

Proyecto #3 Microelectrónica

Axel Gabriel Hernández Masís

Carné: B73690

José Pablo Camacho Cerdas

Carné: B71463

I. INTRODUCCIÓN

Una función hash es aquella la cual asigna datos de tamaño arbitrario a valores de tamaño fijo, es decir un algoritmo el cual toma un bus de datos y lo transforma en una nueva serie de caracteres con una longitud de salida fija.

II. EJECUCIÓN DE ETAPAS DE FLUJO DE DISEÑO EN ARQUITECTURAS

En la tercera parte del proyecto se realizó una descripción de las métricas obtenidas en los flujos de diseño utilizando la herramienta **Qflow**. En las siguientes secciones se presentan métricas, gráficos y tiempos obtenidos para cada arquitectura diseñada las cuales en los proyectos anteriores se presentaron con optimización en área y rendimiento.

III. OPTIMIZACIÓN EN ÁREA

Para esta optimización se utilizó la misma arquitectura propuesta en la primera parte del proyecto, se presenta a continuación:

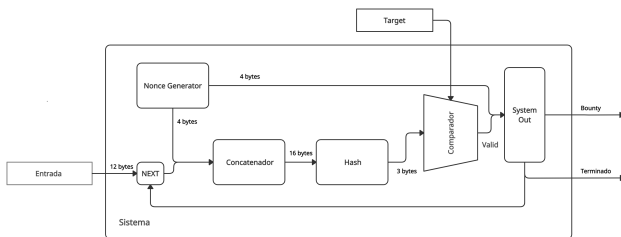


Figura 1: Arquitectura optimizada en área. Creación propia

También es importante anotar que se utilizó el diseño de RTL utilizado para la segunda parte del proyecto.

III-A. Etapas de flujo de diseño

Para esta sección se dio la ejecución de las etapas de diseño mediante el tool **Qflow** hasta el DRC el cual falló debido a que se encontraron errores de DRC en el diseño. A continuación se presenta una tabla con la duración de las etapas del flujo de diseño:

Etapas	Inicio	Fin	Duración
Synthesis	14:30:24	14:30:30	00:00:06
Placement	14:30:37	14:36:08	00:06:29
STA	14:36:08	14:36:09	00:00:01
Route	14:36:09	14:42:11	00:06:02
Post-Route STA	14:42:11	14:42:12	00:00:01
Migration	14:42:12	14:42:58	00:00:46
DRC	14:42:58	14:43:58	00:01:00

Tabla I: Duración de etapas de diseño para optimización en área. Creación propia.

Seguido a esto es importante anotar en la última etapa de DRC, se encontraron **7 errores de DRC** en el layout.

III-B. Mapa de densidad de celdas

Para esta parte del proyecto se realizó un mapa de densidad de celdas a partir del archivo `systemcomplet.def` generado por el tool, en este caso propiamente se utilizó el lenguaje **python** con el cual se desarrolló una lógica para obtener los datos y graficarlos con el fin de poder reutilizar este script con cualquier archivo `.def` generado por esta herramienta para algún diseño distinto al presente.

La lógica consta en realizar un análisis de los datos del archivo, cuando se encuentra el string **COMPONENTS** por primera vez se levanta un flag el cual representa el inicio de la toma de datos, seguido a esto se toman todos los datos que no tengan en su línea la palabra **FILL** las cuales como lo dice el enunciado no se deben de tomar en cuenta para realizar el mapa de densidad. Seguido a esto toda celda que no posea esta palabra va a ser analizada y se tomarán los datos que se encuentran entre paréntesis, los cuales representan las coordenadas X, Y en el plano que representa el área del diseño. Por último se añaden estos valores en dos listas, una para cada coordenada tanto X como Y, por último con estas listas se realiza una graficación de tipo mapa de calor la cual representa colores amarillos e intensos para una mayor densidad de celdas y colores más oscuros para una menor densidad, el código fuente adjunto posee los comentarios y funcionalidad del código de forma más detallada. Para este tipo de mapas de densidad se utilizó la librería `scipy` y mediante el `Gaussian KDE` se pudo realizar la distribución de las celdas por medio de las coordenadas obtenidas. A continuación se presentan los mapas generados para representar la densidad de las celdas, donde los ejes X y Y representan respectivamente los ejes en los cuales se posicionan las celdas en el diseño:

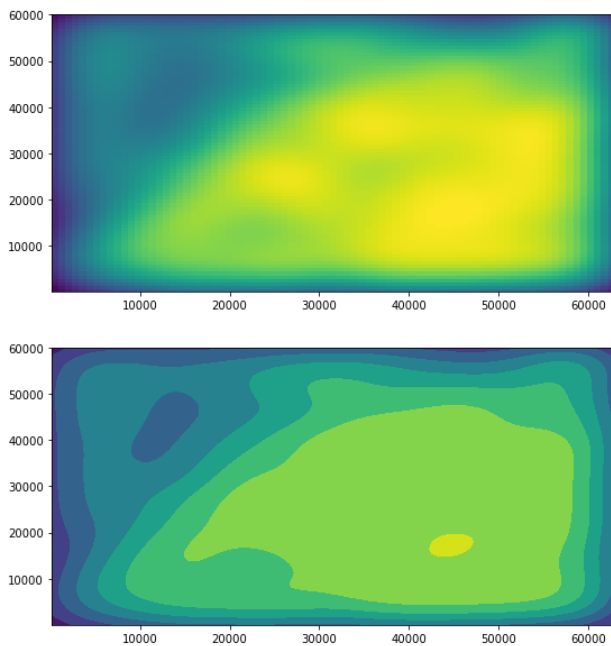


Figura 2: Densidad de celdas para el diseño con optimización de área. Creación propia

De la figura 2 podemos notar una mayor densidad de celdas en la parte derecha inferior del diseño, esto se debe a la forma en que la herramienta procesa la localización de las celdas para obtener un mejor rendimiento.

III-C. Mapa de congestión de metal 4 - Área

Para la sección de congestión de metal 4 se quiere determinar el promedio de tracks de metal disponibles por cuadrado y calcular el porcentaje de tracks usados, para esto se realizaron 3 mapas distintos, el primer mapa el cual describe la longitud del metal y como es distribuido en el sistema, el segundo son los bloques divididos con el fin de obtener el numero de metales dentro de cada celda y el último es un mapa el cual indica el promedio de celdas dentro de cada bloque.

Para la implementación de los mapas de calor, este se desarrolla mediante el lenguaje de programación **python** en el archivo **metal_area**, esto determina los valores dentro del metal 4, los valores fueron tomados al igual que la sección de celdas del archivo **systemcomplete.def** haciendo el inicio de su búsqueda en NETS y tomando las filas que poseyeran el nombre **NEW METAL4**, así mismo mediante un ciclo for y un convertidor de string a enteros. Estos valores se dividieron en 3 vectores, X1,X2 y Y1, dando la longitud de la celda y la posición en Y de estos, teniendo un total de 2528 metales distribuidos en todo el bloque. La Figura 3 muestra la posición de metal 4 en el mapa.

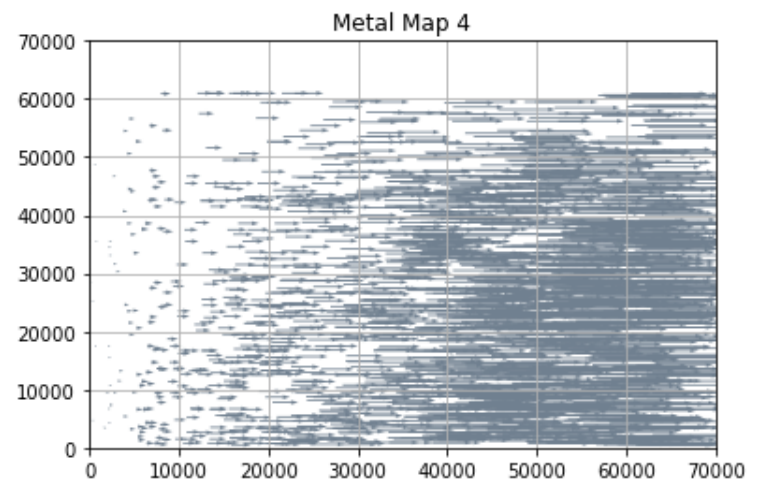


Figura 3: Mapa de metal. Creación propia

Como se observa en la Figura 3 los metales se posicionan en mayor parte en la derecha donde se observan el mayor numero de celdas.

Con esto, se pueden dividir el mapa en secciones y con esto contar el numero de celdas dentro de cada bloque, cabe recalcar que este es contado aunque entre parcialmente al bloque. Para la implementación de los bloques se crearon matrices 20x20 y se creo un ciclo for en el cual suma un valor si lo encuentra dentro del radio del tamaño, en este caso sería de 3000x3000, además para contar los metales que pasan celdas, estos fueron sumados de manera manual y agregados al código con el fin de sumar la matriz de celdas con las celdas encontradas, con lo cual se llegó a lo mostrado en la Figura 4

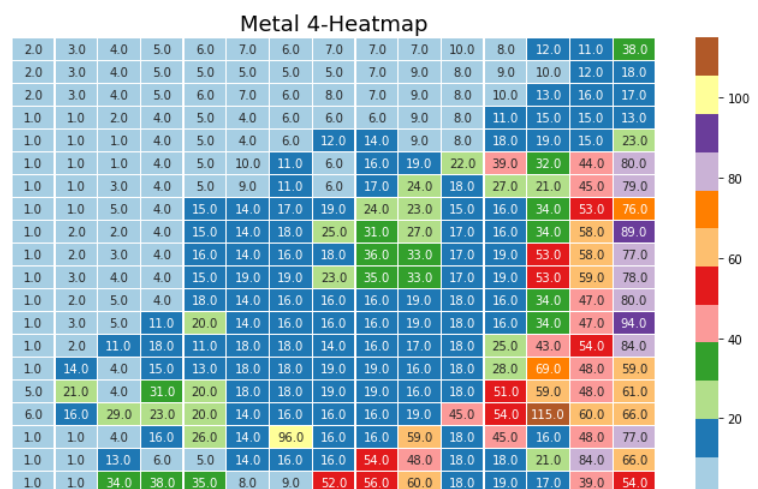


Figura 4: Mapa de calor- Cantidad de metal 4. Creación propia

Se observa en la Figura 4, al igual que la Figura 3 determina que la mayor cantidad de metal se encuentra en la esquina inferior derecha, así mismo se ve que llega a tener en un bloque hasta 115 metales, mientras que en el sector

inferior izquierdo, se encuentran bloques con un sólo metal 4.

Así mismo para la sección porcentual de metales por bloque, se toma la celda mayor como referencia 1 y las demás son divididas a partir de esta, esto ya que como se observa en la sección opcional este tipo de metales está alejado de los pines.

$$Porcentaje_{celdas} = \frac{N * 100}{N_{max}}$$

Donde N es el numero de metales por bloque, Nmax es el valor mayor de metales en un bloque y 100 determina el parámetro de porcentajes se va a obtener de cada bloque.

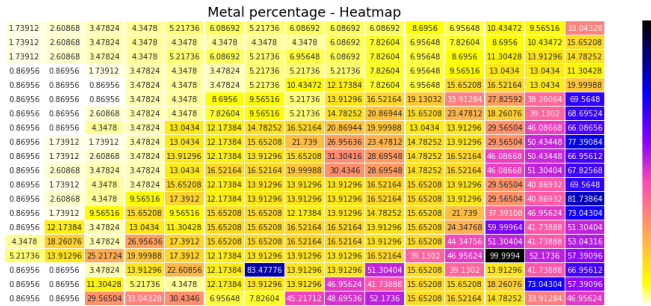


Figura 5: Mapa de calor- Porcentaje metal 4. Creación propia

En la Figura 5 se muestra el porcentaje de metales 4 en los bloques y el el costado derecho la tabla de cero a 100 de este.

IV. OPTIMIZACIÓN EN DESEMPEÑO

Para esta optimización se utilizó la misma arquitectura propuesta en la primera parte del proyecto, se presenta a continuación:

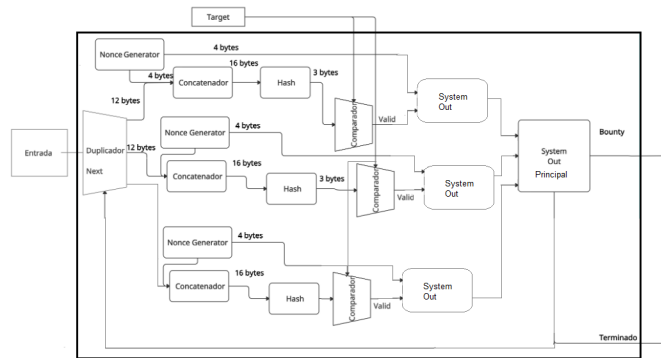


Figura 6: Bloque Desempeño. Creación propia

También es importante anotar que se utilizó el diseño de RTL utilizado para la segunda parte del proyecto.

IV-A. Etapas de flujo de diseño

Para esta sección se dio la ejecución de las etapas de diseño mediante el tool **Qflow** hasta el DRC el cual falló debido a que se encontrar errores de DRC en el diseño. A continuación se presenta una tabla con la duración de las etapas del flujo de diseño:

Etapas	Inicio	Fin	Duración(s)
Synthesis	15:24:10	15:24:28	00:00:18
Placement	15:24:39	15:54:52	00:30:13
STA	15:54:52	15:54:53	00:00:01
Route	15:54:53	16:26:15	00:31:22
Post-Route STA	16:26:16	16:26:18	00:00:02
Migration	16:26:18	16:30:46	00:04:28
DRC	16:30:46	16:31:46	00:01:00

Tabla II: Duración de etapas de diseño para optimización en desempeño. Creación propia.

Seguido a esto es importante anotar en la última etapa de DRC, se encontraron **38 errores de DRC** en el layout.

IV-B. Mapa de densidad de celdas

Para esta parte del proyecto se realizó un mapa de densidad de celdas a partir del archivo *systemcomplet.def* generado por el tool, en este caso propiamente se utilizó el lenguaje **python** con el cual se desarrolló una lógica para obtener los datos y graficarlos con el fin de poder reutilizar este script con cualquier archivo .def generado por esta herramienta para algún diseño distinto al presente. La lógica es la misma presentada en la optimización en área ya que se espera que este script sea universal y se pueda utilizar con cualquier diseño. A continuación se presentan los mapas generados para representar la densidad de las celdas, donde los ejes X y Y representan respectivamente los ejes en los cuales se posicionan las celdas en el diseño:

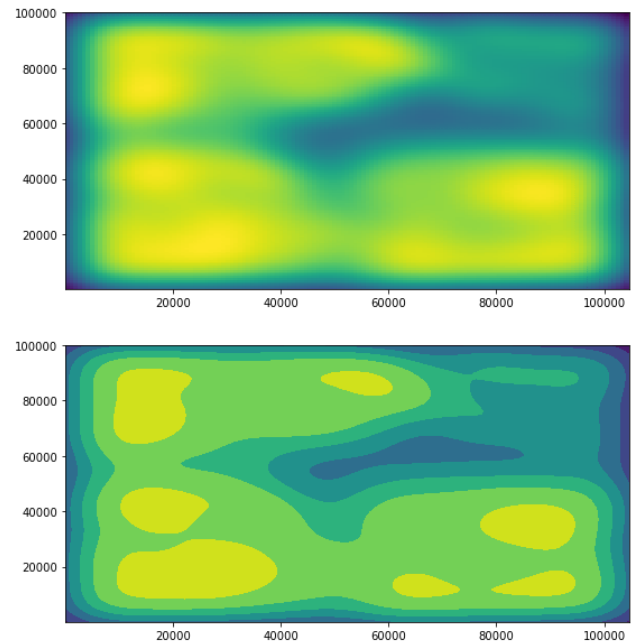


Figura 7: Densidad de celdas para el diseño con optimización de rendimiento. Creación propia

De la figura 7 podemos notar una mayor densidad de celdas en la parte derecha superior del diseño, esto se debe a la forma en que la herramienta procesa la localización de las celdas para obtener un mejor rendimiento. Se puede notar como dato interesante que se crearon 3 sistemas en paralelo

para aumentar el rendimiento del diseño y como se puede notar, existen 3 clusters de celdas que se notan a simple vista los cuales muy probablemente vana representar esta lógica de diseño.

IV-C. Mapa de congestión de metal 4 - Desempeño

La implementación del código es la misma que la de la sección III-C con lo cual se llega a los siguientes resultados:

En la Figura 8 se observa como posee mayor cantidad de celdas me metal 4 que en la sección de área, así mismo se puede observar donde se encuentra la mayor concentración de metal.

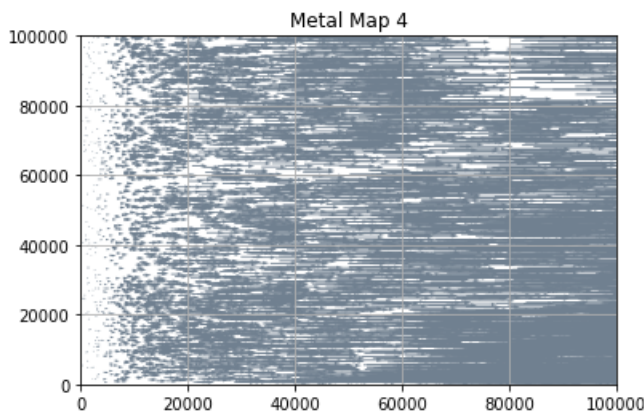


Figura 8: Mapa de metal4-Desempeño. Creación propia

En la figura 9 se observa cuantos metales están distribuidos dentro de cada bloque.

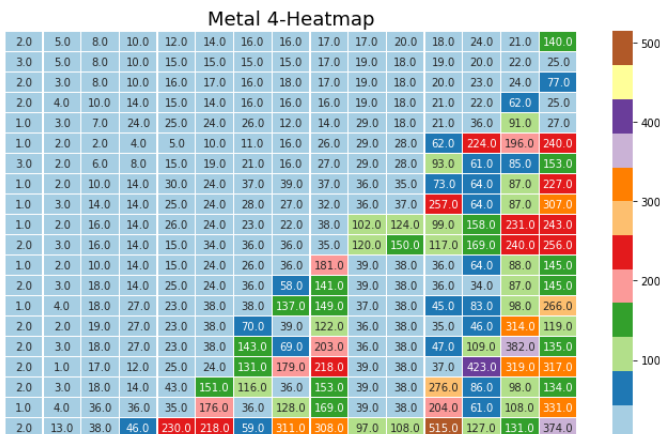


Figura 9: Cantidad de metal 4-Desempeño. Creación propia

En la figura 8 se observa el porcentaje metales distribuidos dentro de cada bloque, tomando como referencia el bloque con más metales.

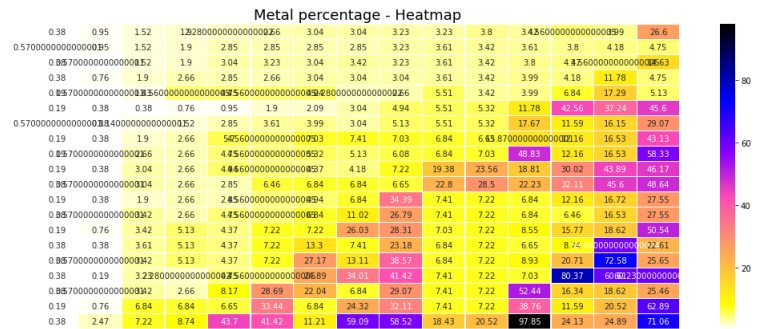


Figura 10: Porcentaje Metal4- Desempeño. Creación propia

Así mismo como se observa en la figura 10 el mapa de calor posee limitaciones con lo cual no pueden ser mostrados por la cantidad de decimales posibles, sin embargo estos fueron impresos en la terminal para exponer el porcentaje de metales por cada bloque. Esto se puede observar en la Figura 11.

```
[[ 0.38 0.95 1.52 1.9 2.28 2.66 3.04 3.04 3.23 3.23 3.8 3.42
4.56 3.99 26.6 ]
[ 0.57 0.95 1.52 1.9 2.85 2.85 2.85 2.85 3.23 3.61 3.42 3.61
3.8 4.18 4.75 ]
[ 0.38 0.57 1.52 1.9 3.04 3.23 3.04 3.42 3.23 3.61 3.42 3.8
4.37 4.56 14.63 ]
[ 0.38 0.76 1.9 2.66 2.85 2.66 3.04 3.04 3.04 3.61 3.42 3.99
4.18 11.78 4.75 ]
[ 0.19 0.57 1.33 4.56 4.75 4.56 4.94 2.28 2.66 5.51 3.42 3.99
6.84 17.29 5.13 ]
[ 0.19 0.38 0.38 0.76 0.95 1.9 2.09 3.04 4.94 5.51 5.32 11.78
42.56 37.24 45.6 ]
[ 0.57 0.38 1.14 1.52 2.85 3.61 3.99 3.04 5.13 5.51 5.32 17.67
11.59 16.15 29.07 ]
[ 0.19 0.38 1.9 2.66 5.7 4.56 7.03 7.41 7.03 6.84 6.65 13.87
12.16 16.53 43.13 ]
[ 0.19 0.57 2.66 2.66 4.75 4.56 5.32 5.13 6.08 6.84 7.03 48.83
12.16 16.53 58.33 ]
[ 0.19 0.38 3.04 2.66 4.94 4.56 4.37 4.18 7.22 19.38 23.56 18.81
30.02 43.89 46.17 ]
[ 0.38 0.57 3.04 2.66 2.85 4.56 6.84 6.84 6.65 22.8 28.5 22.23
32.11 45.6 48.64 ]
[ 0.19 0.38 1.9 2.66 2.85 4.56 4.94 6.84 34.39 7.41 7.22 6.84
12.16 16.72 27.55 ]
[ 0.38 0.57 3.42 2.66 4.75 4.56 6.84 11.02 26.79 7.41 7.22 6.84
6.46 16.53 27.55 ]
[ 0.19 0.76 3.42 5.13 4.37 7.22 7.22 26.03 28.31 7.03 7.22 8.55
15.77 18.62 50.54 ]
[ 0.38 0.38 3.61 5.13 4.37 7.22 13.3 7.41 23.18 6.84 7.22 6.65
8.74 59.66 22.61 ]
[ 0.38 0.57 3.42 5.13 4.37 7.22 27.17 13.11 38.57 6.84 7.22 8.93
20.71 72.58 25.65 ]
[ 0.38 0.19 3.23 2.28 4.75 4.56 24.89 34.01 41.42 7.41 7.22 7.03
80.37 60.61 60.23 ]
[ 0.38 0.57 3.42 2.66 8.17 28.69 22.04 6.84 29.07 7.41 7.22 52.44
16.34 18.62 25.46 ]
[ 0.19 0.76 6.84 6.84 6.65 33.44 6.84 24.32 32.11 7.41 7.22 38.76
11.59 20.52 62.89 ]
[ 0.38 2.47 7.22 8.74 43.7 41.42 11.21 59.09 58.52 18.43 20.52 97.85
24.13 24.89 71.06 ]]
```

Figura 11: Porcentaje Metal4 2- Desempeño. Creación propia

V. RUBRO OPCIONAL

V-A. Mapas de densidad de pines

Para esta parte del proyecto se realizó un mapa de densidad de pines a partir del archivo `systemcomplet.def` como lo realizado en los puntos anteriores.

La lógica consta en realizar un análisis de los datos del archivo, cuando se encuentra el string **PINS** por primera vez se levanta un flag el cual representa el inicio de la toma de datos, seguido a esto se toman todos los datos que tengan en su línea la palabra **PLACED** las cuales como lo dice el enunciado no se deben de tomar en cuenta para realizar el mapa de densidad de pines hasta que se encuentra de nuevo la palabra **END PINS** que representa el fin de los

datos relacionados a la localización de pines. Seguido a esto toda celda que posea esta palabra va a ser analizada y se tomarán los datos que se encuentran entre paréntesis, los cuales representan las coordenadas X, Y en el plano que representa el área del diseño. Por último se añaden estos valores en dos listas, una para cada coordenada tanto X como Y, por último con estas listas se realiza una graficación de tipo mapa de calor la cual representa colores amarillos e intensos para una mayor densidad de celdas y colores más oscuros para una menor densidad. A continuación se presentan los mapas de calor:

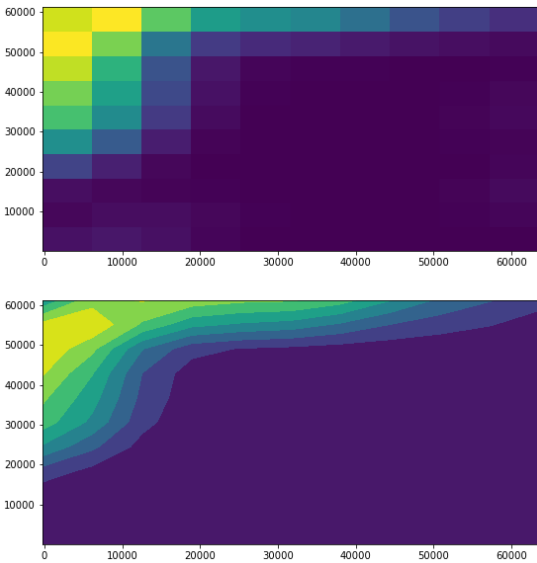


Figura 12: Densidad de pines para el diseño con optimización de área. Creación propia

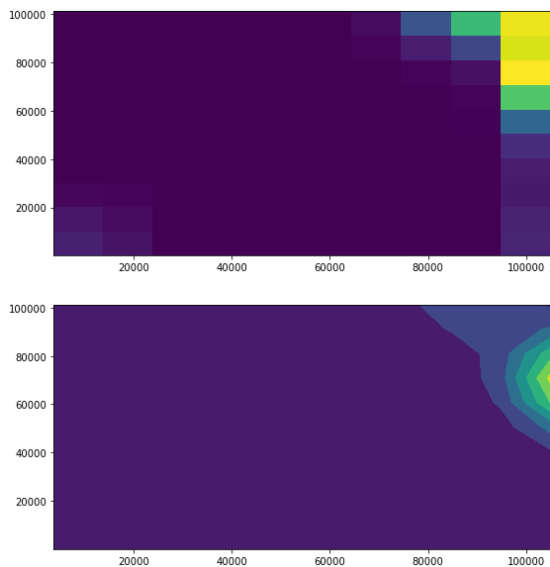


Figura 13: Densidad de pines para el diseño con optimización de rendimiento. Creación propia

De las figuras 12 y 13 podemos notar una mayor densidad

de pines en los extremos, estos practicamente se distribuyen en las zonas externas del diseño ya que representan la comunicación con el mundo exterior, lo que nos da como resultado una correcta interpretación de los datos. También es claro que los pines se encuentran agrupados normalmente en los lugares en los que no existe gran cantidad de celdas, como se puede ver en las figuras 2 y 7, ya que normalmente se destina un espacio para la entrada de señales del mundo exterior al diseño.

VI. CONCLUSIONES

Como se pudo observar en las tablas de duración de etapas, los tiempos de ejecución para cada una de estas es mayor conforme aumenta el tamaño del diseño, en este caso el diseño con optimización en rendimiento tuvo un aumento significativo específicamente en la etapa de placement y routing. Se recomienda realizar este tipo de procesos con anticipación con el fin de poder entender y analizar las etapas así como sus archivos tipo log los cuales proveen de mucha información útil.

Con respecto a los mapas de densidad de celdas, se pudo observar una correcta interpretación de las coordenadas así como de el mapa gráfico en sí, donde los colores más intensos representan una mayor densidad de celdas en ese punto del plano del diseño. Se pudo notar que la optimización en rendimiento posee 3 grandes clusters de acumulación de celdas los cuales muy probablemente representan los 3 sistemas que se conectan en paralelo para obtener un mayor rendimiento a la hora de procesar las operaciones.

Para la sección del mapa de congestión del metal 4, se realizó de manera exitosa el mapa de calor del metal 4, así mismo se determinaron la cantidad de metales de cada bloque y además se obtuvo de manera exitosa el porcentaje de estos metales por bloque. Así mismo, se observó la ubicación de la mayoría de metales, esto respecto a que deben estar alejados de los pines y tienen que tener conexión directa con las celdas. Además se puede observar que el número de metales aumenta en el Heatmap de desempeño lo cual es normal ya que para este se usaron 3 circuitos en paralelo, aumentando el número de metales.

Gracias a el flujo de diseño y las etapas provistas por Qflow, se obtuvieron obtener métricas realistas sobre ambas arquitecturas implementadas así como idea bastante clara de la densidad de celdas y metales en el diseño realizado.

VII. CÓDIGO FUENTE

El código fuente relacionado al presente proyecto se encuentra en el repositorio: <https://github.com/jose-543/Proyecto3-Microelectronica.git>, este mismo repositorio posee un archivo README el cual presenta las instrucciones para compilar y ejecutar el código así como los archivos de tipo .log y .DEF generados por la herramienta Qflow en donde se pueden encontrar más detalles sobre las distintas etapas del flujo de diseño.