

Parte 3 del proyecto: Place&Route, DRC y análisis por medio de mapas de calor

Isaac F. Fonseca Segura
Escuela de Ingeniería Eléctrica
Universidad de Costa Rica
San José, Costa Rica
Email: isaac.fonsecasegura@ucr.ac.cr

Resumen—En este proyecto se lleva a cabo un breve análisis de los tiempos requeridos en las etapas de place&route y de los DRC's que se obtienen después de estas etapas, pero principalmente el proyecto se enfoca en la generación de mapas de densidad de celdas, nets y pines basados en los datos obtenidos del archivo `.def`.

Index Terms—placement, routing, DRC, heatmap, cells, nets, pins, microelectronics.

INTRODUCCIÓN

En esta tercera entrega del proyecto se ejecutan las etapas de place & route para obtener un análisis de DRC (Design Rule Check) por medio de la herramienta Qflow [1]. Luego se usan los datos del archivo `.def` para extraer información sobre las celdas, nets y pines, y con esta información generar mapas de calor. Para ello se usó la función *hist2d* [2] de la biblioteca *matplotlib* [3], la cual funciona con el lenguaje de programación *python* [4]. Para capturar los datos necesarios del archivo `.def` se usó una combinación de los comandos *sed* y *grep*, junto con el uso de expresiones regulares *regex*, en un script de *bash*. Luego para manipular esta información se usó la biblioteca *pandas* [5], y para facilitar las operaciones numéricas se utilizó la biblioteca *numpy* [6]. Una vez con la información filtrada se procedió a generar los mapas de calor.

29 de junio, 2021

I. EXPLICACIÓN

I-A. Generación de mapas de calor

Para generar los mapas de calor a partir del archivo `.def` primero fue necesario filtrar toda la información necesaria. Esto se logró con un pequeño **script de bash**, el cual **se ejecuta de la siguiente manera**: `./script archivo.def`, por ejemplo `./script micro_ucr_hash.def`. Este script genera archivos precedidos por el nombre *coordenadas* los cuales contienen las coordenadas necesarias de celdas, nets y pines para hacer los mapas de calor.

Una vez con todas las coordenadas requeridas se procesó la información en *python* por medio de *pandas* y *numpy* primordialmente, y se generó los mapas de calor por medio de histogramas bidimensionales con *hist2d* de *matplotlib*.

El proyecto está dividido en 2 directorios, **area** y **performance**. En cada uno de estos se encuentra tanto el script en el directorio `datos_def`, como el código fuente en el directorio `src`. Además se incluyeron directorios para capturar las imágenes de los mapas de calor y guardar los datos obtenidos de las etapas iniciales.

Para **generar los mapas de calor es necesario correr el comando** `python main.py` en el directorio `src`. Una vez ejecutado se generan los mapas de calor secuencialmente en pantalla, al cerrar uno se abre el siguiente. Todos se guardan automáticamente en el directorio `img`. Además se despliega la información de la cuadrícula y el número de tracks por cuadrado en esta en la terminal.

Las funciones implementadas tienen como sufijo *hm*, es decir *heatmap*. Cada una recibe una serie de parámetros similares, como la dirección del archivo con las coordenadas correspondientes, las coordenadas *x* y *y* la cuadrícula (que al mismo tiempo son los *bins* del histograma). También reciben el factor de multiplicación de unidad del archivo `.def` y una dirección dónde se guardará la imagen del mapa de calor. La única excepción es la función que genera el mapa de densidad de pines, ya que debe recibir el offset del área del diseño.

I-B. Funciones

I-B1. cells_hm: Simplemente obtiene las coordenadas de las celdas y hace un histograma con estas. Si en un cuadro hay una coordenada de celda eso significa que la celda está en ese lugar de la cuadrícula.

I-B2. nets_hm: Quizás la más complicada de todas, y no representativa completamente debido a la naturaleza de una línea (una serie infinita de puntos). Analiza las coordenadas inicial y final de cada segmento de metal. Si el metal atraviesa la cuadrícula entonces se agrega una nueva coordenada en ese punto y así se llena el histograma con los puntos donde hay metal, en este caso metal 4. Además requiere un poco de manipulación de información para tener los datos necesarios, por ejemplo, reemplazar los valores “*” con la coordenada correcta.

I-B3. pins_hm: Similarmente a *cells_hm*, recoje las coordenadas de los pines y hace un histograma con estas. La

dificultad es que las coordenadas del diseño tienen un offset negativo, por lo que algunos valores son negativos. Por esto requiere el offset, para sumarlo a estas coordenadas negativas y que sean válidas en el mapa de calor, el cual comienza en la coordenada (0,0).

I-B4. main: En el archivo main se agregan todos los datos relevantes al diseño y directorios para las funciones a llamar, también se genera la cuadrícula basado en la extensión física del diseño. Además se hacen cálculos sencillos para presentarlos en terminal. Entre estos cálculos está el de **Tracks por cuadrado de la cuadrícula**, el cual corresponde a tomar el alto del cuadrado y la división por track dada en el archivo .def, y dividirlos. Es decir, si el cuadrado mide 10 μm de alto y la separación entre tracks es de 1 μm , entonces hay 10 tracks en ese cuadrado.

II. RESULTADOS

En esta sección se muestran los distintos resultados obtenidos para ambos diseños propuestos en la primera parte del proyecto. Las mediciones de tiempos se hicieron manualmente, cronometrando los tiempos de ejecución.

II-A. Diseño optimizado en área

Para el diseño optimizado en área los tiempos fueron cortos, debido a que este diseño es muy pequeño. La etapa que consumió más tiempo fue la etapa de routing. Se obtuvieron 2 violaciones de DRC's.

Medición	Tiempo [min:seg]
Placement	2:06
Routing	3:05
PR STA	0:06
Migration	1:02
DRC	0:08
Cantidad de DRCs = 2	

En las Figuras 1, 2 y 3 se muestran los diferentes mapas de calor obtenidos. Para estos mapas se utilizó una división de 50x50.

II-B. Diseño optimizado en rendimiento

Para el diseño optimizado en desempeño los tiempos mucho mayores, no hubo una relación lineal, el tiempo no fue 4 veces mayor (este diseño es aproximadamente 4 veces más grande que el optimizado en área), si no que se asemejó más a un crecimiento exponencial. La etapa que consumió más tiempo fue la etapa de routing. Se obtuvieron 9 violaciones de DRC's.

Medición	Tiempo [min:seg]
Placement	17:03
Routing	44:15
PR STA	0:41
Migration	15:16
DRC	1:28

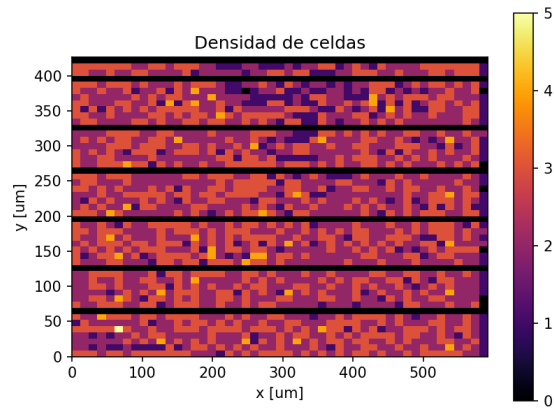


Figura 1. Mapa de densidad de celdas para el diseño optimizado en área.

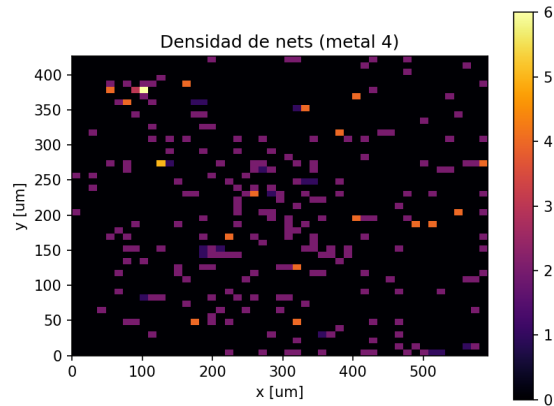


Figura 2. Mapa de densidad de nets (metal 4) para el diseño optimizado en área.

Cantidad de DRCs = 9

En las Figuras 4, 5 y 6 se muestran los diferentes mapas de calor obtenidos. Para estos mapas se utilizó una división de 100x100.

III. CONCLUSIÓN

Lo primero que se pudo observar de esta etapa del proyecto es que los tiempos de place&route pueden aumentar mucho con un poco más de complejidad en el diseño. Por lo que si un diseño tiene muchos componentes es bueno reservar con anticipación el tiempo que se tardará ejecutando estas etapas.

En ambos diseños aparecieron diferentes violaciones de DRC's, para el diseño optimizado en área se encontraron 2 violaciones y para el diseño optimizado en rendimiento se

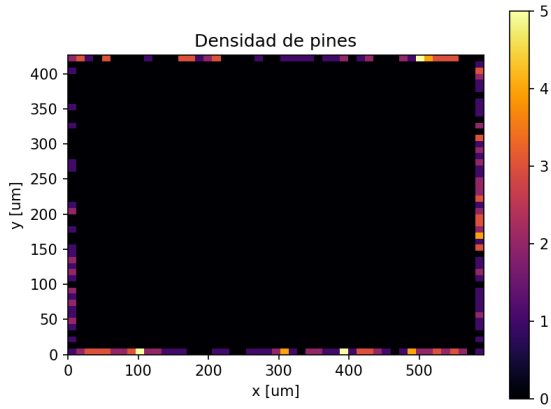


Figura 3. Mapa de densidad de pines para el diseño optimizado en área.

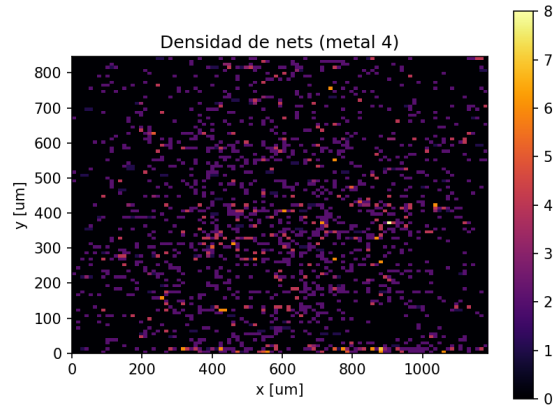


Figura 5. Mapa de densidad de nets (metal 4) para el diseño optimizado en rendimiento.

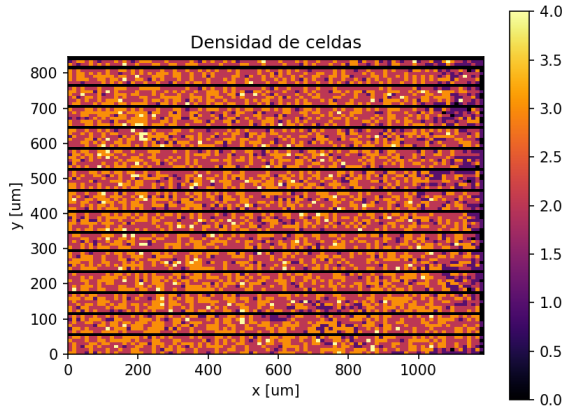


Figura 4. Mapa de densidad de celdas para el diseño optimizado en rendimiento.

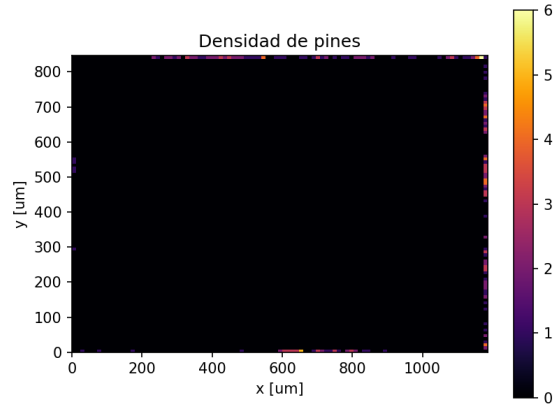


Figura 6. Mapa de densidad de pines para el diseño optimizado en rendimiento.

encontraron 9 violaciones. Si tuviera que asumir de primera mano, teniendo en cuenta que el diseño de rendimiento contiene 4 módulos muy similares a cada módulo del diseño de área, diría que en el diseño optimizado en rendimiento se replican 2 violaciones por módulo, es decir 8 violaciones, y la última puede que pertenezca al módulo de control que contiene este diseño, para un total de 9 violaciones.

Respecto a los mapas de calor, o densidad, fue muy interesante ver cómo están distribuidos los materiales en el área del diseño, el placement. Se distinguen claramente en los mapas de densidad de celdas las secciones en las cuales se ubican las celdas estándar y por donde iría la red de potencia para alimentar estas celdas (refiriéndome a las secciones oscuras). Además estos mapas de densidad de celdas fueron bastante representativos, por lo que con seguridad se puede aproximar

la cantidad de celdas en una región. Por ejemplo, en el diseño optimizado en rendimiento la máxima cantidad de celdas en un cuadro es de 5, como se observa en la barra de color en la Figura 4.

Una justificación para las dimensiones de la cuadrícula para cada diseño fue observar que el diseño optimizado en área entra en alrededor de $600 \times 430 \mu\text{m}^2$, lo cual se puede ubicar en el diseño optimizado en rendimiento y observar más o menos dónde entra cada submódulo.

En los mapas de densidad de metal se pueden ver los pequeños segmentos de metal a través del diseño, ubicados a las alturas definidas el archivo `.def`, y además se puede observar una representación de cuánto metal pasa por cada sección de la cuadrícula. La densidad de metal 4 parece ser relativamente baja si se calcula que: entre segmento y

segmento hay $0.8 \mu\text{m}$ (esto se obtuvo al hacer las diferencias entre las coordenadas x inicial y final de cada segmento). Cada cuadro tiene un largo de alrededor de $12 \mu\text{m}$ (en el diseño optimizado en área), con alrededor de 11 tracks por cuadro, es decir unos $132 \mu\text{m}$ de tracks por cuadro. La máxima cantidad de segmentos completos sería de 3 ($6/2$, por el punto inicial y final), es decir $3 \cdot 0,8 = 2,4$, lo cual es un 1.82% de utilización *máxima* por cuadrado, lo cual es bajo.

Finalmente en los mapas de densidad de pines se observan *claramente* las posiciones donde estarían ubicados los pines en el diseño, y se observa la máxima cantidad de pines en una región específica. Por ejemplo, en el diseño optimizado en área la máxima densidad de pines fue de 5 pines en una región. Además se observa que ambos diseños la herramienta tiene predilección por ubicar los pines arriba y a la derecha, después abajo y finalmente a la izquierda del diseño.

REFERENCIAS

- [1] T. Edwards, *Qflow*. dirección: <https://github.com/RTimothyEdwards/qflow>.
- [2] *matplotlib.pyplot.hist2d*. dirección: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist2d.html.
- [3] *Visualization with Python*. dirección: <https://matplotlib.org/>.
- [4] *Welcome to Python.org*. dirección: <https://www.python.org/>.
- [5] *Pandas*. dirección: <https://pandas.pydata.org/>.
- [6] *Numpy*. dirección: <https://numpy.org/>.