Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica

IE0411 - Microelectrónica: Sistemas en Silicio I

Tarea 4

Prof. Javier Pacheco Brito

Por:
Marco Chacon Soto, B61868

16 de noviembre de 2020

Índice

Índice de figuras

1. Introducción

El siguiente reporte contiene los resultados obtenidos del anális de los módulos arbiter.v y uart.v con la herramienta Qflow. Entre los datos obtenidos se obtienen el área, la cantidad de compuertas lógicas utilizadas y el timing de Flop to Flop y pin to Flop.

Se incluyen también las observaciones de los resultados obtenidos, el repositorio del git donde se encuentran los archivos y las recomendaciones.

2. Resultados Obtenidos

2.1. Arbiter parte A: Librería osu035.

Para esta sección se crean dos carpetas, una para cada módulo. Dentro de dichas carpetas se corre el siguiente comando

```
qflow sta synthesize place route arbiter
```

Luego, se revisa el archivo synth.log, el cuál contiene el número de compuertas y los resultados de los timings, entre otras cosas. Además de este archivo es necesario revisar el archivo arbiter.rtl.v, el cuál contiene las compuertas sintetizadas para este módulo.

2.1.1. Compuertas

Los resultados obtenidos para el módulo arbiter.v se muestran a continuación.
En las siguientes figuras se muestra la cantidad de compuertas que requiere el diseño, la cantidad de compuertas de la librería ABC que utiliza Qflow y la cantidad de compuertas que se encuentran en el archivo arbiter.rtl.v respectivamente

```
=== arbiter ===

   Number of wires:                 81
   Number of wire bits:             81
   Number of public wires:          18
   Number of public wire bits:      18
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:                 71
     $_AND_                         17
     $_AOI3_                         6
     $_AOI4_                         6
     $_DFF_P_                        8
     $_MUX_                          2
     $_NAND_                         6
     $_NOR_                          6
     $_NOT_                          8
     $_OAI3_                         2
     $_OAI4_                         3
     $_OR_                           7
```

Figura 1: Cantidad de compuertas que requiere el diseño del módulo arbiter.v. Simulado en Qflow

```
6.1.2. Re-integrating ABC results.
ABC RESULTS:            AND2X2 cells:        4
ABC RESULTS:           AOI21X1 cells:        5
ABC RESULTS:           AOI22X1 cells:        4
ABC RESULTS:             INVX1 cells:       11
ABC RESULTS:           NAND2X1 cells:        6
ABC RESULTS:           NAND3X1 cells:        4
ABC RESULTS:            NOR2X1 cells:        6
ABC RESULTS:            NOR3X1 cells:       12
ABC RESULTS:           OAI21X1 cells:        8
ABC RESULTS:             OR2X2 cells:        1
ABC RESULTS:        internal signals:       55
ABC RESULTS:          input signals:       13
ABC RESULTS:         output signals:        8
```

Figura 2: Cantidad de compuertas que utiliza la herramienta para implementar el diseño del módulo arbiter.v. Simulado en Qflow

Las siguientes compuertas se obtuvieron al analizar el archivo arbiter.rtl.v, el cuál se encuentra dentro de la carpeta synthesis

- **INVX1**: 9
- **INVX2**: 1
- **INVX4**: 1
- **BUFX2**: 4
- **OR2X2**: 1
- **NOR2X1**: 6
- **NOR3X1**: 12
- **AND2X2**: 4
- **NAND3X1**: 4
- **NAND2X1**: 6
- **OAI21X1**: 8
- **AOI21X1**: 5
- **AOI22X1**: 4
- **DFFPOSX**: 8

2.1.2. Paths y timings

A continuación se presentan los resultados de los paths con mayor y menor retardo, así como la frecuencia máxima del circuito para el módulo arbiter.v

```
Top 20 maximum delay paths:
Path DFFPOSX1_6/CLK to DFFPOSX1_7/D delay 1830.95 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_7/D delay 1815.13 ps
Path DFFPOSX1_5/CLK to DFFPOSX1_7/D delay 1796.83 ps
Path DFFPOSX1_6/CLK to DFFPOSX1_8/D delay 1796.01 ps
Path DFFPOSX1_6/CLK to DFFPOSX1_6/D delay 1795.77 ps
Path DFFPOSX1_6/CLK to DFFPOSX1_5/D delay 1795.77 ps
Path DFFPOSX1_7/CLK to DFFPOSX1_7/D delay 1787.01 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_8/D delay 1783.18 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_6/D delay 1781.6 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_5/D delay 1781.6 ps
Path DFFPOSX1_5/CLK to DFFPOSX1_8/D delay 1762.03 ps
Path DFFPOSX1_5/CLK to DFFPOSX1_6/D delay 1761.78 ps
Path DFFPOSX1_5/CLK to DFFPOSX1_5/D delay 1761.78 ps
Path DFFPOSX1_7/CLK to DFFPOSX1_8/D delay 1755.23 ps
Path DFFPOSX1_7/CLK to DFFPOSX1_6/D delay 1753.63 ps
Path DFFPOSX1_7/CLK to DFFPOSX1_5/D delay 1753.63 ps
Path DFFPOSX1_2/CLK to DFFPOSX1_6/D delay 1704.88 ps
Path DFFPOSX1_2/CLK to DFFPOSX1_5/D delay 1704.88 ps
Path DFFPOSX1_2/CLK to DFFPOSX1_8/D delay 1703.83 ps
Path DFFPOSX1_2/CLK to DFFPOSX1_7/D delay 1613.71 ps
Computed maximum clock frequency (zero slack) = 546.164 MHz
```

Figura 3: Paths con mayor retardo, Flop to Flop y frecuencia máxima. Simulado en Qflow

```
Top 20 minimum delay paths:
Path DFFPOSX1_5/CLK to output pin gnt0 delay 278.609 ps
Path DFFPOSX1_6/CLK to output pin gnt1 delay 345.816 ps
Path DFFPOSX1_7/CLK to output pin gnt2 delay 350.183 ps
Path DFFPOSX1_8/CLK to output pin gnt3 delay 361.89 ps
Path DFFPOSX1_4/CLK to DFFPOSX1_4/D delay 430.124 ps
Path DFFPOSX1_4/CLK to DFFPOSX1_3/D delay 466.82 ps
Path DFFPOSX1_2/CLK to DFFPOSX1_2/D delay 538.854 ps
Path DFFPOSX1_7/CLK to DFFPOSX1_7/D delay 578.254 ps
Path DFFPOSX1_3/CLK to DFFPOSX1_1/D delay 580.83 ps
Path DFFPOSX1_3/CLK to DFFPOSX1_2/D delay 580.83 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_8/D delay 590.334 ps
Path DFFPOSX1_3/CLK to DFFPOSX1_4/D delay 599.257 ps
Path DFFPOSX1_3/CLK to DFFPOSX1_3/D delay 599.776 ps
Path DFFPOSX1_6/CLK to DFFPOSX1_1/D delay 627.244 ps
Path DFFPOSX1_7/CLK to DFFPOSX1_2/D delay 632.104 ps
Path DFFPOSX1_5/CLK to DFFPOSX1_5/D delay 634.022 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_1/D delay 647.256 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_2/D delay 647.256 ps
Path DFFPOSX1_6/CLK to DFFPOSX1_6/D delay 698.23 ps
Path DFFPOSX1_1/CLK to DFFPOSX1_1/D delay 726.618 ps
Design meets minimum hold timing.
```

Figura 4: Paths con menor retardo, Flop to Flop. Simulado en Qflow

```
Top 20 maximum delay paths:
Path input pin req1 to DFFPOSX1_7/D delay 1165.19 ps
Path input pin req3 to DFFPOSX1_7/D delay 1135.62 ps
Path input pin req1 to DFFPOSX1_8/D delay 1130.61 ps
Path input pin req1 to DFFPOSX1_6/D delay 1129.21 ps
Path input pin req1 to DFFPOSX1_5/D delay 1129.21 ps
Path input pin req0 to DFFPOSX1_7/D delay 1128.69 ps
Path input pin req3 to DFFPOSX1_8/D delay 1104.08 ps
Path input pin req3 to DFFPOSX1_6/D delay 1101.32 ps
Path input pin req3 to DFFPOSX1_5/D delay 1101.32 ps
Path input pin req2 to DFFPOSX1_7/D delay 1099.25 ps
Path input pin req0 to DFFPOSX1_8/D delay 1094.28 ps
Path input pin req0 to DFFPOSX1_6/D delay 1092.86 ps
Path input pin req0 to DFFPOSX1_5/D delay 1092.86 ps
Path input pin req2 to DFFPOSX1_8/D delay 1067.9 ps
Path input pin req2 to DFFPOSX1_6/D delay 1065.11 ps
Path input pin req2 to DFFPOSX1_5/D delay 1065.11 ps
Path input pin req1 to DFFPOSX1_3/D delay 836.768 ps
Path input pin req1 to DFFPOSX1_4/D delay 836.768 ps
Path input pin req3 to DFFPOSX1_3/D delay 820.093 ps
Path input pin req3 to DFFPOSX1_4/D delay 820.093 ps
```

Figura 5: Paths con mayor retardo, pin to Flop. Simulado en Qflow

```
Top 20 minimum delay paths:
Path input pin clk to DFFPOSX1_8/CLK delay 0 ps
Path input pin clk to DFFPOSX1_7/CLK delay 0 ps
Path input pin clk to DFFPOSX1_6/CLK delay 0 ps
Path input pin clk to DFFPOSX1_5/CLK delay 0 ps
Path input pin clk to DFFPOSX1_4/CLK delay 0 ps
Path input pin clk to DFFPOSX1_3/CLK delay 0 ps
Path input pin clk to DFFPOSX1_2/CLK delay 0 ps
Path input pin clk to DFFPOSX1_1/CLK delay 0 ps
Path input pin rst to DFFPOSX1_8/D delay 53.6356 ps
Path input pin rst to DFFPOSX1_5/D delay 53.6356 ps
Path input pin rst to DFFPOSX1_6/D delay 53.6356 ps
Path input pin rst to DFFPOSX1_7/D delay 53.6356 ps
Path input pin req0 to DFFPOSX1_5/D delay 168.29 ps
Path input pin rst to DFFPOSX1_1/D delay 194.934 ps
Path input pin rst to DFFPOSX1_2/D delay 194.934 ps
Path input pin req1 to DFFPOSX1_7/D delay 276.463 ps
Path input pin req1 to DFFPOSX1_6/D delay 389.615 ps
Path input pin req3 to DFFPOSX1_7/D delay 404.149 ps
Path input pin req3 to DFFPOSX1_5/D delay 420.25 ps
Path input pin req3 to DFFPOSX1_6/D delay 420.25 ps
```

Figura 6: Paths con menor retardo, pin to Flop. Simulado en Qflow

2.1.3. Layout y área

Por último se obtienen los datos de área y se presenta el layout obtenido a partir de utilizar la herramienta Qflow

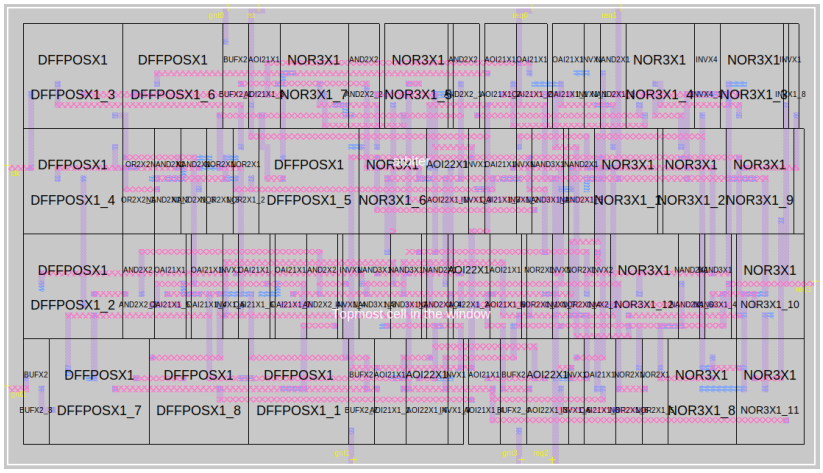

Figura 7: Layout del módulo arbiter. Simulado en Qflow

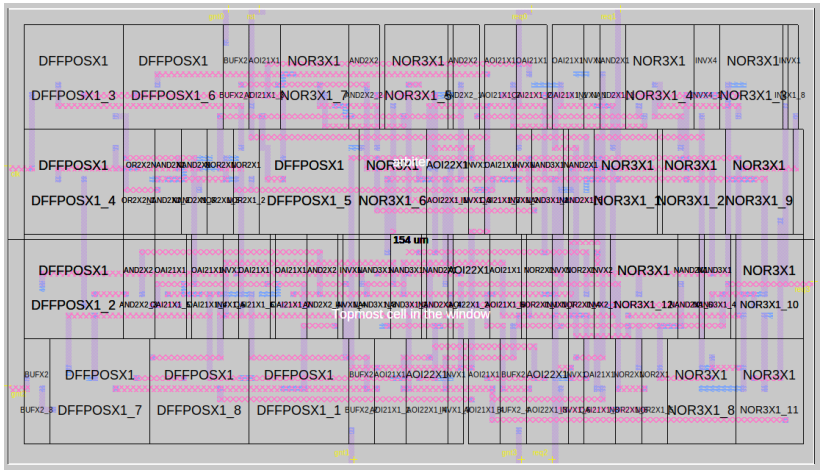

Figura 8: Longitud horizontal del layout del módulo arbiter. Simulado en Qflow

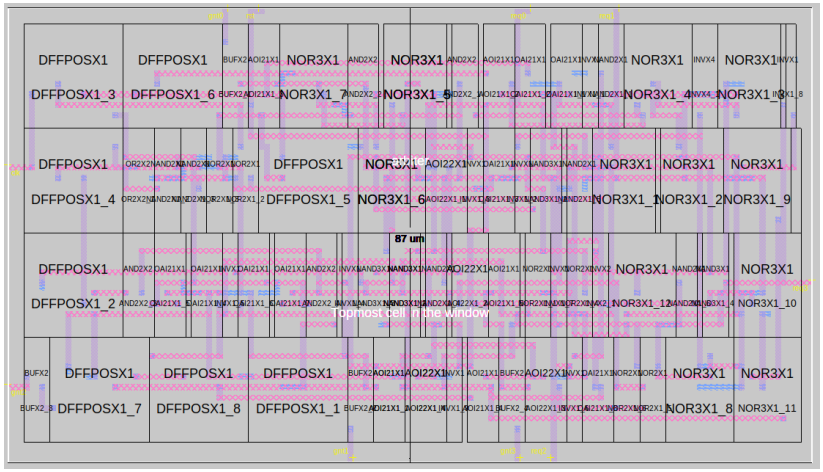

Figura 9: Longitud vertical del layout del módulo arbiter. Simulado en Qflow

2.2. Arbiter parte B: Librería osu050.

Para esta sección fue necesario modificar el archivo qflow_varsh.sh y volver a correr el código utilizado en la parte A, sin embargo, esta vez no se utiliza el comando route ya que presenta errores.

2.2.1. Compuertas

La cantidad de compuertas que requiere el diseño, las implementadas por Qflow de la librería ABC y las obtenidas a partir del archivo arbiter.rtl.v se muestran a continuación

```
=== arbiter ===

   Number of wires:                 81
   Number of wire bits:             81
   Number of public wires:          18
   Number of public wire bits:      18
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:                 71
     $_AND_                         17
     $_AOI3_                         6
     $_AOI4_                         6
     $_DFF_P_                        8
     $_MUX_                          2
     $_NAND_                         6
     $_NOR_                          6
     $_NOT_                          8
     $_OAI3_                         2
     $_OAI4_                         3
     $_OR_                           7
```

Figura 10: Cantidad de compuertas que requiere el diseño del módulo arbiter.v (librería osu050). Simulado en Qflow

```
6.1.2. Re-integrating ABC results.
ABC RESULTS:            AND2X2 cells:        4
ABC RESULTS:           AOI21X1 cells:        5
ABC RESULTS:           AOI22X1 cells:        4
ABC RESULTS:             INVX1 cells:       11
ABC RESULTS:           NAND2X1 cells:        6
ABC RESULTS:           NAND3X1 cells:        4
ABC RESULTS:            NOR2X1 cells:        6
ABC RESULTS:            NOR3X1 cells:       12
ABC RESULTS:           OAI21X1 cells:        8
ABC RESULTS:             OR2X2 cells:        1
ABC RESULTS:        internal signals:       55
ABC RESULTS:           input signals:       13
ABC RESULTS:          output signals:        8
```

Figura 11: Cantidad de compuertas que utiliza la herramienta para implementar el diseño del módulo arbiter.v (librería osu050). Simulado en Qflow

Las siguientes compuertas se obtuvieron al analizar el archivo arbiter.rtl.v, el cuál se encuentra dentro de la carpeta synthesis

- **INVX1**: 10
- **INVX2**: 1
- **BUFX2**: 4
- **OR2X2**: 1
- **NOR2X1**: 6
- **NOR3X1**: 12
- **AND2X2**: 4
- **NAND2X1**: 6
- **NAND3X1**: 4
- **OAI21X1**: 8
- **AOI21X1**: 5
- **AOI22X1**: 4
- **DFFPOSX1**: 8

2.2.2. Paths y timings

A continuación se presentan los resultados de los paths con mayor y menor retardo, así como la frecuencia máxima del circuito para el módulo arbiter.v

```
Top 20 maximum delay paths:
Path DFFPOSX1_6/CLK to DFFPOSX1_8/D delay 2049.88 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_8/D delay 2036.67 ps
Path DFFPOSX1_5/CLK to DFFPOSX1_8/D delay 2031.81 ps
Path DFFPOSX1_7/CLK to DFFPOSX1_8/D delay 2012.96 ps
Path DFFPOSX1_6/CLK to DFFPOSX1_7/D delay 2009.83 ps
Path DFFPOSX1_5/CLK to DFFPOSX1_7/D delay 1991.57 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_7/D delay 1991.35 ps
Path DFFPOSX1_6/CLK to DFFPOSX1_6/D delay 1984.84 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_6/D delay 1971.7 ps
Path DFFPOSX1_7/CLK to DFFPOSX1_7/D delay 1967.44 ps
Path DFFPOSX1_5/CLK to DFFPOSX1_6/D delay 1966.77 ps
Path DFFPOSX1_2/CLK to DFFPOSX1_8/D delay 1966.03 ps
Path DFFPOSX1_6/CLK to DFFPOSX1_5/D delay 1948.94 ps
Path DFFPOSX1_7/CLK to DFFPOSX1_6/D delay 1947.98 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_5/D delay 1934.05 ps
Path DFFPOSX1_5/CLK to DFFPOSX1_5/D delay 1930.85 ps
Path DFFPOSX1_7/CLK to DFFPOSX1_5/D delay 1910.31 ps
Path DFFPOSX1_2/CLK to DFFPOSX1_6/D delay 1900.93 ps
Path DFFPOSX1_2/CLK to DFFPOSX1_7/D delay 1881.32 ps
Path DFFPOSX1_2/CLK to DFFPOSX1_5/D delay 1866.48 ps
Computed maximum clock frequency (zero slack) = 487.834 MHz
```

Figura 12: Paths con mayor retardo, Flop to Flop y frecuencia máxima (librería osu050). Simulado en Qflow

```
Top 20 minimum delay paths:
Path DFFPOSX1_5/CLK to output pin gnt0 delay 393.043 ps
Path DFFPOSX1_6/CLK to output pin gnt1 delay 456.969 ps
Path DFFPOSX1_7/CLK to output pin gnt2 delay 460.766 ps
Path DFFPOSX1_8/CLK to output pin gnt3 delay 470.764 ps
Path DFFPOSX1_4/CLK to DFFPOSX1_4/D delay 552.315 ps
Path DFFPOSX1_4/CLK to DFFPOSX1_3/D delay 586.33 ps
Path DFFPOSX1_7/CLK to DFFPOSX1_7/D delay 666.929 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_8/D delay 675.989 ps
Path DFFPOSX1_2/CLK to DFFPOSX1_2/D delay 679.509 ps
Path DFFPOSX1_3/CLK to DFFPOSX1_1/D delay 680.629 ps
Path DFFPOSX1_3/CLK to DFFPOSX1_2/D delay 680.629 ps
Path DFFPOSX1_3/CLK to DFFPOSX1_4/D delay 695.861 ps
Path DFFPOSX1_3/CLK to DFFPOSX1_3/D delay 703.837 ps
Path DFFPOSX1_6/CLK to DFFPOSX1_1/D delay 747.034 ps
Path DFFPOSX1_7/CLK to DFFPOSX1_2/D delay 750.992 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_1/D delay 759.138 ps
Path DFFPOSX1_8/CLK to DFFPOSX1_2/D delay 759.138 ps
Path DFFPOSX1_5/CLK to DFFPOSX1_5/D delay 760.612 ps
Path DFFPOSX1_1/CLK to DFFPOSX1_5/D delay 814.991 ps
Path DFFPOSX1_1/CLK to DFFPOSX1_8/D delay 814.991 ps
Design meets minimum hold timing.
```

Figura 13: Paths con menor retardo, Flop to Flop (librería osu050). Simulado en Qflow

```
Top 20 maximum delay paths:
Path input pin req1 to DFFPOSX1_8/D delay 1301.74 ps
Path input pin req3 to DFFPOSX1_8/D delay 1276.92 ps
Path input pin req1 to DFFPOSX1_7/D delay 1274.66 ps
Path input pin req0 to DFFPOSX1_8/D delay 1266.87 ps
Path input pin req1 to DFFPOSX1_6/D delay 1249.8 ps
Path input pin req3 to DFFPOSX1_7/D delay 1244.84 ps
Path input pin req2 to DFFPOSX1_8/D delay 1242.19 ps
Path input pin req0 to DFFPOSX1_7/D delay 1239.58 ps
Path input pin req1 to DFFPOSX1_5/D delay 1232.56 ps
Path input pin req3 to DFFPOSX1_6/D delay 1225.08 ps
Path input pin req0 to DFFPOSX1_6/D delay 1214.93 ps
Path input pin req2 to DFFPOSX1_7/D delay 1209.9 ps
Path input pin req3 to DFFPOSX1_5/D delay 1206.19 ps
Path input pin req0 to DFFPOSX1_5/D delay 1197.67 ps
Path input pin req2 to DFFPOSX1_6/D delay 1190.35 ps
Path input pin req2 to DFFPOSX1_5/D delay 1171.42 ps
Path input pin req1 to DFFPOSX1_3/D delay 927.774 ps
Path input pin req1 to DFFPOSX1_4/D delay 927.774 ps
Path input pin req3 to DFFPOSX1_3/D delay 910.445 ps
Path input pin req3 to DFFPOSX1_4/D delay 910.445 ps
```

Figura 14: Paths con mayor retardo, pin to Flop (librería osu050). Simulado en Qflow

```
Top 20 minimum delay paths:
Path input pin clk to DFFPOSX1_8/CLK delay 0 ps
Path input pin clk to DFFPOSX1_7/CLK delay 0 ps
Path input pin clk to DFFPOSX1_6/CLK delay 0 ps
Path input pin clk to DFFPOSX1_5/CLK delay 0 ps
Path input pin clk to DFFPOSX1_4/CLK delay 0 ps
Path input pin clk to DFFPOSX1_3/CLK delay 0 ps
Path input pin clk to DFFPOSX1_2/CLK delay 0 ps
Path input pin clk to DFFPOSX1_1/CLK delay 0 ps
Path input pin rst to DFFPOSX1_8/D delay 60.8009 ps
Path input pin rst to DFFPOSX1_5/D delay 60.8009 ps
Path input pin rst to DFFPOSX1_6/D delay 60.8009 ps
Path input pin rst to DFFPOSX1_7/D delay 60.8009 ps
Path input pin req0 to DFFPOSX1_5/D delay 190.933 ps
Path input pin rst to DFFPOSX1_1/D delay 230.582 ps
Path input pin rst to DFFPOSX1_2/D delay 230.582 ps
Path input pin req1 to DFFPOSX1_7/D delay 322.404 ps
Path input pin req3 to DFFPOSX1_7/D delay 450.8 ps
Path input pin req1 to DFFPOSX1_6/D delay 464.153 ps
Path input pin req3 to DFFPOSX1_5/D delay 469.716 ps
Path input pin req3 to DFFPOSX1_6/D delay 469.716 ps
```

Figura 15: Paths con menor retardo, pin to Flop (librería osu050). Simulado en Qflow

2.2.3. Layout y área

Por último se obtienen los datos de área y se presenta el layout obtenido a partir de utilizar la herramienta Qflow

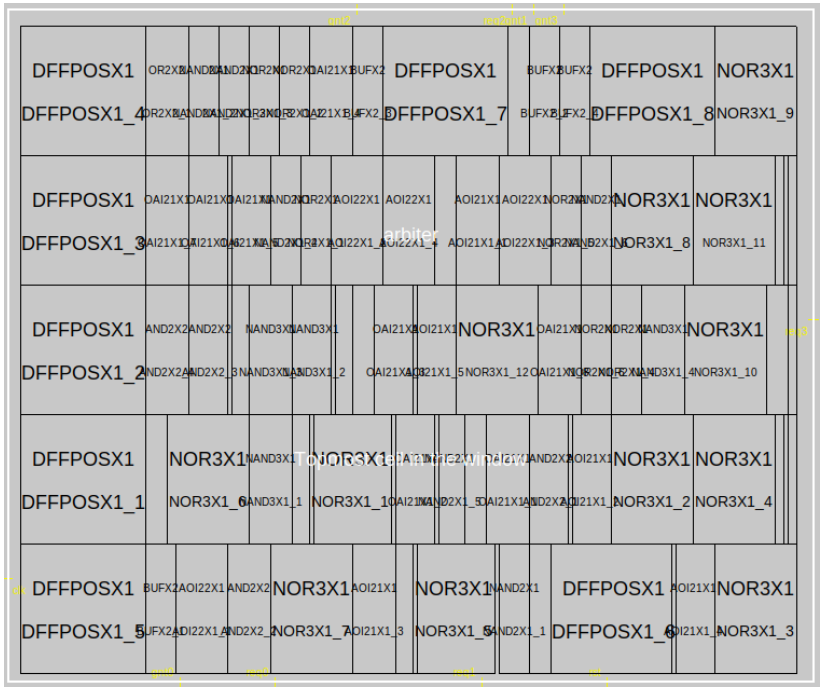

Figura 16: Layout del módulo arbiter (librería osu050). Simulado en Qflow

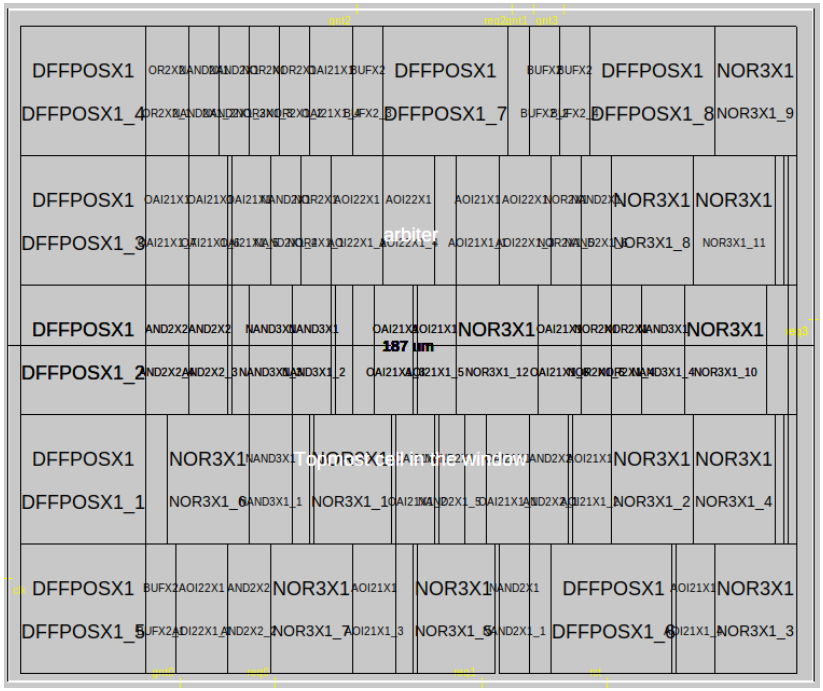

Figura 17: Longitud horizontal del layout del módulo arbiter (librería osu050). Simulado en Qflow

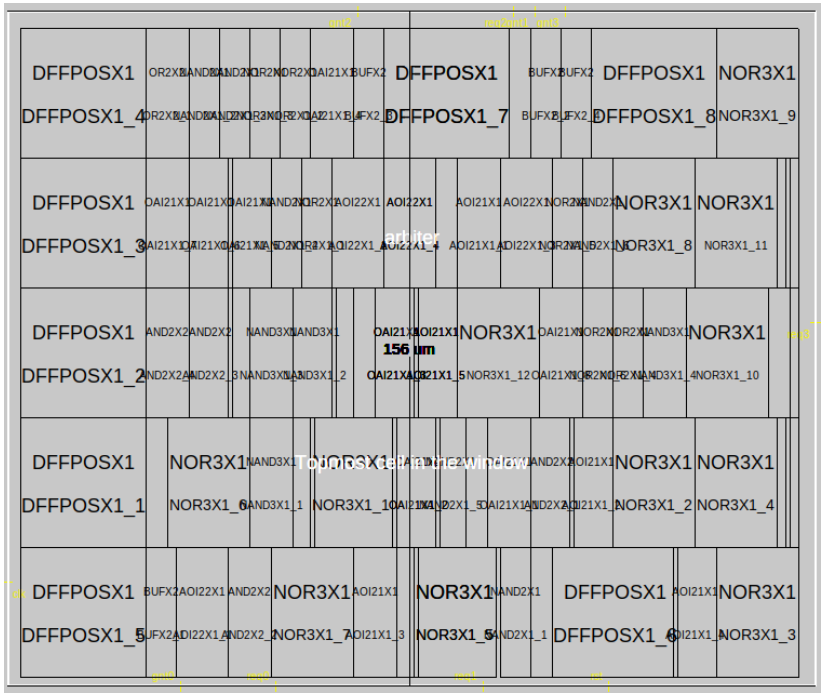

Figura 18: Longitud vertical del layout del módulo arbiter (librería osu050). Simulado en Qflow

2.3. Uart parte A: Librería osu035.

Para esta sección se repite el procedimiento anterior, esta vez con el módulo uart.v.

2.3.1. Compuertas

Los resultados obtenidos para el módulo uart.v se muestran a continuación.
En las siguientes figuras se muestra la cantidad de compuertas que requiere el diseño, la cantidad de compuertas de la librería ABC que utiliza Qflow y la cantidad de compuertas que se encuentran en el archivo uart.rtl.v respectivamente

```
=== uart ===

   Number of wires:                229
   Number of wire bits:            296
   Number of public wires:          21
   Number of public wire bits:      58
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:                280
     $_AND_                         47
     $_AOI3_                         4
     $_DFF_PP0_                     37
     $_DFF_PP1_                      5
     $_MUX_                        101
     $_NAND_                         3
     $_NOR_                         16
     $_NOT_                         20
     $_OAI3_                        10
     $_OAI4_                         8
     $_OR_                          15
     $_XNOR_                         2
     $_XOR_                         12
```

Figura 19: Cantidad de compuertas que requiere el diseño del módulo uart.v. Simulado en Qflow

```
6.1.2. Re-integrating ABC results.
ABC RESULTS:            AND2X2 cells:        6
ABC RESULTS:           AOI21X1 cells:       11
ABC RESULTS:           AOI22X1 cells:        1
ABC RESULTS:             INVX1 cells:       45
ABC RESULTS:            MUX2X1 cells:        8
ABC RESULTS:           NAND2X1 cells:       30
ABC RESULTS:           NAND3X1 cells:       25
ABC RESULTS:            NOR2X1 cells:       24
ABC RESULTS:            NOR3X1 cells:        1
ABC RESULTS:           OAI21X1 cells:       40
ABC RESULTS:           OAI22X1 cells:        2
ABC RESULTS:             OR2X2 cells:        3
ABC RESULTS:          XNOR2X1 cells:        1
ABC RESULTS:        internal signals:      198
ABC RESULTS:          input signals:       54
ABC RESULTS:         output signals:       41
```

Figura 20: Cantidad de compuertas que utiliza la herramienta para implementar el diseño del módulo uart.v. Simulado en Qflow

Las siguientes compuertas se obtuvieron al analizar el archivo uart.rtl.v, el cuál se encuentra dentro de la carpeta synthesis

- **INVX1**: 37
- **INVX2**: 6
- **INVX4**: 1
- **INVX8**: 1
- **BUFX2**: 18
- **BUFX4**: 8
- **OR2X2**: 3
- **NOR2X1**: 24

- **NOR3X1**: 1
- **XNOR2X1**: 1
- **AND2X2**: 6
- **NAND2X1**: 30
- **NAND3X1**: 25
- **OAI21X1**: 40
- **OAI22X1**: 2
- **AOI21X1**: 11
- **AOI22X1**: 1
- **MUX2X1**: 8
- **DFFSR**: 42

2.3.2. Paths y timings

A continuación se presentan los resultados de los paths con mayor y menor retardo, así como la frecuencia máxima del circuito para el módulo uart.v

```
Top 20 maximum delay paths:
Path DFFSR_13/CLK to DFFSR_1/D delay 2166.55 ps
Path DFFSR_42/CLK to DFFSR_28/D delay 2150.91 ps
Path DFFSR_42/CLK to DFFSR_29/D delay 2150.91 ps
Path DFFSR_11/CLK to DFFSR_1/D delay 2137.38 ps
Path DFFSR_42/CLK to DFFSR_27/D delay 2116.67 ps
Path DFFSR_33/CLK to DFFSR_28/D delay 2105.3 ps
Path DFFSR_33/CLK to DFFSR_29/D delay 2105.3 ps
Path DFFSR_42/CLK to DFFSR_26/D delay 2077.23 ps
Path DFFSR_42/CLK to DFFSR_25/D delay 2077.23 ps
Path DFFSR_33/CLK to DFFSR_27/D delay 2071.51 ps
Path DFFSR_42/CLK to DFFSR_24/D delay 2039.64 ps
Path DFFSR_33/CLK to DFFSR_26/D delay 2033.19 ps
Path DFFSR_33/CLK to DFFSR_25/D delay 2033.19 ps
Path DFFSR_12/CLK to DFFSR_1/D delay 2013.62 ps
Path DFFSR_32/CLK to DFFSR_28/D delay 2001.81 ps
Path DFFSR_32/CLK to DFFSR_29/D delay 2001.81 ps
Path DFFSR_33/CLK to DFFSR_24/D delay 1995.18 ps
Path DFFSR_42/CLK to DFFSR_36/D delay 1993.87 ps
Path DFFSR_32/CLK to DFFSR_27/D delay 1967.91 ps
Path DFFSR_42/CLK to DFFSR_31/D delay 1962.26 ps
Computed maximum clock frequency (zero slack) = 461.564 MHz
```

Figura 21: Paths con mayor retardo, Flop to Flop y frecuencia máxima. Simulado en Qflow

```
Top 20 minimum delay paths:
Path DFFSR_40/CLK to DFFSR_41/D delay 322.421 ps
Path DFFSR_15/CLK to output pin rx_data[0] delay 472.527 ps
Path DFFSR_20/CLK to output pin rx_data[5] delay 472.527 ps
Path DFFSR_16/CLK to output pin rx_data[1] delay 472.527 ps
Path DFFSR_17/CLK to output pin rx_data[2] delay 472.527 ps
Path DFFSR_18/CLK to output pin rx_data[3] delay 472.527 ps
Path DFFSR_19/CLK to output pin rx_data[4] delay 472.527 ps
Path DFFSR_23/CLK to output pin rx_empty delay 492.445 ps
Path DFFSR_21/CLK to output pin rx_data[6] delay 494.127 ps
Path DFFSR_22/CLK to output pin rx_data[7] delay 494.127 ps
Path DFFSR_1/CLK to output pin tx_out delay 503.018 ps
Path DFFSR_15/CLK to DFFSR_15/D delay 505.611 ps
Path DFFSR_20/CLK to DFFSR_20/D delay 505.611 ps
Path DFFSR_16/CLK to DFFSR_16/D delay 505.611 ps
Path DFFSR_17/CLK to DFFSR_17/D delay 505.611 ps
Path DFFSR_18/CLK to DFFSR_18/D delay 505.611 ps
Path DFFSR_19/CLK to DFFSR_19/D delay 505.611 ps
Path DFFSR_29/CLK to DFFSR_20/D delay 525.843 ps
Path DFFSR_1/CLK to DFFSR_1/D delay 527.71 ps
Path DFFSR_21/CLK to DFFSR_21/D delay 539.429 ps
Design meets minimum hold timing.
```

Figura 22: Paths con menor retardo, Flop to Flop. Simulado en Qflow

```
Top 20 maximum delay paths:
Path input pin rx_enable to DFFSR_28/D delay 1211.19 ps
Path input pin rx_enable to DFFSR_29/D delay 1211.19 ps
Path input pin rx_enable to DFFSR_27/D delay 1175.79 ps
Path input pin rx_enable to DFFSR_26/D delay 1136.17 ps
Path input pin rx_enable to DFFSR_25/D delay 1136.17 ps
Path input pin rx_enable to DFFSR_24/D delay 1098.9 ps
Path input pin rx_enable to DFFSR_36/D delay 1037.83 ps
Path input pin rx_enable to DFFSR_31/D delay 1020.23 ps
Path input pin rx_enable to DFFSR_39/D delay 962.742 ps
Path input pin tx_enable to DFFSR_14/D delay 924.315 ps
Path input pin rx_enable to DFFSR_37/D delay 883.216 ps
Path input pin tx_enable to DFFSR_13/D delay 875.66 ps
Path input pin tx_enable to DFFSR_1/D delay 854.066 ps
Path input pin rx_enable to DFFSR_38/D delay 772.709 ps
Path input pin rx_enable to DFFSR_23/D delay 700.758 ps
Path input pin tx_enable to DFFSR_2/D delay 650.07 ps
Path input pin ld_tx_data to DFFSR_3/D delay 626.462 ps
Path input pin ld_tx_data to DFFSR_4/D delay 626.462 ps
Path input pin ld_tx_data to DFFSR_5/D delay 626.462 ps
Path input pin ld_tx_data to DFFSR_6/D delay 626.462 ps
```

Figura 23: Paths con mayor retardo, pin to Flop. Simulado en Qflow

```
Top 20 minimum delay paths:
Path input pin txclk to DFFSR_14/CLK delay 0 ps
Path input pin txclk to DFFSR_13/CLK delay 0 ps
Path input pin txclk to DFFSR_12/CLK delay 0 ps
Path input pin txclk to DFFSR_11/CLK delay 0 ps
Path input pin txclk to DFFSR_10/CLK delay 0 ps
Path input pin txclk to DFFSR_9/CLK delay 0 ps
Path input pin txclk to DFFSR_8/CLK delay 0 ps
Path input pin txclk to DFFSR_7/CLK delay 0 ps
Path input pin txclk to DFFSR_6/CLK delay 0 ps
Path input pin txclk to DFFSR_5/CLK delay 0 ps
Path input pin txclk to DFFSR_4/CLK delay 0 ps
Path input pin txclk to DFFSR_3/CLK delay 0 ps
Path input pin txclk to DFFSR_2/CLK delay 0 ps
Path input pin txclk to DFFSR_1/CLK delay 0 ps
Path input pin rx_in to DFFSR_40/D delay 0 ps
Path input pin ld_tx_data to DFFSR_2/D delay 83.5282 ps
Path input pin tx_data[5] to DFFSR_8/D delay 145.099 ps
Path input pin tx_data[4] to DFFSR_7/D delay 145.099 ps
Path input pin tx_data[3] to DFFSR_6/D delay 145.099 ps
Path input pin tx_data[2] to DFFSR_5/D delay 145.099 ps
```

Figura 24: Paths con menor retardo, pin to Flop. Simulado en Qflow

2.3.3. Layout y área

Por último se obtienen los datos de área y se presenta el layout obtenido a partir de utilizar la herramienta Qflow

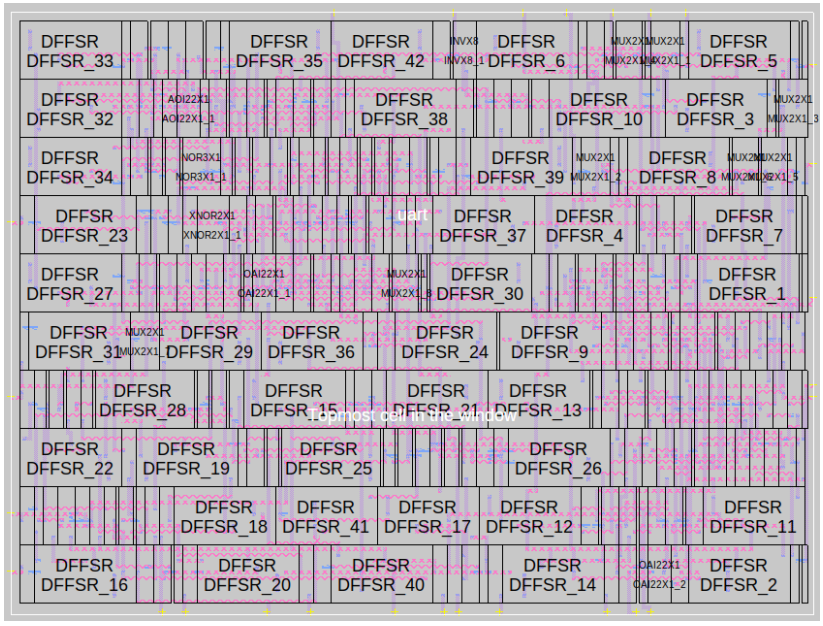

Figura 25: Layout del módulo uart. Simulado en Qflow

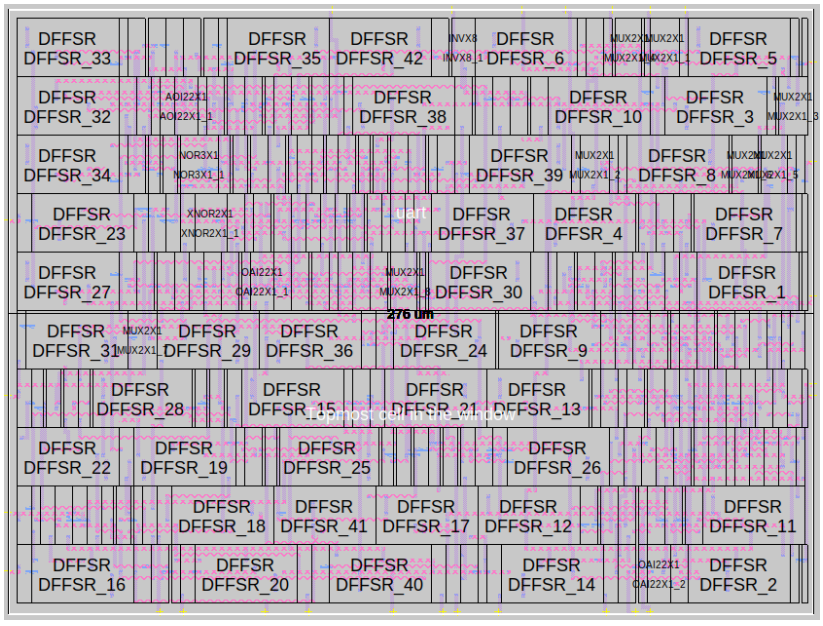

Figura 26: Longitud horizontal del layout del módulo uart. Simulado en Qflow

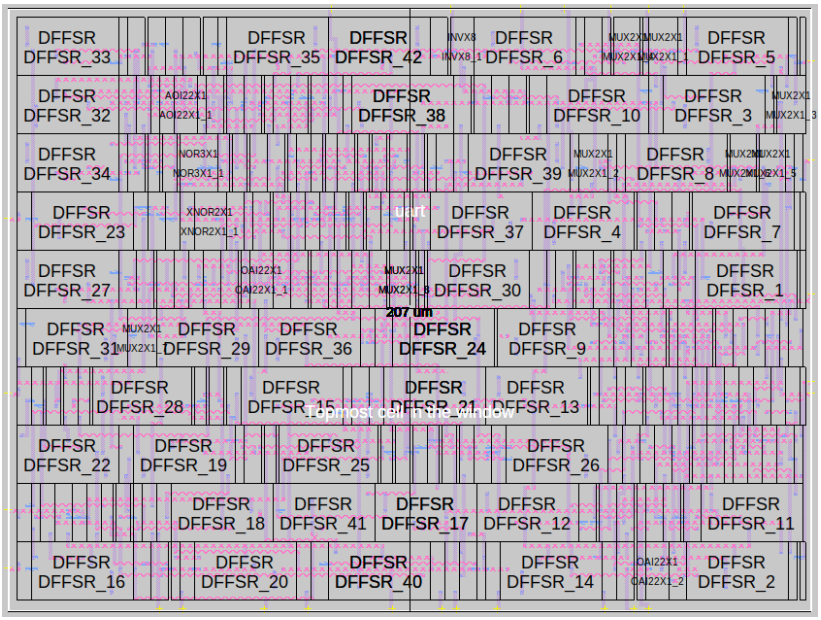

Figura 27: Longitud vertical del layout del módulo uart. Simulado en Qflow

2.4. Uart parte B: Librería osu050.

Para esta sección fue necesario modificar el archivo qflow_varsh.sh y volver a correr el código utilizado en la parte A, sin embargo, esta vez no se utiliza el comando route ya que presenta errores.

2.4.1. Compuertas

La cantidad de compuertas que requiere el diseño, las implementadas por Qflow de la librería ABC y las obtenidas a partir del archivo uart.rtl.v se muestran a continuación

```
=== uart ===

   Number of wires:                229
   Number of wire bits:            296
   Number of public wires:          21
   Number of public wire bits:      58
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:                280
     $_AND_                         47
     $_AOI3_                         4
     $_DFF_PP0_                     37
     $_DFF_PP1_                      5
     $_MUX_                        101
     $_NAND_                         3
     $_NOR_                         16
     $_NOT_                         20
     $_OAI3_                        10
     $_OAI4_                         8
     $_OR_                          15
     $_XNOR_                         2
     $_XOR_                         12
```

Figura 28: Cantidad de compuertas que requiere el diseño del módulo uart.v (librería osu050). Simulado en Qflow

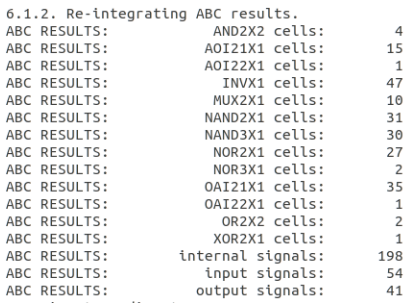

Figura 29: Cantidad de compuertas que utiliza la herramienta para implementar el diseño del módulo uart.v (librería osu050). Simulado en Qflow

Las siguientes compuertas se obtuvieron al analizar el archivo uart.rtl.v, el cuál se encuentra dentro de la carpeta synthesis

- **INVX1**: 42
- **INVX2**: 4
- **INVX8**: 1
- **BUFX2**: 20
- **BUFX4**: 6
- **OR2X2**: 2
- **NOR2X1**: 27
- **NOR3X1**: 2
- **XOR2X1**: 1
- **AND2X2**: 4
- **NAND2X1**: 31
- **NAND3X1**: 30
- **OAI21X1**: 35
- **OAI22X1**: 1
- **AOI21X1**: 15
- **AOI22X1**: 1
- **MUX2X1**: 10
- **DFFSR**: 42

2.4.2. Paths y timings

A continuación se presentan los resultados de los paths con mayor y menor retardo, así como la frecuencia máxima del circuito para el módulo uart.v

```
Top 20 maximum delay paths:
Path DFFSR_39/CLK to DFFSR_1/D delay 7507.98 ps
Path DFFSR_36/CLK to DFFSR_1/D delay 7392.28 ps
Path DFFSR_39/CLK to DFFSR_11/D delay 7344.08 ps
Path DFFSR_37/CLK to DFFSR_1/D delay 7228.1 ps
Path DFFSR_36/CLK to DFFSR_11/D delay 7213.98 ps
Path DFFSR_36/CLK to DFFSR_24/D delay 7184.83 ps
Path DFFSR_38/CLK to DFFSR_1/D delay 7096.03 ps
Path DFFSR_39/CLK to DFFSR_24/D delay 7071.34 ps
Path DFFSR_37/CLK to DFFSR_11/D delay 7049.81 ps
Path DFFSR_37/CLK to DFFSR_24/D delay 7020.65 ps
Path DFFSR_39/CLK to DFFSR_23/D delay 6945.94 ps
Path DFFSR_39/CLK to DFFSR_42/D delay 6927.77 ps
Path DFFSR_38/CLK to DFFSR_11/D delay 6917.73 ps
Path DFFSR_36/CLK to DFFSR_27/D delay 6914.24 ps
Path DFFSR_38/CLK to DFFSR_24/D delay 6888.58 ps
Path DFFSR_42/CLK to DFFSR_1/D delay 6860.11 ps
Path DFFSR_36/CLK to DFFSR_42/D delay 6804.13 ps
Path DFFSR_39/CLK to DFFSR_27/D delay 6795.56 ps
Path DFFSR_36/CLK to DFFSR_23/D delay 6794.5 ps
Path DFFSR_33/CLK to DFFSR_1/D delay 6793.15 ps
Computed maximum clock frequency (zero slack) = 133.192 MHz
```

Figura 30: Paths con mayor retardo, Flop to Flop y frecuencia máxima (librería osu050). Simulado en Qflow

```
Top 20 minimum delay paths:
Path DFFSR_40/CLK to DFFSR_41/D delay 399.627 ps
Path DFFSR_29/CLK to DFFSR_1/D delay 513.64 ps
Path DFFSR_26/CLK to DFFSR_39/D delay 518.83 ps
Path DFFSR_20/CLK to output pin rx_data[3] delay 543.397 ps
Path DFFSR_21/CLK to output pin rx_data[4] delay 543.397 ps
Path DFFSR_17/CLK to output pin rx_data[0] delay 543.397 ps
Path DFFSR_22/CLK to output pin rx_data[5] delay 543.397 ps
Path DFFSR_18/CLK to output pin rx_data[1] delay 543.397 ps
Path DFFSR_23/CLK to output pin rx_data[6] delay 543.397 ps
Path DFFSR_19/CLK to output pin rx_data[2] delay 543.397 ps
Path DFFSR_1/CLK to DFFSR_42/D delay 568.488 ps
Path DFFSR_4/CLK to DFFSR_22/D delay 603.336 ps
Path DFFSR_35/CLK to DFFSR_19/D delay 613.282 ps
Path DFFSR_31/CLK to DFFSR_17/D delay 616.553 ps
Path DFFSR_3/CLK to DFFSR_5/D delay 618.28 ps
Path DFFSR_27/CLK to DFFSR_2/D delay 620.404 ps
Path DFFSR_1/CLK to output pin rx_empty delay 627.103 ps
Path DFFSR_24/CLK to DFFSR_35/D delay 679.901 ps
Path DFFSR_8/CLK to DFFSR_8/D delay 685.048 ps
Path DFFSR_28/CLK to DFFSR_15/D delay 685.846 ps
Design meets minimum hold timing.
```

Figura 31: Paths con menor retardo, Flop to Flop (librería osu050). Simulado en Qflow

```
Top 20 maximum delay paths:
Path input pin uld_rx_data to DFFSR_1/D delay 4397.1 ps
Path input pin uld_rx_data to DFFSR_11/D delay 4218.8 ps
Path input pin uld_rx_data to DFFSR_24/D delay 3977.18 ps
Path input pin tx_enable to DFFSR_1/D delay 3790.58 ps
Path input pin uld_rx_data to DFFSR_27/D delay 3704.94 ps
Path input pin tx_enable to DFFSR_11/D delay 3572.08 ps
Path input pin uld_rx_data to DFFSR_33/D delay 3462.79 ps
Path input pin tx_enable to DFFSR_24/D delay 3330.59 ps
Path input pin uld_rx_data to DFFSR_42/D delay 3152.85 ps
Path input pin uld_rx_data to DFFSR_6/D delay 3138.16 ps
Path input pin uld_rx_data to DFFSR_23/D delay 3133.67 ps
Path input pin tx_enable to DFFSR_27/D delay 3052.75 ps
Path input pin uld_rx_data to DFFSR_17/D delay 3032.65 ps
Path input pin uld_rx_data to DFFSR_5/D delay 2989.48 ps
Path input pin uld_rx_data to DFFSR_3/D delay 2989.48 ps
Path input pin uld_rx_data to DFFSR_4/D delay 2989.48 ps
Path input pin uld_rx_data to DFFSR_26/D delay 2989.48 ps
Path input pin tx_enable to DFFSR_33/D delay 2856.16 ps
Path input pin uld_rx_data to DFFSR_22/D delay 2835.18 ps
Path input pin uld_rx_data to DFFSR_21/D delay 2835.18 ps
```

Figura 32: Paths con mayor retardo, pin to Flop (librería osu050). Simulado en Qflow

```
Top 20 minimum delay paths:
Path input pin txclk to DFFSR_14/CLK delay 0 ps
Path input pin txclk to DFFSR_13/CLK delay 0 ps
Path input pin txclk to DFFSR_12/CLK delay 0 ps
Path input pin txclk to DFFSR_11/CLK delay 0 ps
Path input pin txclk to DFFSR_10/CLK delay 0 ps
Path input pin txclk to DFFSR_9/CLK delay 0 ps
Path input pin txclk to DFFSR_8/CLK delay 0 ps
Path input pin txclk to DFFSR_7/CLK delay 0 ps
Path input pin txclk to DFFSR_6/CLK delay 0 ps
Path input pin txclk to DFFSR_5/CLK delay 0 ps
Path input pin txclk to DFFSR_4/CLK delay 0 ps
Path input pin txclk to DFFSR_3/CLK delay 0 ps
Path input pin txclk to DFFSR_2/CLK delay 0 ps
Path input pin txclk to DFFSR_1/CLK delay 0 ps
Path input pin rx_in to DFFSR_40/D delay 0 ps
Path input pin tx_data[1] to DFFSR_22/D delay 145.592 ps
Path input pin rx_enable to DFFSR_28/D delay 171.107 ps
Path input pin tx_data[0] to DFFSR_21/D delay 176.392 ps
Path input pin rx_enable to DFFSR_42/D delay 177.683 ps
Path input pin rxclk to DFFSR_39/CLK delay 199.663 ps
```

Figura 33: Paths con menor retardo, pin to Flop (librería osu050). Simulado en Qflow

2.4.3. Layout y área

Por último se obtienen los datos de área y se presenta el layout obtenido a partir de utilizar la herramienta Qflow

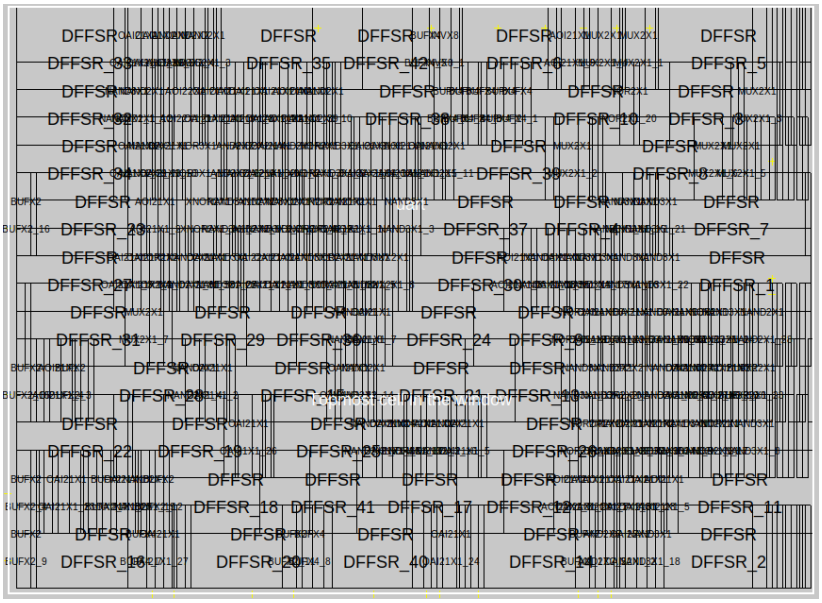

Figura 34: Layout del módulo uart (librería osu050). Simulado en Qflow

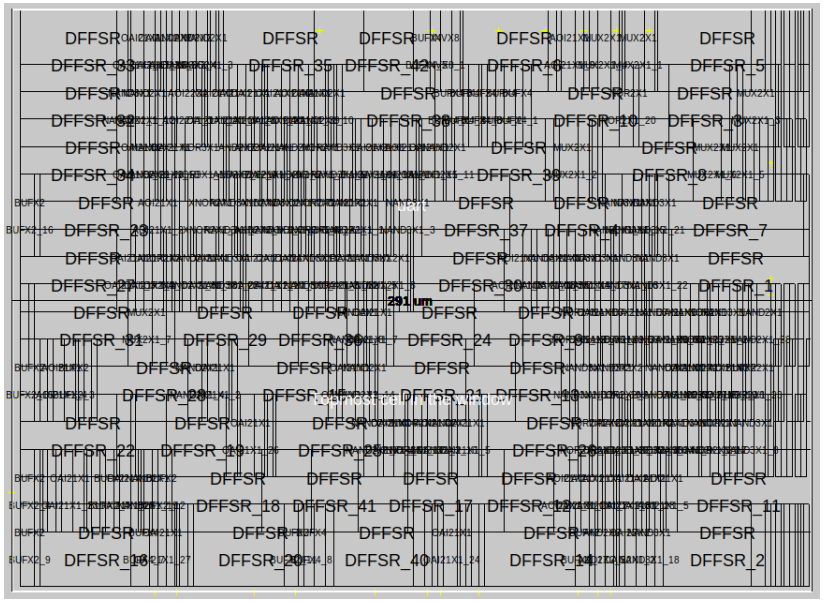

Figura 35: Longitud horizontal del layout del módulo uart (librería osu050). Simulado en Qflow

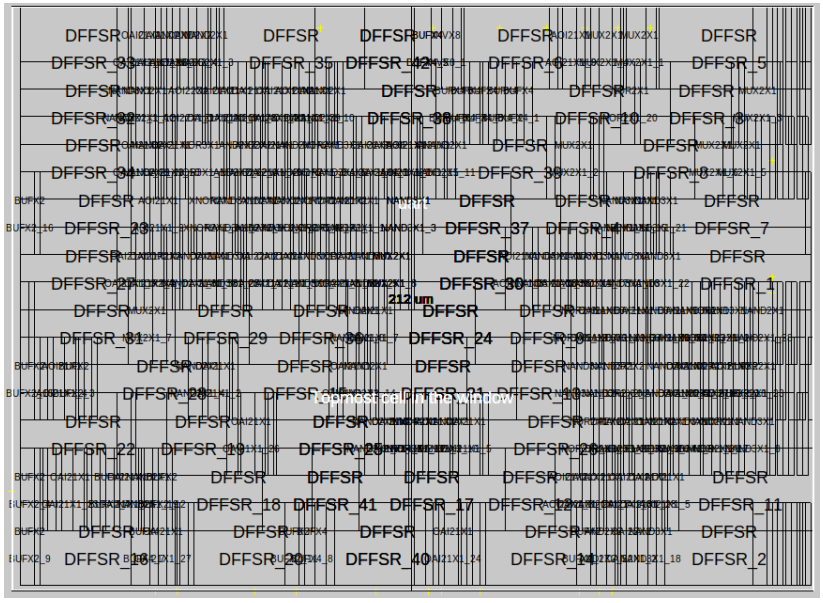

Figura 36: Longitud vertical del layout del módulo uart (librería osu050). Simulado en Qflow

3. Análisis de resultados

3.1. OSU0350

Inicialmente se observa que la cantidad de compuertas utilizadas por el rtl para cada archivo no concuerda con la cantidad de compuertas propuestas por la librería ABC, sin embargo, al leer el archivo synth.log nos podemos dar cuenta de que esta cantidad de compuertas cambia, ya que luego de parsear el archivo “osu05_stdcells” el log informa que algunas compuertas cambian. Es por esto

que se decide utilizar la cantidad de compuertas del rtl como la final.

En total, se cuentan 69 compuertas para el módulo arbiter sintetizado y 265 para el uart. Por lo cuál se espera que el área del uart sea mucho mayor que el del arbiter. Dicha área se calcula a partir de las figuras 8 y 9 para el arbiter, obteniendo un área de 1.34x10^{-8} m^2 y a partir de las figuras 26 y 27 para uart, obteniendo un área de 5.71x10^{-8} m^2 para uart, 4.2 veces mayor que el área del arbiter debido a la gran cantidad de compuertas que utiliza.

Por otra parte, en la sección "Paths y timings" se obtienen los retardos para los paths Flop to Flop y Pin to Flop. También se obtiene la frecuencia máxima de operación a partir del mayor delay Flop to Flop, los resultados obtenidos muestran una frecuencia máxima de 546.164 MHz para arbiter y de 461.564 MHz para uart. Esto se puede justificar al observar los delays máximos de arbiter, los cuáles son menores (1830.95 ps) que los de uart (2166.55 ps), ya que arbiter utiliza una menor cantidad de lógica secuencial entre cada Flop, mientras que uart, como lo indica su cantidad de compuertas, utiliza mucha más lógica secuencial.

3.2. OSU050

Utilizando la librería OSU050, el número de compuertas que utiliza arbiter incrementa a 73, mientras que uart aumenta a 274. Por parte del área, arbiter ahora tendrá un area de 2.9172x10^{-8} m^2, la cuál es significativamente mayor con respecto al área obtenida utilizando la librería osu0350, esto puede deberse a que las compuertas de la librería osu050 tienen mayor tamaño o bien que las compuertas extra que fueron agregadas son compuertas compuestas como las AOI21X1 y AOI22X1, las cuáles cuentan con mayor área. Para uart, el área obtenida es de 6.1692x10^{-8} m^2, la cuál también aumento al cambiar de librería.

Por otra parte, los retardos máximos tanto para arbiter como para uart aumentan, esto afecta directamente la frecuencia máxima, la cuál es reducida en ambos diseños, sin embargo, el diseño uart es el que se ve mayormente afectado, ya que su delay máximo incrementa de 2166.55 ps a 7507.98 ps, es decir, casi 3.5 veces. El aumento de área provoca un aumento en los retardos, sin embargo, este aumento no es proporcional ya que como se puede observar, el diseño de arbiter duplicó su área pero el delay sólo se vio afectado con un leve incremento, mientras que uart tuvo un leve incremento en su área pero un incremento muy significativo en delays para los paths con mayor retardo y por lo tanto su frecuencia máxima de operación se vio reducida, mientras que en los paths de menor retardo hubo un aumento muy leve (60 ps).

Para ambos diseños, la herramienta indica que estos cumplen con todos los tiempos de hold.

4. Conclusiones y recomendaciones

A partir del análisis de resultados se llega a las siguientes conclusiones

- Se logra obtener la cantidad de compuertas de cada librería que son utilizadas para los diferentes diseños
- Se obtienen características de timings a partir de la herramienta Qflow

- Se observa cómo los diseños tienen diferentes retardos y como estos retardos cambian cuando se utiliza otra librería, que dicho cambio no es lineal sino que un diseño se ve más afectado que el otro.

También se recomienda lo siguiente

- Leer la documentación del software para poder tener un mejor entendimiento de los comandos
- Trabajar de forma ordenada mediante un repositorio de git
- Cerrar el layout después de medir el largo o el ancho, ya que estas medidas se sobreponen y no son legibles si se intenta realizar ambas al mismo tiempo.