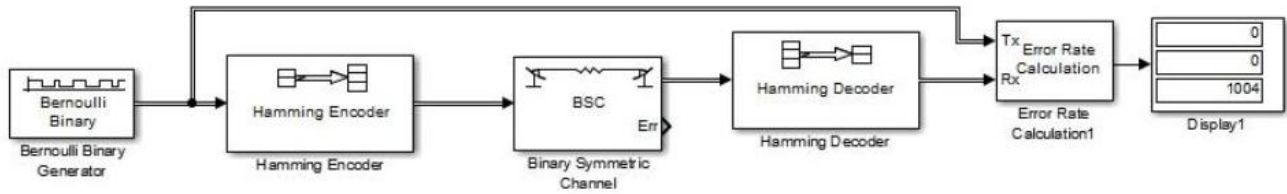


Figura 25: sistema de comunicación digital con canal binario simétrico y codificador de canal de Hamming.

Simulación del sistema con codificación

Ejecutar varias simulaciones del sistema utilizando los mismos valores de probabilidad de error que los utilizados en el punto 2. Graficar los resultados de los dos casos de ejecución, con y sin codificador de canal (Probabilidad de error configurada en el canal vs. Probabilidad de error obtenida después del decodificador).

Partiendo del código anterior, es posible agregar el codificador y decodificador Hamming. Para lograr esto, basta con usar la función **encode**, la cual se encarga de codificar en Hamming los valores binarios proporcionados por el generador, recibiendo como parámetros los valores de **n** y **k**. El resultado es guardado en la variable **code**, como se muestra en el siguiente código:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Hamming Encoder %%%%%%%%%%%
for c=1:H_k:dL
    code = [code encode(data(c:c+3),H_n,H_k,'hamming/fmt',gfprimdf(3))];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

De igual forma, se sigue utilizando de igual manera el código del canal binario simétrico, guardando cada resultado de cada probabilidad de error en la matriz **nDataC**. Finalmente, para decodificar cada resultado, basta con utilizar la función **decode**, la cuál se aplicará para los 10 resultados, en donde se reciben como parámetros también los valores de **n** y de **k**.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Hamming Decoder %%%%%%%%%%%
for c=1:length(prob)
    dcode = [];
    for d=1:H_n:length(code)
        dcode = [dcode decode(nDataC(c,d:d+6),H_n,H_k,'hamming/fmt',gfprimdf(3))];
    end
    nDataD(c,:) = dcode;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

El bit error rate es calculado de la misma forma que en el código anterior. El código completo del script es el siguiente:

```

nDataC = [];
nDataD = [];
BER = [];

```



```

code = [];
prob = [0.005 0.01 0.015 0.025 0.05 0.1 0.15 0.2 0.25 0.3];
H_n = 7;
H_k = 4;

%%%%%%%%%Binary Generator%%%%%%%%%
data=round(rand(1,1024));
dL=length(data);
%%%%%%%%%

%%%%%%%%%Hamming Encoder%%%%%%%%%
for c=1:H_k:dL
    code = [code encode(data(c:c+3),H_n,H_k,'hamming/fmt',gfprimdf(3))];
end
%%%%%%%%%

%%%%%%%%%Binary symmetric channel%%%
for c=1:length(prob)
    ndata = bsc(code,prob(c));
    nDataC(c,:) = ndata;
end
%%%%%%%%%

%%%%%%%%%Hamming Decoder%%%%%%%%%
for c=1:length(prob)
    dcode = [];
    for d=1:H_n:length(code)
        dcode = [dcode decode(nDataC(c,d:d+6),H_n,H_k,'hamming/fmt',gfprimdf(3))];
    end
    nDataD(c,:) = dcode;
end
%%%%%%%%%

%%%%%%%%%Error Rate Calculation%%%%%%%%%
for c=1:length(prob)
    value=0;
    for d=1:dL;
        if(nDataD(c,d)~=data(d))
            value=value+1;
        end
    end
    BER = [BER (value/dL)*100];
end
%%%%%%%%%

```

Resultados

La comparación de los errores obtenidos entre no usar codificación y usar codificación Hamming se muestra en la siguiente tabla:

	P = 0.005	P = 0.01	P = 0.015	P = 0.025	P = 0.05	P = 0.1	P = 0.15	P = 0.2	P = 0.25	P = 0.3
w/o coding	0.58	1.66	0.97	2.92	5.17	11.13	14.94	20.89	25.39	31.73
Hamming	0	0	0.09	1.17	2.14	6.44	13.08	20.5	26.66	34.17

Finalmente, la siguiente gráfica muestra la comparación de los dos métodos, graficando la probabilidad de error configurada en el canal contra la probabilidad de error obtenida después del decodificador.

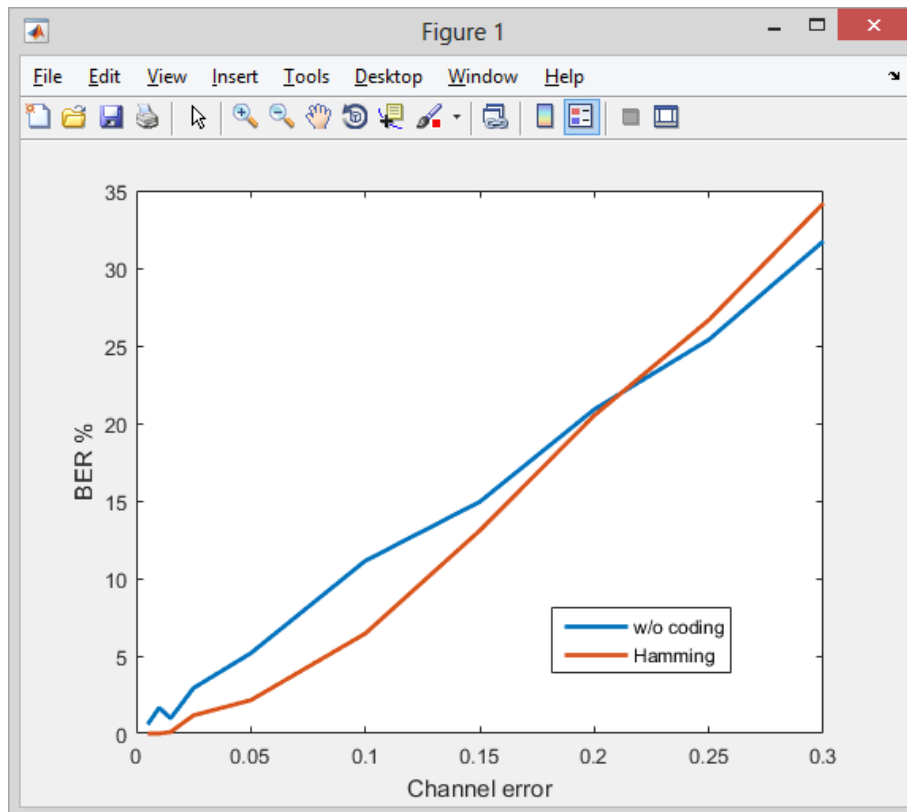


Figura 26: prob. de error en el canal VS prob. de error obtenida después del decodificador.

Conclusiones

Analizar el comportamiento del sistema y los resultados, explicando las características particulares del codificador de Hamming. Justificar adecuadamente las conclusiones.

Al observar la tabla anterior y la gráfica de la figura 3 es claramente visible que la codificación Hamming es muy útil para reducir la probabilidad de error obtenida después del decodificador. Sin embargo, cabe destacar que esto ocurre sólo para un rango de las probabilidades de error en el canal, dado que cuando esta probabilidad llega a volverse muy grande, entonces el resultado de la codificación/decodificación Hamming llegará a ser mayormente erróneo a que si no se utilizara ningún tipo de codificación. Esto se debe principalmente a que las distancias mínimas del código Hamming se tornan más cortas por el simple hecho de la adición de ruido, corrompiendo los datos; a menor distancia, menor capacidad de detectar y corregir errores. Como conclusión, se vuelve a remarcar que, para probabilidades de error que no excedan de cierto límite en el canal de comunicación, es altamente recomendable utilizar la codificación Hamming.

Diseño de un PCB

Contenidos

<i>Creación de proyecto y librerías utilizando la herramienta Allegro Design Entry CIS</i>	<i>36</i>
<i>Creación de footprints y pads con Pad Designer, PCB editor y LP Calculator</i>	<i>45</i>
<i>Creación de esquemáticos</i>	<i>52</i>
<i>Diseño del layout</i>	<i>59</i>
<i>Conclusiones</i>	<i>66</i>

Creación de proyecto y librerías utilizando la herramienta Allegro Design Entry CIS

Allegro Design Entry CIS es una herramienta la cual tiene la utilidad de hacer proyectos creando librerías de símbolos, así como esquemáticos para cualquier tarjeta. A continuación se presentan los pasos para realizar un proyecto, así como para conocer las opciones con las que cuenta dicha herramienta.

1. Creación de proyecto.

Al abrir Allegro CIS, el usuario visualizará la siguiente ventana, con los menús File, View, Tools, Edit, Options, Window y Help. Para crear un nuevo proyecto, se deberá elegir la opción File > New > Project.

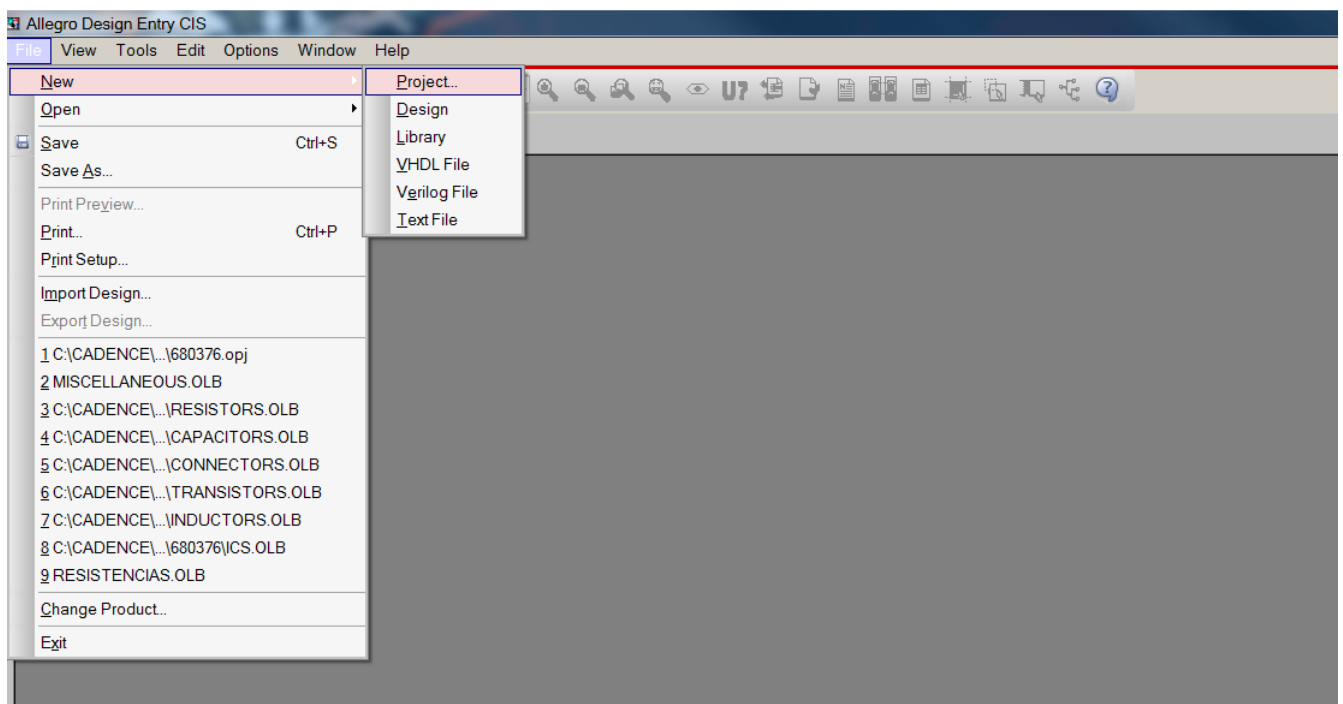


Figura 27: Allegro CIS.

Se deberá especificar el nombre del proyecto, y elegir la opción de PC Board Wizard. Después, dar click en OK. Una nueva ventana aparecerá preguntando por simulación, no necesaria por el momento; dar click en next. Otra ventana aparecerá preguntando si se desean agregar librerías existentes al proyecto. Dado que se crearán las librerías por el usuario, no hay necesidad de agregar esto, por lo que se dará click en finish.

Una vez que se ha creado el proyecto, al mismo tiempo se habrá creado un diseño con el mismo nombre que se le dio proyecto. Uno o más diseños pueden estar dentro del mismo proyecto, de tal manera que cada diseño pertenezca a una tarjeta distinta, pero que dichas tarjetas estén relacionadas de cierta forma. Un diseño nuevo puede crearse en la opción File > New > Design.

2. Creación de librerías.

Creado el proyecto, el usuario podrá visualizar la pantalla mostrada a continuación.

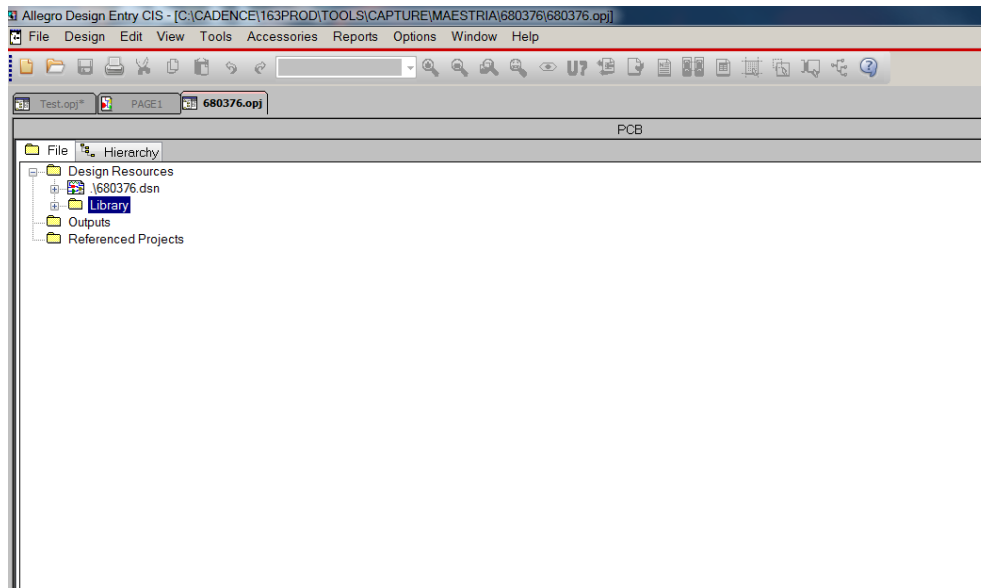


Figura 28: creación de un proyecto.

Una librería contendrá la información lógica de cada símbolo a agregar en un proyecto. Es decir, el número de pines con los que cuenta, tipo de pin (input, output, power, etc.), nombre de pin, footprint a utilizar, prefijos, etcétera. Para crear un símbolo nuevo, se debe seleccionar la opción File > New > Library. Una vez hecho esto, aparecerá un folder nuevo dentro de Library con el nombre library1. Si se desea renombrar la librería, se debe hacer click derecho sobre el folder, y seleccionar Save As.

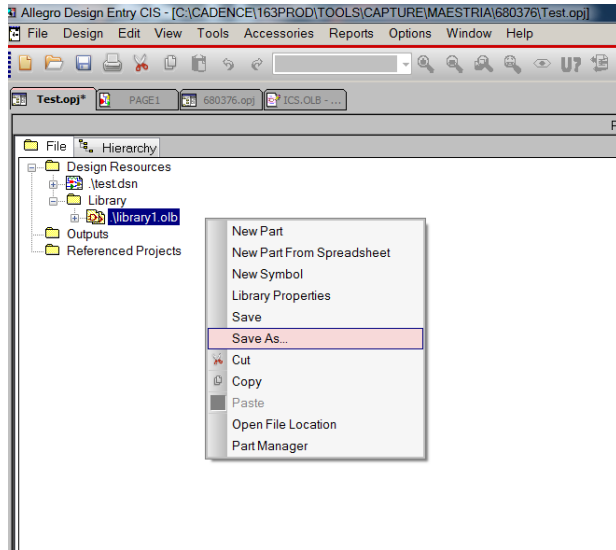


Figura 29: guardando una librería.

Para este tutorial, se han creado las siguientes librerías: resistors, capacitors, inductors, ICs, transistors, connectors, miscellaneous. Cada una de ellas puede contener uno o más símbolos (por ejemplo, la librería resistors puede incluir varias resistencias de distintos valores).

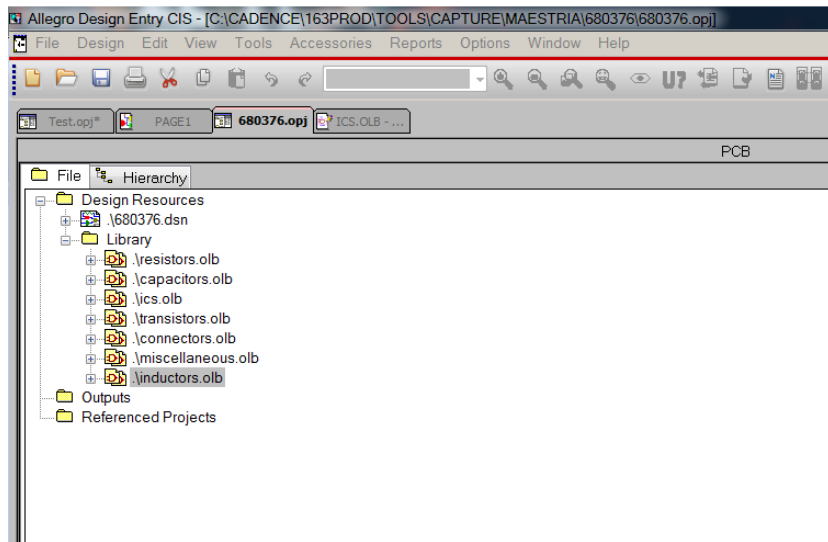


Figura 30: librerías para distintos componentes.

Para empezar a crear símbolos, se debe dar click derecho en la librería deseada y elegir la opción New Part (la opción New Part from Spreadsheet será explicada más adelante). Una ventana aparecerá pidiendo la siguiente información:

- Nombre del símbolo.
- Prefijo
- PCB Footprint.
- Partes por símbolo.
- Numeración del símbolo.

Para la creación del primer símbolo, un IC conteniendo dos FETs, se llenaran los datos mencionados con los siguientes valores:

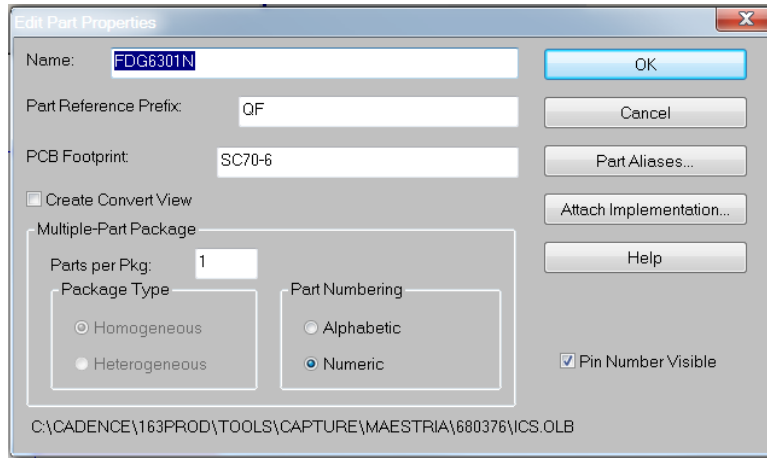


Figura 31: creación de un símbolo lógico.

El nombre del símbolo debe contener el número de parte. En el caso de resistencias, capacitores o inductores, es buena idea también agregar más especificaciones, tales como su valor de resistencia/capacitancia, etcétera. El prefijo es el nombre que tendrá el símbolo una vez agregado al esquemático, seguido de una numeración, dependiendo de la cantidad que se llegue a utilizar el mismo símbolo. La opción Parts per Package permite al usuario definir si un mismo símbolo contará con partes replicadas. Como ejemplo, considérese un integrado con cuatro compuertas AND; se replicarán cuatro veces dichas compuertas las cuales pertenecerán al mismo símbolo, la única diferencia entre compuertas será el número de pin. Para este ejemplo, solo se utilizará 1 parte por paquete. Una vez que se haya llenado toda esta información, se procederá a dar click en OK.

A continuación, se deberá crear el dibujo del símbolo. El usuario visualizará un lienzo desplegando solamente un cuadro interlineado, conteniendo el prefijo ya definido anteriormente.

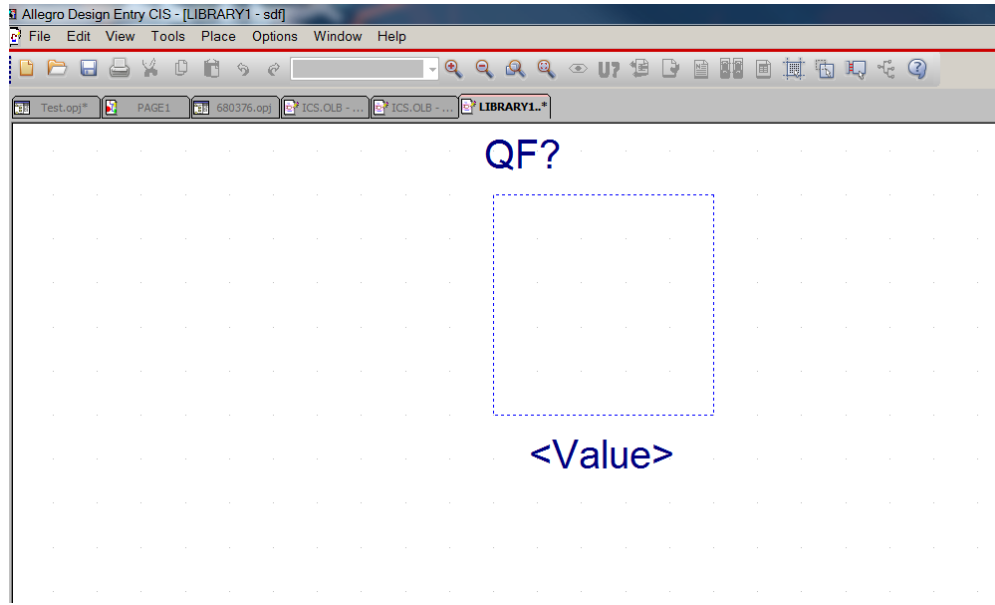


Figura 32: dibujo del símbolo lógico.

Se debe proceder a editar la etiqueta Value, dando doble click sobre ella y poniendo como valor también el número de parte.

En la parte derecha de la pantalla, el usuario visualizará varias herramientas necesarias para la creación del dibujo del símbolo. Por el momento, las opciones principales a usar serán Place Rectangle, Place Eclipse, Place Line y Place Pin.

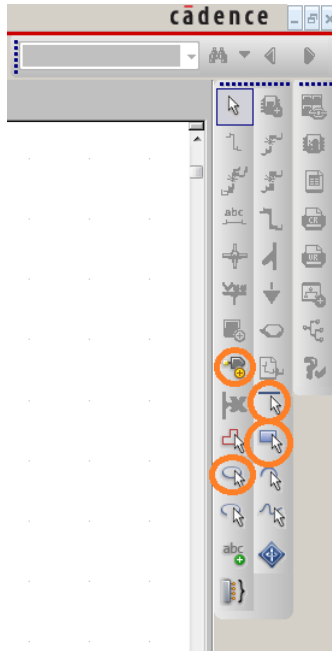


Figure 33: menú de opciones.

Para realizar el dibujo de un integrado de seis pines, lo usual es agregar un rectángulo dentro del cuadro interlineado con la opción Place Rectangle (se recomienda que tanto el rectángulo como el cuadro interlineado tengan las mismas dimensiones). Una vez hecho esto, se procederá a agregar los seis pines del integrado utilizando la opción Place Pin, la cual pedirá la siguiente información.

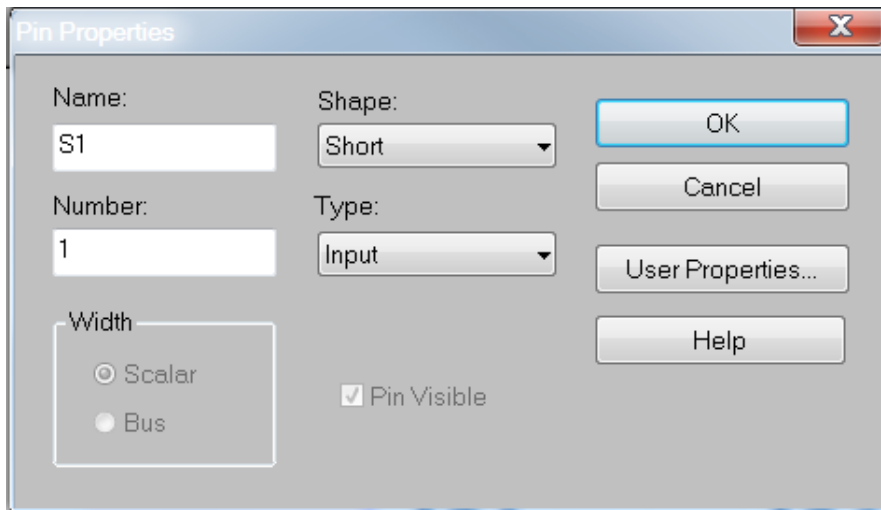


Figura 34: agregando pins a un símbolo.

Todo pin debe llevar un cierto nombre y número, dicha información puede ser obtenida de la hoja de datos del componente. Al mismo tiempo, debe definirse de qué tipo de pin se trata, ya sea input, output, power, etc. Dentro de la opción shape, se recomienda elegir la opción Short, ya que esto facilita el realizar las conexiones en los esquemáticos. Después de haber puesto esta información y dar click en OK,

el usuario podrá elegir con el mouse la ubicación deseada del pin en el borde del cuadro interlineado del lienzo. Una vez agregados los seis pines, el símbolo deberá lucir parecido al mostrado en la siguiente imagen.

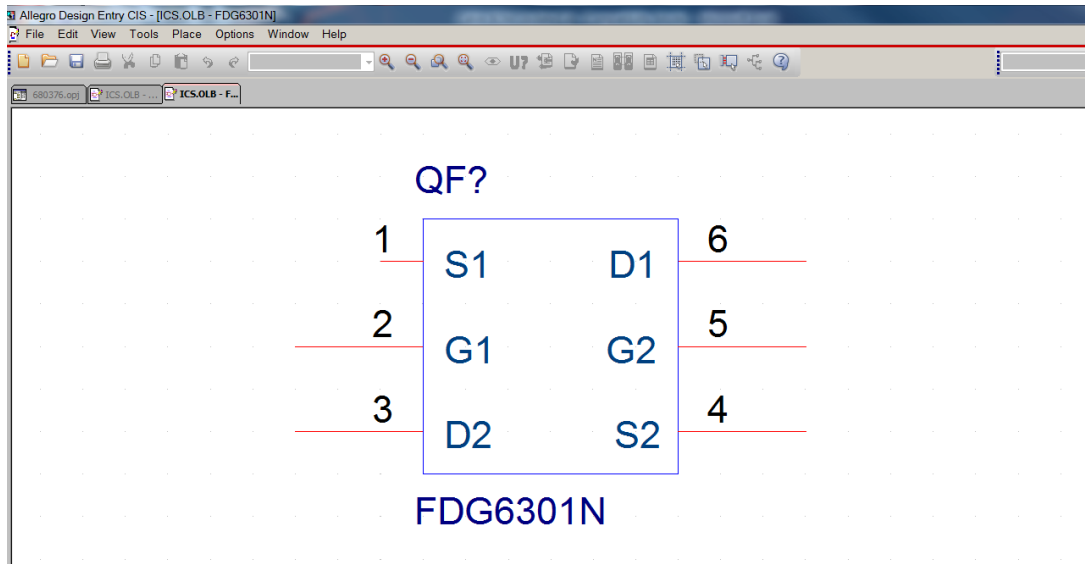


Figure 35: símbolo lógico creado.

De esta manera, se ha terminado de crear el primer símbolo lógico dentro de la librería deseada. Como se mencionó anteriormente, existe una forma alterna para crear símbolos al hacer click derecho en una librería, eligiendo la opción New Part From Spreadsheet.

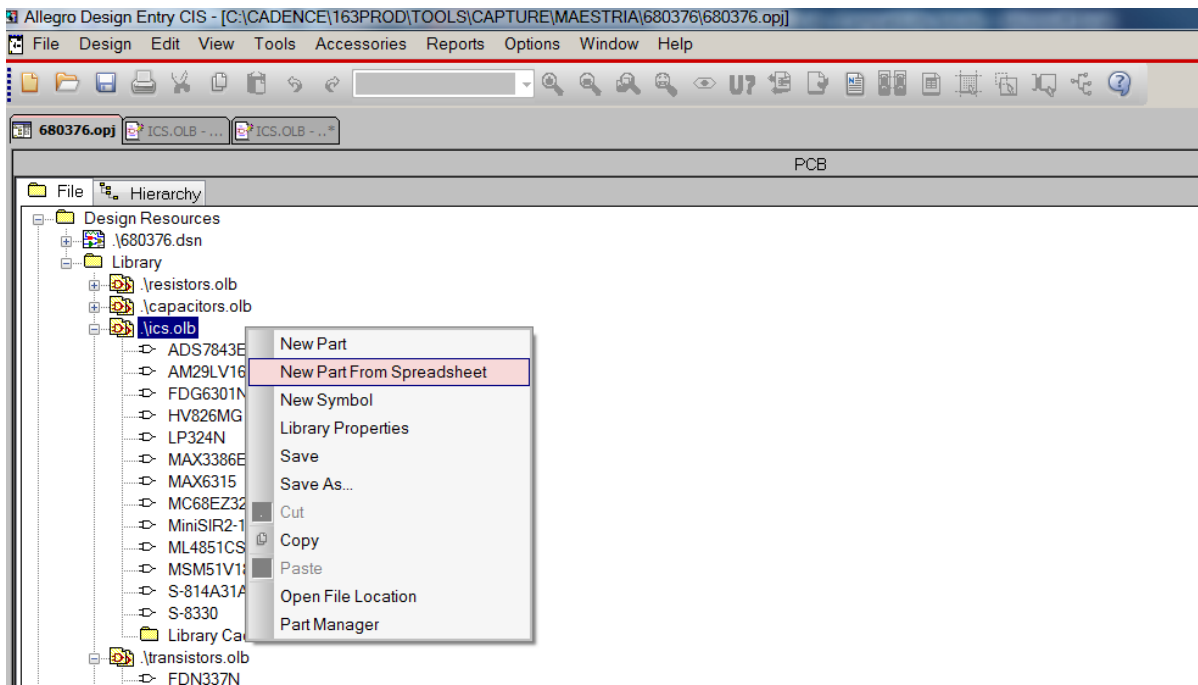


Figura 36: creación alterna de un símbolo.

Dicha opción tiene la ventaja de realizar con mayor rapidez símbolos que contengan un gran número de pines, tales como CPUs o memorias. Al haberla elegido, la siguiente ventana se desplegará.

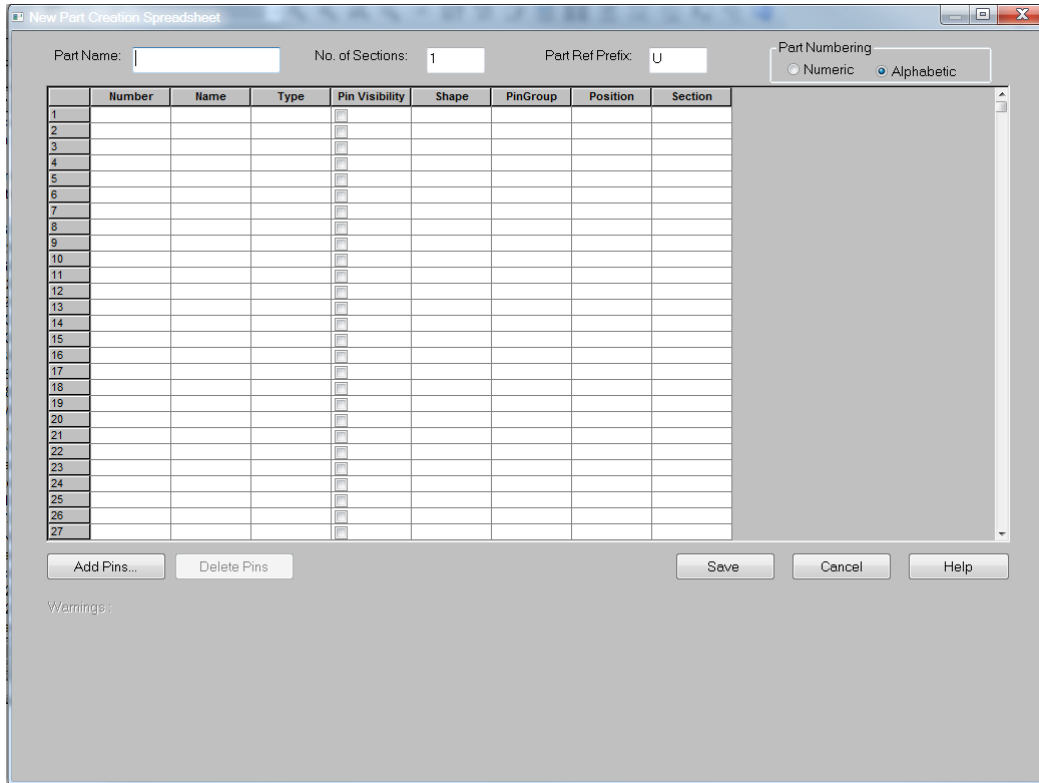


Figura 37: lista de pins.

Existen 6 columnas que serán de nuestro interés por ahora, las cuales son: Number, Name, Type, Pin Visibility, Shape y Position. Tal como sus nombres lo indican, en las columnas de Number y Name se debe poner el nombre y número de cada respectivo pin; en la columna Type se debe elegir de qué tipo de pin se trata (input, output, bidirectional, etc.); Pin Visibility será usado solamente para habilitar la visibilidad de los datos del pin. En la columna Shape se recomienda elegir Short, como se mencionó anteriormente, para facilitar la conexión en esquemáticos. Y finalmente, la columna de Position definirá la ubicación del pin con respecto al dibujo creado por la herramienta, ya sea izquierda, derecha, arriba o abajo. El dibujo del símbolo creado siempre resultará ser un rectángulo, y mientras más pines se agreguen, mayor será la dimensión de dicha figura. Finalmente, antes de dar click en el botón Save, se deberá especificar el Part Name y el prefijo en la parte superior de la ventana. Como ejemplo, en la figura siguiente se muestra una memoria con 50 pines creada fácilmente con esta herramienta.

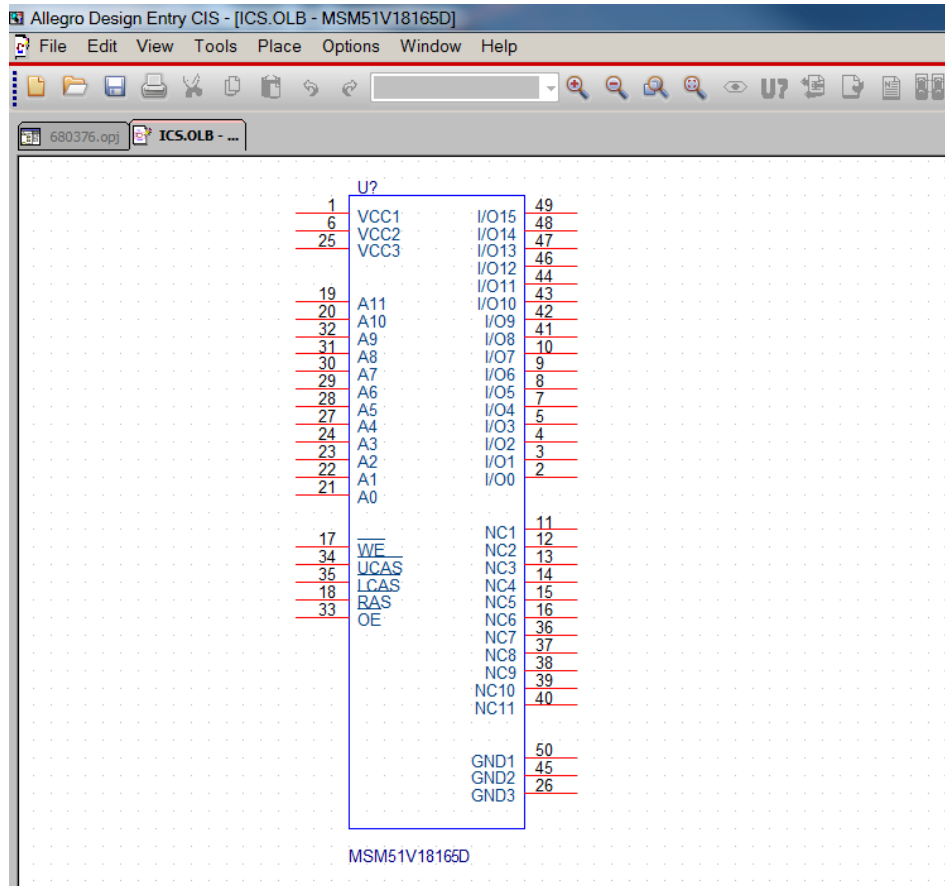


Figura 38: símbolo lógico creado con un número significativo de pins.

Los pines aún pueden ser movidos o desplazados en el dibujo como el usuario lo desee, utilizando el mouse, seleccionando los pines a mover y después arrastrándolos hacia su nueva ubicación.

Creación de footprints y pads con Pad Designer, PCB editor y LP Calculator

En esta sección se muestran los pasos para saber cómo crear símbolos físicos (footprints), así como los pads que éstos mismos utilizarán. Para llevar a cabo dicha tarea, se debe contar con las siguientes herramientas:

- Allegro PCB Editor.
- Pad Designer.
- LP Calculator.

Además de esto, para cada footprint a crear se debe contar con la respectiva hoja de datos del fabricante, ya que aquí se muestra toda la información referente a dimensiones físicas del componente, tales como largo, ancho, alto, número de pines, etcétera, esenciales para la creación del símbolo físico. Como ejemplo, en los siguientes pasos se mostrará a detalle cómo realizar el footprint para un controlador touchscreen.

1. Consulta de la hoja de datos del fabricante.

La hoja de datos puede ser obtenida en Internet para cada componente si se cuentan con datos tales como el fabricante y el número de parte. Para el caso del controlador touchscreen, dichos datos corresponden a Burr Brown, siendo el número de parte ADS7843E. Una vez que se cuenta dicha hoja, se debe proceder a buscar la sección de dimensiones del componente, la cual, para este caso, es la mostrada en la siguiente imagen:

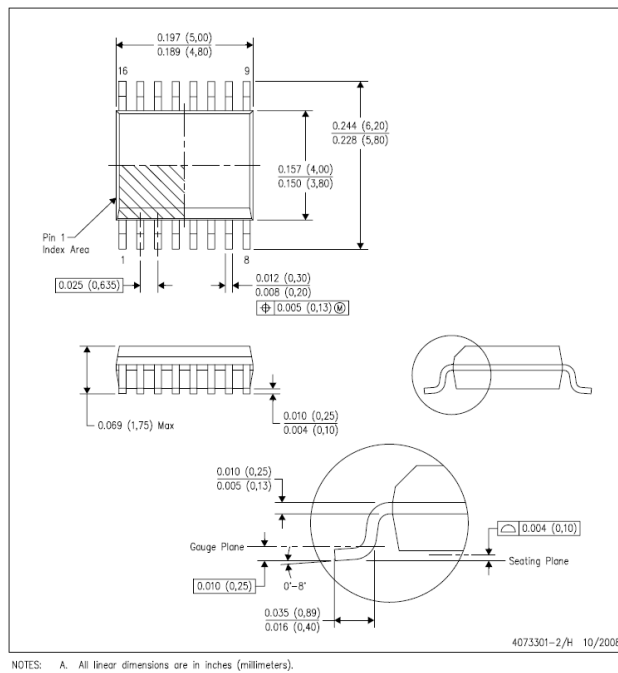


Figura 39: hoja de datos.

Se debe tener mucho cuidado en saber qué unidades son las manejadas en las dimensiones. Al momento de la creación del footprint, se deben seguir exactamente las mismas unidades, es decir, no hacer ningún tipo de conversión, ya que cualquier décima menospreciada puede impactar para lograr un diseño libre de fallas. Para la creación de cualquier símbolo, se debe contar con las siguientes dimensiones:

- Largo y ancho del cuerpo del componente (outline).
- Largo y ancho del pin.
- Pitch entre pines dentro de la misma hilera (distancia de centro a centro de un pin a otro pin contiguo).
- Pitch entre hileras de pines (distancia de centro a centro de una hilera de pines a otra).

Si se observa detenidamente la imagen de arriba, se podrán identificar las dimensiones mencionadas en los primeros tres puntos anteriores. Sin embargo, no se cuenta con el pitch entre hileras de pines; esta dimensión por lo general no suele estar especificada en gran parte de las hojas de datos, y desafortunadamente, es necesaria para la creación del símbolo al utilizar la herramienta PCB Editor. Cuando llegue a presentarse este caso, será necesario auxiliarse del programa LP Calculator, el cual nos permitirá obtener dicho dato, a cambio de hacerle saber todas las dimensiones mencionadas en los primeros tres puntos, además de las siguientes ya mostradas en la hoja de datos:

- Altura del componente.
- Distancia entre hileras de pines de orilla a orilla.
- Longitud de la parte del pin que hace contacto con la superficie (hasta el codo del pin).
- Distancia en altura desde la superficie del pin hasta el cuerpo del componente.

2. LP Calculator.

Una vez que se abre el programa, se debe dar click en el botón Calculate, para después elegir el tipo de componente del cual se desean obtener dimensiones (en este caso, un SOIC). Una vez seleccionado, aparecerá la siguiente pantalla:

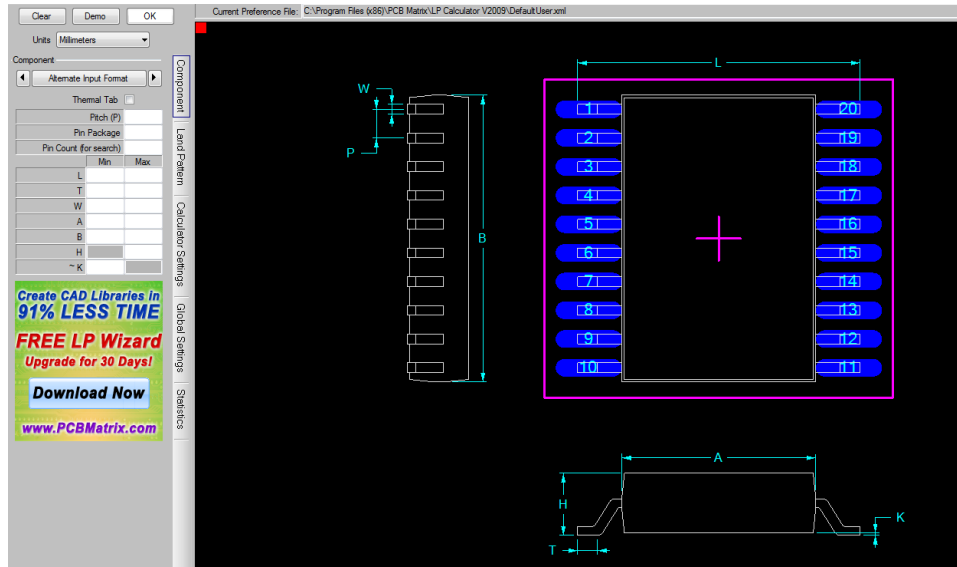


Figura 40: LP Calculator.

El usuario deberá poner todas las dimensiones ya mencionadas en el panel mostrado del lado izquierdo. Una vez hecho esto, se deberá dar click en el botón OK, y justamente en la sección Land Pattern (opción justamente a un lado del panel de la izquierda) se obtendrá como resultado el pitch de centro a centro entre hileras de pines. Cabe mencionar que una vez que se haya usado esta herramienta, se propondrán también nuevas dimensiones para el largo y ancho del pin (también mostradas dentro de Land Pattern), las cuales tomarán prioridad sobre las dimensiones ya vistas en la hoja de datos.

Siguiendo el ejemplo del controlador touchscreen, el panel de la izquierda deberá ser llenado de la siguiente forma:

Pitch (P)	0.635	
Pin Package	16	
Pin Count (for search)	16	
	Min	Max
L	5.80	6.20
T	0.40	0.89
W	0.20	0.30
A	3.80	4.00
B	4.80	5.00
H		1.75
~K	0.10	

Figura 41: dimensiones del footprint.

Una vez que se dé click en OK, se deberá seleccionar la opción Land Pattern, en donde se podrán visualizar las nuevas dimensiones sugeridas para el largo y ancho del pin, así como el pitch entre hileras de pines.

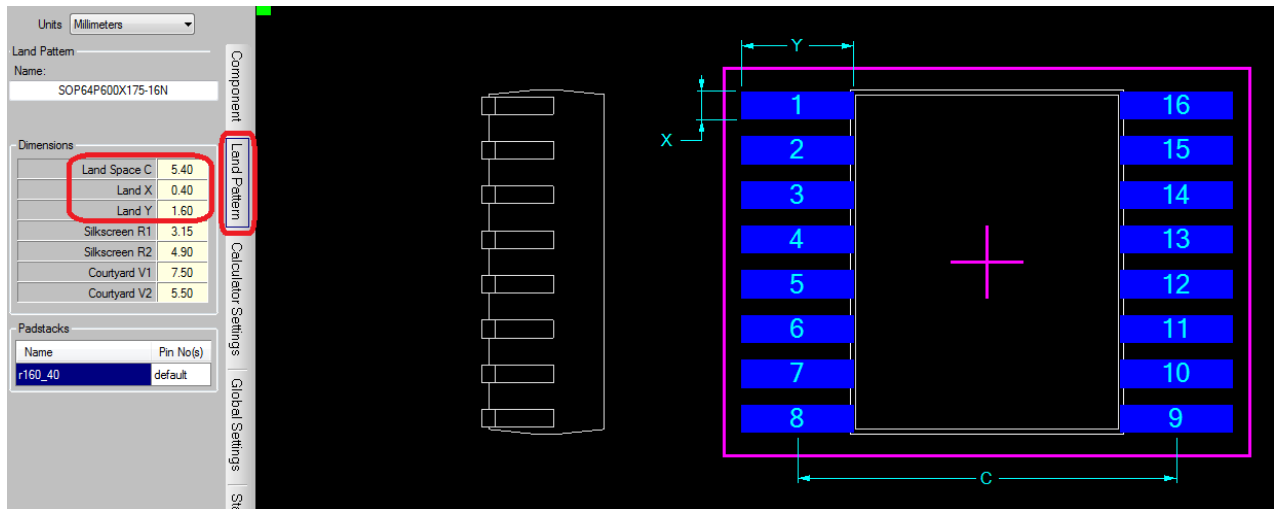


Figura 42: Land Pattern.

Ahora que ya se cuenta con todas las dimensiones requeridas, se puede proceder a la creación del footprint. Sin embargo, previo a este paso, se deben crear antes los pins utilizando la herramienta Pad Designer.

3. Pad Designer.

Esta herramienta tiene el único propósito de generar archivos conteniendo un pin (pad) con dimensiones específicas. Los archivos de pads pueden ser referenciados tantas veces como se requiera para uno o más símbolos, sin importar si se tratan de empaquetados iguales o diferentes. Como regla de dedo, todos los pads creados deberán seguir las mismas unidades de medición. Para el proyecto en curso, se elegirá como unidad la milésima de pulgada (mil). La convención del nombre a seguir para los archivos de pads será: forma del pad, la dimensión más larga y después la dimensión más corta. Una vez que se abra el programa, el usuario visualizará la siguiente ventana:

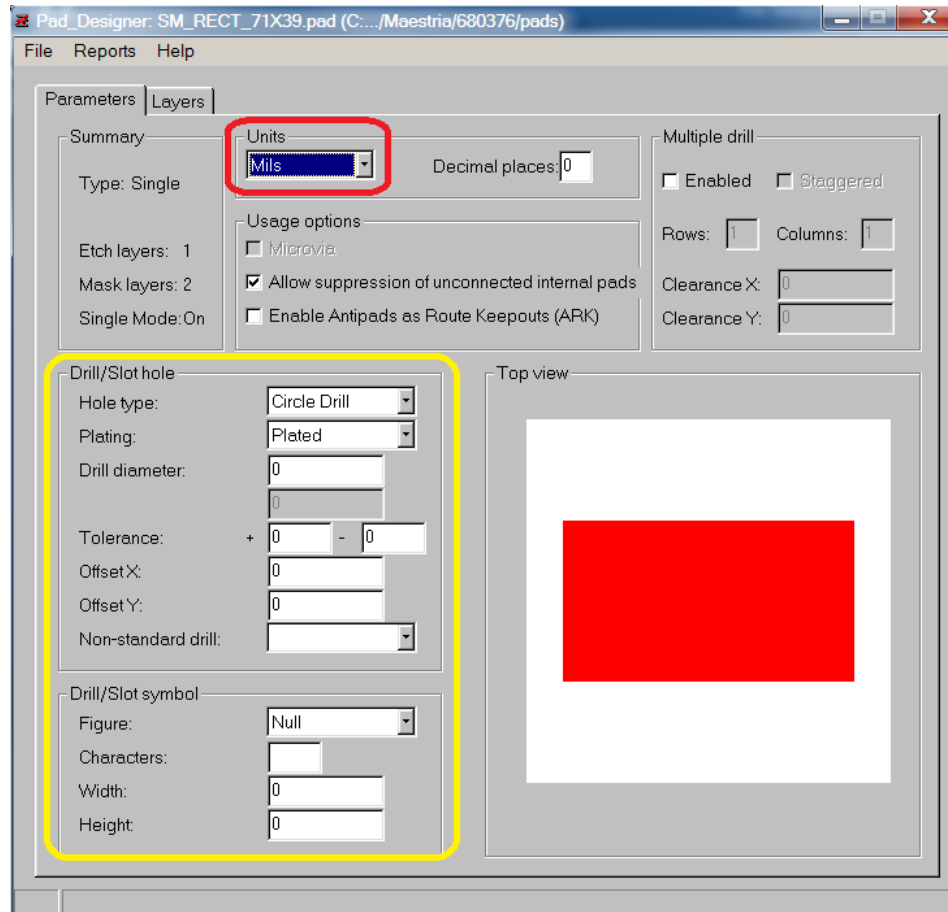


Figura 43: Pad Designer.

Dentro de la pestaña Parameters (pestaña actual) se deberán elegir las unidades, así como el número de decimales a utilizar. Para la creación de pads thru hole, se deberán especificar los datos dentro del recuadro amarillo. Dado que los pads a crear en este ejemplo son SMT (surface mount), estas opciones serán ignoradas y se pasará a la pestaña de Layers, mostrada en la siguiente imagen:

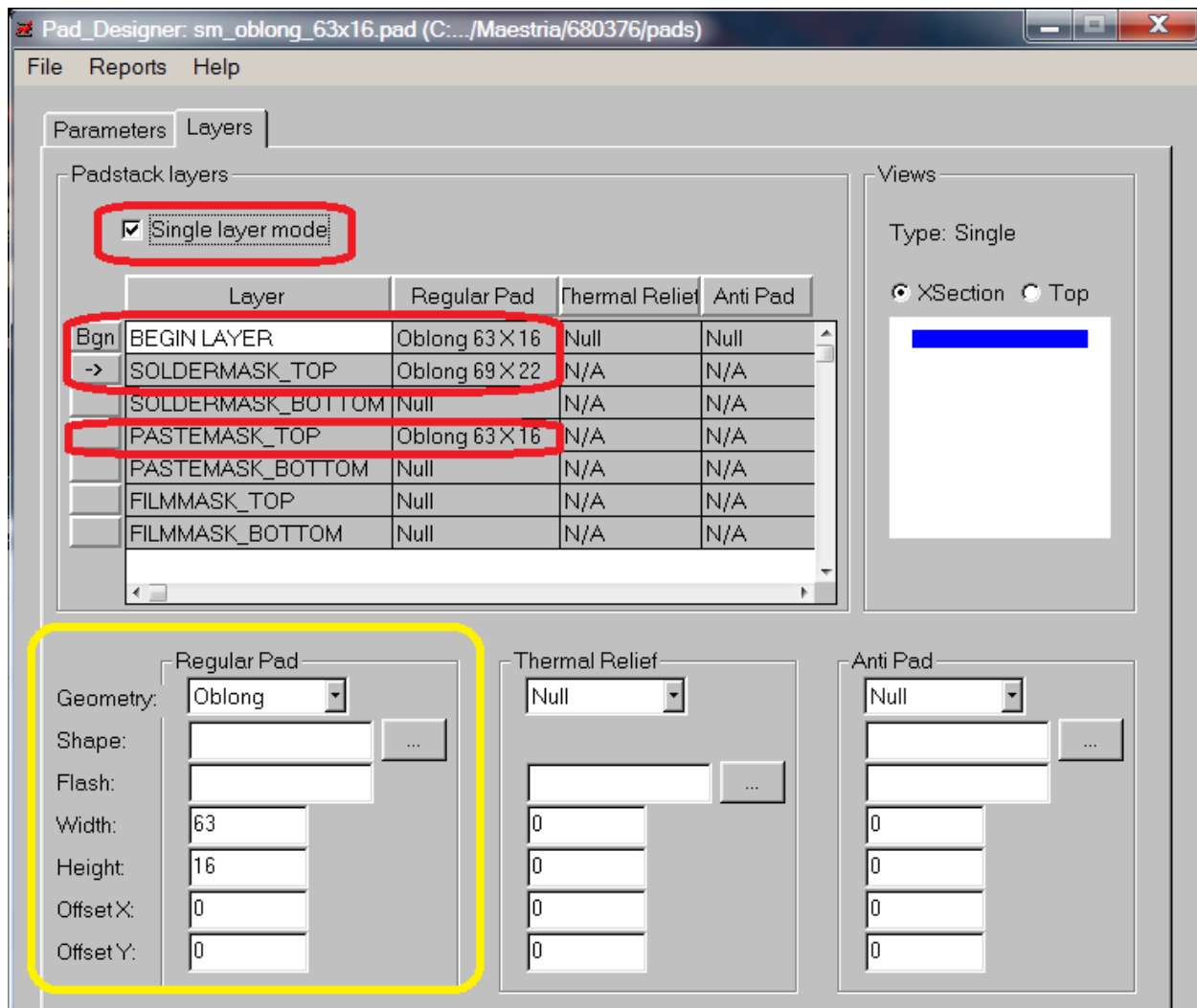


Figura 44: definición de subclases y geometrías.

Se deberá seleccionar la opción Single Layer Mode, ya que con esto se denota que se está generando un pad SMT. Así mismo, se deberán poner las dimensiones del pad para las opciones Begin Layer, Soldermask_TOP y Pastemask_TOP. Tomando las dimensiones del pad obtenidas del LP calculator y haciendo la conversión de mm a mils, se tiene que el pad debe medir 63*16 mils. Dichos datos deben ser agregados en la sección dentro del recuadro amarillo en la imagen. Además de esto, se debe elegir una geometría para el pad. Por convención, para cualquier símbolo, solamente el pin 1 deberá tener una forma rectangular, mientras que el resto de los pines deberán tener la forma de oblong; dada esta regla, se procederá a crear no uno sino dos archivos de pads, ambos con las mismas dimensiones de largo y ancho. Por último, como regla de dedo, a las dimensiones para la opción de Soldermask_TOP se les deberán sumar 6 mils tanto al largo como al ancho del pin, a menos que la hoja de datos del componente indique lo contrario. Para guardar los cambios, basta con ir al menú File y seleccionar la opción Save as.

4. PCB Editor.

El último paso para la creación de un footprint consiste en utilizar la herramienta PCB Editor. Al abrir el programa, se deberá ir al menú File y elegir la opción New. Aparecerá una nueva ventana en donde se pregunta por el Drawing Name y el Drawing Type. Para Drawing Name se deberá poner el nombre del empaquetado (en este caso SSOP-16), mientras que en Drawing Type se cuenta con varias opciones a elegir. Para la creación de símbolos existen dos opciones, las cuales son Package Symbol y Package Symbol (Wizard); la primera opción dejará al usuario crear desde cero el footprint deseado, y tendrá que agregar manualmente todas las partes que lo conforman, tales como pines, outline, reference designator, silkscreen, etcétera. Por otro lado, la segunda opción cuenta con un Wizard el cual guiará paso a paso al usuario a la creación del footprint, en donde se darán a elegir varios empaquetados, y se preguntarán por las dimensiones requeridas para poder crear dicho símbolo. Si el footprint a crear se encuentra o se asimila a alguno de los empaquetados presentes en el Wizard, entonces se recomienda utilizar esta opción, de lo contrario, se deberá usar la opción Package Symbol. Para este ejemplo, el controlador touchscreen se asemeja en empaquetado a un SOIC (small outline integrated circuit). Dicho empaquetado está presente dentro de las opciones en Package Symbol Wizard, por lo que se procederá a seguir con el Wizard en vez de crear el footprint manualmente.

Una vez elegido el empaquetado, el wizard pedirá cargar un template. Cadence cuenta por default con uno, el cual puede ser utilizado para la creación del símbolo. Una vez que se de click en el botón Load Template, se procederá a dar click en Next. Después de esto, se preguntarán las unidades a utilizar dentro del wizard y las unidades para crear el footprint, así como el prefijo para el reference designator; las unidades a elegir serán milímetros (con tres décimas), mientras que el prefijo del ref. des. será U*. Una vez dada esta información, en la siguiente pantalla se preguntará por las dimensiones del componente, las cuales, de acuerdo a los datos obtenidos hasta ahora, serán las siguientes:

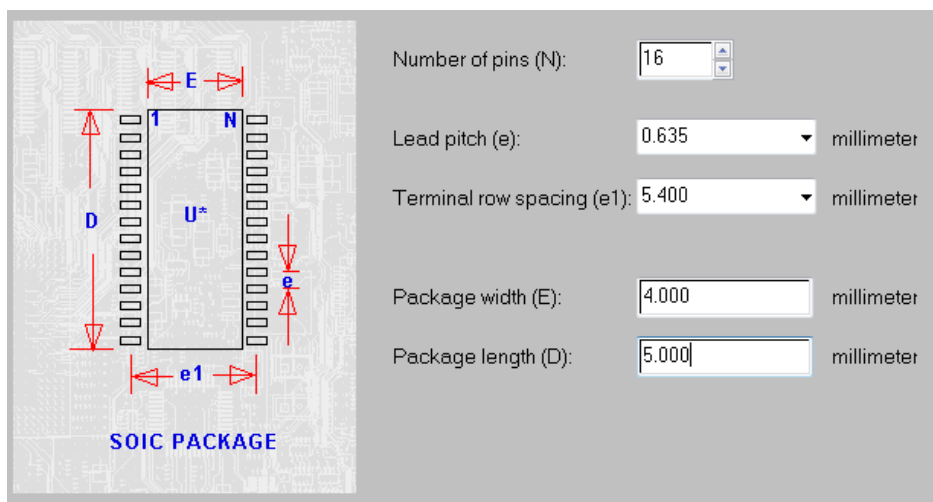


Figura 45: wizard para la creación de un footprint.

Después de dar click en Next, se pedirán a continuación los pads a utilizar en el footprint. Como se mencionó anteriormente, para el pin 1 se utilizará el pad rectangular, mientras que para el resto se usará oblong.

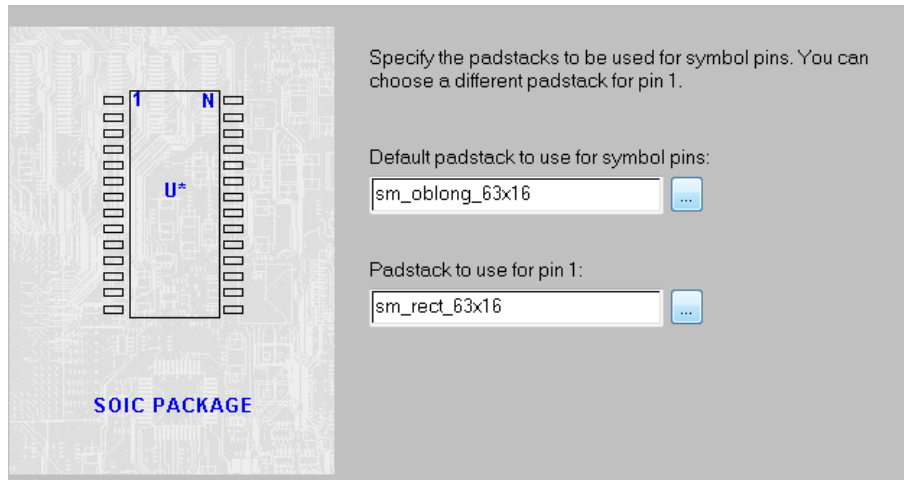


Figura 46: pads a utilizar en el footprint.

Después de dar click en Next, se pedirá elegir la ubicación del origen del símbolo. Se recomienda que se elija como origen el centro del cuerpo. También se preguntará si se desea que el wizard genere un archivo compilado del footprint, denominado archivo psm. Dicho archivo es el que será utilizado para poder agregar el símbolo a un board file. Finalmente, al dar click una vez más en Next, aparecerá la pantalla final del wizard, en donde se menciona que se crearán los archivos dra y psm; al dar click en el botón finish, el footprint habrá sido generado. Para guardarlo, basta con ir al menú File y elegir la opción Save as.

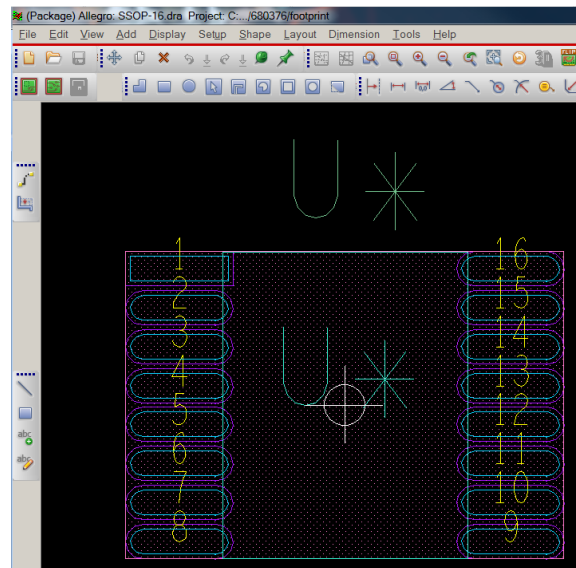


Figura 47: footprint creado.

En caso de haber elegido la opción de creación de símbolo manual, el usuario deberá estar familiarizado con las opciones del PCB editor y conocer cuáles son y dónde están ubicadas las opciones para realizar tareas tales como agregar pines, agregar reference designators, agregar shapes, líneas, texto, etcétera.

Creación de esquemáticos

En esta sección se presenta cómo realizar los esquemáticos, de tal manera que se tenga el proyecto listo para utilizarse en un board file, mediante la creación de un netlist.

El proyecto debe contar con las hojas de esquemáticos suficientes para contener todas conexiones que se deseen realizar, incluyendo también todos los símbolos lógicos de los cuales partirán dichas conexiones. Dentro de la pestaña principal del proyecto, al hacer click derecho en el diseño, se desplegará la opción para crear un esquemático nuevo; al seleccionarlo, se creará un nuevo folder dentro del diseño, el cual contendrá el número de hojas que el usuario desee agregar, como se muestra en la siguiente imagen:

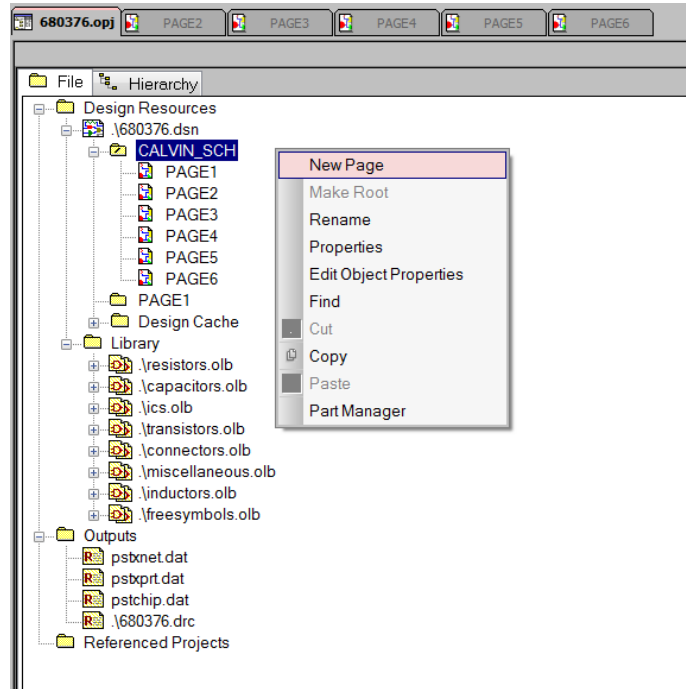


Figura 48: creando un esquemático.

Una vez dentro de una página de esquemático, se puede proceder a hacer alguna de las siguientes tareas:

Agregar un símbolo lógico

Realizar una conexión de un símbolo a otro dentro de la misma página.

Realizar una conexión que vaya a otra página del esquemático.

Realizar conexiones por medio de buses.

Nombrar señales

Agregar conexiones de PWR o GND.

Todas estas opciones pueden ser seleccionadas desde el panel ubicado a la derecha de la pantalla, como se muestra en la siguiente imagen:

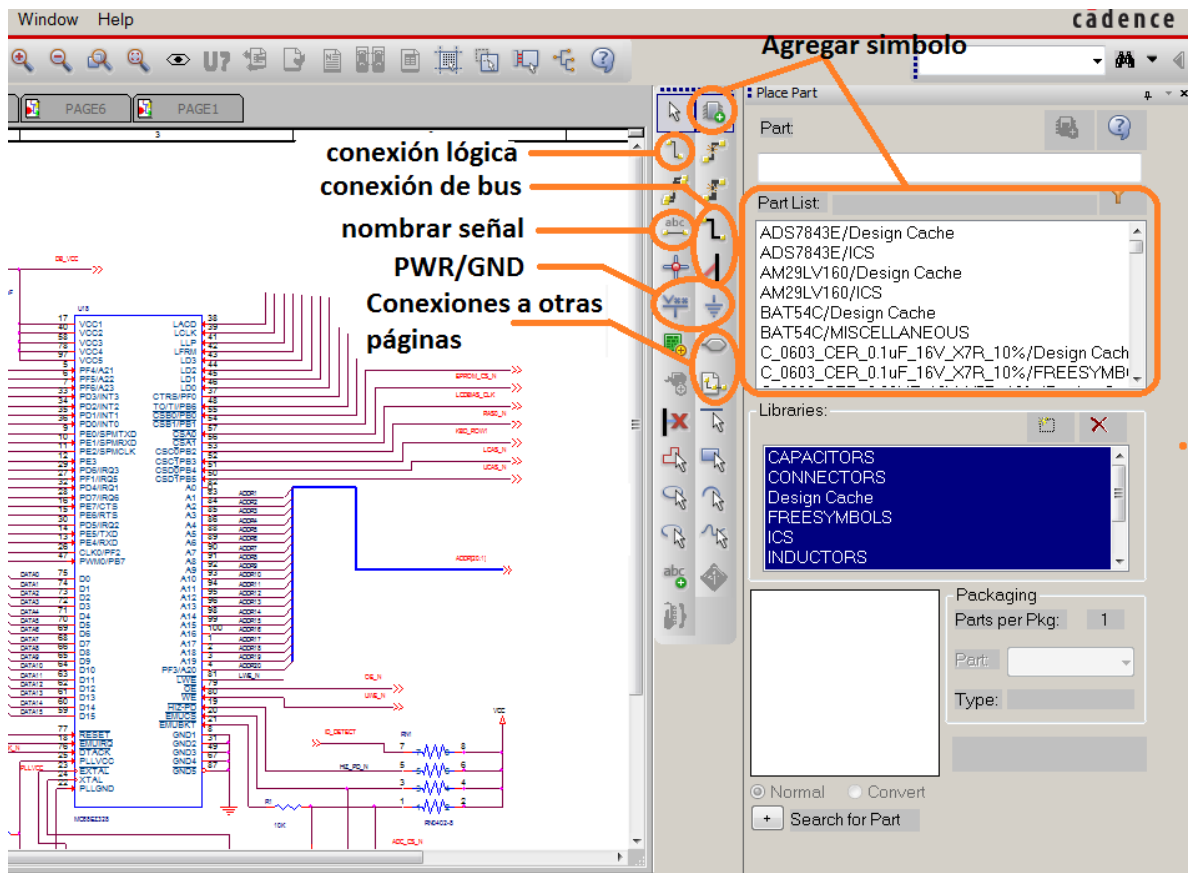


Figura 49: menú de opciones.

Para empezar a realizar conexiones, es necesario contar previamente con símbolos dentro de las hojas. Como se muestra en la imagen de arriba, al seleccionar la opción de agregar símbolo (place part), aparecerá una lista de partes del lado derecho. El usuario puede agregar la cantidad que desee para la parte o partes que sean necesarias en el diseño; basta con dar doble click en una de ellas para después colocarla dentro de la hoja del esquemático. Posteriormente, el usuario podrá realizar conexiones entre los símbolos que se hayan agregado dentro de una misma hoja usando la opción de la conexión lógica (place wire); se deberá seleccionar un pin de alguno de estos símbolos y llevarlo al pin destino con el mismo cursor del mouse. Si se desea hacer una conexión hacia otra página del esquemático, en ese caso, se deberá utilizar la opción de conexiones a otras páginas, nombradas en la herramienta como Place Port o Place off-page connector. La diferencia entre ambas opciones radica en que para Place Port, se debe tomar en cuenta el tipo de pin al que se está haciendo la conexión, ya sea input, output, bidireccional, etcétera; de no respetarse esto, pueden generarse errores al momento de querer crear el net list. Por otro lado, la opción de Place off-page connector simplemente realizará la conexión a otra página sin importar el tipo de pin que se trate. Al utilizar cualquiera de estas dos opciones, el usuario deberá editar el nombre de la señal, haciendo doble click en el nombre de la señal actual (portxxx para place port o offpageleftxxx para off page connectors), como se muestra en la siguiente imagen.

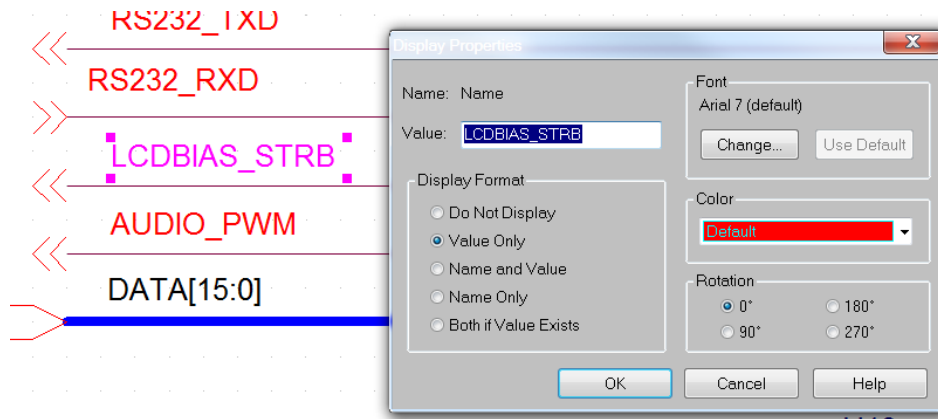


Figura 50: editar nombres de señales.

En el caso de querer realizar de manera simultánea la conexión de varias señales, comúnmente aplicado en señales de memoria como direcciones o datos, se puede proceder a utilizar las opciones de conexión de bus, las cuales son Place Bus y Place Bus Entry. La primera opción llevará como tal a todas las señales que se deseen incluir dentro del bus deseado, y la manera de usarse es idéntica a la opción de Place Wire, siendo la única diferencia el ancho de la línea en el dibujo del esquemático; sin embargo, para conectar el bus a los pines deseados, se deberá utilizar la opción de Place Bus Entry, en donde se deberá hacer contacto entre el pin y el bus o entre un wire conectado al pin y el bus, como se muestra en la siguiente imagen:

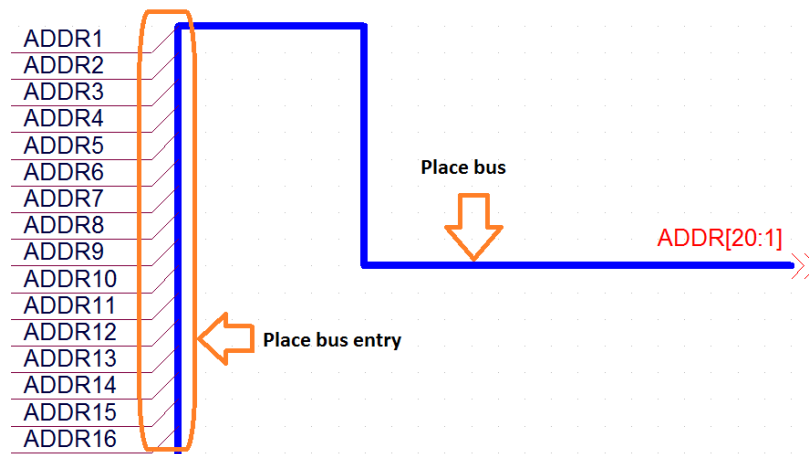


Figura 51: buses.

En cuanto la opción de nombrar señales, como ya se mencionó anteriormente, si se utiliza un puerto o un off-page connector, basta con dar doble click en el nombre que viene por default con dicha opción y cambiarlo al que desee el usuario. Por otro lado, si la señal viaja dentro de la misma página de esquemático, basta con utilizar la opción de nombrar señal (place net alias); dicha opción pedirá el nombre de la señal deseada, con lo cual una vez que se haya otorgado, el usuario deberá arrastrar dicho nombre hasta el wire deseado usando el cursor.

Finalmente, una opción más necesaria en cualquier esquemático son las conexiones a PWR y GND, las cuales pueden ser agregadas utilizando las opciones de Place Power o Place Ground. Al seleccionarlas,

se desplegará una lista con todos los tipos de símbolos para ambas conexiones, de acuerdo a lo que desee el usuario.

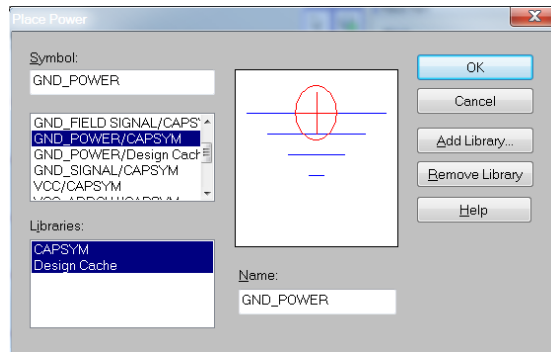


Figura 52: conexiones de PWR y GND.

Con las opciones ya mencionadas hasta este punto, es posible proceder a la creación completa de un esquemático utilizando todos los símbolos lógicos que fueron realizados anteriormente. Suponiendo que el usuario haya terminado sus esquemáticos y desee crear el netlist para poder llevar dichas conexiones lógicas a un board file, previamente se deberá correr un Design Rule Check; dicha opción será visible al seleccionar el esquemático en la pestaña del proyecto y después eligiendo el menú de Tools:

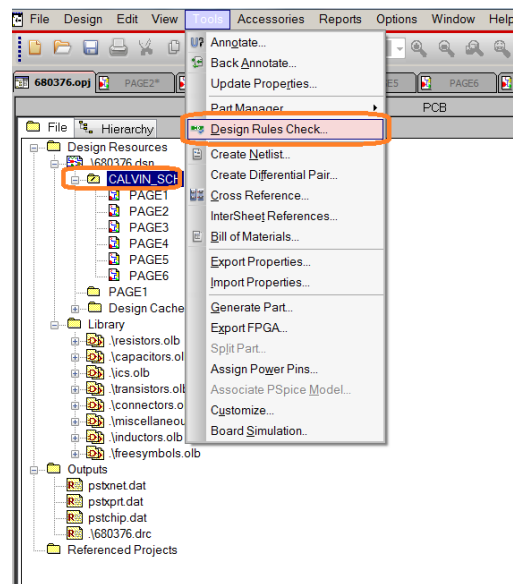


Figura 53: Design Rule Check.

Al seleccionarla, se abrirá una nueva ventana en donde se pedirá un set de reglas para este chequeo, las cuales pueden quedarse con la selección por default, en donde se marcarán en el esquemático todos los errores del diseño que deben de arreglarse previo a la generación del netlist. Se deberán correr múltiples chequeos hasta que el diseño esté libre de errores y, preferentemente, también de warnings, como se muestra en la siguiente imagen. Una vez logrado esto, se podrá proceder a la creación del netlist.

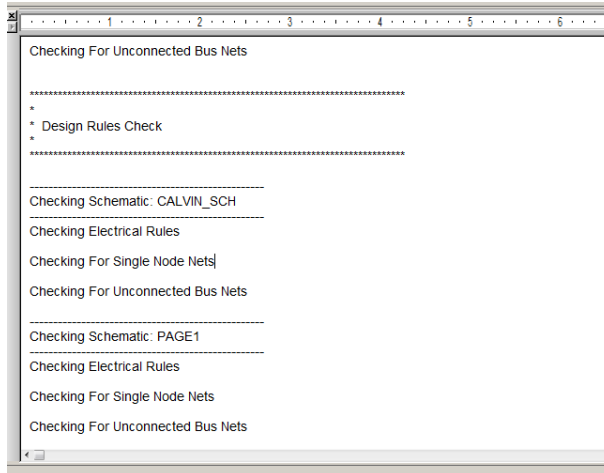


Figura 54: log del chequeo.

La creación del netlist se realiza dentro del mismo menú de Tools, en la opción Create Netlist. Aparecerá la ventana mostrada en la siguiente imagen, en donde deberá especificarse qué board file será utilizado de input y cuál de output. El resto de las opciones pueden dejarse con sus valores default. Una vez que se dé click en OK, se procederá a crear el netlist. Cualquier error que haya quedado en el diseño del esquemático, así como cualquier conflicto que exista con algún footprint podrá aparecer en este proceso e impedirá la creación de dicho netlist, por lo que deberán ser arreglados antes de volver a repetir este paso. En caso de que se haya realizado de manera exitosa, se abrirá el PCB Editor y el usuario podrá comenzar a realizar el layout de su diseño.

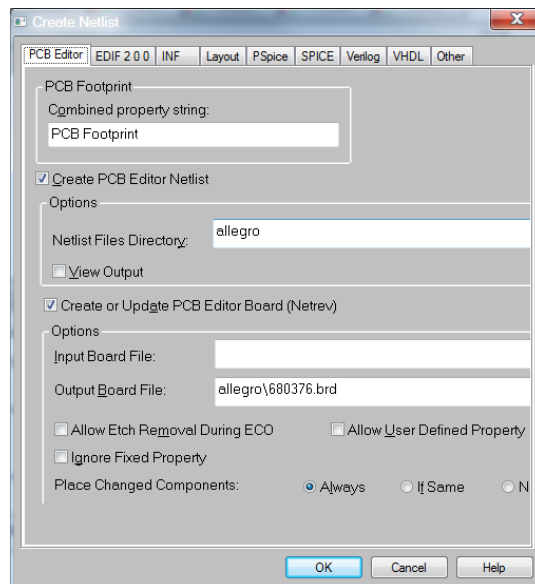


Figura 55: creación del netlist.

Un netlist creado sin errores ni warnings deberá mostrar un estatus de compilación similar al obtenido a continuación del ejercicio asignado en clase:

```
----- Summary Statistics -----

netrev run on Apr 1 1:36:09 2013
DESIGN NAME : '680376'
PACKAGING ON Jan 3 2011 16:39:28

COMPILE 'logic'
CHECK_PIN_NAMES OFF
CROSS_REFERENCE OFF
FEEDBACK OFF
INCREMENTAL OFF
INTERFACE_TYPE PHYSICAL
MAX_ERRORS 500
MERGE_MINIMUM 5
NET_NAME_CHARS '#%&()*+-./:=>?@[^_`]'
NET_NAME_LENGTH 24
OVERSIGHTS ON
REPLACE_CHECK OFF
SINGLE_NODE_NETS ON
SPLIT_MINIMUM 0
SUPPRESS 20
WARNINGS ON

No error detected
No oversight detected
No warning detected

cpu time    1:08:26
elapsed time 0:00:36

*** Done ***
```

Ready

Figura 56: log del netlist.

Diseño del layout

Por último, en esta sección se explica una metodología para realizar el diseño del layout de la tarjeta en curso, de la cual ya se han elaborado los esquemáticos y se cuenta con los footprints requeridos. Para esto, se parte de la suposición de que el usuario ya tiene conocimiento de la herramienta PCB Editor.

Partiendo desde ceros en una tarjeta nueva, las tareas a realizar deben ser las siguientes en el orden mencionado:

1. Placement de los componentes primarios, es decir, aquellos que ocupen una región más grande en la tarjeta, que por lo general son conectores, CPUs, memorias, etc.
2. Placement de componentes secundarios (resistencias, capacitores, inductores, transistores, etc.).
3. Creación de reglas en el constraint manager.
4. Fanout de todos los componentes en la tarjeta.
5. Ruteo de todas las señales.
6. Fine tuning del ruteo.
7. Limpiar DRCs.
8. Acomodo del silkscreen.

Por lo general, siempre se tendrá la ayuda de un ingeniero mecánico para facilitar el placement de todos los componentes más grandes, con el fin de evitar interferencias entre dichos componentes. Para nuestro ejemplo, se ha proporcionado un archivo PDF en donde se sugiere una ubicación para todos ellos. Posteriormente, se debe proceder a hacer un placement preliminar de todas aquellas partes restantes; para esto, uno debe auxiliarse siempre de los esquemáticos, ya que aquí se podrá visualizar en donde deben ser colocados todos estos símbolos. En las siguientes imágenes se muestra una sección del esquemático, así como un placement sugerido en el layout.

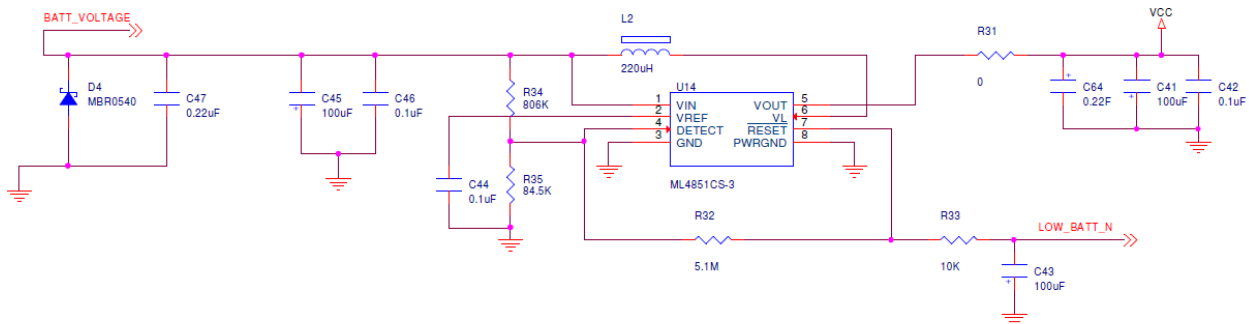


Figura 57: página de esquemático.

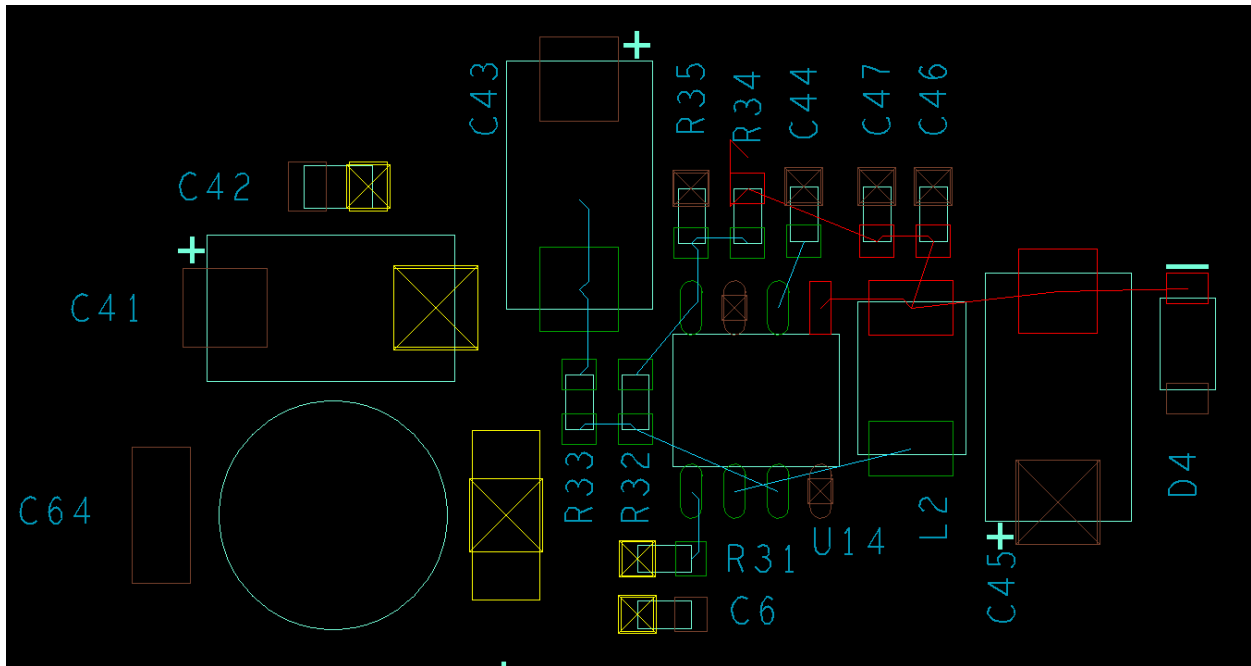


Figura 58: placement sugerido.

En tercer lugar debe realizarse la implementación de todas las reglas de ruteo en el Constraint Manager. Para este proyecto, dichas reglas deben aplicarse a las secciones Physical y Spacing. Para Physical, las reglas consisten básicamente en el ancho de los trazos, tipo de vía a utilizar, longitud máxima para neck mode, entre otros detalles. Se pueden crear múltiples sets de reglas y aplicar el indicado a diferentes conjuntos de señales. Para el caso de Spacing, las reglas consisten en el espaciamiento mínimo que debe existir entre trazos, trazo y vía, trazo y pin, trazo y plano, etcétera. De la misma manera, pueden crearse múltiples sets de reglas de espaciamiento y aplicarse al conjunto correspondiente de señales. Por lo general, estas reglas siempre deben ser proporcionadas por el dueño de cada interfaz de la tarjeta.

Las señales siempre deben agruparse en grupos, los cuales son definidos como Net Class. Cada Net Class puede/debe incluir todas las señales que pertenecen a una misma interfaz. De esta manera, será más fácil la asignación de reglas físicas y de espaciamiento, ya que solo se asignarán a la clase, en vez de señal por señal.

Una vez implementadas las reglas, el siguiente paso sugerido es hacer el fanout de todos los componentes en la tarjeta. Se sugiere realizar esto previo al ruteo de las señales ya que éste puede verse impactado por el hecho de agregar vías más adelante, lo cual requeriría volver a editar todos los trazos cercanos a pines de componentes, ya sea en layers internos o externos. El fanout puede realizarse de forma manual, utilizando la herramienta Add Connect y creando un trazo partiendo del pin seleccionado, para después agregar una vía. También puede realizarse de manera automática utilizando la herramienta Create Fanout, en la cual puede especificarse cómo se desea tener el arreglo de vías y si se quiere hacer fanout pin por pin, o del símbolo completo. Finalmente, para aquellas señales que

puedan ser ruteadas en el mismo layer en el que se encuentra el símbolo, ya sea top o bottom, la vía puede ser borrada y continuar con el trazo.

Posteriormente se debe proceder al ruteo de todas las señales en la tarjeta. Este paso resultará en el mayor consumo de tiempo del ingeniero, ya que gran parte del diseño se define en este paso. Algunos pasos a definir en este punto son los siguientes:

- Rutear señales en layers internos o externos, de acuerdo a las restricciones que puedan tener algunas señales, así como a la facilidad de ruteo.
- Definir ruteo horizontal y ruteo vertical por layer. No se recomienda tener ambos en el mismo más que como última opción.
- Saber qué señales son las que requieren tener referencia de GND, que por lo regular, son todas aquellas de alta velocidad.
- Crear los planos de power para todos los rieles existentes en la tarjeta. Para rieles con poca demanda de corriente, pueden sustituirse los planos por trazos lo suficientemente anchos.
- Realizar fine tuning para señales que lo requieran, el cual puede consistir en cumplir longitudes mínimas/máximas, así como en emparejar longitudes (por lo regular para señales de memoria).

Todos estos deberán ser realizados con el criterio del ingeniero de CAD, siempre cumpliendo con las reglas especificadas para cada interfaz. En la siguiente imagen se muestra un ejemplo de ruteo de señales de datos de memoria, las cuales fueron emparejadas en longitud utilizando trombones:

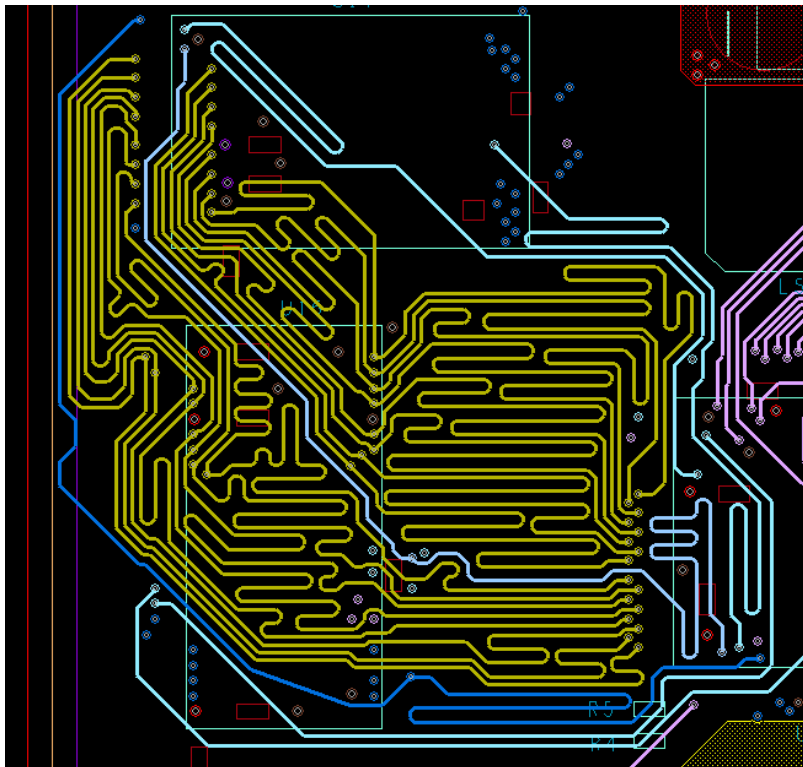


Figura 59: ruteo de señales de memoria con fine tuning.

A lo largo del proceso de ruteo pueden llegar a cometerse violaciones, las cuales son mostradas en la tarjeta con marcadores denominados DRCs (Design Rule Check). Dichas violaciones pueden ocurrir por violar alguna regla configurada en el constraint manager, ya sea física o de espaciamento, tal como el no tener el ancho correcto para un trazo, no tener espaciamento suficiente entre trazos o entre trazos y vías, etcétera. Se recomienda que los DRCs vayan siendo limpiados a lo largo del proceso de ruteo, ya que esto puede ayudar a visualizar áreas críticas de la tarjeta, las cuales son caracterizadas por contar con poco espacio para rutear cierto número de señales.

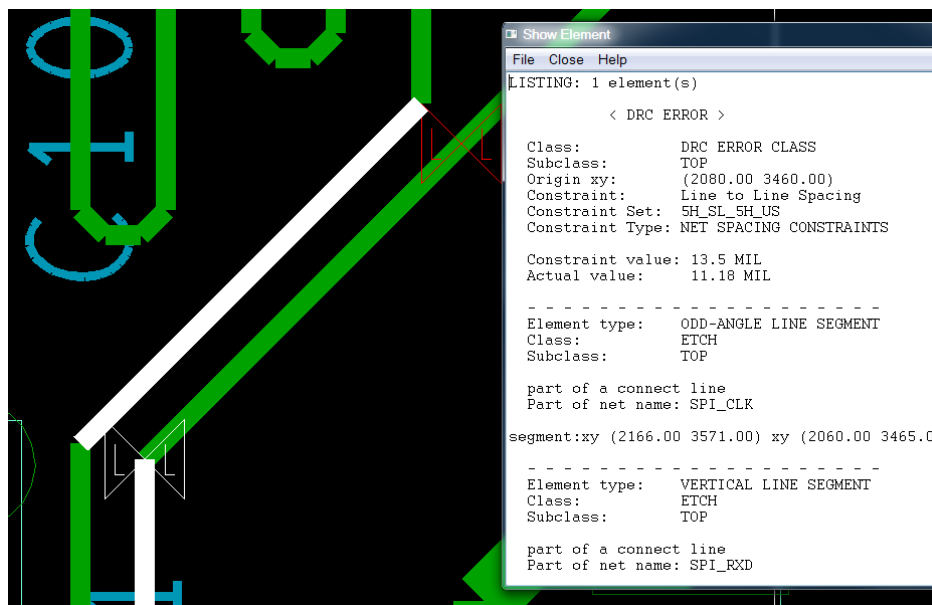


Figura 60: DRCs presentes en el layout.

Finalmente, una vez que se tienen ruteadas todas las señales en la tarjeta y se hayan limpiado todos los DRCs, puede procederse al acomodo del silkscreen para todos los componentes colocados en top y bottom. Debe seleccionarse un tamaño de texto lo suficientemente grande como para que pueda ser visualizado al final en la tarjeta real. También debe tenerse cuidado de que el silkscreen no se traslape con ningún pin o vía, ya que esto puede afectar a la visibilidad del texto, así como en el hecho de afectar el área de contacto que puedan tener los pines de un componente con el solder paste.

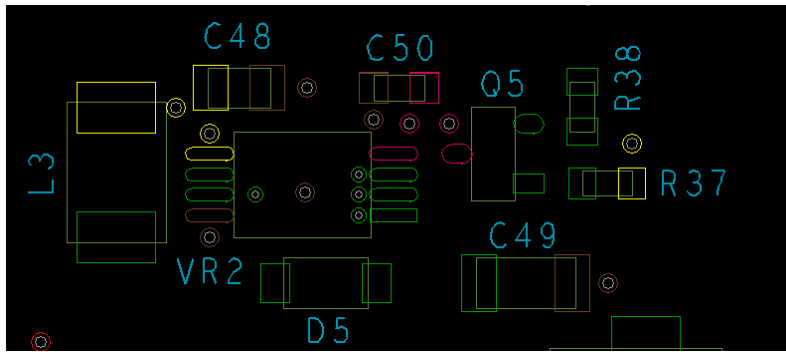


Figura 61: acomodo de silkscreen.

Luego de haber realizado todos estos pasos, la tarjeta debería estar prácticamente lista para fabricarse en términos de manufactura. Para cerciorarse de que todas las señales han sido ruteadas y de que todos los DRCs han sido limpiados, puede consultarse el estatus de la tarjeta en el menú Display:

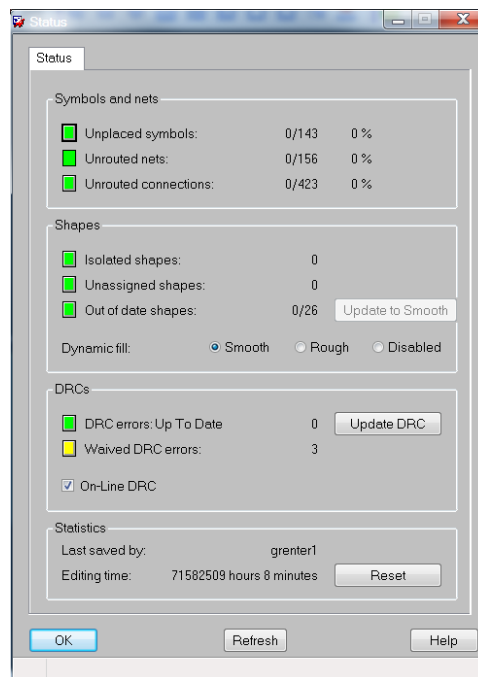


Figura 62: estatus de la tarjeta.

En las siguientes imágenes se muestra el diseño completo del PCB, el cual corresponde a una Memory Riser, en donde se tienen dos DIMMs por canal, y 3 canales por controlador de memoria, así como los reguladores de voltaje necesarios.

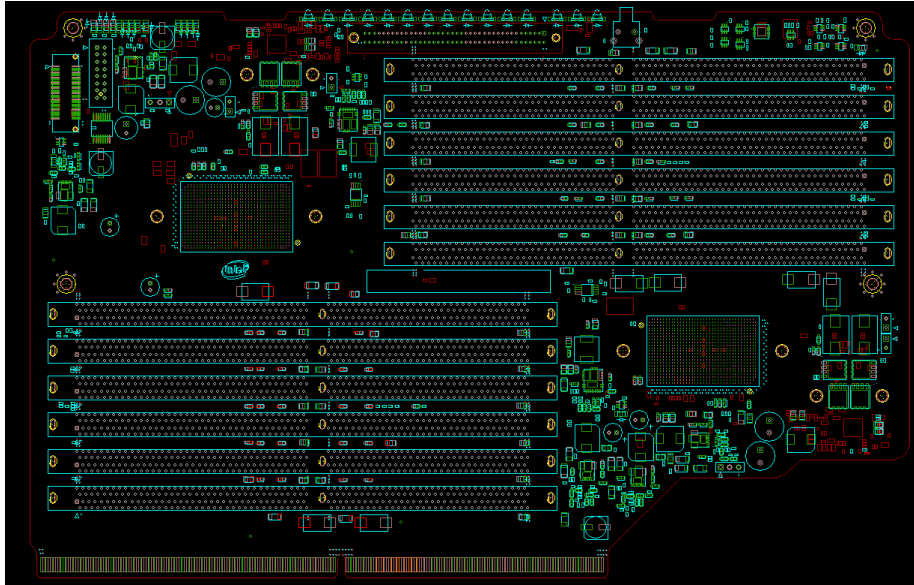


Figura 63: layout del placement.

El stack up diseñado es de 16 capas, el cual se muestra en la siguiente imagen:

Layout Cross Section										
	Subclass Name	Type	Material	Thickness (MIL)	Conductivity (mho/cm)	Dielectric Constant	Loss Tangent	Negative Artwork	Shield	Width (MIL)
1		SURFACE	AIR			1	0			
2	TOP	CONDUCTOR	COPPER	2	595900	1	0	<input type="checkbox"/>		6.00
3		DIELECTRIC	FR-4	2.7	0	4.2	0.035			
4	L2_GND1	PLANE	COPPER	1.3	595900	1	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
5		DIELECTRIC	FR-4	4	0	4.2	0.035			
6	L3_SIG1	CONDUCTOR	COPPER	1.3	595900	1	0	<input type="checkbox"/>		5.00
7		DIELECTRIC	FR-4	3.7	0	4.2	0.035			
8	L4_Pwr1	PLANE	COPPER	1.3	595900	1	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
9		DIELECTRIC	FR-4	4	0	4.2	0.035			
10	L5_SIG2	CONDUCTOR	COPPER	1.3	595900	1	0	<input type="checkbox"/>		5.00
11		DIELECTRIC	FR-4	3.7	0	4.2	0.035			
12	L6_GND2	PLANE	COPPER	1.3	595900	1	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
13		DIELECTRIC	FR-4	4	0	4.5	0.035			
14	L7_VOLTAGE_SIG	PLANE	COPPER	1.3	595900	4.5	0.035	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
15		DIELECTRIC	FR-4	28	0	4.5	0.035			
16	L8_VOLTAGE_SIG	PLANE	COPPER	1.3	595900	4.5	0.035	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
17		DIELECTRIC	FR-4	4	0	4.2	0.035			
18	L9_GND3	PLANE	COPPER	1.3	595900	1	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
19		DIELECTRIC	FR-4	3.7	0	4.2	0.035			
20	L10_SIG3	CONDUCTOR	COPPER	1.3	595900	1	0	<input type="checkbox"/>		5.00
21		DIELECTRIC	FR-4	4	0	4.2	0.035			
22	L11_Pwr2	PLANE	COPPER	1.3	595900	1	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
23		DIELECTRIC	FR-4	3.7	0	4.2	0.035			
24	L12_SIG4	CONDUCTOR	COPPER	1.3	595900	1	0	<input type="checkbox"/>		5.00
25		DIELECTRIC	FR-4	4	0	4.2	0.035			
26	L13_GND4	PLANE	COPPER	1.3	595900	1	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
27		DIELECTRIC	FR-4	2.7	0	4.2	0.035			
28	BOTTOM	CONDUCTOR	COPPER	2	595900	1	0	<input type="checkbox"/>		6.00
29		SURFACE	AIR			1	0			

Figura 64: stackup del PCB.

El ruteo de memoria en una de las capas internas del PCB, con ajuste de longitud se muestra en la siguiente imagen:

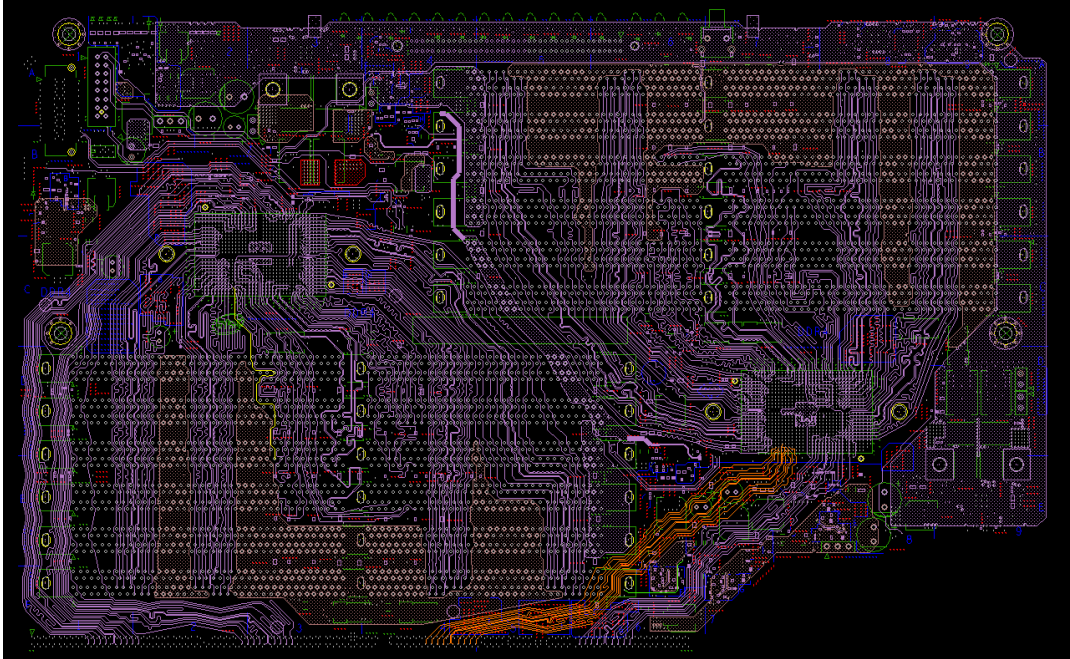


Figura 65: layout del ruteo.

Se utilizaron dos capas dedicadas para la distribución de planos de poder, de acuerdo a los diferentes rieles necesarios en el diseño. Una de las dos capas es mostrada a continuación:

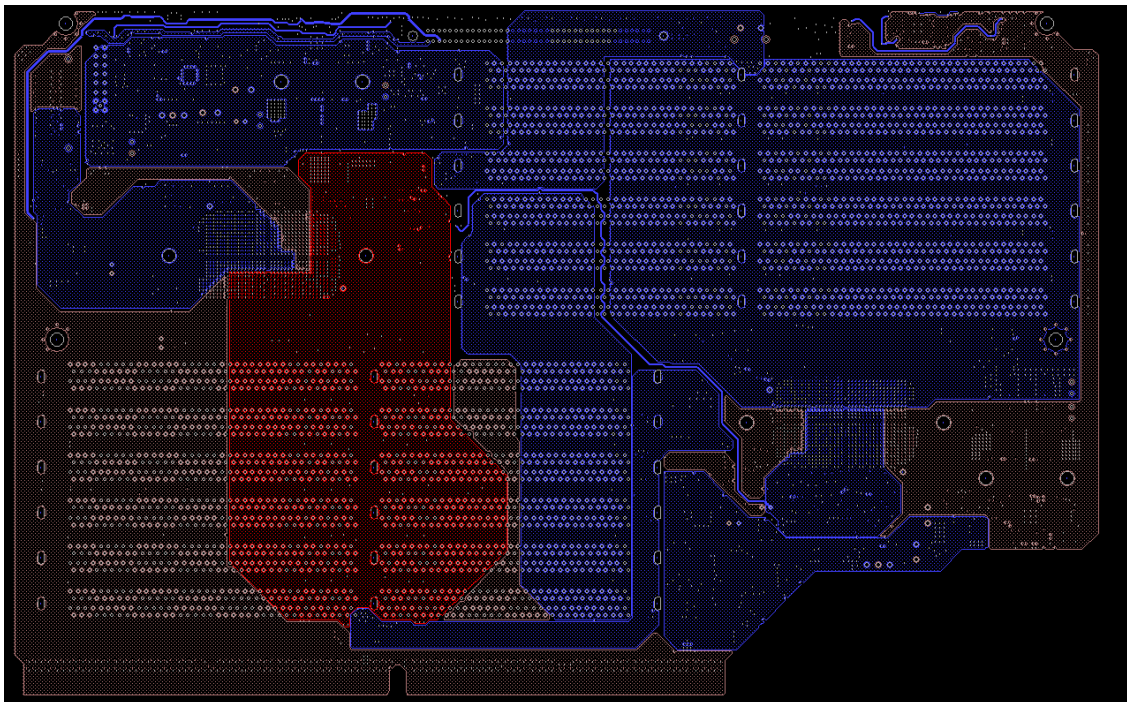


Figura 66: layout de planos de PWR.

Y finalmente, el acomodo del silkscreen.

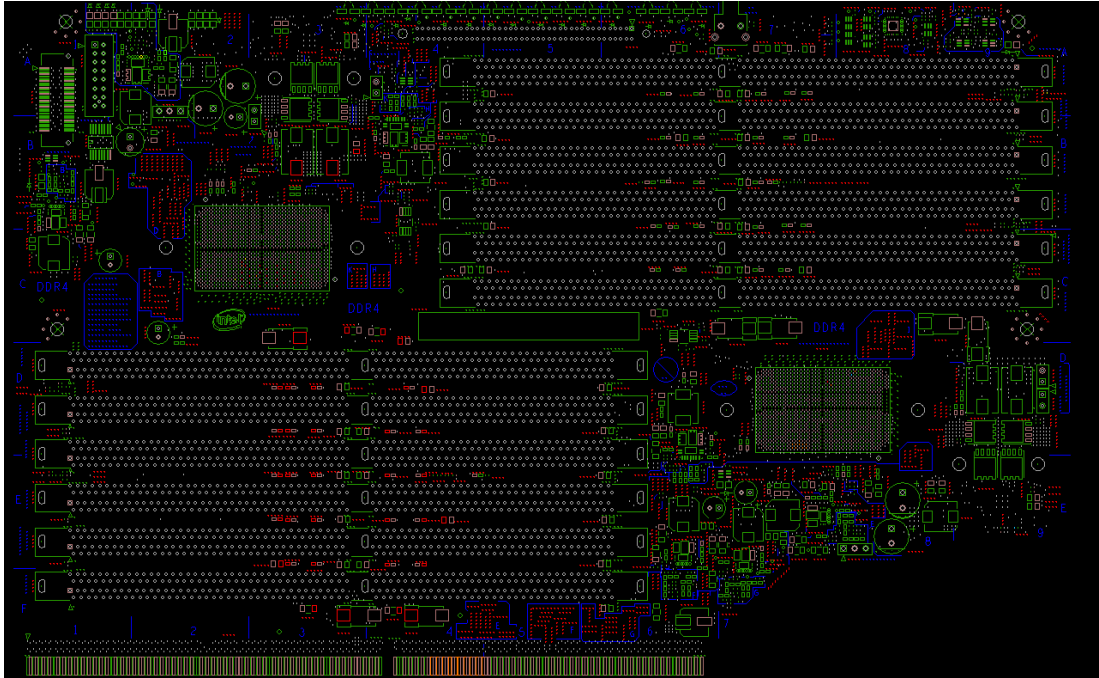


Figura 67: silkscreen de la tarjeta.

Conclusiones

A pesar de ya haber contado con conocimientos de layout desde el principio del curso, estuve motivado a tomar la materia con tal de adquirir los conocimientos relacionados a la creación de librerías y esquemáticos. Siento que la materia fue muy buena en ese aspecto y ahora tengo un mayor entendimiento de cómo consiste un proyecto en su totalidad. A pesar de que en mi trabajo se utilice otro software de Cadence para realizar estas tareas, es importante tener en claro los conceptos y los pasos a realizar, con lo cual me siento con mayor capacidad y con la determinación de dominar este software también eventualmente (PCB Design HDL). De igual forma, todos los conceptos vistos en clase me han sido de gran utilidad, ya que han ayudado a reforzar mi conocimiento de PCBs y tener ahora una mayor versatilidad como ingeniero de layout en mi trabajo.