Politecnico di Torino

Collegio di Elettronica, Telecomunicazioni e Fisica

Laboratory 1 : Design and implementation of a digital filter

Report for the course
Integrated Systems Architecture

Master degree in Electronics Engineering

Authors: Group 09

January 13, 2024

Contents

CHAPTER 1

Part 1: Reference model development

1.1 Matlab model

1.1.1 IIR Direct Form II - Filter specifications

The goal of this activity has been to design an Infinite Impulse Response Filter (IIR) in direct form II, given that our group number is odd(i.e. $p = 0$).
The specifications of this filter were obtained following the algorithm described in the lab's descriptions file. Given the surnames of the group members reverse ordered, as shown below, it follows that $x = 7$ and $y = 6$.

1. Laouibi
2. Giunti
3. Capruzzi

Thus the specifications obtained are the followings:

- Order of the filter N is computed as:

$$N = 2^p \cdot [(x \bmod 2) + 1] + 6 \cdot p = 2 \tag{1.1}$$

- Number of bits n_b is obtained as :

$$n_b = (y \bmod 7) + 8 = 14 \tag{1.2}$$

- Cut-off frequency $f_c = 2kHz$;
- Sampling frequency of the samples to process $f_s = 10kHz$;
- Maximum Total Harmonic Distortion (THD)=-30 dB.

Then the filter that we need to implement computes the output samples as follows:

$$y[n] = \sum_{k=0}^{N} b_k \cdot x[n-k] - \sum_{k=1}^{N} a_k \cdot y[n-k] \tag{1.3}$$

Where:

- $y[n]$ is the output sample;

- $x[n]$ is the input sample;
- N is the order of the filter previously computed;
- a_k and b_k are the coefficients computed externally that define the frequency response of the filter.

1.1.2 Matlab model description

Given that we have a second order filter, we need three coefficients for the feed-forward part, namely b_0, b_1 and b_2 and two for the feed-back one, a_1 and a_2.

To compute them meeting the required specifications, a simple Matlab code was developed. More in detail Matlab calculates these coefficients first with a floating point arithmetic, then we perform a quantization step to prepare these coefficients to be implemented in a digital design working in fixed point.

The coefficients of the filter have one bit for the integer part and 13 bits for the fractional part (Q1.13 format). They correspond to the integer values shown in Table 1.1.

The quantization has been performed considering the bit-width of our design ($n_b = 14$). A plot of the frequency response of the filter for both floating point and fixed point coefficients has been obtained and depicted in Figure 1.1, where we can notice that the two curves, related to the floating point coefficients (blue curve) and the fixed point ones (red curve), nearly overlap. There is a small shift as expected since some precision is lost in the quantization step, but the overall behaviour is identical and the frequency response is approximately -3 dB for the two curves at the normalized cut-off frequency, that is the one expected, in fact:

$$f_c{}^{norm} = \frac{f_c}{f_{nyquist}} = \frac{2kHz}{5kHz} = 0,4 \tag{1.4}$$

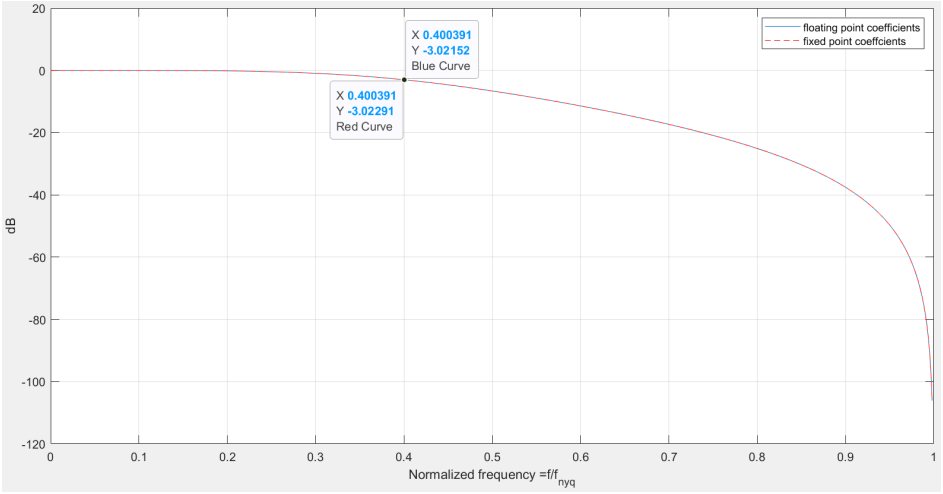

Figure 1.1: Plot of the frequency response for both floating point and fixed point coefficients.

Lastly, to test the filter design in Matlab, two sinusoidal waves have been created, one at a frequency in the band of the filter, at 500 Hz, and the other at an out of band frequency of 4500 Hz. These two

Coefficient name	Real value	Quantized value
b_0	0,2065	1692
b_1	0,4131	3384
b_2	0.2065	1692
a_1	0.3696	3028
a_2	0.1958	-1604

Table 1.1: IIR's filter coefficients

waves are sampled at the sampling frequency (10 kHz), averaged at each point and given as input to the filter, which outputs a filtered signal, as shown in Figure 1.2. The result of the filtering operation has been quantized in 14 bits and saved in an output text file, as well as the input, which will be needed in the next steps for comparison purposes.

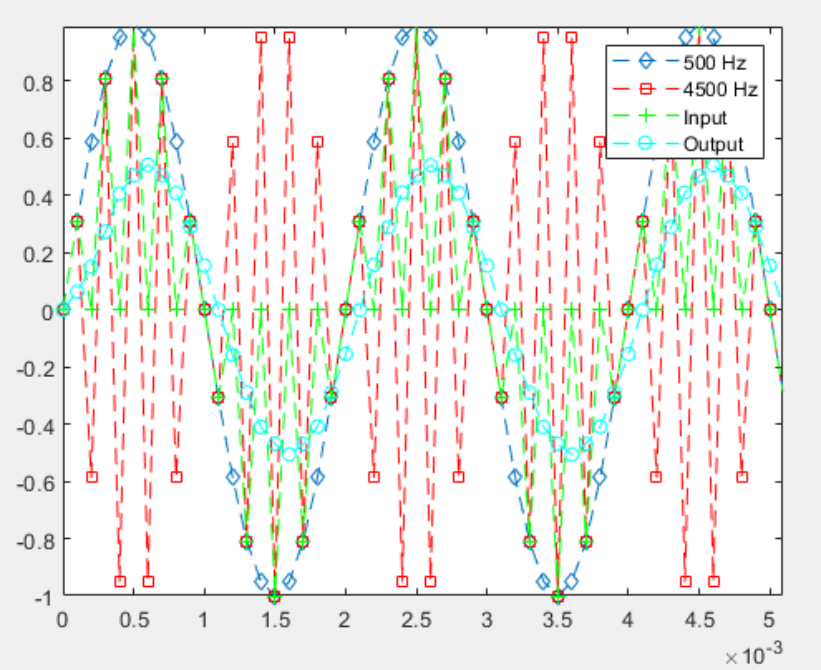

Figure 1.2: Plot of the generated waveforms, where input is given by the average of the 2 individual tones. The filter is filtering the out of band frequency.

1.2 C model and THD

The Matlab script described above gives us the coefficients of the filter and the input waveform values correctly sampled, with a dynamic already scaled to suit a representation in 14 bits, that is the bit-width we need to work with.

After that the fixed point model is developed as a C program, that is necessary for two reasons:

- Evaluating the performance of the fixed point implementation by comparing the results of C implementation to the Matlab ones, since in Matlab operations are performed with infinite

precision while in our architecture we have a finite precision in order to limit the total area.

- This will be used as a reference model to develop, debug and test the digital filter as an hardware architecture.

1.2.1 C model description

The C code implemented receives the input samples as a text file, whose name is given in the command line, then if the file is opened correctly and the shift amount is correct (more on this later), it enters in a loop that reads the file line by line and gives the samples one by one to the function named "myfilter" that, as the name suggests, filters them by applying the equation 1.3, with the coefficients received from matlab; then the output of myfilter is returned back and written in an output text file whose name is also given as a parameter from the command line.

The function "myfilter" is composed of four parts:

- The first part cleans the intermediate registers (represented by vector the sw[N]), if it's the first input received from the main, to be sure to start from correct point;
- The second part is a loop that computes the feed-back and the feed-forward components of the filter for the samples stored in the registers x[n-1] and x[n-2];
- The third part sums the input sample to the feed-back part, multiplies it by b_0 and it sums the result to the feed-forward part;
- finally, the intermediate registers are updated with the values computed before.

1.2.2 THD and Precision

The goal of the fixed point architecture is to implement the filtering required limiting the area as much as possible: to do so we can reduce the precision of the results of the multiplication, since the multipliers are the components that consumes the largest portion of area in our design, in fact the bit-width of the results doubles at each operation.

Since truncating the bit-width will cause a loose of precision, we need a figure of merit to understand the entity of the error on the results: to this purpose the Total Harmonic Distortion (THD) can be used, as the name suggests, it measures the harmonic distortion present in a signal (i.e. the harmonics of the fundamental frequency introduced by the truncation).

The minimum number of bits we need to truncate after each multiplication to allow all operators to work at 14 bits is 13 LSBs and one MSB, since the MSB bits of the result are related to the sign and so the first bit will be always '0'. In this way the result of multiplications will be in 14 bits and then we will need to cut one bit from the result of the addition to have it also in 14 bits, in this way the THD of the results is -76.93 dB, this is the minimum THD we can get: i.e. this is the maximum precision we can get from this fixed point implementation since all operators can work with 14 bits.

By increasing the number of bits truncated to the multiplication result, the THD increases. The maximum THD that we can accept is -30 dB, as indicated ny the specifications. To respect this limit the maximum truncation we can do is of 20 bits, which results in a THD of -37.24 dB.

In this way the overall precision is reduced, however internally we can work with 7 bits of precision: that results in a huge reduction of the hardware required, as explained in next sections.

1.3 Comparison between the two implementations

In figure 1.3 we can appreciate that the results produced by the C fixed point implementation are less accurate than those of Matlab (floating point), as expected, however the difference is small compared to big advantage in terms of area saving realized by truncating 20 bits.

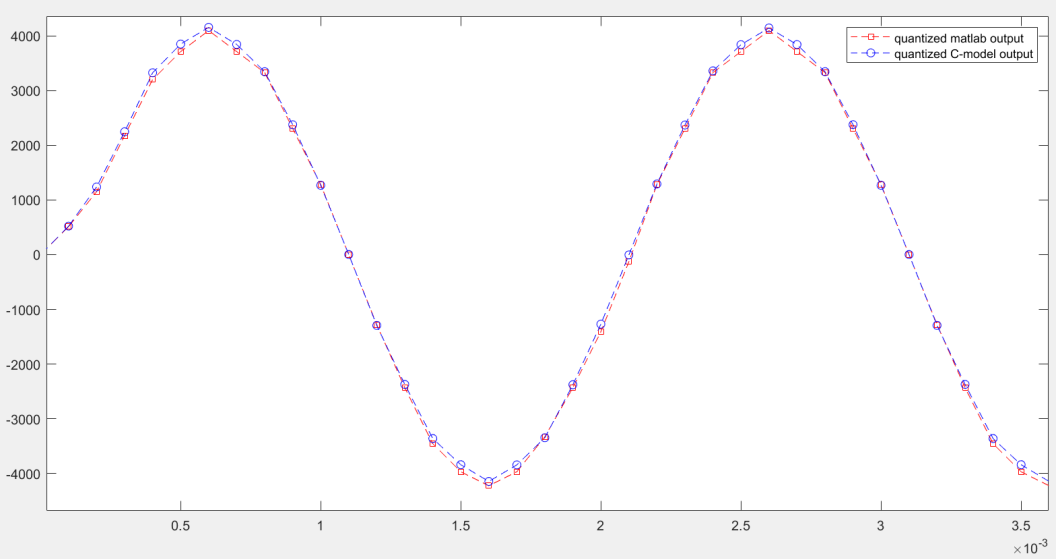

Figure 1.3: Plot of the generated waveforms, where "Quantized Matlab output" is the output generated and quantized on 14 bits and "Quantized C model output" is the output produced by C model that is represented on 14 bits too.

CHAPTER 2

VLSI implementation

2.1 Architecture description

The block scheme of the architecture of the filter, shown in Figure 2.1, was derived from equation 1.3 and it is composed by:

- One Subtractor of 14 bits;
- Two adders of 7 bits;
- One adder of 8 bits;
- Five multipliers of 14 bits.
- Two intermediate Registers to produce delayed versions of input;
- Input and output Registers.

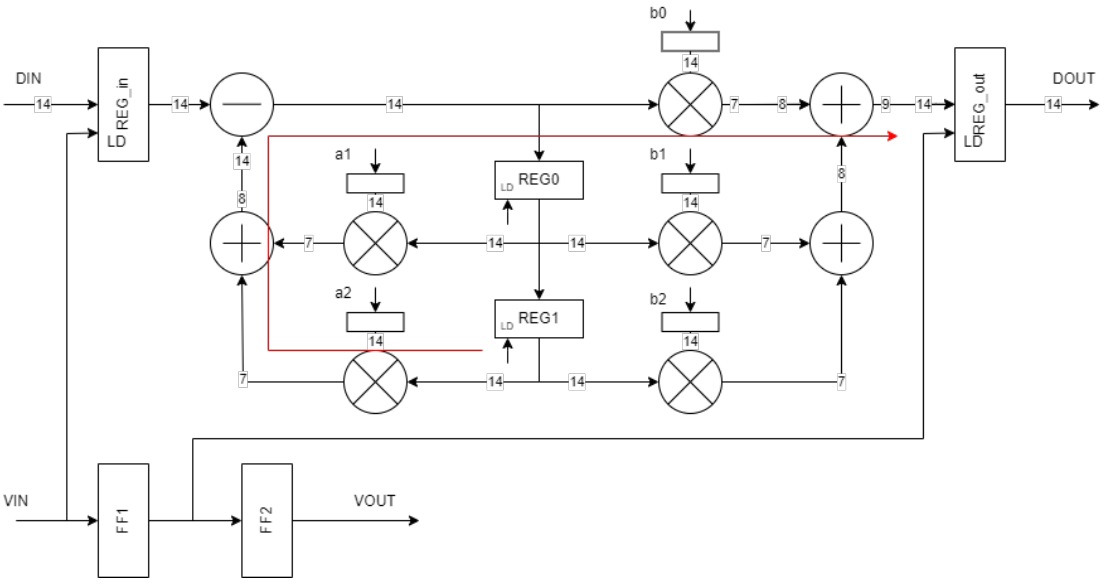

Figure 2.1: Schematic of the IIR filter as implemented in VHDL, with each line indicating the number of bits of that connection. The red line indicates the critical path.

In Figure 2.2 we can see the expected timing diagram of this architecture, with results of the processing of the first samples of the signal generated from Matlab: the results at the output DOUT are the same as the ones generated by the fixed point C model, and it works also taking into account when VIN moves from '1' to '0' and then back to '1'.

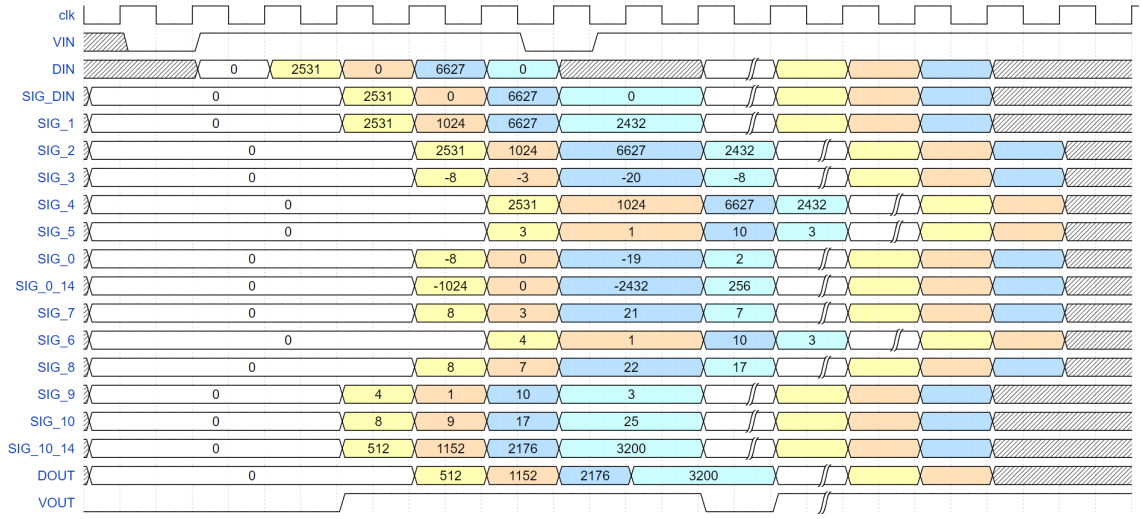

Figure 2.2: Timing diagram analysis.

As can be seen in Figures 2.1 and 2.2, compared to the basic implementation of the filter, the schematic has been modified to suit a real working condition of an IIR Direct Form II filter: all inputs and outputs of the filter are loaded by registers, moreover, this filter works only when an external validation signal (VIN), which could be generated by a preceding unit, rises to indicate that the input data (DIN) is valid and can be processed: as a result this signal has been given as load of the input data register. This architecture handles internally also the generation of the valid out signal (VOUT), that is the result of the VIN delayed by 2 clock cycles, that is the time needed by the filter to produce a valid output data (DOUT). The component receiving the DOUT in this way accepts the filtered signal only when VOUT is at logic value '1'.

2.2 VHDL model

Using the block scheme shown in Figure 2.1, the implementation of the IIR filter in VHDL was developed in six different files that describe each component represented in the block diagram, plus a top entity that connects the components together according to the schematic. The components implemented are the following:

- Adder;
- Subtractor;
- Multiplier;
- Register with Enable;
- Register with no Enable;
- Flip Flop;

All these components have been coded in VHDL language using a behavioural description; here we report a brief explanation of the non standard modifications introduced in these components:

- Inside the Adder file, there is a part dedicated to the cases of overflow: in a few words it only verifies if the signs of the inputs are the same and the output has a sign that is different from them it rises the overflow flag, and if this is the case then the output is omitting the LSB of the result, otherwise, if the overflow flag is low then the result discards the MSB;
- Inside the Subtractor file, we just invert the sign of the second operand and then an addition is performed, moreover the overflow verification is performed in the same way as the Adder case;
- The output of the multiplier omits the MSB of the result because the most significant two bits are related to the sign, thus the MSB is always '0', and it also truncates 20 LSB, that is the truncation we implemented to save area;
- The register with no Enable was used only for coefficients (only for simulation purposes, see **"es1v3.0_description.pdf"**), while the register with the enable was used for signals.

To verify this architecture against the fixed point C model we used a test bench composed by the following elements:

- Clock generator: a component that generates the clock;
- Data maker: a component that provides the coefficients and the input samples, generated by Matlab and provided in a text file, one by one to the filter. In addition to this it also generates VIN signal and an end simulation signal;
- Data sink: a component that reads the results generated by C model, provided in a text file, and reading the results generated by our filter it verifies them one by one, and if it detects some errors it reports them;
- Finally, a test bench component that is connecting all these elements to the the top entity file of our design.

2.3 Simulation

In this phase the previous architecture has been simulated. In particular all the instructions written in the files **"es1v3.0_description.pdf"** and **"documents.pdf"** have been followed in order to perform correctly the simulation process using Questasim.

The VHDL files of the filter have been put into the folder named **src**, while test bench files (mixed VHDL/Verilog) have been put into **tb** folder. Regarding the files produced by Questasim and the text file with input data (**samples.txt**), they have been inserted into **sim**, the main directory for the simulation and precisely into the folder **txt_files**. In order to run Questasim with the proper settings, it has been exploited a script file named **run_sim.do**.

IMPORTANT NOTE: from this moment on, for sake of simplicity, the precise location of the useful files will not be reported in this report, but it is well explained in each folder by a text file named **READ_ME.txt** which also contains a brief description on what the file is used for.

To process all the samples the simulation lasts 38.80 us, keeping in mind that in this phase a clock period equal to $T_{clk} = 100\ ns$ has been used as an example. Note that this time interval has been

taken from the first rising edge of the clock sampling VOUT = '1' to the last rising edge of the clock sampling VOUT = '1'. A snapshot from 0 ns to 1600 ns is shown in Figure 2.3 to highlight the behaviour of the filter when VIN moves from 0 to 1 and viceversa.

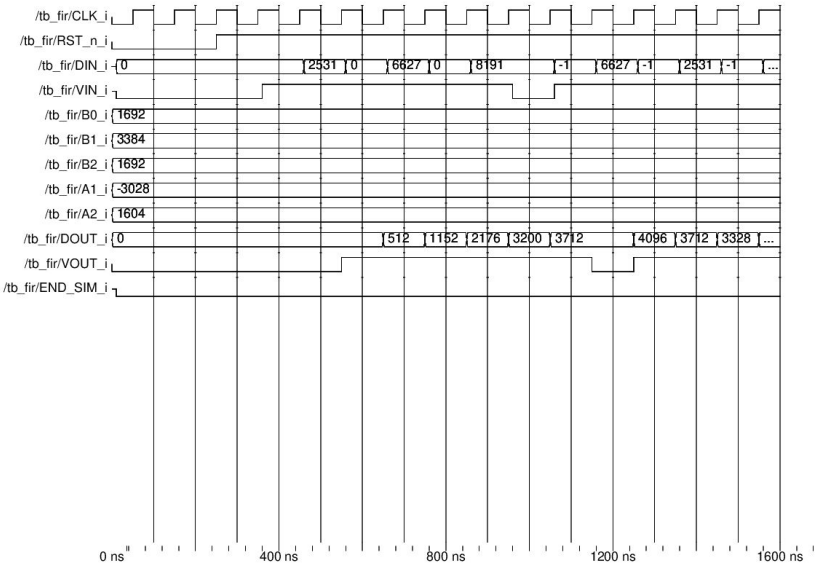

Figure 2.3: Waveform for $T_{clk} = 100\ ns$

In addition to that, it has been verified that the results given by the VHDL model are equal to the results of the C model. It can be easily seen by the comparison between the files **results_c.txt** and **results_hdl.txt**.

2.4 Logic synthesis

In order to perform the synthesis with Synopsys Design Compiler, all the steps indicated in the files **"documents.pdf"** and **"es1v3.0_description.pdf"** have been followed. Then the maximum achievable clock frequency has been found by changing the constraints on the period of the applied clock, until the timing report has given an overall slack equal to 0 with the minimum clock period. Subsequently also the area has been found by mean of the command "report_area".

In the following pictures, there are the snapshots relative to the most meaningful lines of the timing report (Figure 2.4) at the maximum frequency and the report of the area in the same conditions (Figure 2.5).

Then the clock frequency has been set to $f_M/4 = 71.43\ MHz$ with a clock period of 14 ns, where f_M corresponds to the maximum frequency previously obtained: about 278 MHz. Note that the precise quarter frequency is $f_M/4 = 69.44\ MHz$ with a corresponding period of 14.4 ns. Despite this, it has been preferred to make a synthesis with a period in the resolution of 1 ns, this because this is the timescale at which our design is set. Attempts have been made in order to change the timescale to an order of at least 100 ps for setting the precise clock period of 14.4 ns, but this sort of operation gave problems in Questasim which at its turn linked errors related to the library **"nangate45"**. For this reason, it has been preferred to set the clock period to 14 ns, knowing that working precisely at

```
clock MY_CLK (rise edge)                                  3.60       3.60
clock network delay (ideal)                               0.00       3.60
clock uncertainty                                        -0.07       3.53
REG_out/Q_reg[6]/CK (DFFR_X1)                             0.00       3.53 r
library setup time                                       -0.04       3.49
data required time                                                   3.49
---------------------------------------------------------------------------
data required time                                                   3.49
data arrival time                                                   -3.49
---------------------------------------------------------------------------
slack (MET)                                                          0.00
```

Figure 2.4: Timing report at f_M= 278 MHz

```
****************************************
Report : area
Design : myIIR2
Version: S-2021.06-SP4
Date   : Sat Nov 19 14:07:21 2022
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /eda/dk/nangate45/synopsys/NangateOpenCellLibrary_typical_ecsm.db)

Number of ports:                         1078
Number of nets:                          4754
Number of cells:                         3480
Number of combinational cells:           3312
Number of sequential cells:               128
Number of macros/black boxes:               0
Number of buf/inv:                        663
Number of references:                      28

Combinational area:              4757.675998
Buf/Inv area:                     474.543999
Noncombinational area:            682.290022
Macro/Black Box area:               0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                 5439.966020
Total area:                 undefined
1
```

Figure 2.5: Area report at f_M= 278 MHz

quarter frequency is not an "hard limit" but it is just an operating condition that is set in order to work in a range of frequency with a slack that is necessarily positive.
The corresponding area of the analyzed cell in these new conditions has been reported in Figure 2.6: it is easy to see that at maximum clock frequency, the area of the cell is higher than in the case $f_M/4$ because of the different implementation of the blocks used by the synthesizer to speed-up the architecture and make it compliant with the given constraints.

All the practical explanations regarding the location of the scripting files and the results in the delivered folders are leaved to the previously quoted **READ_ME.txt** file.

The netlist has been then verified via simulation, setting properly the new clock frequency. As a matter of fact, new files have been created in the **tb** folder: **clk_gen_fm4.vhd** and **data_sink_fm4.vhd**. This because in order to verify the design at quarter frequency it was necessary to change the clock period inside the clk_gen.vhd file and the data_sink.vhd was modified to output the filtered samples in a new file named **results_hdl_fm4.txt**.

Subsequently, it has been reported in Figure 2.7, the new waveform for the simulation at quarter frequency in an interval from 0 ns to 210 ns.

It can be demonstrated that also this time, by comparing the files **results_c.txt** and **results_hdl_fm4.txt** the C-model results and the VHDL results for the filtered samples are the same. Moreover, the amount

```
****************************************
Report : area
Design : myIIR2
Version: S-2021.06-SP4
Date   : Sat Nov 19 14:03:48 2022
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /eda/dk/nangate45/synopsys/NangateOpenCellLibrary_typical_ecsm.db)

Number of ports:                         1078
Number of nets:                          3782
Number of cells:                         2396
Number of combinational cells:           2230
Number of sequential cells:               128
Number of macros/black boxes:               0
Number of buf/inv:                        355
Number of references:                      21

Combinational area:              4037.614018
Buf/Inv area:                     210.140000
Noncombinational area:            680.960022
Macro/Black Box area:               0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                 4718.574040
Total area:                 undefined
1
```

Figure 2.6: Area report with $f_M/4$

```
clock MY_CLK (rise edge)                                 14.00      14.00
clock network delay (ideal)                               0.00      14.00
clock uncertainty                                        -0.07      13.93
REG_out/Q_reg[7]/CK (DFFR_X1)                             0.00      13.93 r
library setup time                                       -0.04      13.89
data required time                                                  13.89
--------------------------------------------------------------------------
data required time                                                  13.89
data arrival time                                                   -9.18
--------------------------------------------------------------------------
slack (MET)                                                          4.70
```

Figure 2.7: Timing report with $f_M/4$

of time required to complete the simulation is 5.43 us.

Regarding the estimation of the power consumption, both Questasim and Synopsys Design Compiler have been exploited. First of all, Questasim has been launched by mean of a set of commands in the file **run_verif.do** that, apart from performing the simulation at quarter frequency, it saves in the file **myIIR2_syn.vcd** all the switching information. Then these information have been converted in a .saif file to be used by Synopsys Design Compiler. After these operations, the power consumption estimation has been performed by Synopsys using the commands in file **pow_cons_est.syn** and the results are saved in a .txt file (look at the readme file). From that last file it can be seen that the total used power is about 390.99 uW. All the most important previous results are summarized in the Table 2.1.

| $f_M/4 = 71.43MHz$ | $A = 4718.57um^2$ | $P = 390.99uW$ | T= 5.43 us |

Table 2.1: Most important results of the synthesis with SYNOPSYS where A is the area, P is the power consumption and T is the simulation time.

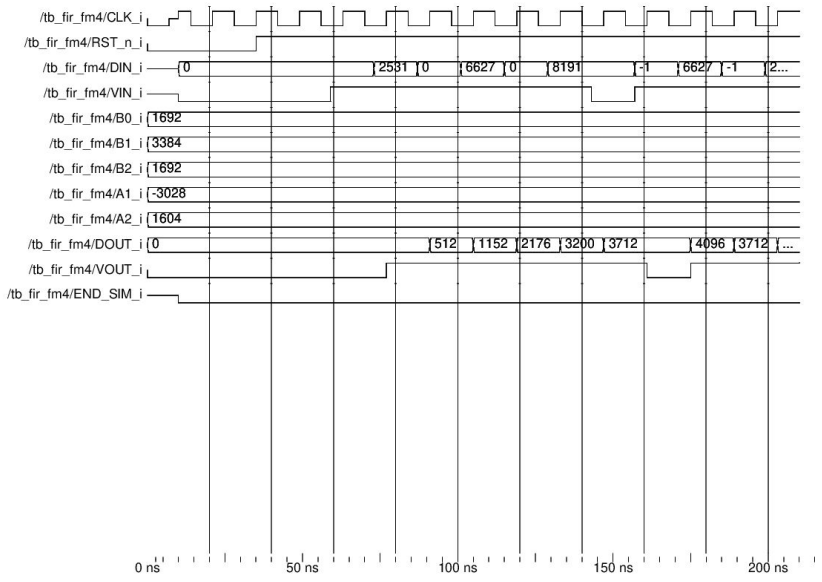

Figure 2.8: Waveform with $f\_M/4$

2.5 Place and route

In this section it has been reported the results of the Place and Route of the design at quarter frequency using as the Cadence Innovus instrument. It is important to remark that all the relevant files produced by the software have been located in the folder **innovus**. To perform the place and route operations all the instructions reported in files **"es1v3.0_description.pdf"** and **"documents.pdf"** have been followed and then, exploiting the command file produced by Innovus, two scripts have been created in order to perform respectively the Place and Route of the architecture and the following power estimation in "batch-mode".

Regarding the area of the circuit, Innovus gives the possibility to print many sorts of reports, so by reporting the "gate count", the overall area of the circuit has been obtained. In Figure 2.9 have been reported the gate area and the total area of "myIIR2" module.

```
Gate area 0.7980 um^2
Level 0 Module myIIR2                          Gates=      5778 Cells=    2243 Area=    4610.8 um^2
```

Figure 2.9: Circuit area after Place and Route design with $f\_M/4$

After this operation, the design has been verified using Questasim, so a new command file named **innovus.do** has been created that has been used for the purpose of simulation. Note that the filtered samples are wrote in the file **results_hdl_fm4_innovus.txt** that have to be compared with the filtered samples of the C-model in **results_c.txt**. It is easy to see that the two files match. From what concerns the amount of time required to complete the simulation, it has been verified that it is 5.43 us. In Figure 2.10 there is a snapshot of the waveform from the beginning until 210 ns, useful to verify the correct behavior of the design.

In order to evaluate the power consumption, as the previous case with Synopsys, first the switching activity has been recorded and saved in a file named **design.vcd**. Then this file has been uploaded by

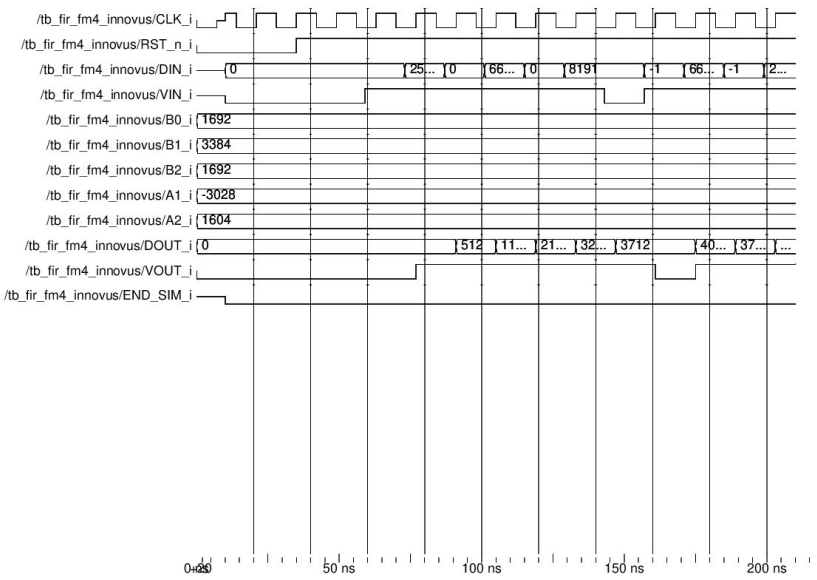

Figure 2.10: Waveform after Place and Route design with $f\_M/4$

Innovus that at this time produced an output file with the power report named **power_report_new**. As it can be seen by looking at this report, the total power of the circuit is 0.4006 mW (400.6 uW).

No timing violation for both setup and hold modes of the design with Innovus, it can be checked looking at files **myIIR2_postRoute.slk** and **myIIR2_postRoute_hold.slk**. It can be seen that the slack is always positive so there are no timing violations as expected.

In order to summarize the most important results of the place and route, Table 2.2 can be consulted.

$$\mid f_M/4 = 71.43MHz \mid A = 4610.8um^2 \mid P = 400.6uW \mid T = 5.43us \mid$$

Table 2.2: Place and Route results: A is the area, P is the power consumption and T is the simulation time.

2.6 Explanations, comparisons and comments

As it can be observed, we have two sets of results regarding the area of the filter, its simulation time and its power estimation at 71.43 MHz: the first has been obtained with the synthesis tool Synopsys Design Compiler while the second one with Cadence Innovus, used for the Place and Route process. These results, highlighted respectively in Table 2.1 and in Table 2.2, obviously are very close because they refer to the same architecture designed in VHDL but not exactly the same (apart from the simulation time that depends in both cases from Questasim).

The reason is that Synopsys, given as input the VHDL design, produces as output the netlist of the input module, so a gate level representation of it with a certain technological node (in our case 45 nm) and estimate the power consumption and area starting from this netlist. On the other hand, Innovus produces the same kinds of report starting from the Place and route process given the netlist as input, so basically it "places" the logic elements of the netlist in a limited amount of space and "routes" them, so decides the design of all wires needed to connect the placed components. This

difference implies that the results given by Innovus are closer to the real conditions of a possible implementation of the filter because it takes into account also the line capacitances of real interconnections while the results given by Synopsys are more theoretical.

CHAPTER 3

Advanced architecture development

3.1 IIR Filter with the J-look-ahead improvement

3.1.1 Theoretical derivation:

The previously realized architecture has been improved using the J-look-ahead technique, with J = 1. This technique allows us to apply the universal techniques (e.g. retiming and pipelining) in a more efficient way to increase the throughput of the architecture.

The starting point is to compute the sample $y[n+1]$, using the equation 1.3, as shown in the following:

$$y[n+1] = \sum_{k=0}^{N} b_k \cdot x[n+1-k] - \sum_{k=1}^{N} a_k y[n+1-k] = \tag{3.1}$$

$$y[n+1] = b_0 \cdot x[n+1] + b_1 \cdot x[n] + b_2 \cdot x[n-1] - a_1 \cdot y[n] - a_2 \cdot y[n-1]) \tag{3.2}$$

Then we substitute the $y[n]$ term with the value of equation 1.3. It follows that:

$$y[n+1] = b_0 \cdot x[n+1] + b_1 \cdot x[n] + b_2 \cdot x[n-1] - a_1 \cdot \{\sum_{k=0}^{N} b_k \cdot x[n-k] - \sum_{k=1}^{N} a_k y[n-k]\} - a_2 \cdot y[n-1] \tag{3.3}$$

Then expanding it :

$$\begin{aligned} y[n+1] = {} & b_0 \cdot x[n+1] + b_1 \cdot x[n] + b_2 \cdot x[n-1] - a_2 \cdot y[n-1] + \\ & - a_1 \cdot (b_0 \cdot x[n] + b_1 \cdot x[n-1] + b_2 \cdot x[n-2] - a_1 \cdot y[n-1] - a_2 \cdot y[n-2]) \end{aligned} \tag{3.4}$$

Collecting the terms referred to the same sample we obtain:

$$\begin{aligned} y[n+1] = {} & b_0 \cdot x[n+1] + (b_1 - a_1 \cdot b_0) \cdot x[n] + (b_2 - a_1 \cdot b_1) \cdot x[n-1] - (a_1 \cdot b_2) \cdot x[n-2] + \\ & + (a_1^2 - a_2) \cdot y[n-1] + a_1 \cdot a_2 \cdot y[n-2]) \end{aligned} \tag{3.5}$$

Then :

$$\begin{aligned} y[n] = {} & b_0 \cdot x[n] + (b_1 - a_1 \cdot b_0) \cdot x[n-1] + (b_2 - a_1 \cdot b_1) \cdot x[n-2] - (a_1 \cdot b_2) \cdot x[n-3] + \\ & + (a_1^2 - a_2) \cdot y[n-2] + a_1 \cdot a_2 \cdot y[n-3]) \end{aligned} \tag{3.6}$$

Finally we substitute the coefficients as follow:

- ${b_0}^{LK} = b_0$;
- ${b_1}^{LK} = (b_1 - a_1 \cdot b_0)$;
- ${b_2}^{LK} = (b_2 - a_1 \cdot b_1)$;
- ${b_3}^{LK} = -(a_1 \cdot b_2)$;
- ${a_2}^{LK} = -(a_1^2 - a_2)$;
- ${a_3}^{LK} = -(a_1 \cdot a_2)$;

Thus we obtain:

$$y[n] = {b_0}^{LK} \cdot x[n] + {b_1}^{LK} \cdot x[n-1] + {b_2}^{LK} \cdot x[n-2] + {b_3}^{LK} \cdot x[n-3] - ({a_2}^{LK} \cdot y[n-2] + {a_3}^{LK} \cdot y[n-3]) \tag{3.7}$$

3.1.2 C model and THD:

C model

The modifications introduced are:

- First of all new coefficients need to be computed as explained above. To avoid the quantization error as much as possible, we choose to compute them by using the real values of the original filter, then we quantize them on 14 bits.
- Then, instead of having the vector that mimics the registers of two elements only, we increase it to have three elements since here we need to process three delayed versions of the original sample x[n-1], x[n-2], and x[n-3].
- Finally, we changed the number of cycles to three to compute the contributions related to the three delayed versions of the signal, and we used an additional coefficient ${a_1}^{LK}$ that we made equal to 0 to make the contribution coming from it null, since, as described by the equation 3.7, the output y[n] is computed without any contribution from y[n-1].

Comments on the results and THD

The results obtained by this fixed point architecture are different from the original fixed point one, and this is mainly due to the fact that the new coefficients of the filter obtained after the J-look-ahead improvement cannot be computed exactly, in fact they need additional bits of precision, that translate in bigger multipliers and consequently bigger area for the overall circuit.

The solution adopted is to truncate them to the nearest value, this introduces additional distortions to the filtered signal, in fact, if we maintain the same truncation strategy at the output of the multipliers, explained in section 1.2, and we measure the THD of the new filtered signal, we obtain -35.08 dB, that is bigger by almost 2 dB with respect to the original version that was of -37.24 dB, so we can expect that the distortion is higher in this case but it is still below the specification limit that is -30 dB, so we can accept it.

3.1.3 Architecture:

The architecture depicted in Figure 3.1 was derived using the equation 3.7. We can see that considering the original version of the filter, we have changed the coefficients and added: one Multiplier, one Adder, and one Register.

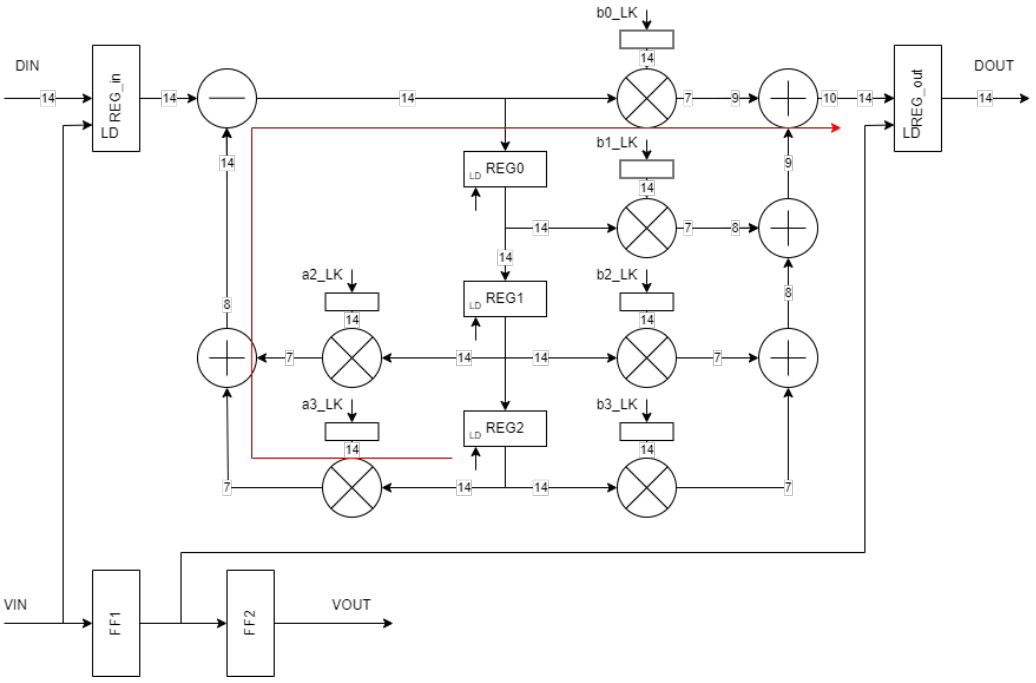

Figure 3.1: Schematic of the IIR filter: only J-look-ahead applied. The red arrow indicates the critical path.

3.1.4 VHDL model:

The VHDL used the same components described in section 2.2, we only need to instantiate the new components, as described by the block scheme in Figure 3.1, in the top entity file.
Finally, we need to change the coefficients with the new ones in the data maker component of the test bench.

3.1.5 Simulation:

In this section it has been performed the simulation of the architecture of the filter with the J-look-ahed technique applied. Since the processes done to obtain the results are basically the same as the ones done in section 2.3, only the main results and explanations are reported, remembering that all the useful information about the command files location in the different folders and the relative results are reported in the **READ_ME.txt** file.

As before, firstly it has been performed a simulation for testing the architecture at a clock period of 100 ns. As expected, the two files **results_c.txt** and **results_hdl.txt** have coincident results. Moreover the simulation time has been calculated from the obtained waveform and it is 38.8 us as the previous case. In figure 3.2 it is reported the associated waveform.

3.1.6 Synthesis:

Following the same procedure as before in section 2.4, firstly it has been found the maximum clock frequency. The results for the slack and the area are reported respectively in figure 3.3 and figure 3.4.

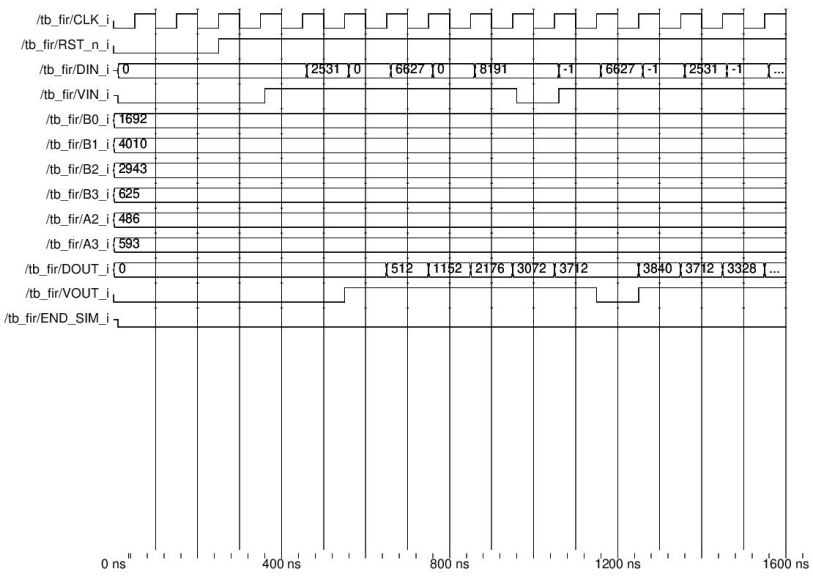

Figure 3.2: Waveform for $T_{clk} = 100\ ns$

```
clock MY_CLK (rise edge)                                  4.00       4.00
clock network delay (ideal)                               0.00       4.00
clock uncertainty                                        -0.07       3.93
REG_out/Q_reg[8]/CK (DFFR_X1)                             0.00       3.93 r
library setup time                                       -0.04       3.89
data required time                                                   3.89
--------------------------------------------------------------------------
data required time                                                   3.89
data arrival time                                                   -3.89
--------------------------------------------------------------------------
slack (MET)                                                          0.00
```

Figure 3.3: Timing report at $f_M = 250\ MHz$

```
****************************************
Report : area
Design : myIIR2
Version: S-2021.06-SP4
Date   : Fri Nov 18 20:13:59 2022
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /eda/dk/nangate45/synopsys/NangateOpenCellLibrary_typical_ecsm.db)

Number of ports:                         1304
Number of nets:                          5253
Number of cells:                         3688
Number of combinational cells:           3485
Number of sequential cells:               156
Number of macros/black boxes:               0
Number of buf/inv:                        553
Number of references:                      24

Combinational area:               5380.382007
Buf/Inv area:                      394.212001
Noncombinational area:             832.580027
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  6212.962034
Total area:                 undefined
1
```

Figure 3.4: Area report at $f_M = 250\ MHz$

Subsequently, the same results are reported, regarding the $f_M/4 = 62.5\ MHz$ frequency. The results given by the report are highlighted in figure 3.5 and figure 3.6.

```
clock MY_CLK (rise edge)                                  16.00      16.00
clock network delay (ideal)                                0.00      16.00
clock uncertainty                                         -0.07      15.93
REG_out/Q_reg[6]/CK (DFFR_X1)                              0.00      15.93 r
library setup time                                        -0.04      15.89
data required time                                                   15.89
---------------------------------------------------------------------------
data required time                                                   15.89
data arrival time                                                    -8.75
---------------------------------------------------------------------------
slack (MET)                                                           7.13
```

Figure 3.5: Timing report at $f_M = 62.5\ MHz$

```
****************************************
Report : area
Design : myIIR2
Version: S-2021.06-SP4
Date   : Fri Nov 18 20:24:33 2022
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /eda/dk/nangate45/synopsys/NangateOpenCellLibrary_typical_ecsm.db)

Number of ports:                         1304
Number of nets:                          4516
Number of cells:                         2836
Number of combinational cells:           2634
Number of sequential cells:               156
Number of macros/black boxes:               0
Number of buf/inv:                        384
Number of references:                      24

Combinational area:              4799.172025
Buf/Inv area:                     218.918001
Noncombinational area:            829.920027
Macro/Black Box area:               0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                 5629.092051
Total area:                 undefined
1
```

Figure 3.6: Area report at $f_M = 62.5\ MHz$

After that, the simulation of the netlist at quarter frequency has been performed. As the previous case, it has been added three files: **clk_gen_fm4.vhd** for setting the new period of the clock that is $T_{CLK} = 16\ ns$, **data_sink_fm4.vhd** to output the filtered samples in a new file named **results_hdl_fm4.txt**, and **tb_fir_fm4.v** to include in the design the 2 previous new files. Also this time the results of C-model and HDL one are the same, and the simulation time is of 6.21 us. It follows the waveform for the simulation at quarter frequency in an interval from 0 ns to 260 ns (figure 3.7).

Regarding the power estimation all the previously explained steps at section 2.4 have been followed, obtaining in the end a power consumption of 390.67 uW as stated in figure 3.8 . In order to see all the command files and printed reports, please refer to the already quoted **READ_ME.txt** file in the main folder of the section.

All the most important previous results are summarized in the table 3.1

| $f_M/4 = 62.5MHz$ | $A = 5629.09um^2$ | $P = 390.67uW$ | $T = 6.21us$ |

Table 3.1: Most important results of the synthesis with SYNOPSYS where A is the area, P is the power consumption and T is the simulation time.

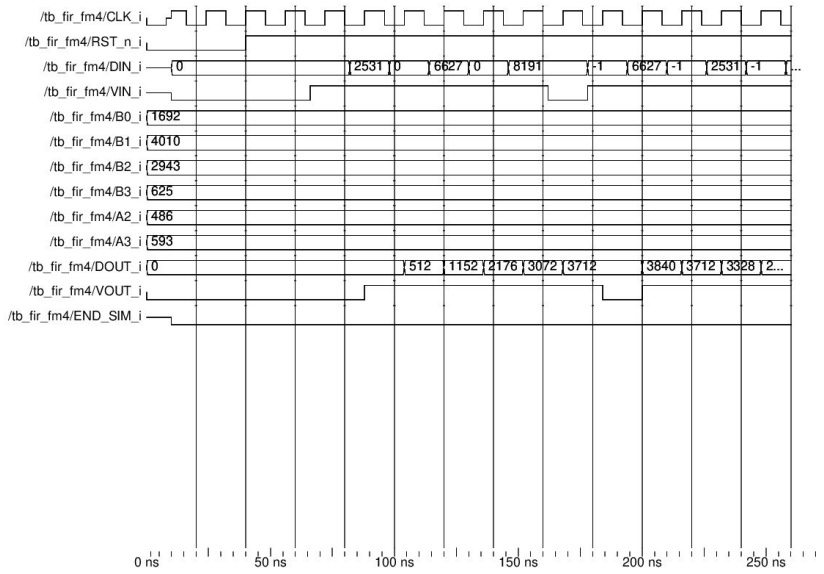

Figure 3.7: Waveform at $f_M = 62.5\ MHz$

```
Design : myIIR2
Version: S-2021.06-SP4
Date   : Fri Nov 18 21:16:16 2022
****************************************


Library(s) Used:

    NangateOpenCellLibrary (File: /eda/dk/nangate45/synopsys/NangateOpenCellLibrary_typical_ecsm_nowlm.db)


Operating Conditions: typical   Library: NangateOpenCellLibrary
Wire Load Model Mode: top

Design        Wire Load Model            Library
------------------------------------------------
myIIR2                5K_hvratio_1_1    NangateOpenCellLibrary


Global Operating Voltage = 1.1
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ns
    Dynamic Power Units = 1uW    (derived from V,C,T units)
    Leakage Power Units = 1nW


  Cell Internal Power  = 172.8767 uW   (63%)
  Net Switching Power  = 102.7778 uW   (37%)
                         ---------
Total Dynamic Power    = 275.6545 uW  (100%)

Cell Leakage Power     = 115.0133 uW


                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000            0.0000            0.0000            0.0000  (   0.00%)
memory             0.0000            0.0000            0.0000            0.0000  (   0.00%)
black_box          0.0000            0.0000            0.0000            0.0000  (   0.00%)
clock_network      0.0000            0.0000            0.0000            0.0000  (   0.00%)
register          66.2866           14.7152        1.3772e+04           94.7743  (  24.26%)
sequential         0.0000            0.0000            0.0000            0.0000  (   0.00%)
combinational    106.5902           88.0626        1.0124e+05          295.8934  (  75.74%)
--------------------------------------------------------------------------------------------------
Total            172.8768 uW       102.7778 uW     1.1501e+05 nW       390.6677 uW
```

Figure 3.8: Power report from Synopsys

3.1.7 Place and Route: J-look-ahead IIR Filter

Also for the Place and Route the same procedure has been followed: one initial script for obtaining the design at routed level, a simulation of the new netlist and a final command script to perform the power estimation.

The area of the circuit has been estimated with the help of the Gate Count function of Innovus and can be visualized in figure 3.9.

Then in the following picture is presented the resulting waveform given by the simulation of the netlist

```
Gate area 0.7980 um^2
Level 0 Module myIIR2                                Gates=      6974 Cells=    2725 Area=    5565.5 um^2
```

Figure 3.9: Area report with Innovus

obtained with Innovus (figure 3.10). As expected, looking at the text files **results_hdl_fm4_innovus.txt** and **results_c.txt** it can be verified that this is a match of the results and it has been also calculated the simulation time that is 6.21 us as in the case of the simulation of the netlist with Synopsys.

Then, after the computation of the .vcd file by mean of Questasim, the power estimation has

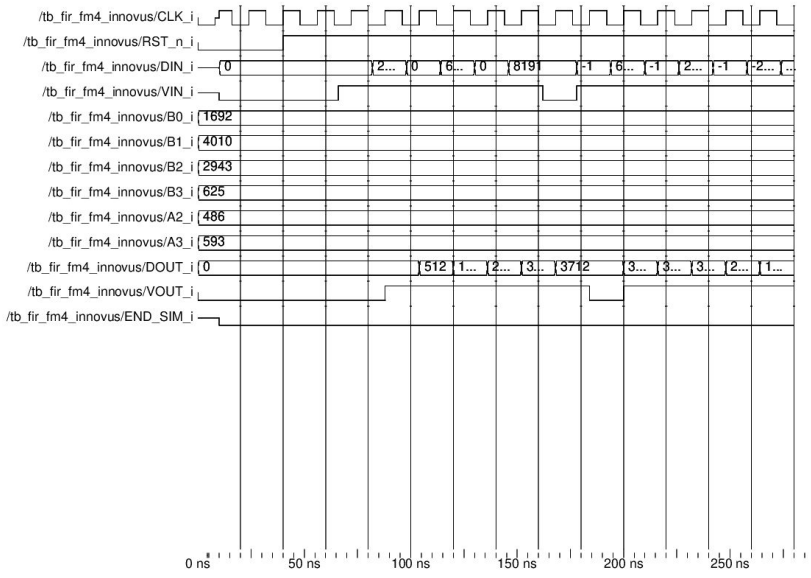

Figure 3.10: Waveform after Place and Route design at quarter frequency

been performed and it has given the following results at figure 3.11.

As previously done for Synopsys, in the end it has been reported a table with the summary of all the interested measurements done with Innovus (table 3.2).

FINAL CONSIDERATION: note that the results obtained with the J-look ahead of this section and the results obtained in the section where the filter has been analyzed without techniques applied are very close. Moreover, it can be seen that the maximum frequency is lower in the case of the J-look ahead (250 MHz versus 278 MHz). This result highlights the fact that the J-look ahead is just a way

```
*
*   Power Units = 1mW
*
*   Time Units = 1e-09 secs
*
*       report_power -outfile power_report_new -sort total
*
-------------------------------------------------------------------------------


Total Power
-------------------------------------------------------------------------------
Total Internal Power:        0.17811794        43.6950%
Total Switching Power:       0.11629590        28.5291%
Total Leakage Power:         0.11322574        27.7759%
Total Power:                 0.40763957
-------------------------------------------------------------------------------


Group                          Internal   Switching     Leakage       Total  Percentage
                               Power      Power         Power         Power  (%)
-------------------------------------------------------------------------------------
Sequential                      0.07056     0.01638     0.01377      0.1007       24.71
Macro                                 0           0           0           0           0
IO                                    0           0           0           0           0
Combinational                    0.1056     0.08383     0.09936      0.2888       70.84
Clock (Combinational)          0.001995     0.01609   9.169e-05     0.01817       4.458
Clock (Sequential)                    0           0           0           0           0
-------------------------------------------------------------------------------------
Total                            0.1781      0.1163      0.1132      0.4076         100
-------------------------------------------------------------------------------
```

Figure 3.11: Power report of the netlist after Place and Route at quarter frequency

$$\mid f_M/4 = 62.5MHz \mid A = 5565.5um^2 \mid P = 407.6uW \mid T = 6.21us \mid$$

Table 3.2: Most important results of the synthesis with INNOVUS where A is the area, P is the power consumption and T is the simulation time.

to modify the architecture of the circuit in order to apply other techniques that are useful in order to improve the performance such as retiming and pipelining, that will be applied in the next section.

3.2 IIR Filter with the J-look-ahead and the universal techniques improvements

3.2.1 Theoretical derivation:

The J-look ahead technique doesn't introduce any improvement in terms of speed or area, rather in most cases it lowers the speed of the circuit and increases the occupied area, as demonstrated in the previous section, because it introduces new combinational elements that worsen the critical path, even though in our case the critical path remains unchanged, as can be seen by comparing Figure 2.1 and 3.1 Instead, it gives us more margin to apply the universal methods such as pipelining and retiming.

We applied retiming and pipelining to our architecture in the following way:

- Through retiming the 3 registers in the middle arch are split and moved after the multiplication and addition operations both in the feedback and feed-forward sections (the three registers in the middle are now three registers for each side: feedback and feedforward);
- The circuit can be now cut in the middle with a pipeline register, since this cut forms a feed forward cut-set;
- Since there is no branch including a multiplication with a a_1 coefficient, among the three registers located in the feedback section, two are in series with no other element in-between: one of them can be moved back in the middle arch through retiming, before the multiplications;

- As last step, instead of having 2 registers in the middle, one that is a functional register and the other the pipeline register, the pipeline register can function as a substitute of the functional register, that is thus dropped, leaving intact the proper function of the circuit.

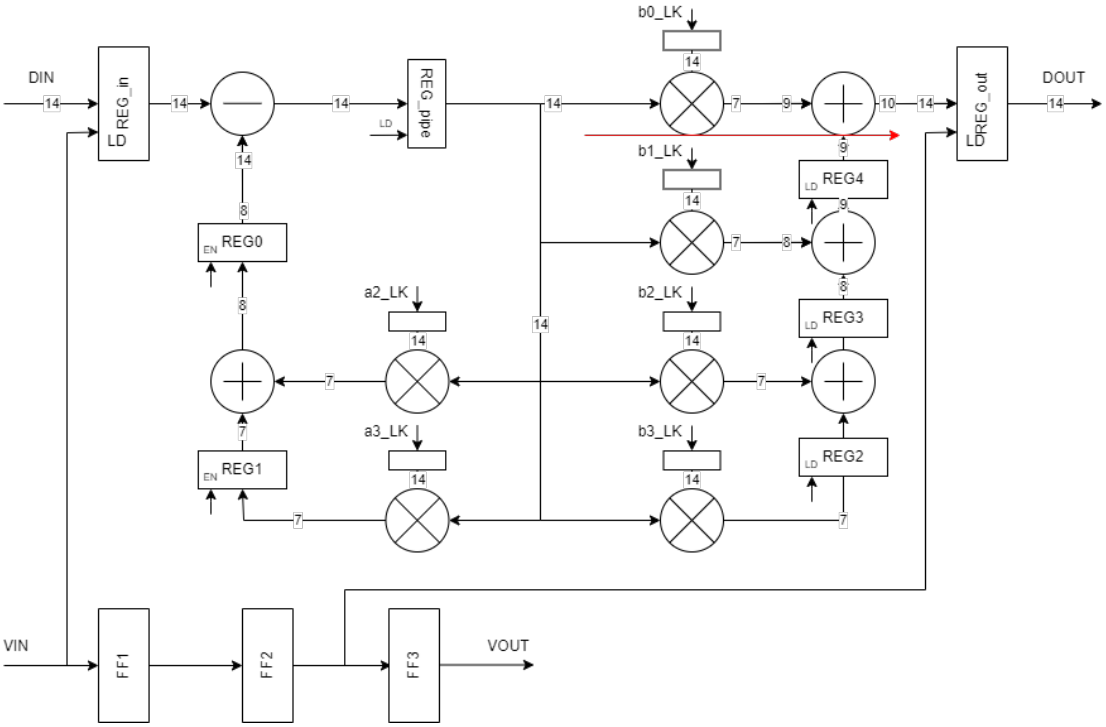

Figure 3.12: Schematic of the IIR filter as implemented in VHDL. J-look-ahead and universal techniques applied. The red arrow indicates the critical path.

With this choice we are able to cut with registers the critical path, which in the only J-look-ahead scheme (Figure 3.1) is the one passing through 2 multipliers and 3 adders in the feedback path (indicated by the red arrow), so that with the new architecture there is no path where the signal has to pass through more than one multiplier and one adder, before encountering a register. In this way the critical path is more than cut in half, resulting in a higher expected frequency with respect to the non-pipelined version. This has been achieved with a cost in area only equal to 3 more registers and one flip flop.

Notice also that the delay of the filter is now equal to 3 clock cycles, so also the generation of the valid out (VOUT) signal has been changed.

Finally, there is no need to change the C model to adapt it to this version since the results will be the same as the original J-look-ahead version, as for the VHDL code the top entity needs to be changed to be compliant with the new architecture derivation, shown in figure 3.12.

3.2.2 Simulation: J-look-ahead and pipelined IIR Filter

In this section it has been performed the simulation of the architecture of the filter with the J-look-ahed technique applied in the previous section and also the pipeline and the retiming. Since the processes are always the same, only the main results and explanations are reported, remembering that all the

useful information about the command files location in the different folders and the relative results are reported in the **READ_ME.txt** file.

Firstly it has been performed a simulation for testing the architecture at a clock period of 100 ns. As expected, the two files **results_c.txt** and **results_hdl.txt** have coincident results. Moreover the simulation time has been calculated from the obtained waveform and it is 38.8 us as the previous case. In figure 3.13 it is reported the associated waveform.

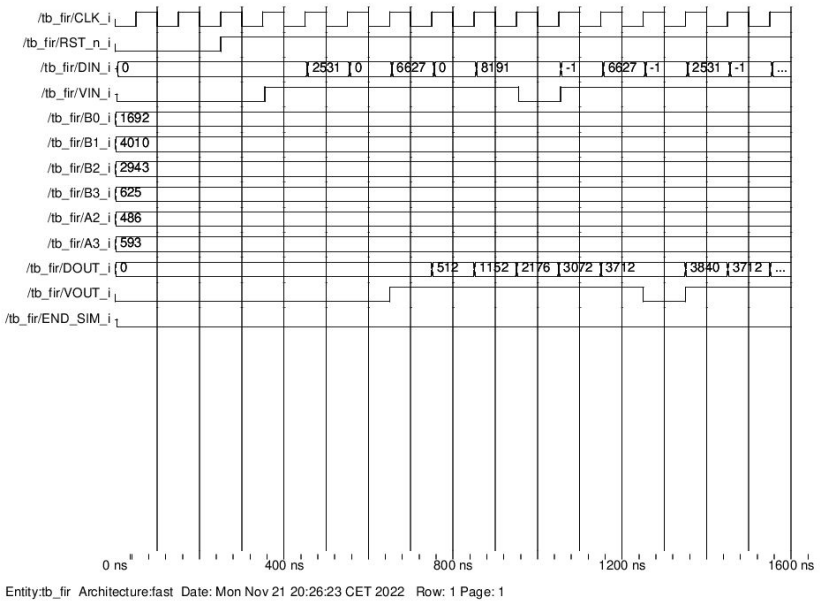

Figure 3.13: Waveform for $T_{clk} = 100\ ns$

3.2.3 Synthesis: J-look-ahead and pipelined IIR Filter

Following the same procedure as before, firstly it has been found the maximum clock frequency that is 357.14 MHz. The results for the slack and the area are reported respectively in figure 3.14 and figure 3.15.

```
clock MY_CLK (rise edge)                                 2.80       2.80
clock network delay (ideal)                              0.00       2.80
clock uncertainty                                       -0.07       2.73
REG_out/Q_reg[8]/CK (DFFR_X1)                            0.00       2.73 r
library setup time                                      -0.04       2.69
data required time                                                  2.69
--------------------------------------------------------------------------
data required time                                                  2.69
data arrival time                                                  -2.69
--------------------------------------------------------------------------
slack (MET)                                                         0.00
```

Figure 3.14: Timing report at $f_M = 357.14\ MHz$

Subsequently, the same results are reported, regarding the $f_M/4 = 90.9\ MHz$ frequency. The results given by the report are highlighted in figure 3.16 and figure 3.17.

IMPORTANT NOTE: like in the first case of the filter architecture where no technique was

```
****************************************
Report : area
Design : myIIR2
Version: S-2021.06-SP4
Date   : Sat Nov 19 18:19:15 2022
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /eda/dk/nangate45/synopsys/NangateOpenCellLibrary_typical_ecsm.db)

Number of ports:                         1401
Number of nets:                          5441
Number of cells:                         3707
Number of combinational cells:           3458
Number of sequential cells:               199
Number of macros/black boxes:               0
Number of buf/inv:                        577
Number of references:                      35

Combinational area:               5394.480010
Buf/Inv area:                      386.231999
Noncombinational area:            1064.000035
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  6458.480044
Total area:                 undefined
1
```

Figure 3.15: Area report at $f_M = 357.14\ MHz$

```
clock MY_CLK (rise edge)                                 11.00      11.00
clock network delay (ideal)                               0.00      11.00
clock uncertainty                                        -0.07      10.93
REG_out/Q_reg[7]/CK (DFFR_X1)                             0.00      10.93 r
library setup time                                       -0.04      10.89
data required time                                                  10.89
---------------------------------------------------------------------------
data required time                                                  10.89
data arrival time                                                   -4.00
---------------------------------------------------------------------------
slack (MET)                                                          6.89
```

Figure 3.16: Timing report at $f_M = 90.9\ MHz$

```
****************************************
Report : area
Design : myIIR2
Version: S-2021.06-SP4
Date   : Sat Nov 19 18:30:11 2022
****************************************

Library(s) Used:

    NangateOpenCellLibrary (File: /eda/dk/nangate45/synopsys/NangateOpenCellLibrary_typical_ecsm.db)

Number of ports:                         1401
Number of nets:                          4801
Number of cells:                         3011
Number of combinational cells:           2762
Number of sequential cells:               199
Number of macros/black boxes:               0
Number of buf/inv:                        428
Number of references:                      30

Combinational area:               4918.074024
Buf/Inv area:                      260.680000
Noncombinational area:            1058.680034
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)

Total cell area:                  5976.754058
Total area:                 undefined
1
```

Figure 3.17: Area report at $f_M = 90.9\ MHz$

applied, also in this one, the clock period at quarter frequency should have been different: 11.2 ns instead of 11 ns. The reason of this choice regards the resolution as deeply explained in section 2.4. After that, the simulation of the netlist at quarter frequency has been performed. As the previ-

ous case, it has been added three files: **clk_gen_fm4.vhd** for setting the new period of the clock that is $T_{CLK} = 11\ ns$, **data_sink_fm4.vhd** to output the filtered samples in a new file named **results_hdl_fm4.txt**, and **tb_fir_fm4.v** to include in the design the 2 previous new files. Also this time the results of C-model and HDL one are the same, and the simulation time is of 3.88 us. It follows the waveform for the simulation at quarter frequency in an interval from 0 ns to 1600 ns (figure 3.13).

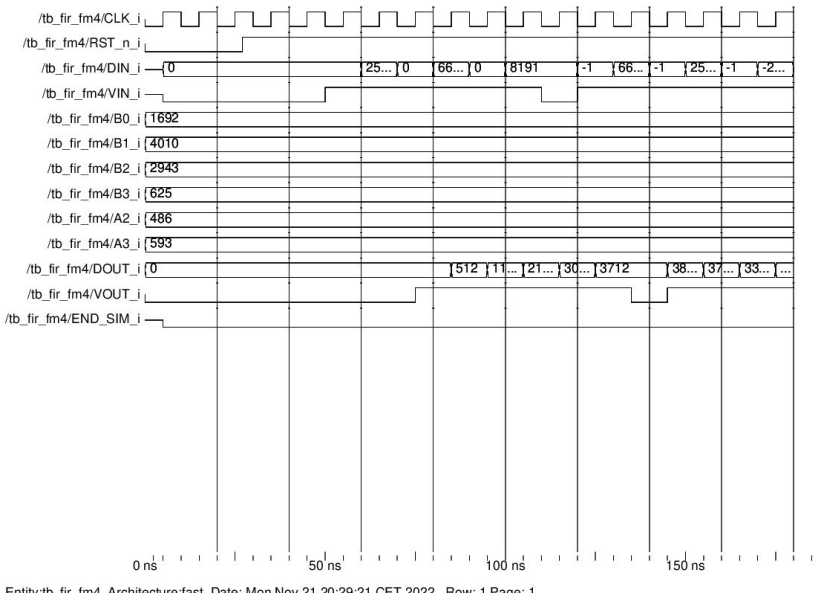

Figure 3.18: Waveform at $f_M = 90.9\ MHz$

IMPORTANT NOTE: in this case the tco defined in the **data_maker.vhd** file has been changed from 10 ns to 5 ns. This because it has been preferred to avoid having a clock period (11 ns) close to the combinatorial delay (10 ns) at which the incoming samples are given. As a matter of fact, the first time the tco has been leaved as it was but the results of **results_c.txt** and **results_hdl_fm4.vhd** were different from a certain sample. This because having a combinatorial delay close to the clock period made the filter to "jump" some samples that entered the filter with 1 clock cycle of delay.
Regarding the power estimation all the previously explained steps have been followed, obtaining in the end a power consumption of 613.79 uW as stated in figure 3.19 . In order to see all the command files and printed reports, please refer to the already quoted **READ_ME.txt** file in the main folder of the section.
All the most important previous results are summarized in the table 3.3

$$\left|\ f_M/4 = 90.9MHz\ \right|\ A = 5976.75um^2\ \left|\ P = 613.79uW\ \right|\ T = 3.88us\ \left|\right.$$

Table 3.3: Most important results of the synthesis with SYNOPSYS where A is the area, P is the power consumption and T is the simulation time.

3.2.4 Place and Route: J-look-ahead and pipelined IIR Filter

Also for the Place and Route the same procedure has been followed: one initial script for obtaining the design at routed level, a simulation of the new netlist and a final command script to perform the power estimation.

```
Design        Wire Load Model            Library
------------------------------------------------
myIIR2                5K_hvratio_1_1    NangateOpenCellLibrary


Global Operating Voltage = 1.1
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ns
    Dynamic Power Units = 1uW    (derived from V,C,T units)
    Leakage Power Units = 1nW


  Cell Internal Power  = 314.7069 uW   (64%)
  Net Switching Power  = 177.0444 uW   (36%)
                         ---------
Total Dynamic Power    = 491.7513 uW  (100%)

Cell Leakage Power     = 122.0416 uW


                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000            0.0000            0.0000            0.0000  (   0.00%)
memory             0.0000            0.0000            0.0000            0.0000  (   0.00%)
black_box          0.0000            0.0000            0.0000            0.0000  (   0.00%)
clock_network      0.0000            0.0000            0.0000            0.0000  (   0.00%)
register         134.4788            5.4921        1.7493e+04          157.4637  (  25.65%)
sequential         0.0000            0.0000            0.0000            0.0000  (   0.00%)
combinational    180.2284          171.5524        1.0455e+05          456.3292  (  74.35%)
--------------------------------------------------------------------------------------------------
Total            314.7072 uW       177.0445 uW     1.2204e+05 nW       613.7928 uW
```

Figure 3.19: Power report from Synopsys

| $f_M/4 = 90.9MHz$ | $A = 5714.7um^2$ | $P = 613.6uW$ | $T = 3.88us$ |

Table 3.4: Most important results of the synthesis with INNOVUS where A is the area, P is the power consumption and T is the simulation time.

The area of the circuit has been estimated with the help of the Gate Count function of Innovus and can be visualized in figure 3.20.

```
Gate area 0.7980 um^2
Level 0 Module myIIR2                                Gates=      7161 Cells=    2789 Area=    5714.7 um^2
```

Figure 3.20: Area report with Innovus

Then in the following picture is presented the resulting waveform given by the simulation of the netlist obtained with Innovus (figure 3.21). As expected, looking at the text files **results_hdl_fm4_innovus.txt** and **results_c.txt** it can be verified that this is a match of the results and it has been also calculated the simulation time that is 6.21 us as in the case of the simulation of the netlist with Synopsys.

Then, after the computation of the .vcd file by mean of Questasim, the power estimation has been performed and it has given the following results at figure 3.22.

As previously done for Synopsys, in the end it has been reported a table with the summary of all the interested measurements done with Innovus (table 3.4).

FINAL CONSIDERATION: differently from the case in which just the J-look ahead has been applied, now the performances are highly improved because of the application of both the pipelining and the retiming.

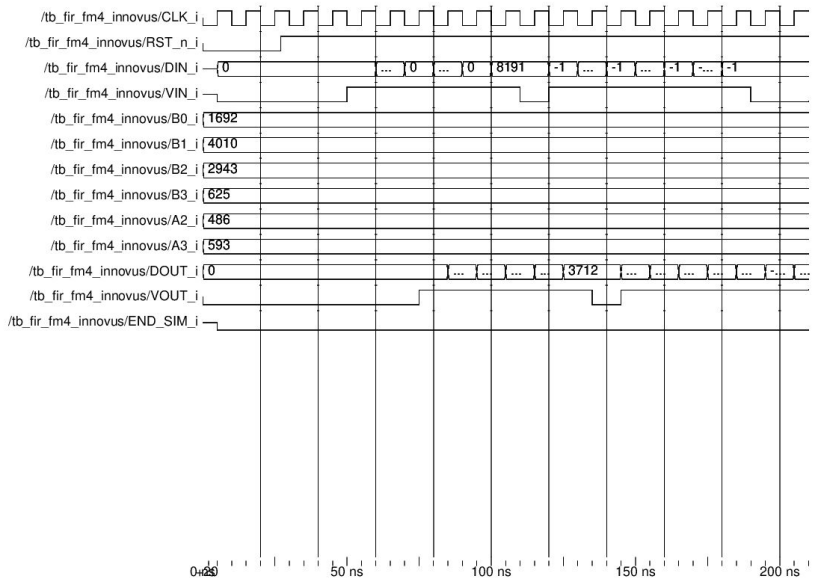

Figure 3.21: Waveform after Place and Route design at quarter frequency

```
*   Power Units = 1mW
*
*   Time Units = 1e-09 secs
*
*       report_power -outfile power_report_new -sort total
*
--------------------------------------------------------------------------------


Total Power
--------------------------------------------------------------------------------
Total Internal Power:        0.30417434        49.5721%
Total Switching Power:       0.19306389        31.4641%
Total Leakage Power:         0.11636171        18.9638%
Total Power:                 0.61359994
--------------------------------------------------------------------------------


Group                           Internal   Switching     Leakage       Total  Percentage
                                Power      Power         Power         Power  (%)
--------------------------------------------------------------------------------------
Sequential                        0.1257     0.00914     0.01534      0.1502       24.48
Macro                                  0           0           0           0           0
IO                                     0           0           0           0           0
Combinational                     0.1752      0.1548      0.1009      0.4309       70.23
Clock (Combinational)            0.003179     0.02916   9.167e-05     0.03243       5.285
Clock (Sequential)                     0           0           0           0           0
--------------------------------------------------------------------------------------
Total                             0.3042      0.1931      0.1164      0.6136         100
--------------------------------------------------------------------------------


Rail                  Voltage   Internal   Switching     Leakage       Total  Percentage
                                Power      Power         Power         Power  (%)
--------------------------------------------------------------------------------------
VDD                       1.1     0.3042      0.1931      0.1164      0.6136         100
```

Figure 3.22: Power report of the netlist after Place and Route at quarter frequency