

IE0411 Microelectrónica -G01-

Tarea 4: Qflow

Jorge Muñoz Taylor (A53863) - jorge.munoztaylor@ucr.ac.cr

II-2020

Índice

1. Desarrollo	3
1.1. Tecnologías	3
2. Resultados	4
2.1. Parte A	4
2.1.1. Determinar el área y frecuencia de cada circuito. Deben aparecer las capturas del layout generado.	4
2.1.2. Determinar el número y el tipo (INVX1, INVX2, etc) de compuertas utilizados para generar el circuito.	5
2.1.3. Mostrar los paths con mayor retardo y los de menor retardo para los Flop to Flop y Pin to Flop.	6
2.2. Parte B	8
2.2.1. Utilice la librería osu050 y realice los pasos de la sección A.	8
2.3. Parte C	12
2.3.1. Genere un informe con una discusión de los resultados de las dos secciones. Se espera que analice la frecuencia, área, número de compuertas, retardos usando los conocimientos adquiridos en el curso.	12
3. Conclusiones	13
4. Recomendaciones	14
Referencias	15

1. Desarrollo

Para la obtención de la frecuencia máxima y las medidas de los circuitos en Qflow se utilizó el comando[1] del código 1.

Código 1

```
qflow sta ARCHIVO
```

Donde ARCHIVO corresponde al nombre del archivo que se va a analizar, cabe resaltar que dependiendo de la versión de Qflow que se tenga instalada se debe colocar o no la extensión del archivo verilog.

Para generar el layout se utilizará el programa magic, con este mismo software se calculará el área de cada circuito. Los comandos necesarios para ello se muestran en el código 2, donde el área se utiliza multiplicando el valor de la longitud horizontal por la longitud vertical del circuito.

Código 2

```
lef read /usr/share/qflow/tech/osu035/osu035_stdcells.lef
def read ../layout/CIRCUITO.def
measure horizontal
measure vertical
```

1.1. Tecnologías

Para mapear los diseños creados en verilog se utilizan bibliotecas que contienen el tipo de compuertas que se utilizarán junto con sus características físicas, para el proyecto se usarán las bibliotecas *OSU*¹ propiamente las OSU035 y OSU050. El número indica el tamaño de los transistores que componen las compuertas de la biblioteca.

¹Acrónimo de *Oklahoma State University*, que corresponde al nombre de la universidad donde fueron desarrolladas estas bibliotecas.

f_{max} [MHz]	461,564
Medida horizontal [μm]	276
Medida vertical [μm]	207
Area [nm^2]	57,132

Cuadro 2: Área y frecuencia máxima del circuito *uart.v*.

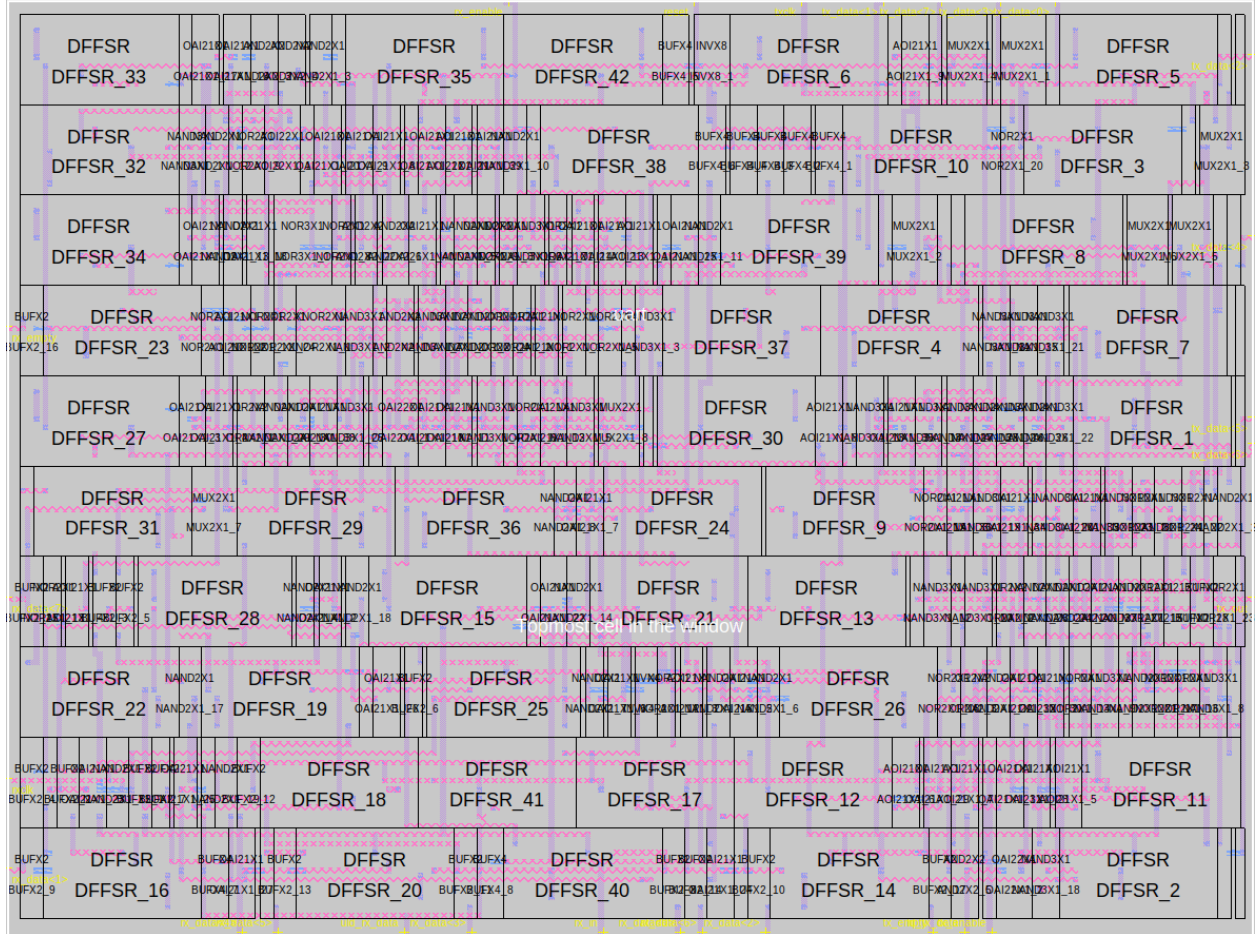


Figura 2: Layout generado del circuito *uart.v* utilizando la biblioteca de componentes por defecto de Qflow.

2.1.2. Determinar el número y el tipo (INVX1, INVX2, etc) de compuertas utilizados para generar el circuito.

El tipo y cantidad de compuertas se obtienen al utilizar el comando *qflow synthesize place route Arbiter/uart* en la terminal de *Ubuntu*. En los cuadros 3 y 4 se muestra el conteo de componentes para los circuitos *Arbiter.v* y *uart.v* respectivamente.

- Arbiter

Compuerta	Cantidad
AND	17
AOI3	6
AOI4	6
MUX	2
NAND	6
NOR	6
NOT	8
OAI3	2
OAI4	3
OR	7

Cuadro 3: Compuertas generadas por Qflow durante el proceso de síntesis para el circuito *Arbiter.v*.

- UART

Compuerta	Cantidad
AND	48
AOI3	4
MUX	101
NAND	3
NOR	19
NOT	22
OAI3	10
OAI4	8
OR	15
XNOR	2
XOR	12

Cuadro 4: Compuertas generadas por Qflow durante el proceso de síntesis para el circuito *uart.v* usando la biblioteca *OSU035*.

2.1.3. Mostrar los paths con mayor retardo y los de menor retardo para los Flop to Flop y Pin to Flop.

- Arbiter

En el cuadro 5 se puede observar que el path *DDFPOSX1_6/CLK to DFFPOSX1_7/D* tiene el máximo retardo *flop to flop* mientras que el path input *pin req1 to DFFPOSX1_7/D* tiene el máximo retardo *pin to flop*.

Para los retardos mínimos, tenemos que para *flop to flop* corresponde al path *DDFPOSX1_4/CLK to DFFPOSX1_4/D* y para el *pin to flop* corresponde al input *pin clk to DFFPOSX1_8/CLK*.

Retardos	Máximo (ps)	Mínimo (ps)
Flop to flop	1830,95	430,124
Pin to flop	1165,19	0

Cuadro 5: Retardos flop-to-flop y pin-to-flop del circuito *Arbiter.v*.

- UART En el cuadro 6 se tiene, para el retardo máximo flop to flop el path *DFFSR_13/CLK to DFSR_1/D* y para el pin to flop el path *input pin rx_enable to DFSR_28/D*.

Para los retardos mínimos tenemos, flop to flop: path *DFFSR_40/CLK to DFSR_41/D*, pin to flop: *path input pin txclk to DFSR_14/CLK*.

Retardos	Máximo (ps)	Mínimo (ps)
Flop to flop	2166,55	322,421
Pin to flop	1211,19	0

Cuadro 6: Retardos flop-to-flop y pin-to-flop del circuito *uart.v*.

2.2. Parte B

2.2.1. Utilice la librería osu050 y realice los pasos de la sección A.

Se realizaron los mismos puntos de la parte A pero en lugar de utilizar la biblioteca de componentes por defecto se utilizó la biblioteca osu050 siguiendo los pasos descritos en el enunciado.

■ Arbiter

Para los retardos máximos (descritos en el cuadro 7) se tiene, flop to flop: *path DFFPOSX1_6/CLK to DFFPOSX1_8/D*, pin to flop: *path input pin req1 to DFFPOSX1_8/D*.

Para los retardos mínimos (descritos en el cuadro 7), flop to flop: *path DFFPOSX1_4/CLK to DFFPOSX1_4/D*, pin to flop: *pin clk to DFFPOSX1_8/CLK*.

Retardos	Máximo (ps)	Mínimo (ps)
Flop to flop	2049,88	552,315
Pin to flop	1301,74	0

Cuadro 7: Retardos flop-to-flop y pin-to-flop del circuito *Arbiter.v* utilizando la biblioteca de componentes *osu050*.

Los valores de frecuencia máxima y medidas del circuito *Arbiter.v* se muestran en el cuadro 8, el *layout* del mismo circuito con la biblioteca *osu050* es el presentado en la figura 3.

f_{max} [Hz]	487,834
Medida horizontal [μm]	190
Medida vertical [μm]	160
Area [nm^2]	30,4

Cuadro 8: Área y frecuencia máxima del circuito *Arbiter.v* usando la biblioteca osu050.

En el cuadro 9 se presenta el conteo de compuertas usadas durante la síntesis.

Compuerta	Cantidad
AND	17
AOI3	6
AOI4	6
MUX	2
NAND	6
NOR	6
NOT	8
OAI3	2
OAI4	3
OR	7

Cuadro 9: Compuertas generadas por Qflow durante el proceso de síntesis para el circuito *Arbiter.v* usando la biblioteca *OSU050*.

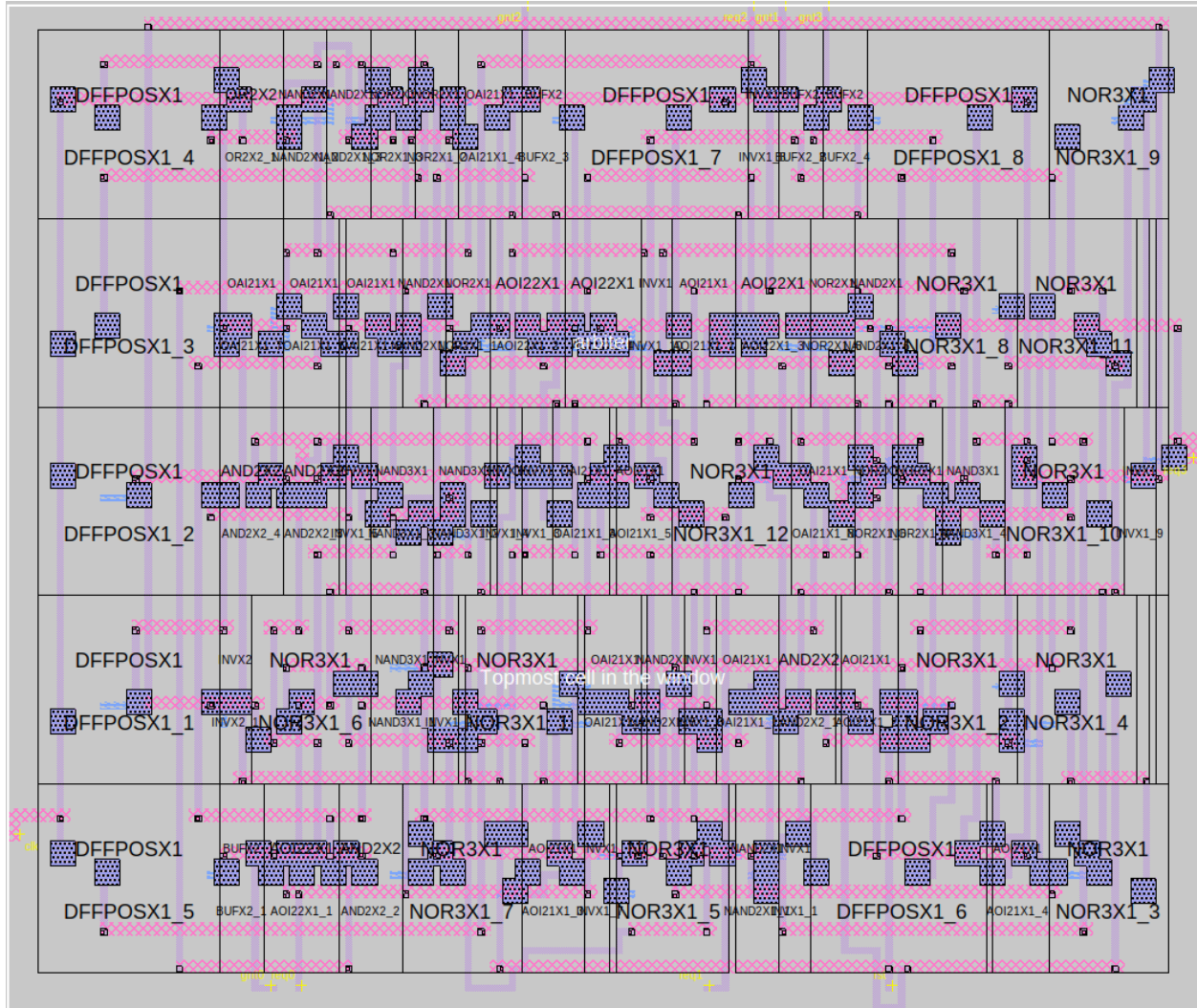


Figura 3: Layout generado del circuito *Arbitrer.v* utilizando la biblioteca de componentes osu050 de Qflow.

■ UART

Para los retardos máximos (cuadro 10), flop to flop: *path DFFSR_33/CLK to DFFSR_28/D*, para pin to flop: *path input pin rx_enable to DFFSR_28/D*.

Para los retardos mínimos (cuadro 10), flop to flop: *path DFFSR_40/CLK to DFFSR_41/D*, para pin to flop: *path input pin txclk to DFFSR_14/CLK*.

Retardos	Máximo (ps)	Mínimo (ps)
Flop to flop	2669,63	399,627
Pin to flop	1302,42	0

Cuadro 10: Retardos flop-to-flop y pin-to-flop del circuito *uart.v* utilizando la biblioteca de componentes osu050.

La frecuencia máxima posible del circuito *uart.v* junto con sus medidas están tabuladas en el cuadro 11, su *layout* con el conjunto de componentes de la biblioteca *osu050* se muestra en la figura 4.

f_{max} [Hz]	374,583
Medida horizontal [μm]	292
Medida vertical [μm]	214
Area [nm^2]	62,488

Cuadro 11: Área y frecuencia máxima del circuito *uart.v* usando la biblioteca *osu050*.

En el cuadro 12 se presenta el conteo de compuertas usadas durante la síntesis.

Compuerta	Cantidad
AND	48
AOI3	4
MUX	101
NAND	3
NOR	19
NOT	22
OAI3	10
OAI4	8
OR	15
XNOR	2
XOR	12

Cuadro 12: Compuertas generadas por Qflow durante el proceso de síntesis para el circuito *uart.v* usando la biblioteca *OSU050*.

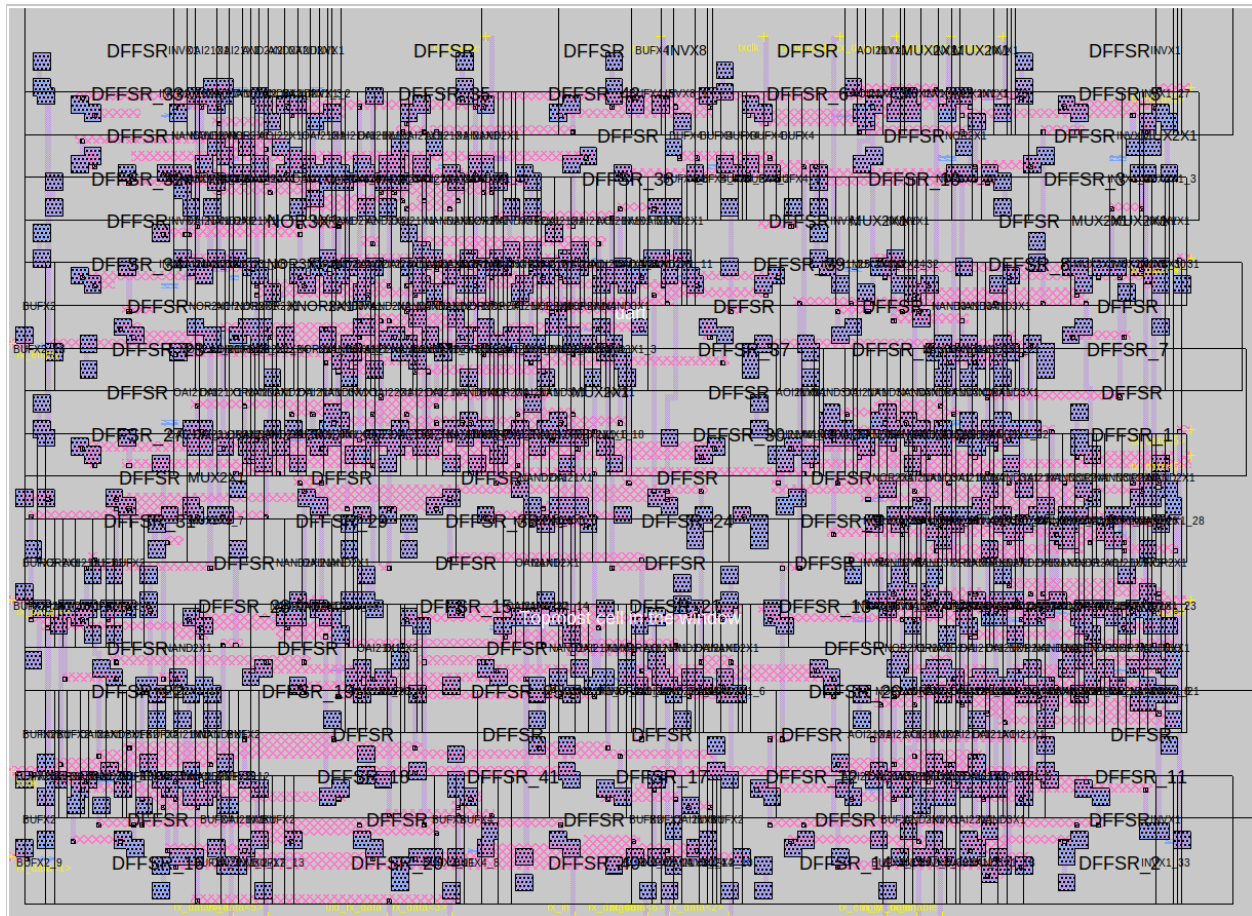


Figura 4: Layout generado del circuito *uart.v* utilizando la biblioteca de componentes osu050 de Qflow.

2.3. Parte C

2.3.1. Genere un informe con una discusión de los resultados de las dos secciones. Se espera que analice la frecuencia, área, número de compuertas, retardos usando los conocimientos adquiridos en el curso.

Se puede notar de los cuadros 1, 2, 8 y 11 hay una relación evidente, el área de los circuitos *arbiter.v-uart.v* es mayor cuando se utiliza la biblioteca *OSU050* lo cual resulta obvio dado que la tecnología implementada en dicha biblioteca utiliza transistores de mayor tamaño que en la biblioteca *OSU035*. Este incremento de tamaño también se ve reflejado en la frecuencia máxima del circuito, en los cuadros mencionados anteriormente también se puede notar como la frecuencia disminuye cuando lo hace el área del circuito. Esto ocurre porque la dimensión de los transistores influye en los tiempos de retardo de las señales afectando así la frecuencia máxima que puede alcanzar el circuito.

El número de compuertas utilizado tanto para el *arbiter* es el mismo para las dos tecnologías utilizadas para la síntesis (sin embargo el número de compuertas colocadas en el layout si cambia entre ambas tecnologías) porque estos datos corresponden al tipo y número de compuertas sin reducir, el proceso de reducción final se da antes de implementar el *layout*; lo mismo ocurre para el circuito *uart* con las dos tecnologías.

De igual forma los retardos *flop to flop* y *pin to flop* son diferentes para cada tecnología, en ambos circuitos los retardos fueron mayores cuando se utilizó la tecnología *OSU050* (excepto para los retardos mínimos *pin to flop* que para ambos circuitos es 0 para ambas tecnologías), la explicación a este resultado recae en el tamaño de los semiconductores, al ser estos de mayor tamaño en la tecnología *OSU050* es esperable que los retardos de sus compuertas sean mayores que con semiconductores más pequeños (la corriente tiene más camino por recorrer además la resistencia aumenta).

3. Conclusiones

- El tamaño de los semiconductores influye directamente en los tiempos de retardo de las compuertas lógicas.
- De igual forma la frecuencia del circuito se ve afectada con la tecnología que se implementa el mismo.
- La teoría estudiada en el curso microelectrónica sobre retardos fue fácilmente apreciable durante los resultados de las síntesis.

4. Recomendaciones

- Tener en cuenta la versión de *Qflow* que se tiene instalada en la máquina, ya que algunos comandos son ligeramente diferentes en base a la versión.
- Conocer las etapas seguidas por *Qflow* para sintetizar los circuitos ya que de esta forma se pueden entender claramente los mensajes desplegados en la terminal.
- Recordar que si no se especifica otra biblioteca de componentes *Qflow* utilizará de forma nativa la biblioteca *osu035*.

Referencias

- [1] TIM. (2019) Qflow tutorial. <http://opencircuitdesign.com/qflow/>.