Visible Image Watermarking

Digital Design and Logical Synthesis for Electric Computer Engineering

(36113611)

Course Project

Digital High-Level Design

Version 0.3

Revision Log

Rev	Description	Done By	Date
0.1	Initial document	Ariel Moshe, Amit Nagar Halevy	30/11/2020
0.2	Digital Changes Functional Verification	Ariel Moshe, Amit Nagar Halevy	05/12/2020
0.3	Synthesis	Ariel Moshe, Amit Nagar Halevy	31/12/2020

Table of Content

1. LIST OF FIGURES

2. LIST OF TABLES

3. INTRODUCTION

3.1 Design Constraints Requirements

Initial system constraints to use in the synthesis flow are as follows:

Constraint Type	Value
Clock Period	At least 100KHz
Clock Max External Latency	Maximum 1 ns
Clock Max Internal Latency	Maximum 1 ns
Clock uncertainty	Maximum 1 ns
Clock transition time	Maximum 1 ns
Input delay	Maximum 2.5 ns
Input transition	Maximum 1 ns
Output delay	Maximum 1 ns
Design area	Smaller than 50,000,000
Input driving cell	Smallest NOT gate
Output load	Load Pin of biggest DFF
Wire load model	tsmc18_wl50

Table 1: Design Constraints Requirements.

3.2 The Used Constraints

False Path			
From Pin/Port	Through Pin	To Pin/Port	Comments
reset	reset	All Registers	In out design the Reset is asynchronous.
Clock Definitions			
Constraint Name	CLK	Comment	
Period [Nano Seconds]	$10K$	$Duty\ Cycle - 50\%$	
Rising edge [Nano Seconds]	1		
Falling edge [Nano Seconds]	1		
Pin/Port name	clk		
Uncertainty	1		
Transition	1		
External Delay [Nano Seconds]			
Pin name	Value	Comment	
All pins	1		
Load [f Farads]			
Pin name	Value	Comment	
D	30.5	DFFSRHQX4 - output drives the largest DFF	
External Driver			
Pin name	Standard Cell Name	Cell Port name	Comment
A,Y	INV	INVXL	The input driver of all inputs is the smallest Inverter gate

Table 2: Used Constraints

4. REPORTS & RESULTS OF SYNTHESIS FLOW

4.1 Synthesis Flow

We change the RTL design according to the lab's requirement and run compilation.

Synthesis flow diagrams are follow below, and refers to all 4 compilations –

1. Without the SAIF input file and without gated clock.
2. With the SAIF input file and without gated clock.
3. Without the SAIF input file and with clock gating.
4. With the SAIF input file with clock gating

The diagrams not include the Control and Registers module, because it was too large to load.

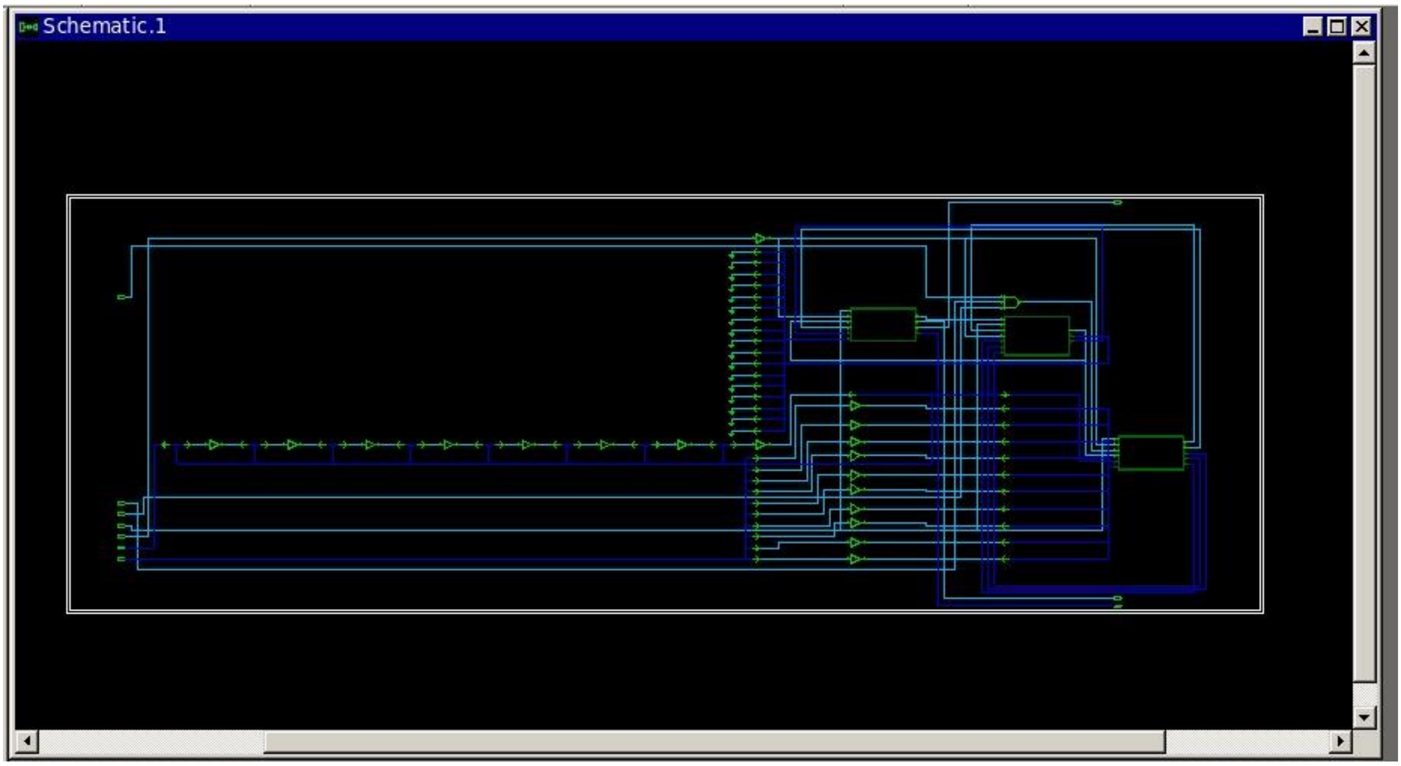
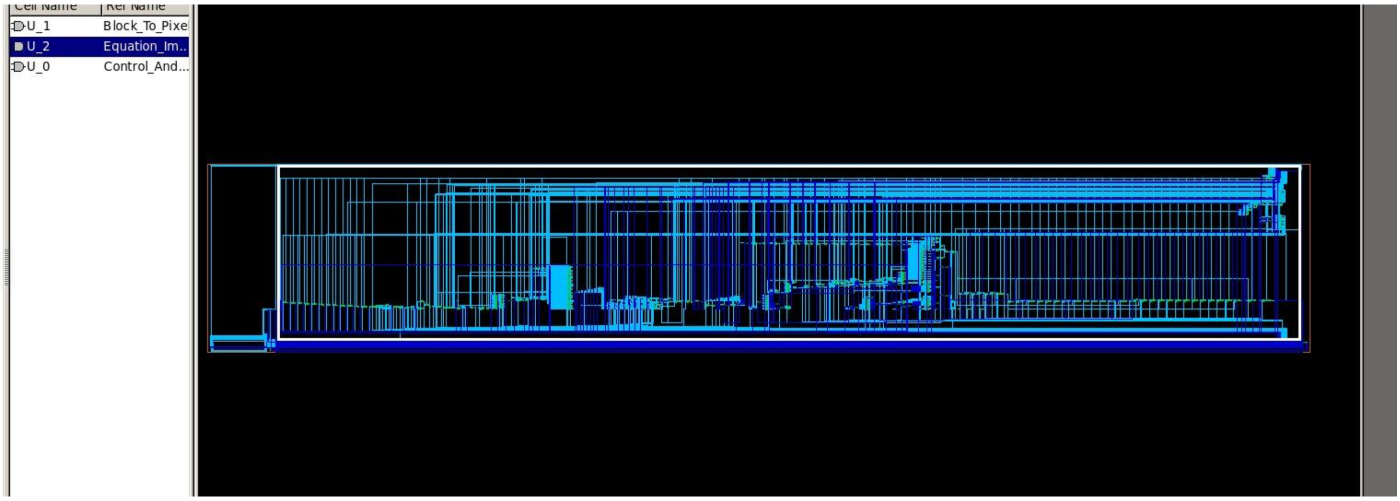


Figure 1: Visible Watermarking – Gate level – Regular Compilation



Cell Name	Ref Name
U_1	Block_To_Pixe
U_2	Equation_Im..
U_0	Control_And...

Figure 2: Equation Implementation – Gate level – Regular Compilation

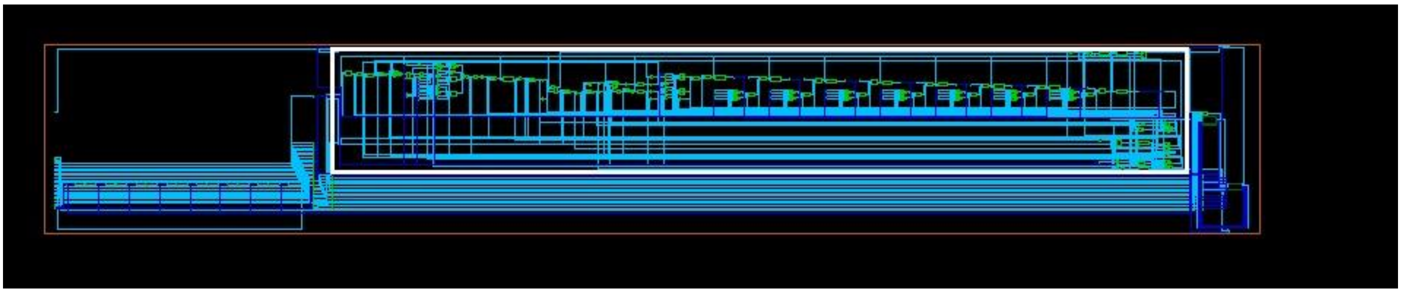
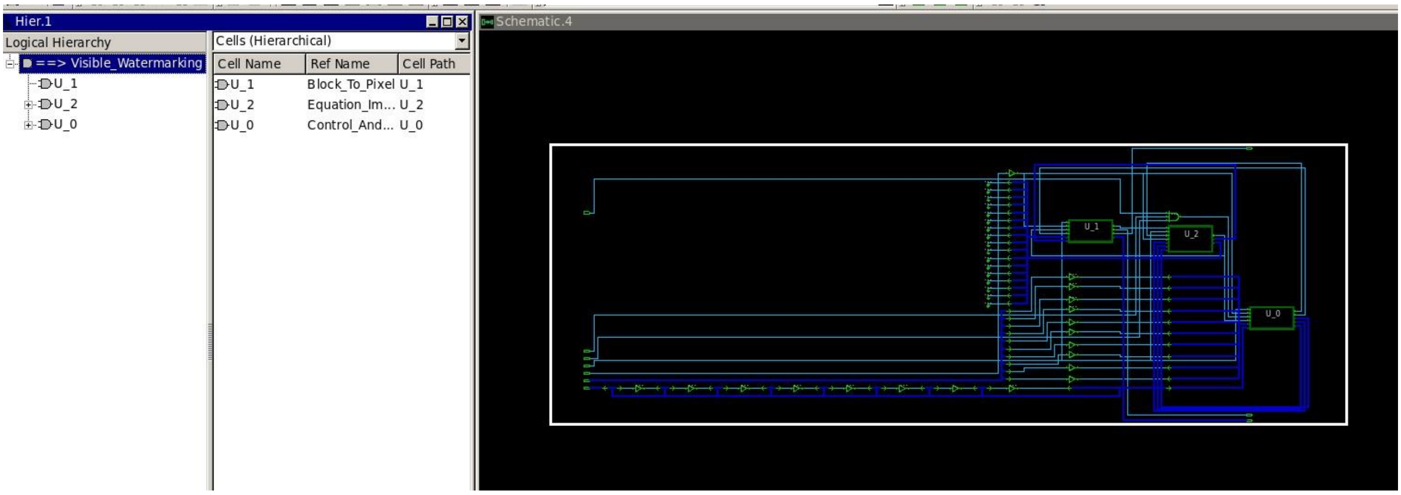


Figure 3: Block to Pixel – Gate level – Regular Compilation



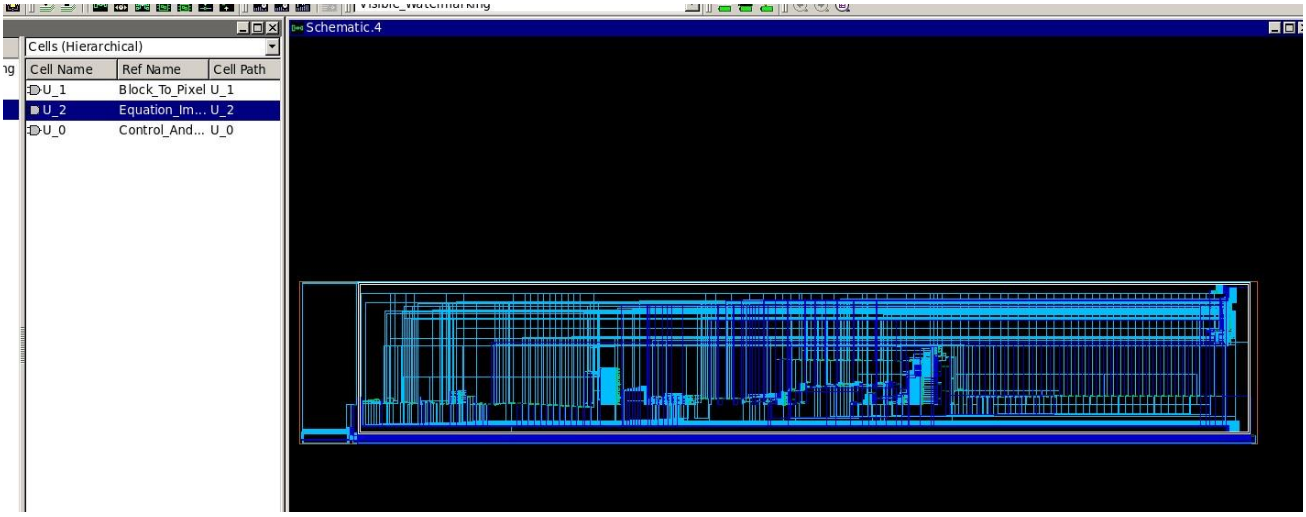
Logical Hierarchy

- ==> Visible_Watermarking
  - U_1
  - U_2
  - U_0

Cells (Hierarchical)

Cell Name	Ref Name	Cell Path
U_1	Block_To_Pixel	U_1
U_2	Equation_Im...	U_2
U_0	Control_And...	U_0

Figure 4: Visible Watermarking – Gate level – W/ SAIF

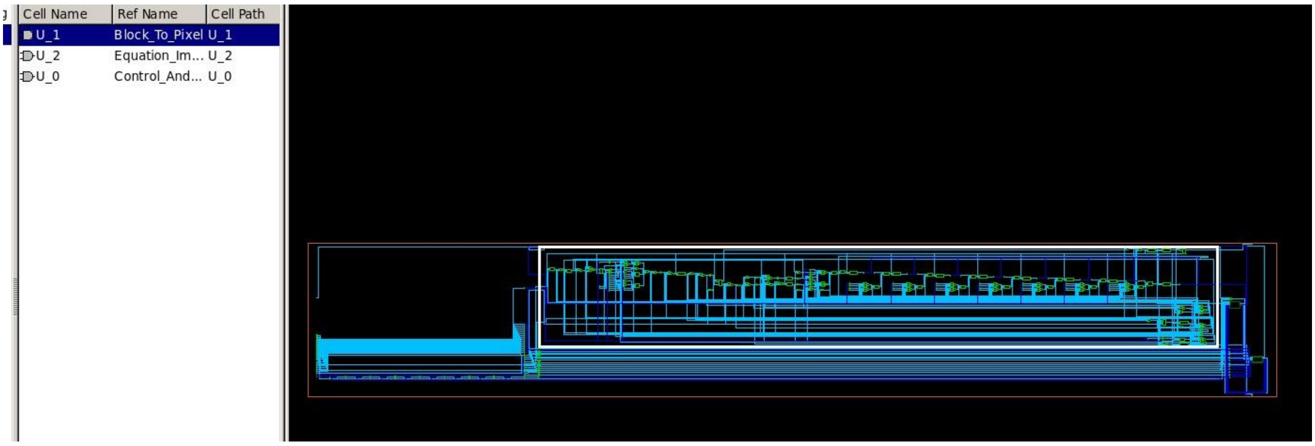


Schematic.4

Cells (Hierarchical)

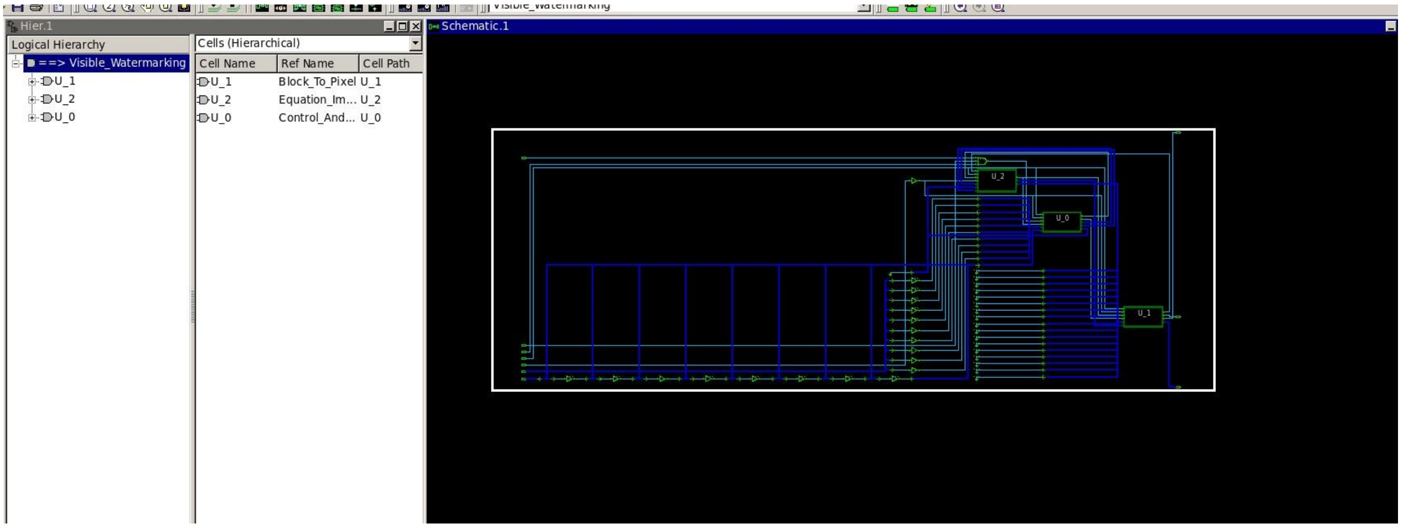
Cell Name	Ref Name	Cell Path
U_1	Block_To_Pixel	U_1
U_2	Equation_Im...	U_2
U_0	Control_And...	U_0

Figure 5: Equation Implementation – Gate level – W/ SAIF



Cell Name	Ref Name	Cell Path
U_1	Block_To_Pixel	U_1
U_2	Equation_Im...	U_2
U_0	Control_And...	U_0

Figure 6: Block to Pixel – Gate level – W/ SAIF



Hier.1

Logical Hierarchy

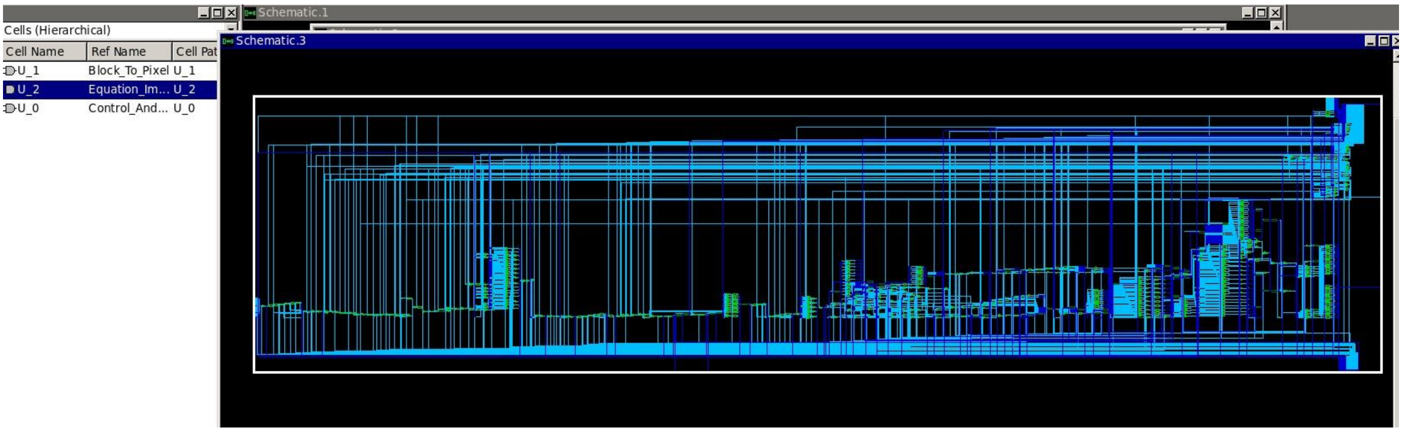
==> Visible_Watermarking
- U_1
- U_2
- U_0

Cells (Hierarchical)

Cell Name	Ref Name	Cell Path
U_1	Block_To_Pixel	U_1
U_2	Equation_Im...	U_2
U_0	Control_And...	U_0

Schematic.1

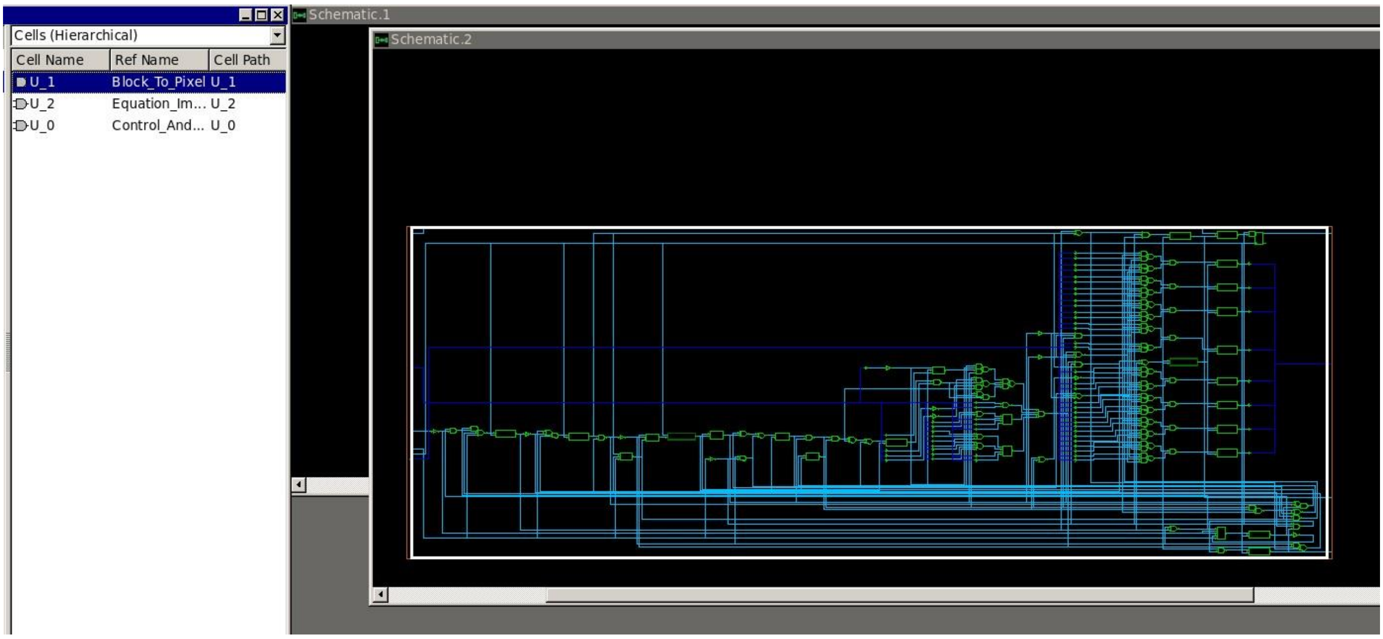
Figure 7: Visible Watermarking – Gate level – W/ CG



Cells (Hierarchical)

Cell Name	Ref Name	Cell Pat
U_1	Block_To_Pixel	U_1
U_2	Equation_Im...	U_2
U_0	Control_And...	U_0

Figure 8: Equation Implementation – Gate level – W/ CG



Cells (Hierarchical)

Cell Name	Ref Name	Cell Path
U_1	Block_To_Pixel	U_1
U_2	Equation_Im...	U_2
U_0	Control_And...	U_0

Figure 9: Block to Pixel – Gate level – W/ CG

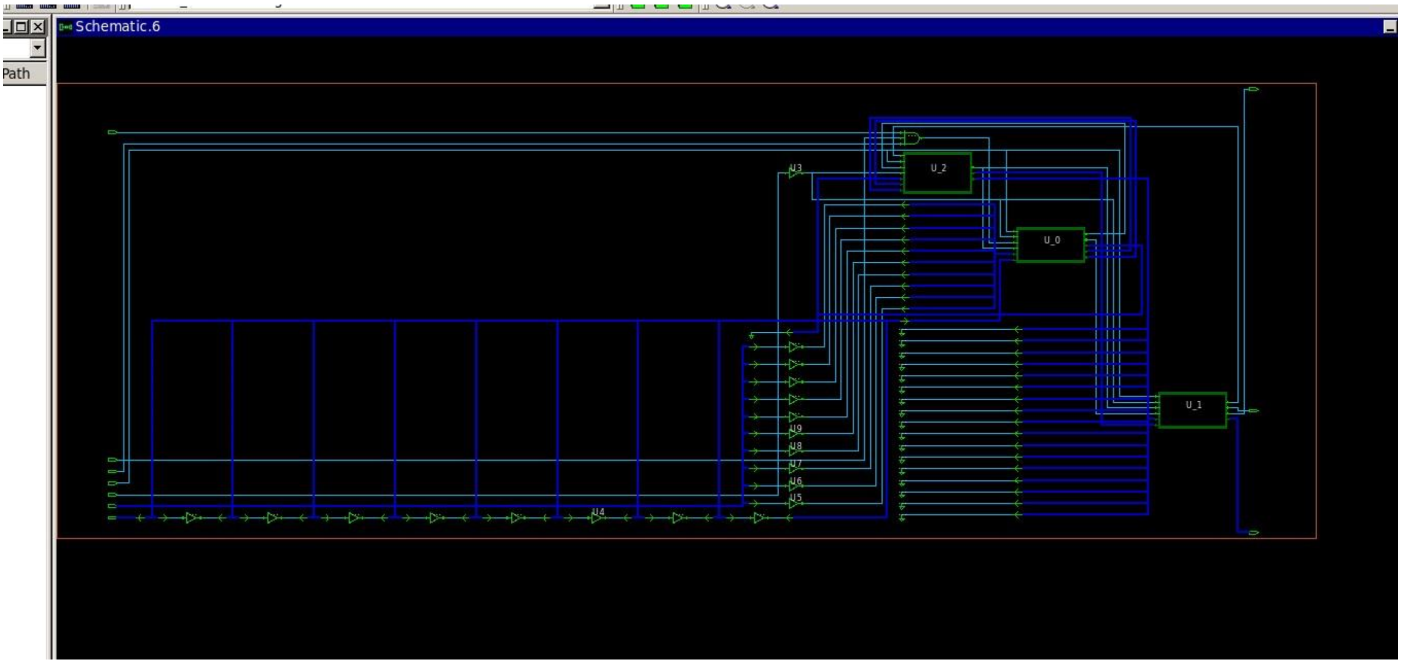
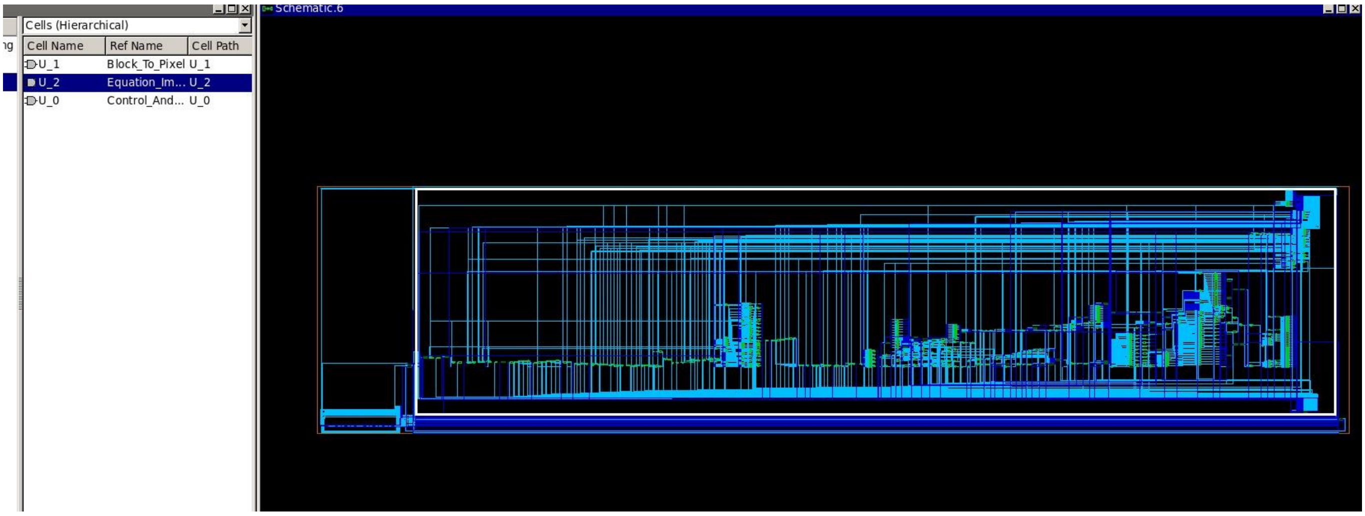


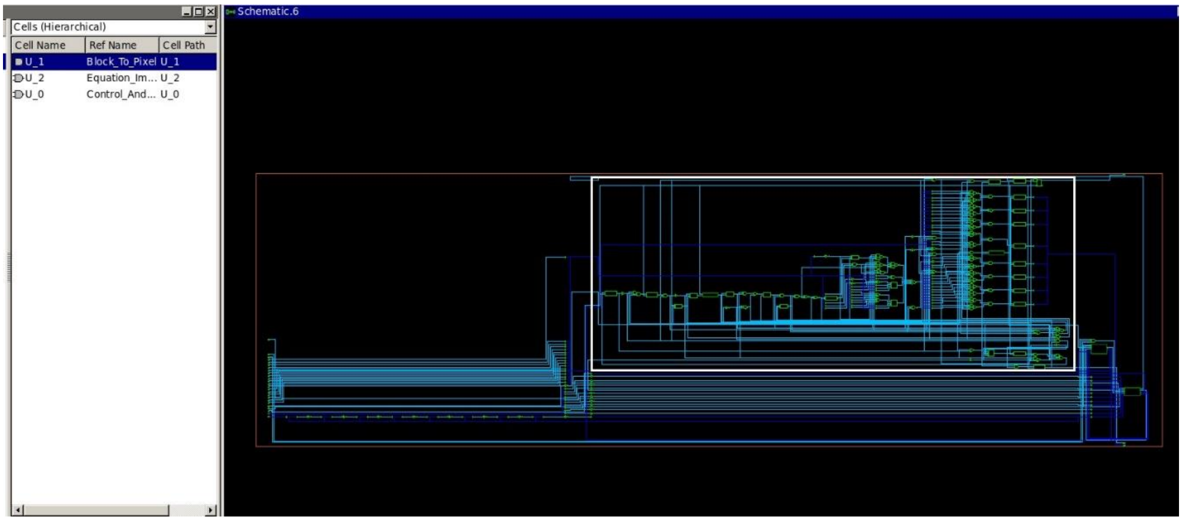
Figure 10: Visible Watermarking – Gate level – W/ SAIF & CG



Cells (Hierarchical)

Cell Name	Ref Name	Cell Path
U_1	Block_To_Pixel	U_1
U_2	Equation_Im...	U_2
U_0	Control_And...	U_0

Figure 11: Equation Implementation – Gate level – W/ SAIF & CG



Cell Name	Ref Name	Cell Path
U_1	Block_To_Pixel	U_1
U_2	Equation_Im...	U_2
U_0	Control_And...	U_0

Figure 12: Block to Pixel – Gate level – W/ SAIF & CG

4.2 Area Report

Design area report is follow below, and refers to all 4 compilations –

5. Without the SAIF input file and without gated clock.
6. With the SAIF input file and without gated clock.
7. Without the SAIF input file and with clock gating.
8. With the SAIF input file with clock gating

The report include a table of all post synthesis components and table of area sizes of different groups of component.

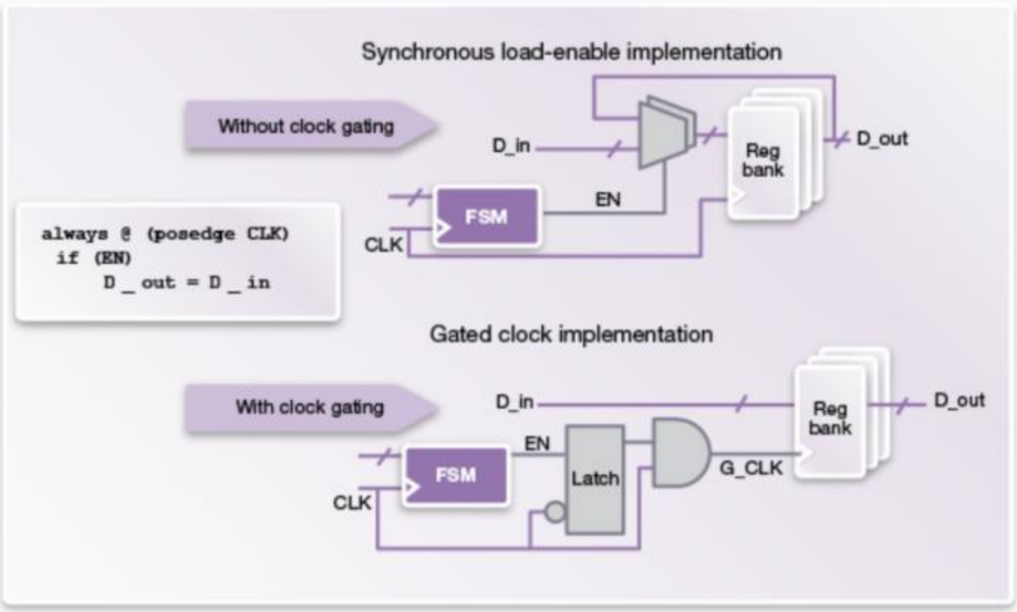
Ports	41
Nets	228
Cells	23
Combinational cells	20
Buf/Inv	19
References	5

Table 3: Number of Components in Synthesis

	Area					
Compilation	Combinational	Buf/Inv	Non-Combinational	Net Interconnect	Total Cell	**Total**
Regular	624,242	30,918	649,659	19,097,040	1,273,901	**20,370,941**
W/ SAIF	624,225	30,958	649,659	19,097,273	1,273,884	**20,371,158**
W/ CG	630,608	30,536	493,368	17,972,358	1,123,977	**19,096,335**
W/ SAIF & CG	630,608	30,536	493,368	17,972,358	1,123,977	**19,096,335**

Table 4: Design Area Report for all 4 Compilations

According to table 4, we can see that the gate level implementation needed only$\sim 20 \cdot 10^{6}[\mu m^{2}]$. In addition clock gating method of compilation significantly reduced the non-combinational and net interconnect areas as expected by reducing the need for mux logic at the input, as illustrated in figure 13.



```
always @ (posedge CLK)
  if (EN)
      D _ out = D _ in
```

Figure 13: Example of dynamic power optimization with clock gating

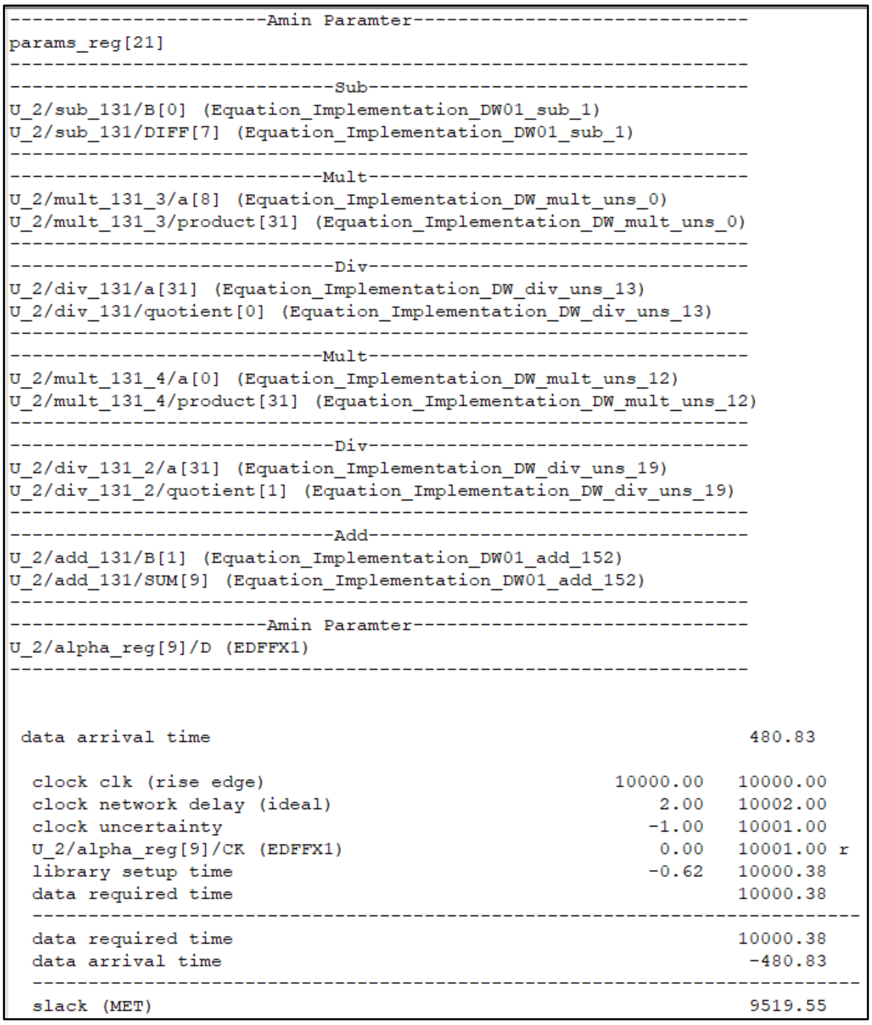
There are couple of ways to save area in addition to DC optimizations –

- Logic sharing – share modules like multiplier, adders and divider to avoid duplication of logic.
- Resource Sharing – using results of different logic modules for other logic operations.
- Multiplication by a constant – using shifters and adders instead of multiplier.

4.3 Timing Report

Short version of the timing report for setup time is follow below.

The first section describe the critical path and second show summary of the timing result – data arrival time and slack. The slack is met (positive) according to the report.

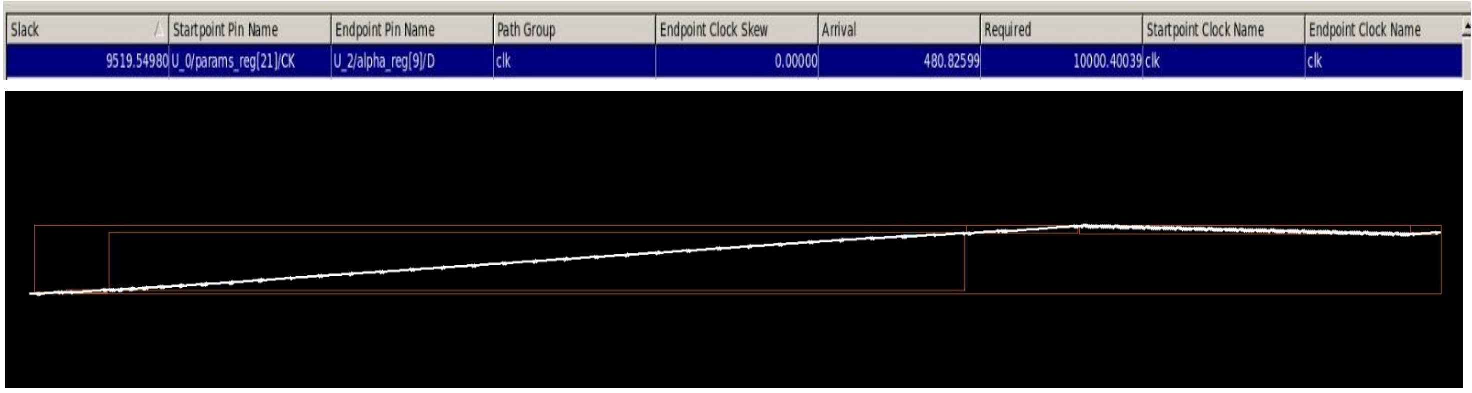


```
-----------------------Amin Paramter------------------------------
params_reg[21]
------------------------------------------------------------------
-----------------------------Sub----------------------------------
U_2/sub_131/B[0] (Equation_Implementation_DW01_sub_1)
U_2/sub_131/DIFF[7] (Equation_Implementation_DW01_sub_1)
------------------------------------------------------------------
----------------------------Mult----------------------------------
U_2/mult_131_3/a[8] (Equation_Implementation_DW_mult_uns_0)
U_2/mult_131_3/product[31] (Equation_Implementation_DW_mult_uns_0)
------------------------------------------------------------------
-----------------------------Div----------------------------------
U_2/div_131/a[31] (Equation_Implementation_DW_div_uns_13)
U_2/div_131/quotient[0] (Equation_Implementation_DW_div_uns_13)
------------------------------------------------------------------
----------------------------Mult----------------------------------
U_2/mult_131_4/a[0] (Equation_Implementation_DW_mult_uns_12)
U_2/mult_131_4/product[31] (Equation_Implementation_DW_mult_uns_12)
------------------------------------------------------------------
-----------------------------Div----------------------------------
U_2/div_131_2/a[31] (Equation_Implementation_DW_div_uns_19)
U_2/div_131_2/quotient[1] (Equation_Implementation_DW_div_uns_19)
------------------------------------------------------------------
-----------------------------Add----------------------------------
U_2/add_131/B[1] (Equation_Implementation_DW01_add_152)
U_2/add_131/SUM[9] (Equation_Implementation_DW01_add_152)
------------------------------------------------------------------
-----------------------Amin Paramter------------------------------
U_2/alpha_reg[9]/D (EDFFX1)
------------------------------------------------------------------


 data arrival time                                          480.83

  clock clk (rise edge)                          10000.00   10000.00
  clock network delay (ideal)                        2.00   10002.00
  clock uncertainty                                 -1.00   10001.00
  U_2/alpha_reg[9]/CK (EDFFX1)                       0.00   10001.00 r
  library setup time                                -0.62   10000.38
  data required time                                        10000.38
  ---------------------------------------------------------------------------
  data required time                                        10000.38
  data arrival time                                          -480.83
  ---------------------------------------------------------------------------
  slack (MET)                                                9519.55
```

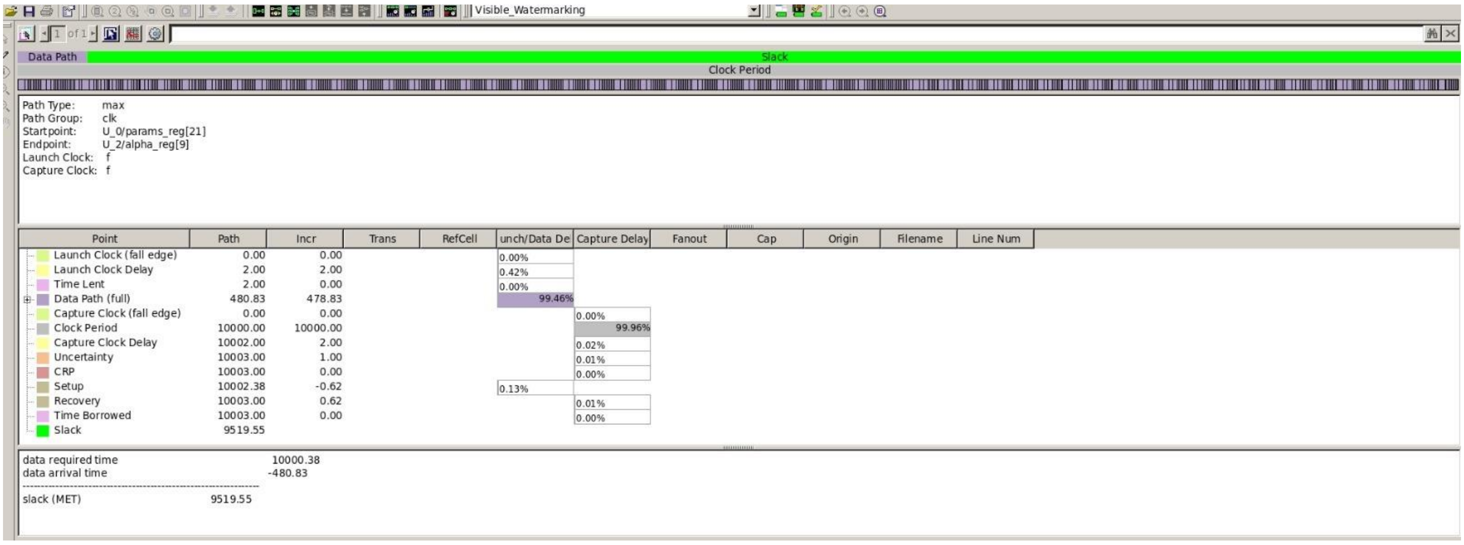
Figure 14: Timing report – short version



Slack	Startpoint Pin Name	Endpoint Pin Name	Path Group	Endpoint Clock Skew	Arrival	Required	Startpoint Clock Name	Endpoint Clock Name
9519.54980	U_0/params_reg[21]/CK	U_2/alpha_reg[9]/D	clk	0.00000	480.82599	10000.40039	clk	clk

Figure 15: Critical Path

Inspector



Path Type: max
Path Group: clk
Startpoint: U_0/params_reg[21]
Endpoint: U_2/alpha_reg[9]
Launch Clock: f
Capture Clock: f

Point	Path	Incr	Trans	RefCell	unch/Data De	Capture Delay	Fanout	Cap	Origin	Filename	Line Num
Launch Clock (fall edge)	0.00	0.00			0.00%						
Launch Clock Delay	2.00	2.00			0.42%						
Time Lent	2.00	0.00			0.00%						
Data Path (full)	480.83	478.83			99.46%						
Capture Clock (fall edge)	0.00	0.00				0.00%					
Clock Period	10000.00	10000.00				99.96%					
Capture Clock Delay	10002.00	2.00				0.02%					
Uncertainty	10003.00	1.00				0.01%					
CRP	10003.00	0.00				0.00%					
Setup	10002.38	-0.62			0.13%						
Recovery	10003.00	0.62				0.01%					
Time Borrowed	10003.00	0.00				0.00%					
Slack	9519.55										

```
data required time                    10000.38
data arrival time                     -480.83
------------------------------------------------
slack (MET)                  9519.55
```

Figure 16: Inspector Analysis

It can be seen that the delay results only from the data path.

Histogram

We chose to check the 20 slowest paths of the design, and the slack results demonstrated by histogram –

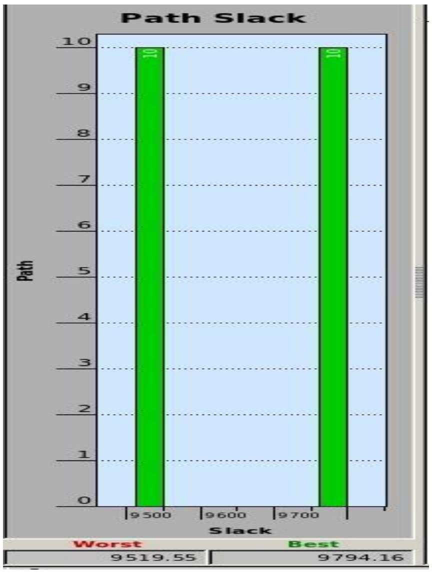


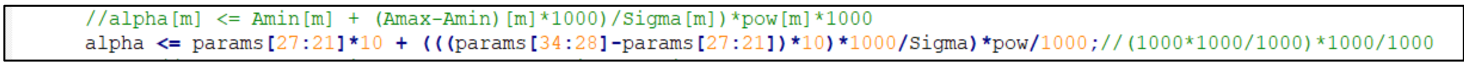
Figure 17: Histogram Analysis

According to the histogram, there are 10 path with the same worst slack.

RTL Code

The critical path source is from the 'Equation_Implementation' module while calculating eq.1 –

$$Eqn.(1): \alpha_k = \alpha_{\min} + \frac{(\alpha_{\max} - \alpha_{\min})}{\sigma_k} \cdot 2^{-(\mu_k - 0.5)^2}$$



```
    //alpha[m] <= Amin[m] + (Amax-Amin)[m]*1000)/Sigma[m])*pow[m]*1000
    alpha <= params[27:21]*10 + (((params[34:28]-params[27:21])*10)*1000/Sigma)*pow/1000;//(1000*1000/1000)*1000/1000
```

Figure 18: RTL Code - Critical Path

Changing the RTL code to preform multiplication by a constant with shifters and adders instead of multiplier, can increase time performance significantly.

4.4 Power Report

SAIF –

The Switching Activity Interchange format (SAIF) file contains toggle counts (number of changes) on the signals of the design. It also contains the timing attributes which specify time durations for signals at level 0, 1, X, or Z. The T0 is the duration of time in the logic 0 state, T1 is the time in logic 1, TX is the time in an unkown state, TC is the total number of transitions (rising and falling combined), and IG is the number of transition glitches during the monitoring. The SAIF file is recommended for power related tasks (i.e., power analysis or power driven implementation).

Clock Gating –

Clock gating is a technique used for reducing dynamic power dissipation, by removing the clock signal when the circuit is not in use. Clock gating saves power by pruning the clock tree, at the cost of adding more logic to a circuit. Pruning the clock disables portions of the circuitry so that the flip-flops in them do not have to switch states. Switching states consumes power. When not being switched, the switching power consumption goes to zero, and only leakage currents are incurred.

We Produced the Switching Activity Interchange Format (SAIF) file from the Questasim simulation that activated on a single image that has been watermarked.

Afterward, we used the SAIF file as input to the Design Compiler (DC) and Compile the design with 4 different variations as following –

1. Without the SAIF input file and without gated clock.
2. With the SAIF input file and without gated clock.
3. Without the SAIF input file and with clock gating.
4. With the SAIF input file with clock gating.

The following table summarize all 4 power reports of the different compilations –

Notes:

- $Cell\ Internal + Net\ Switching = Total\ Dynamic$
- $Total\ Dynamic + Leakage = Total$
- $\mathrm{P}_{Dynamic} = \alpha \cdot f \cdot C \cdot V_{DD}^{2}$
- α – Activity factor – number of signal transitions per cycle.

	Power				
Compilation	$Cell\ Internal$ $[\mu W]$	$Net\ Switching$ $[\mu W]$	$Total\ Dynamic$ $[\mu W]$	$Leakage$ $[\mu W]$	$\boldsymbol{Total}$ $[\boldsymbol{mW}]$
Regular	59.5	7.5	67	46.8	$\mathbf{0.1139}$
W/ SAIF	28,753	1,232	29,986	46.8	$\mathbf{30}$
W/ CG	9	8.5	17.5	48.7	$\mathbf{0.066}$
W/ SAIF & CG	3,025	1,241	4,267	48.7	$\mathbf{4.3159}$

Table 5: Power Report

From the power report, it can be seen that compiling with clock gating, with or without SAIF file, decrees the total dynamic power as expected, because its saves power by eliminating the unnecessary activity associated with reloading register banks and also eliminates the feedback net and multiplexer.

Power savings from clock gating is increased with the size of the logic following the clock-gated register. The greater the fan-out, the better, most effective when width of the register is at least 3. Moreover, Single clock gating cell replaces multiple MUXes, however add latches and other logic that consume more power.

In our design, we used a large register bank, hence it's more efficient to use clock gating compilation.

In addition, compiling with SAIF file in both cases drastically increase the dynamic power compare to without it. This result is unexpected and caused due to some problem with the transition from the HDL design to the gate level.

Compilation with the SAIF file supposed to decrees the dynamic power since it's annotates switching activity information on nets, pins, ports, and cells in the current design and the DC leverages the information in a SAIF file to perform various dynamic power optimizations while generating the netlist.

For example, an addition operation that can be implemented with full adders (FA) in 3 or 4 stages. The 3-stage adder structure may be good for timing, but not for power if one of the primary inputs to the adder structure is toggling enormously. The toggles are propagated downstream, increasing the overall dynamic power. If the DC finds that adders are in timing paths that have enough positive slack to implement the adder in 4 stages, then the alternative structure with the high activity input routed to the very last stage is considered, minimizing unnecessary toggles and reducing dynamic power.

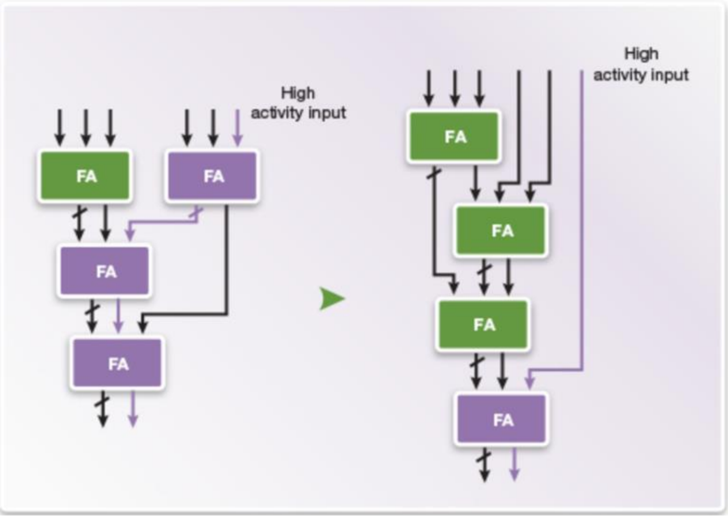
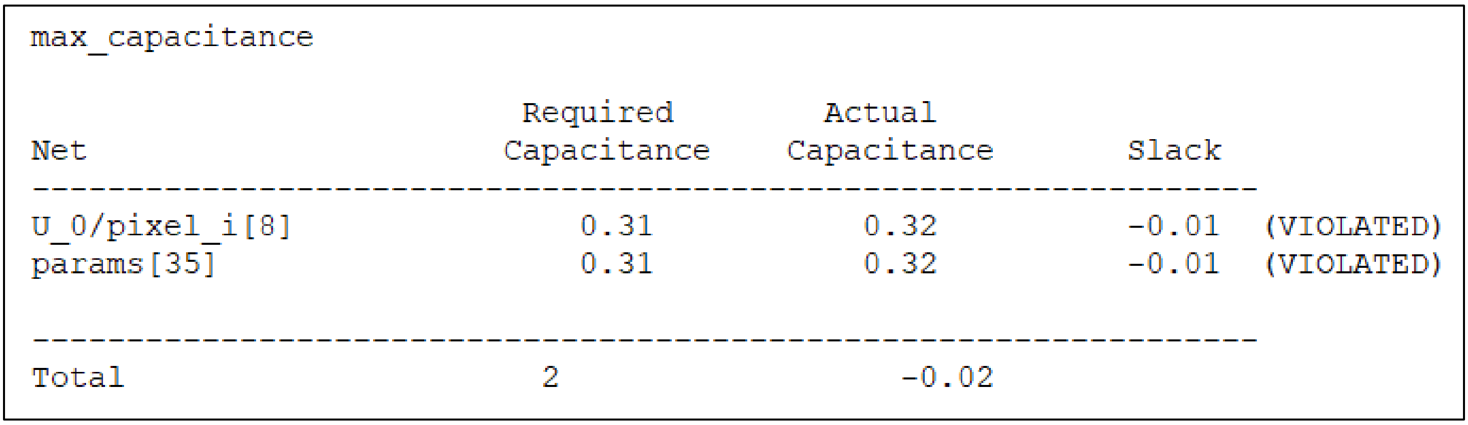


Figure 19: Example of dynamic power optimization using SAIF file

4.5 Constraints Violations

Design constraints report is follow below.



```
max_capacitance

                         Required       Actual
Net                     Capacitance    Capacitance       Slack
-----------------------------------------------------------------
U_0/pixel_i[8]              0.31           0.32          -0.01   (VIOLATED)
params[35]                  0.31           0.32          -0.01   (VIOLATED)

-----------------------------------------------------------------
Total                      2                 -0.02
```

Figure 20: Constraints Report

According to the report there are 2 violations. These violations refer to the input capacitance of the ports and seems that they are insignificant.

5. VERIFICATION RESULTS

In this section, we imported the synthesis outputs (top.v and top.sdf) into the HDL Designer project and run the simulation to show the results of the Gate Level implementation and updated RTL in the HDL-Designer Vs the Golden Model results.

Unfortunately, the Gate Level implementation simulation didn't work and we didn't find the problem while debuging, hence we will show only the simulation result of the updated RTL for the synthesis.

5.1 Golden Model Comparison

The following table shows the result of update RTL compared to the golden model.

# Image	N	M	Mean Squared Error (HDL-Designer)
1	18	2	0.52
2	18	1	0.49
3	14	1	0.48
4	17	1	0.46

Table 6: Mean Squared Errors

Like in Lab2, we chose to check the Mean Squared Error between the HDL-Designer and Gate Level Implementation to the golden model images.

$$MSE_1 = \frac{1}{\#Image\_pixels} \sum_{i=1}^{\#Image\_pixels} \left(pixel_{\text{HDL−Designer}} - pixel_{Golden\ Model}\right)^2$$

$$MSE_2 = \frac{1}{\#Image\_pixels} \sum_{i=1}^{\#Image\_pixels} \left(pixel_{\text{Gate Level Implementation}} - pixel_{Golden\ Model}\right)^2$$

The MSEs received are below 1, which means that the result is very similar to the golden model result.

Human eye test using MATLAB plot of output txt files:

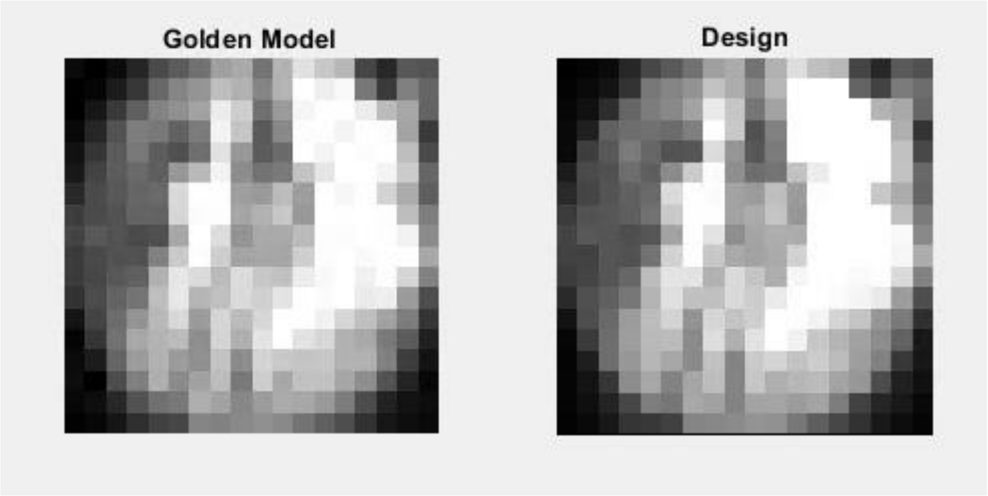


Figure 21: Watermarked Image 1 Comparison.

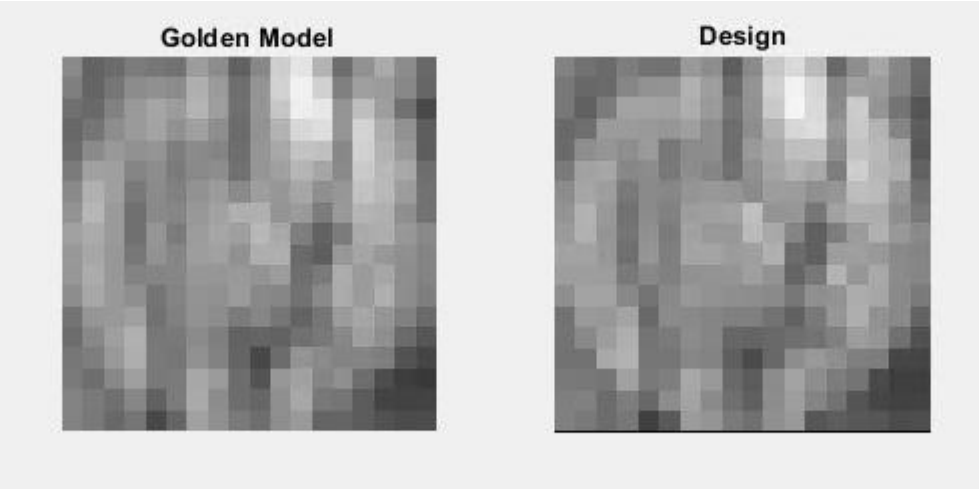


Figure 22: Watermarked Image 2 Comparison.

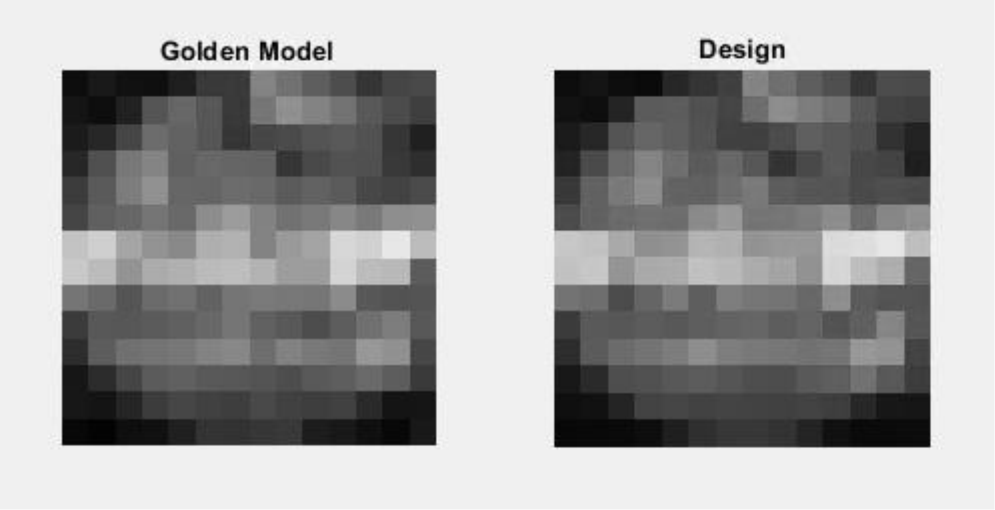


Figure 23: Watermarked Image 3 Comparison.

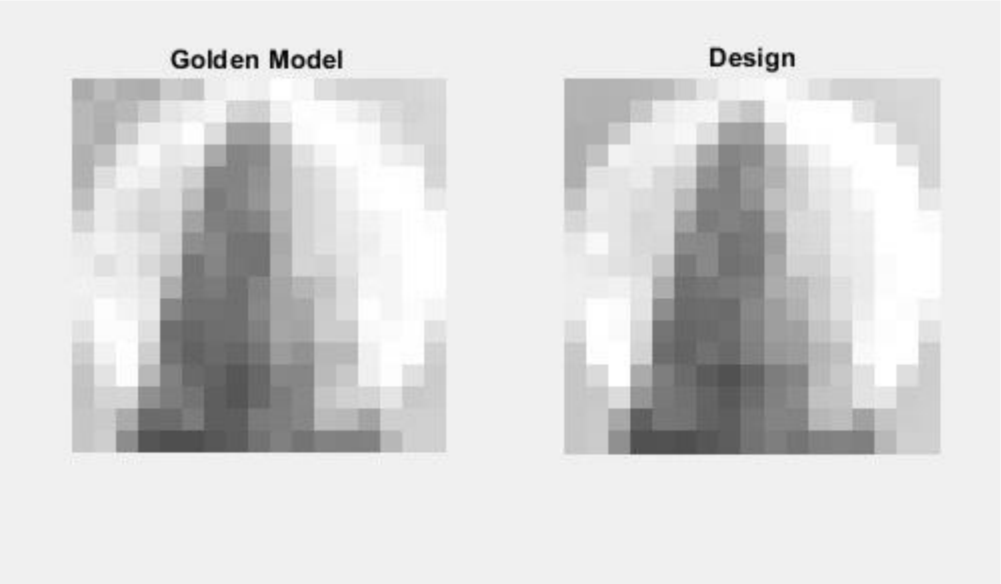


Figure 24: Watermarked Image 4 Comparison.

As one can see, the differences almost unnoticed for different images and parameters as seen in Lab 2, for the RTL design.

5.2 Conclusion

The main differences between the RTL code and the synthesis output file are that in the post synthesis clock inverters were added to keep the clock pulse precision all over the design. In addition, The DC used TSMC 0.18mm Process 1.8-Volt SAGE-XTM Standard Cell Library to implement sequential and combinatorial logic for the synthesis and generated extra signals for each cell from the library.

In terms of area – power – timing (APT) tradeoff, compilation with clock gating was area and power efficient compared to others. Timing report was similar for each of the compilations, furthermore the slack was very high, and therefore we could increase the clock frequency.

The following graph explain the relations between area – power – timing and the trade-off.

If a design has more area, A, available it should be able to perform a given computation in less time, T.

If a design use less power, P, it will perform a given computation in more time, T, and the frequency will be reduced by the cube root of the original (dynamic) power.

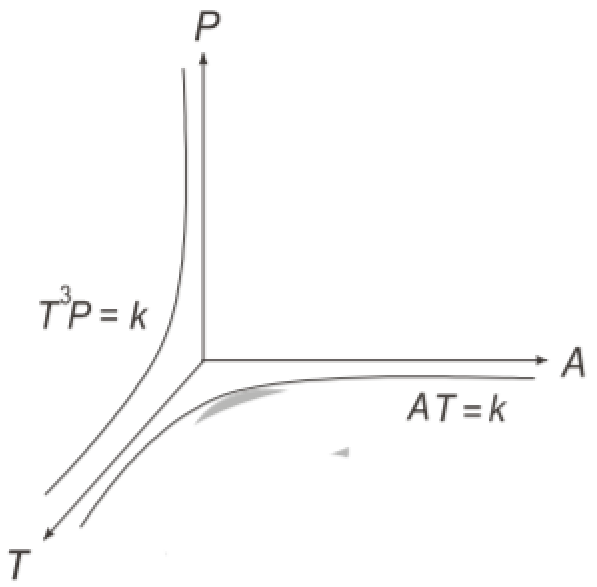


Figure 25: Area-Time-Power Trade-off