Politecnico di Torino

III Facoltà di Ingegneria

Lab 1
Integrated Systems Architecture

Master degree in Electronics Engineering

Authors: Group 25

Francesco Bono, Claudia Golino, Mattia Mirigaldi

Contents

CHAPTER 1

Reference model development

The purpose of the laboratory is to design a FIR filter (since the group number is odd) with cut-off frequency of 2 kHz and sampling frequency of 10 kHz.

1.1 Filter design with Matlab

The first step is launching a matlab script to extract the FIR results when a signal made of two sinusoidal waves at two different frequencies is at the input.
First of all has been computed with the equations in the laboratory description, the filter grade (N) and the number of bits of quantization (n_b).
N = 8 and n_b = 14.
Once updated the scripts with the found filter grade and number of bits, has been run it and have been obtained two .txt files that are respectively the input samples to be used subsequently in the filter and the expected results, both of them quantized on n_b.

1.2 Develop the fixed point model as a C program

The second step aim is to write a fixed point implementation in C language of the FIR in the direct form described by the equation 1.1

$$y = \sum_j x_{i-j} \cdot b_j \tag{1.1}$$

Starting from the IIR direct form c file given as example, has been eliminated the auto aggressive part in order to obtain the direct form of the FIR filter.
To evaluate the correctness of the code has been launched it and the results obtained have been compared with the ones coming out from the matlab script, founding that the results are comparable.

1.3 Evaluate the THD

The Total Harmonic Distortion (THD) of the results y of the fixed point implementation is THD = -73.9954 dB.
The value found is far grater than the one required, indeed in the assignment is required only a maximum THD of -30 dB. Taking advantage of this gap, the THD has been increased by lowering the precision of the filter that translates in decreasing the area of the filter.
Indeed, one possible way to decrease the area is to decrease the number of bits used to represent the

data (done by operating a left shift on the inputs data) that therefore lower the precision.
Empirically has been found that the bitwidth can be reduced up to 7 bits, expected result since the value of the coefficients used in this filter is quite low.
Here below the piece of code modified and after, in the table 1.1, the THDs found.

```
for (i=0; i<NT; i++)
y += ((sx[i]>>shift_val)*(b[i]>>shift_val)) >> (NB-shift_val-1) ;
```

shift_val is an integer variable representing the number of bits shifted used to test several possible shifts.

	No Shift	6bits	7bits	8bits
THD[dB]	-73.99	-38.8	-31.95	-23.58

Table 1.1: THD results

CHAPTER 2

VLSI implementation

2.1 Starting architecture development

The architecture of the FIR filter in VHDL has been done translating the flow of operations done in the c-description of the filter. According to the specifications the FIR is an 8-grade filter where data are quantized on 14 bits, meaning that the architecture
Takes as input :

- Input data "Din" on 14 bits
- 9 coefficients each on 14 bits
- Validation signal Vin that if '1' load a new sample in the architecture
- clock and complemented reset signals

Gives in output :

- Output data "Dout" on 14 bits
- Validation signal Vout that if '1' stores the evaluated Dout

The architecture implemented is shown in the figure 2.1:

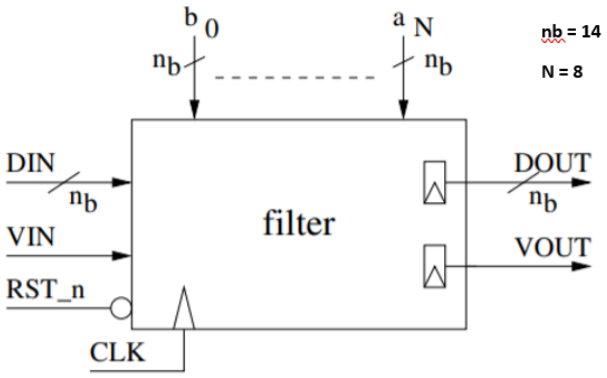

Figure 2.1: External interface of FIR filter

From previous total harmonic distortion (THD) evaluation has been found that using a bitwidth of 8 bits, the filter is able to met the THD requirement($THD < -30dB$).
This result has a strong impact on the area occupied by the filter since the partial operations are implemented using 8-bit operands, instead of the original 14-bit. In the designed architecture are taken the 8 most significant bits of Din and coefficients, are performed the partial operations by taking only the 8 most significant bits of the results, and lastly the obtained result is shift left by 6 positions such that D_{out} is on 14 bits.

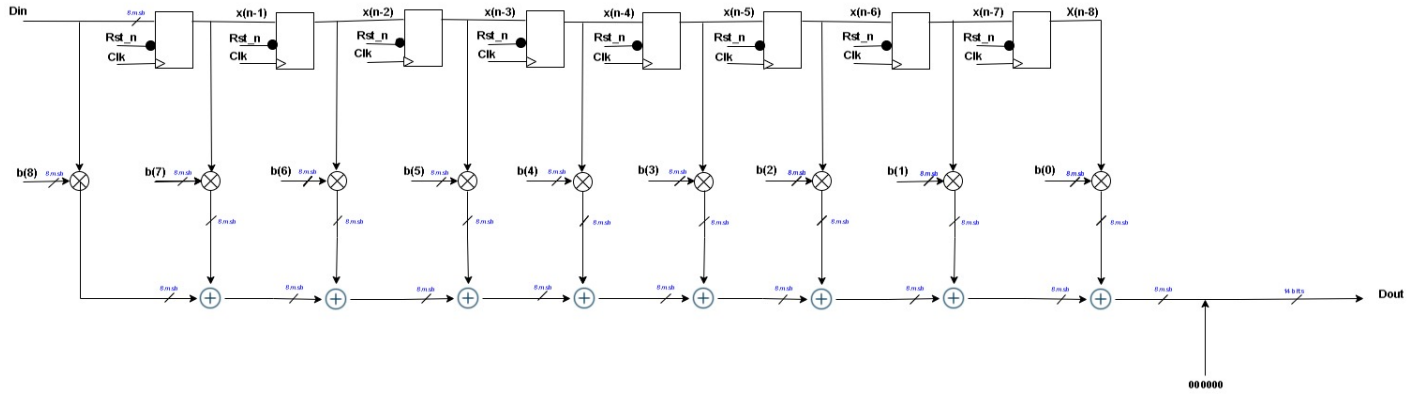

Figure 2.2: Internal architecture of the FIR filter

2.2 Simulation

The simulation was executed creating a testbench in verilog implemented as a module in which are connected four submodules.
These submodules, written in vhdl, are:

- *clk_gen* : it generates clock and reset signals
- *read_input* : it reads from *samples.txt* the inputs
- *DUT* : the filter under test
- *write_output* : it writes on a .txt file the results of the filter computation

To verify the correctness of the designed architecture the testbench has been run on modelsim for 2025 ns, that corresponds to the length of a complete execution. In the figure 2.3 a part of the simulation waves is shown.

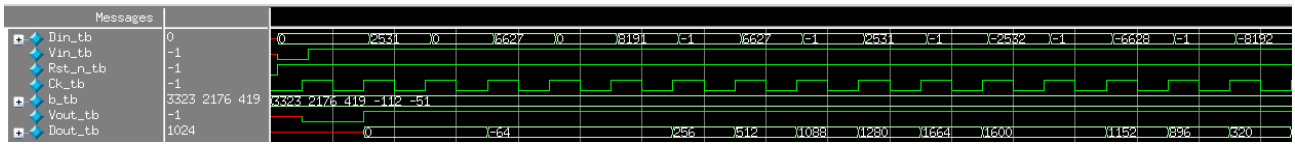

Figure 2.3: FIR Simulation

2.3 Implementation

The implementation of the filter consist in two fundamental steps:

- Synthesizing the circuit, to evaluate the characteristic of the design;
- Place & route, to map and connect the macros the synthesis netlist on the target silicon.

2.3.1 Logic synthesis

The filter has been synthesized with Synopsys Design Compiler (commands are shown in the script). The fundamental steps are the analysis and elaboration of the design, setting of the clock frequency and setting of the constraints, such as the clock uncertainty and the maximum delay of the signals. After the technology was chosen, the design is compiled in order to report its characteristic.
Below are listed the results achieved with the designed filter:

- The maximum clock frequency achieved is $f_{max} \approx 408$ MHz,that correspond to a 2.45 ns clock period to reach a 0 slack;
- Its corresponding total cell area is $A_{f_{max}} \approx 2.72$ Kcells.

Cell area				
	Combinational	Buf/Inv	Non Combinational	Total
f_{max}	2345.32	71.82	375.86	2721.18
$f_{max}/4$	2355.16	87.78	375.86	2731.02

Table 2.1: Area results

Afterwards the clock frequency of our design was set to $f_{clk} = f_{max}/4 \approx 102$ MHz (9.8 ns clock period) and the netlist was tested via simulation for 2025 ns, in order to verify the correctness of the output results, which must be the same as the ones obtained with the VHDL simulation.
Finally the power consumption of your design was estimated with the obtained switching activity.

Power consumption [µW]				
Total	Internal	Switching	Total Dynamic	Leakage
410.61	218.71	140.27	358.98	51.63

Table 2.2: Power results

We can observe that the net switching power and the cell internal power are respectively the 39% and 61% of the total dynamic power. The main power groups are the registers and the combinational part, which respectively consume the 19.18% and the 80.82% of the total power.
Report summary of the logic synthesis,table 2.3:

	Frequency [MHz]	Area	Power consumption [µW]
f_{max}	408	2721.18	-
$f_{max}/4$	102	2731.02	410.61

Table 2.3: Logic synthesis results

2.3.2 Place & Route

The Place and route of the filter was executed with Cadence SOC Innovus.
The main steps of this stage are: specifying the floorplan, inserting power rings,routing power, placing cells, optimizing, placing filler and finally routing cells.
After the design is completed a post routing optimization was applied to achieve the required timing constraints, with the following verification of connectivity and geometry.

The clock frequency of our design was set to $f_{clk} = f_M/4$ and the timing constraints are met, even after the post routing optimization ($slack > 0$).
The reported area is shown in table 2.4
After the post synthesis simulation, the power consumption of the design was estimated, table 2.5

LEVEL	Gates	Gates area [µm^2]	Cells	Area [µm^2]
0	3385	0.7980	1239	2701.5

Table 2.4: Gate count of FIR

Voltage [V]	Power consumption [µW]			
VDD	Total	Internal	Switching	Leakage
1.1	498.6	270.6	178	49.98

Table 2.5: Power results

We can observe that the internal, switching and leakage power are respectively the 54%, 36% and 10% of the total power.The sequential and the combinational groups respectively consume the 16.31% and the 83.69% of the total power.
In the power report there is also a summary for power distribution, where are indicated the highest average and highest leakage powers, the total capacitance and the number of instances in design.
Report summary of the Place & Route, table 2.6:

	Frequency ($f_{max}/4$) [MHz]	Area [µm^2]	Power consumption [µW]
SYNTHESYS	102	2731.02	410.61
DESIGN	102	2701.5	498.60

Table 2.6: Place and Route results

Innovus also allows to extract the value of parasitic capacitance and resistance for each metal wire that are important to analyze the behavior of the circuit over time.
The complete characterization of each net in the design in terms of its capacitance and resistance network can be found in file.spef.
The final schematic of the FIR can be observed in Figure 2.4

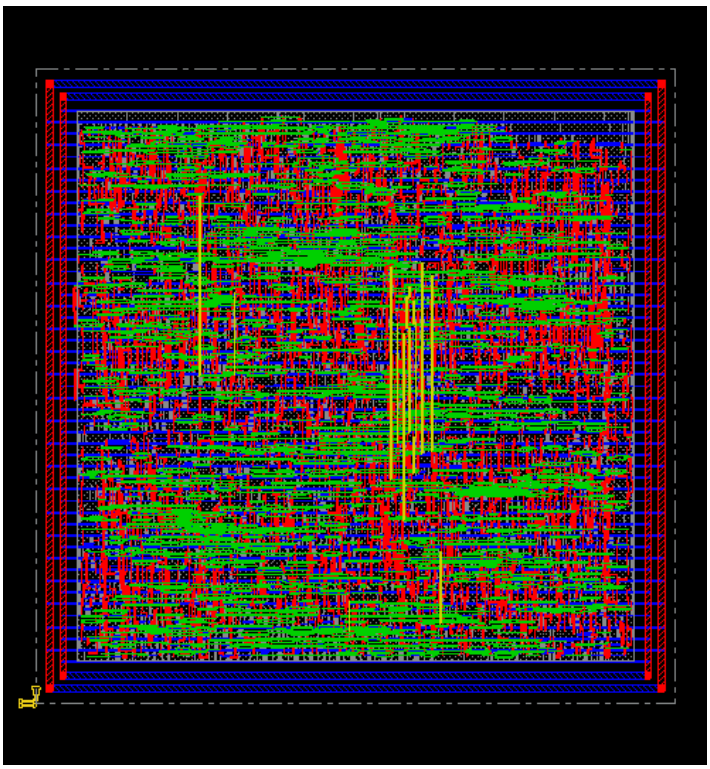

Figure 2.4: Physical layout FIR

2.4 Advanced architecture development

2.4.1 Unfolding

To improve the timing performance and throughput of the FIR filter can be used the unfolding technique that converts a sequential DFG into a parallel one, In this case the architecture has been unfolded with a degree of parallelism of 3.
Interesting to note is that the arcs in the DFG connecting two nodes with no register between the nodes are simply duplicated as it is in the other parallel structures. Instead, given the arc "e" in G (feed-forwad graph of the DFG) connecting nodes u to node v with w registers, allocate in G_p P arcs, going from u_i to v_j with w_i registers according to

$$j = \mod\left(\frac{i+w}{P}\right) \tag{2.1}$$

$$w_i = \left\lfloor \frac{i+w}{P} \right\rfloor \tag{2.2}$$

Where P in this case is equal to 3 since 3-unfolding and i = 0,1,2 (in generale i = 0 ... P-1).
The equations stated before are applied only to the arcs containing a register, that are the ones shown below :

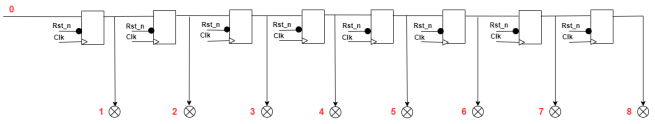

Figure 2.5: Arcs to unfold

In the below tables are reported the results obtained

Arc in the sequential DFG	i [Analyzed G]	j [Destination G]	w_i [Number of regs in the arc]
$0->1$	0	1	0
$0->2$	0	2	0
$0->3$	0	0	1
$0->4$	0	1	1
$0->5$	0	2	1
$0->6$	0	0	2
$0->7$	0	1	2
$0->8$	0	2	2

Table 2.7: Unfoled table for G_0

Arc in the sequential DFG	i [Analyzed G]	j [Destination G]	w_i [Number of regs in the arc]
$0->1$	1	2	0
$0->2$	1	0	1
$0->3$	1	1	1
$0->4$	1	2	1
$0->5$	1	0	2
$0->6$	1	1	2
$0->7$	1	2	2
$0->8$	1	0	3

Table 2.8: Unfoled table for G_1

Arc in the sequential DFG	i [Analyzed G]	j [Destination G]	w_i [Number of regs in the arc]
$0->1$	2	0	1
$0->2$	2	1	1
$0->3$	2	2	1
$0->4$	2	0	2
$0->5$	2	1	2
$0->6$	2	2	2
$0->7$	2	0	3
$0->8$	2	1	3

Table 2.9: Unfoled table for G_2

The architecture obtained with unfolding loads at each clock cycle three data inputs and store three data output, meaning that the obtained throughput is

$$Th = \frac{3}{t_{cp}} \tag{2.3}$$

where $t_{cp} = N * t_+ + t_*$ and in this case $t_{cp} = 8 * t_+ + t_*$

In the parallel DFGs there's a common structure shown in figure 2.6, called in vhdl *processing_unit*

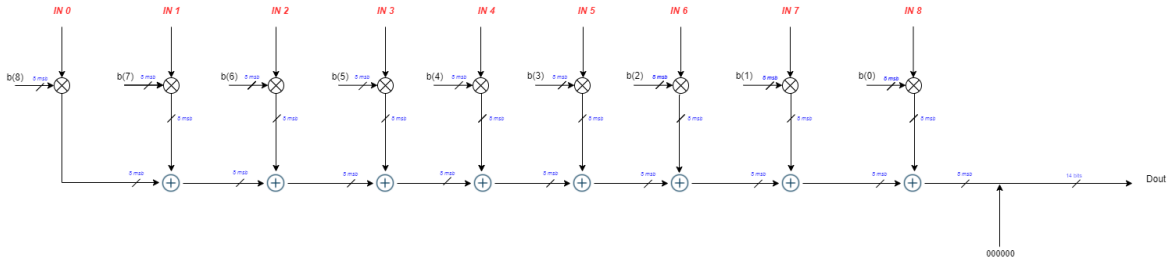

Figure 2.6: Processing unit

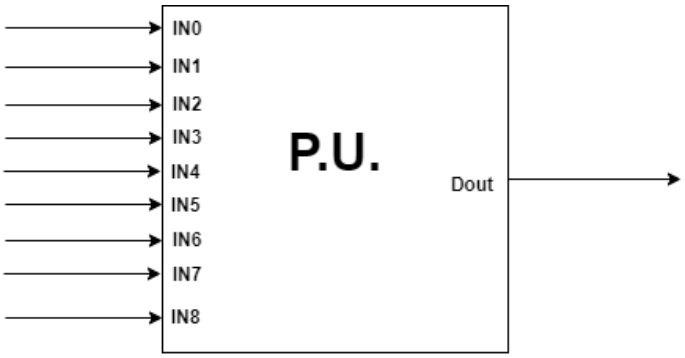

Figure 2.7: Interface of the processing unit

For ease of represention has been chosen to not represent the complete unfolded structure but to represent the individual parallel structures.
The processing unit in the parallel DFG_0 is fed according to table 2.7

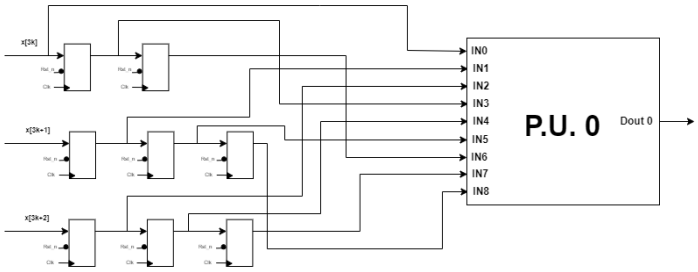

Figure 2.8: DFG_0

The processing unit in the parallel DFG_1 is fed according to table 2.8

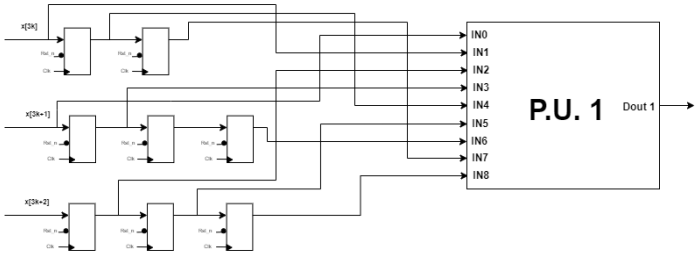

Figure 2.9: DFG_1

The processing unit in the parallel DFG_2 is fed according to table 2.9

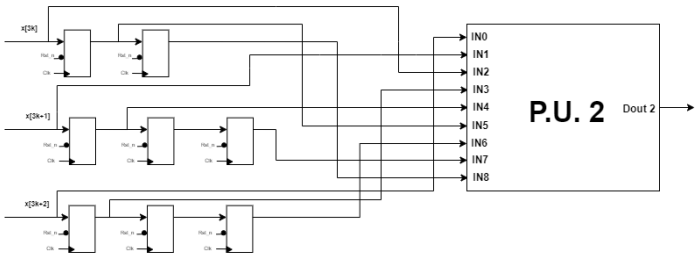

Figure 2.10: DFG_2

2.4.2 Pipelining

Another way to improve the throughput is by pipeling that reduce the critical path. In the picture 2.11 is shown the critical path :

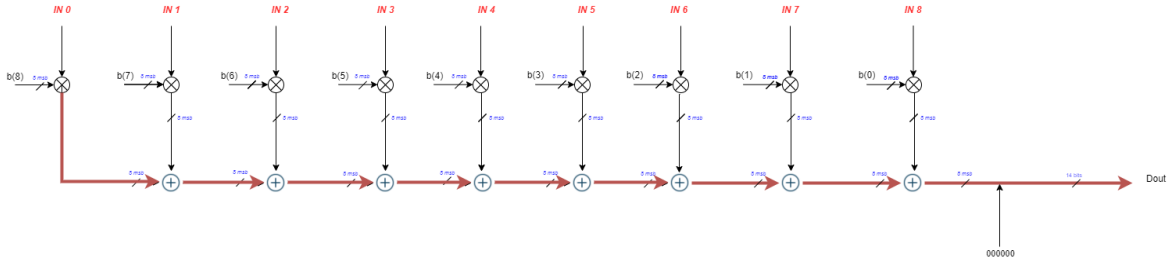

Figure 2.11: critical path in the FIR filter

$$t_{cp} = 8 * t_{+} + t_{*} \tag{2.4}$$

From the analysis done has been found that the max throughput is found by adding a 2-stage pipeline in the critical path.

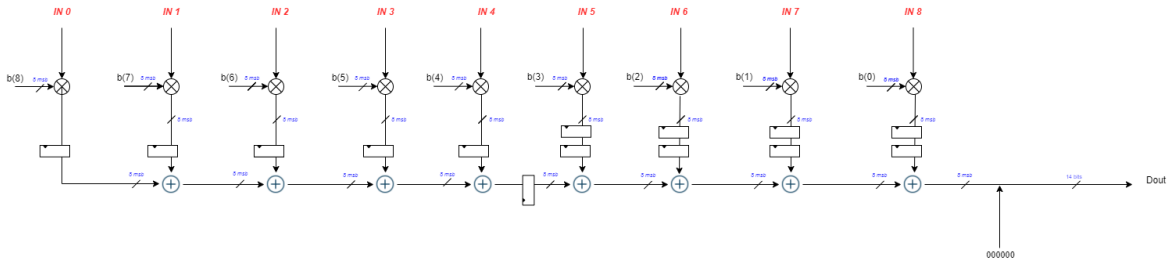

Figure 2.12: 2 stage pipeline

$$t_{cp} = max[4 * t_{+}, t_{*}] \tag{2.5}$$

2.4.3 Logic synthesis

The unfolded and pipelined filter has been synthesized with the same procedure of section 2.3.1, showing the following results.

- The maximum clock frequency achieved is $f_{max} \approx 556$ MHz,that correspond to a 1.8 ns clock period to reach a 0 slack;
- Its corresponding total cell area is $A_{f_{max}} \approx 11.39$ Kcells.

Afterwards the clock frequency of our design was set to $f_{clk} = f_{max}/4 \approx 139$ MHz (7.2 ns clock period) and the netlist was tested via simulation for 695 ns.

Cell area				
	Combinational	Buf/Inv	Non Combinational	Total
f_{max}	8026.82	415.76	3362.24	11389.06
$f_{max}/4$	7899.14	374.26	3362.24	11261.38

Table 2.10: Area results

After the post synthesis simulation, the power consumption of the design was estimated (table 2.11).

Power consumption [mW]					
FIR	Total	Internal	Switching	Total Dynamic	Leakage
Unfolded & pipelined	1.80	1.08	0.51	1.58	0.22

Table 2.11: Power results

We can observe that the net switching power and the cell internal power are respectively the 32% and 68% of the total dynamic power.
The main power groups are the registers and the combinational part, which respectively consume the 39.67% and the 60.33% of the total power.
Report summary of the logic synthesis, table 2.12:

	Frequency [MHz]	Area	Power consumption [mW]
f_{max}	556	11389.06	-
$f_{max}/4$	139	11261.38	1.8

Table 2.12: Logic synthesis results

2.4.4 Place & Route

For the Place and Route of the unfolded and pipelined filter we proceeded as in the previous case.
The clock frequency was set to $f_{clk} = f_M/4$ and also in this case the timing constraints are met before and after the post routing optimization.
The reported area is shown in table 2.13

LEVEL	Gates	Gates area [µm^2]	Cells	Area [µm^2]
0	13791	0.7980	4744	11005.5

Table 2.13: Gate count of FIR unfolded and pipelined

After the routing a simulation was executed with the design optimized by innovus and it worked properly.
In the figure 2.13 the starting waves of the simulation are shown.

Figure 2.13: FIR unfolded post-route simulation

After the simulation the power analysis was extracted. We can notice that the internal power represent the 61.3% of the total power while the switching is the 31.6% and the leakage around the 7%. In this implementation the leakage and the switching power are decreased by almost the 8%.
In the table 2.14 the power summary is highlighted.

Voltage [V]	Power consumption [µW]			
VDD	Total	Internal	Switching	Leakage
1.1	2848.7	1746.4	899.8	202.4

Table 2.14: Power results Fir unfolded and pipelined

Report summary of the Place & Route, table 2.15:

	Frequency ($f_{max}/4$) [MHz]	Area [µm^2]	Power consumption [µW]
Starting design			
SYNTHESYS	102	2731.02	410.61
DESIGN	102	2701.5	498.60
Unfolded & pipelined			
SYNTHESYS UNF.	139	11261.38	1800
DESIGN UNF.	139	11005.5	2848.7

Table 2.15: Place and Route overall results

At the end the final schematic of the unfolded and pipelined FIR is shown in figure 2.14.

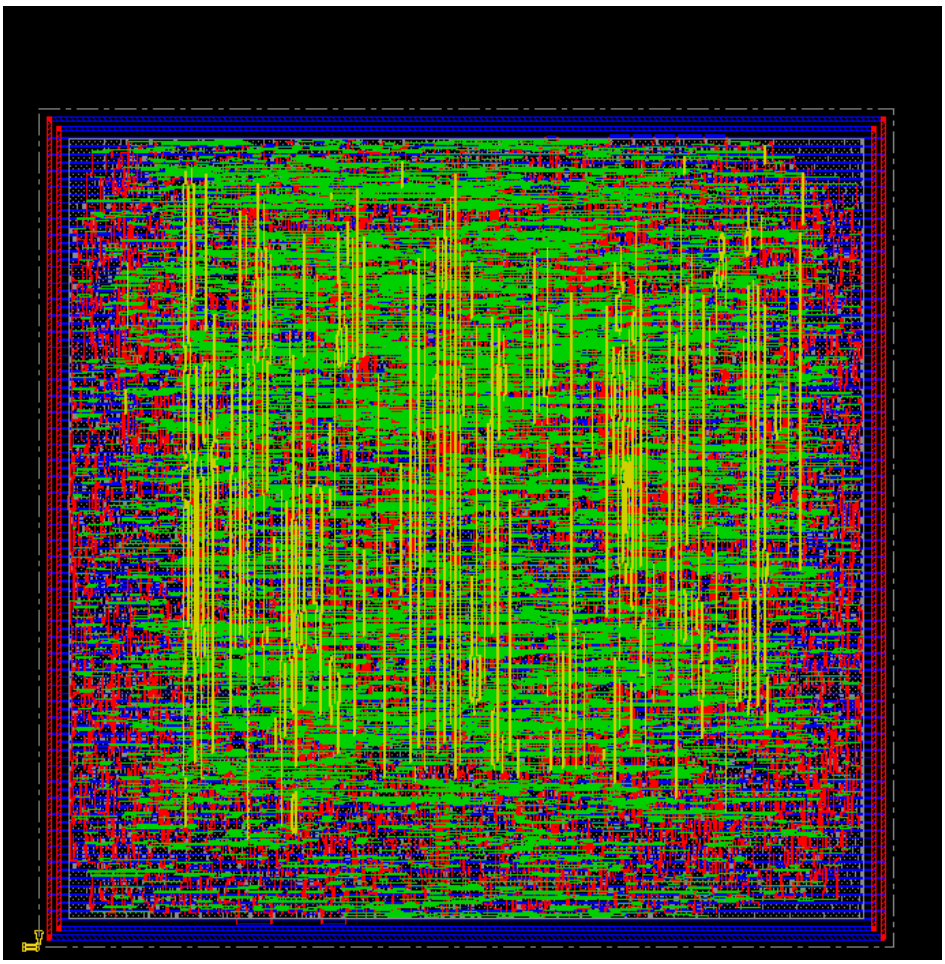

Figure 2.14: Physical layout FIR unfolded and pipelined