Design And Analysis Of Low Power Sequence Generator Module For DNA Fragment Assembly

N. A. A. Seman and A. K. Halim
Department of Electronic, Faculty of Electrical Engineering
University Teknologi MARA, 40450
Shah Alam, Selangor, Malaysia
Email: melyna.as@gmail.com

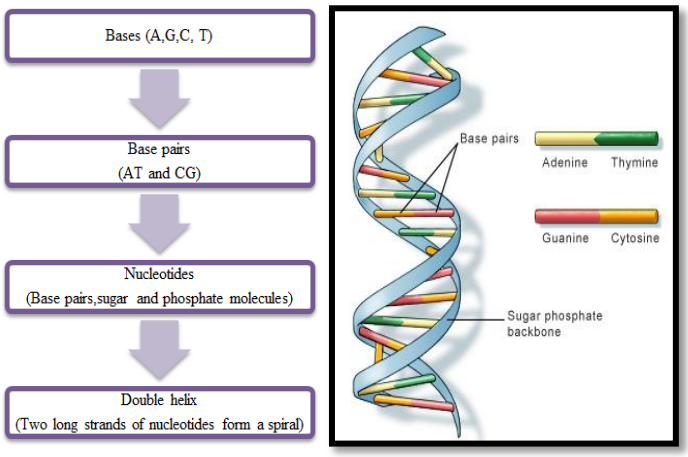
Abstract- **This paper presents the design and analysis of low power Sequence Generator Module (SGM) for DNA Fragment Assembly. The objectives of this project are to construct DNA Fragment Assembly module using Euler algorithm and optimize power consumption of the module. Another objective is to simulate and verify the module in FPGA and ASIC. Power become a primary consideration in design and develops process of a system. DNA fragment assembly system needs a low power consumption since a genome have a large scale of information waiting for decode. Low power techniques were implemented to determine the best approach of standard low power method. The SGM was analyzed using various constraints including clock gating technique to find the lowest power consumption in the module. The design and analysis process successfully done with Xilinx's software, Verilog Compiler Synopsys(VCS), Design Compiler(DC), Power Compiler(PC) and PrimeTime(PT). The major finding of this analysis is the combination of clock gating technique and power compiler constraints contributed the lowest power consumption in SGM by reducing 98% compared to the analysis of SGM without those techniques. A low power SGM had successfully been developed.**

Keywords- Low Power; Power Compiler; Euler Path; Clock Gating; Sequence Generator Module; DNA Fragment Assembly

I. INTRODUCTION

Every single thing in this life is specified by genome. Genome contains biological information that was needed to build and maintain a living thing, including humans. The biological information contained in a genome is encoded in DNA. DNA strand consists of millions of base pairs(bp) that are waiting to be assembled like pieces of puzzle. Different living things have different length and quantity of bases pair. Human DNA is about 3.2 billion of nucleotides in length[1].

DNA stands for deoxyribonucleic acid which is the basic material in humans and organisms. DNA can be found in the cell nucleus and the minority of DNA can be found in mitochrondria. DNA stored information in codes which made up by four chemical bases. The chemical bases are Adenine(A), Thymine(T), Cytosine(C) and Guanine(G). While combination of two bases were called as bases pair.The base A can pair with T while C can pair with G. Each base attach to a sugar and phosphate molecule. The base and both molecules combined and be called as nucleotide. In other words, each single of DNA molecule is a chain of nucleotides unit. Then, nucleotides are arranged in two long strands that form a spiral. The spiral is called double helix. Figure 1(a) shows the flow from DNA bases to double helix form. While Figure 1(b) shows the picture of DNA structure. Its structure looks like a ladder.



(a) (b)

Figure 1 About DNA (a) Flow process of DNA (b) DNA structure like a ladder

A genome must go through sequencing process in order to gather information about the genome. For example, two genomes from different species can be sequence to get the differences and similarity between both species. The bases A, G, C and T were used as DNA coding in sequencing process. Sequencing is a method to determine the exact order of nucleotides in the DNA molecules.

Almost a decade ago, scientist used shotgun method to sequence. Then, a method called as hybridization appeared. Both shotgun and hybridization are classic and traditional method. Shotgun method is a widely used technique to sequence genomes[2]. Shotgun can sequence up to 1000bp. Hybridization is the very first short-read sequencing approach. It is proposed in 1988 based on building microarray containing every possible nucleotide of length k. As the hybridization concept used each DNA fragments as a k-tuple, then, Idury and Waterman mimicked the concept by divide each DNA fragment length into n-k[2].

Recently, researchers were developed those method with many approaches and algorithms. A comparison of DNA fragment assembly also was conducted[3]. A technique that attempts to reconstruct the original sequence from a large number of fragments was called DNA fragment assembly [3]. A mathematical approach called as de Bruijn also was used to assemble k-mers generated by hybridization. In a de Bruijn graph, each edge is called k-mer. K-mer had been observed in the input data and implicitly represents a series of overlapping k-mers that overlap by a length of k-1 [4]. The de Bruijn graphs were first brought to bioinformatics in 1989. In contrast, in a thesis by D.R. Zerbino [5], the parameter *k* which called as k-mer can be a limitation in the analysis using de Bruijn graph framework since there is not many choice of *k* parameter.

The limitation of sequence method is the quantity of fragments that can be assembled. The fragments must be not more than 1000bp. That is why the algorithms were needed to break the genome into smaller parts called as fragments. Two decades ago, the classical approach to fragment assembly DNA sequences was called “overlap-layout-consesus”. The “overlap-layout-consensus” method approached the fragment assembly based on overlap graph which is can be describe as it will need to meet every single pieces of fragments in order to get match. The process using this classical approach took a long time. While, the de Bruijn method produced Eulerian path that will be used to compare which output is better and correct [5].

II. Euler Path Algorithm

Euler Algorithm introduced by Leonhard Euler in 1735[6]. There are two types of Euler. One is Euler path while another one is Euler circuit. An Euler path is a path that uses every edge of a graph exactly once. It starts and ends at di erent vertices. Then, an Euler circuit is a circuit that uses every edge of a graph exactly once. The Euler circuit starts and ends at the same vertex.

Figure 2 below shows the method that had been used in this SGM design to produce the output. This design used Euler path algorithm approach.

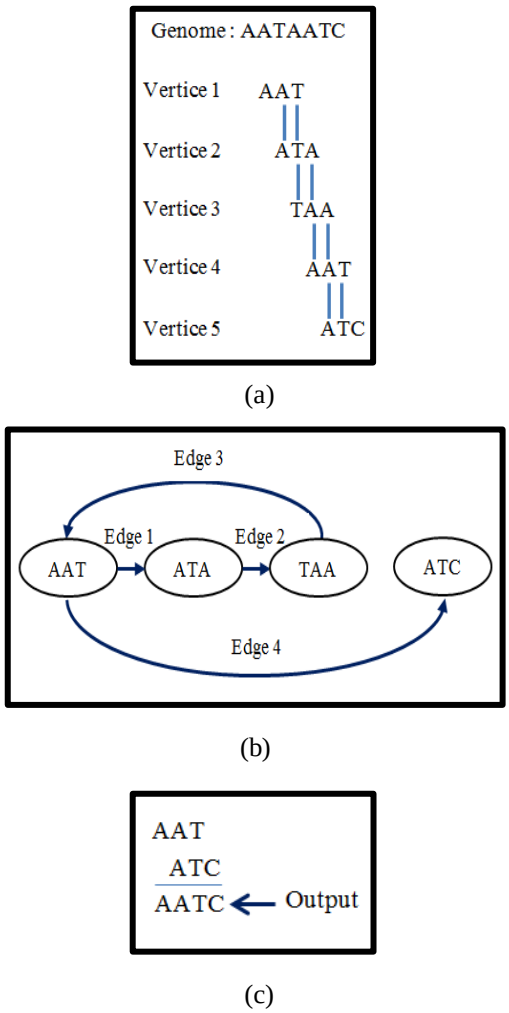


Figure 2 Sequence steps (a) Step 1 (b) Step 2 (c) Step 3

Figure 2(a) shows how a genome was divided to a smaller fragments. The fragments are called vertices. Coincidently, vertice 1 and vertice 4 has same codes. Then, step 2 in Figure 2(b) shows the path that represented the genome. The vertice became node and edges connected between the nodes based on the genome order. The path indicated as edges. The Euler path algorithm functions to find the shortest path to produce the genome output. In Figure 2(c) shows how overlap two vertices produce the output.

III. LOW POWER TECHNIQUES

Every design flow must have power analysis in order to determine the power consumption of the design. A power dissipation consists of dynamic and static power. The dynamic power contributes up to 50% of the total power dissipation[7]. The major contributors of dynamic power in synchronous design are clock frequency and switching activity. Clock switching process consumed approximately 50% of total power and is expected to increase in the next generations of design as the technology size become lower[8]. Static power is the power from transistor leakage current that flows whenever power is been applied to the device[9].

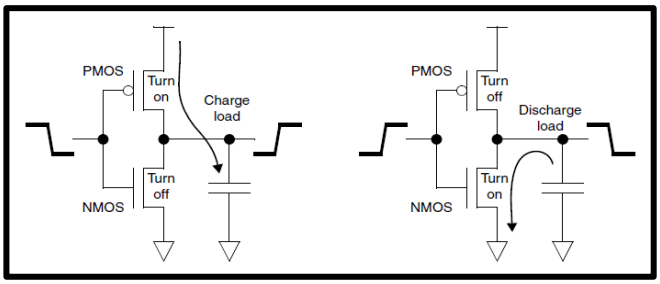


Figure 3 Dynamic power

Figure 3 shows dynamic power flow in a circuit. Each transition in a circuit contributed to energy dissipation. The energy dissipated depends on supply and capacitive load of the net. Since current flows only during logic transitions on the net, a long-term dynamic power dissipation depends on the clock frequency and switching activity[9]. Slower transitions also contributed to power consumption. In other word, power is directly proportional with voltage, frequency and capacitance. The equation of the power is $P=CV^2F$. Reducing the one of the power's components tend to reduce the power consumption. The dynamic power spent up to 50% in clock buffers since clock buffer have the highest toggle rate in the system [10]. This paper focused to reduce dynamic power consumption.

A. Clock Gating

The focus of the SGM analysis is at the major contributors of dynamic power consumption; switching activity and clock frequency. This analysis used variations of clock period in order to determine how others parameters were affected. This timing optimization process showed the best period for the SGM. The main parameters that been affected by clock period are timing slack, area and power consumption. The classic and popular method to reduce dynamic power consumption is clock gating. The clock gating method disabled clock when idle time or there is no activity at that time. So, the lower toggle rate on clock pin tends to reduce more internal power of registers. Since the enabled signal was gated, the clock network has less switching activity, thus consumes less switching power. Another benefit of clock gating is saves area as the clock gating technique eliminating multiplexers in the circuits.

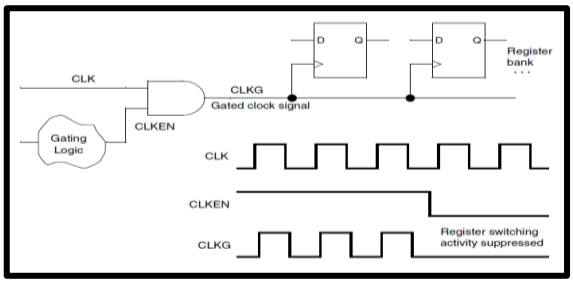


Figure 4 Clock gating method

The clock gating method can be visualized as in Figure 4 above. The input clock and enable enter AND gate produces gated clock signal. The clock transition always ON, when enable was inserted, the clock gated has transitions only when enable is in high condition. When enable is low, the transition in clock gated also low. Indirectly, this technique save power for switching activity as the system will be ON only when there is an activity.

B. Power Compiler(PC)

Power compiler was able to reduce the power consumption by 41%[11]. There are many functions and method can be used in the PC. For example, it can downsizing the gate and pin reordering. There also have standards low power technique and advanced low power technique such as power gating. This design scope is in the standard low power technique. Power compiler gave the infrastructure for power management. It is already have the library models for ASIC design.

The analysis provides visibility into power consumption whereas PC can calculate each cell in the design. The analysis can be done throughout the flow from RTL to GDSII process. The best of PC is the optimization performance that a design can get. The power consumption can be reduced at all levels of abstraction. Designers can always check and optimize it. PC can show any calculation of power that a designer's wants. The power models are switch power, internal power and leakage power. Switch power depends on output load and input transition. This switch

power can be control. However, both internal and leakage power cannot be control since there are characterization that was already be implemented by vendors.

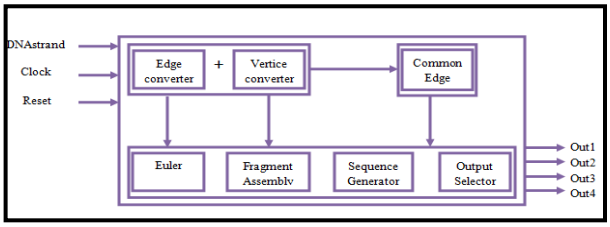
IV. METHODOLOGY

Design and analysis need a good planning and tools to get the best work done. There were four parts in this methodology section. The four parts were proposed methodology, SGM block diagram, FPGA and ASIC design flow. The proposed methodology reviewed the methods used in this project. Modules in SGM and its interaction between each other were shown in SGM block diagram section. The design flow represented the design and analysis process of the SGM.

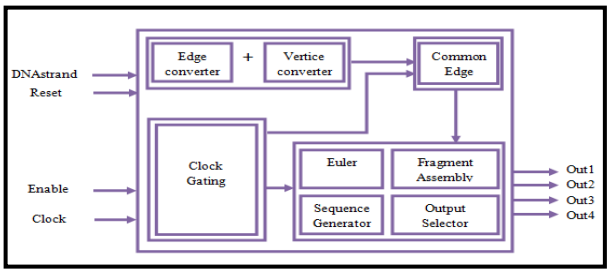
A. Proposed Methodology

The proposed methods were to use Euler path algorithm to construct the SGM using Verilog languange. Then, Verilog programme codes were modified to insert clock gating manually to each module in SGM. Simulations were conducted to verify the Euler path algorithm functionality. Next, power compiler constraints were applied to the design. There were three different versions of SGM. They were normal SGM, SGM with clock gating and SGM with clock gating and Power Compiler. The fourth version compared to the others three versions was compiled with ultra high effort.

B . SGM Block diagram



(a)



(b)

Figure 5 SGM blocks diagram (a) Without clock gating module (b) With clock gating module

There two blocks diagram represented the SGM in Figure 5. Figure 5(a) shows a block diagram without clock gating technique while in Figure 5(b) is a block diagram that was inserted with clock gating technique. Both blocks diagram have same modules. The difference between these blocks diagram is merely clock gating module.

The blocks diagram consists of seven modules. Each module have their own functionality. The modules are, edge and vertice converter, common edge, fragment assembly, sequence generator and output selector modules. Edge and vertice converter separated a genome into edge and vertice. Then, common edge finds the common edge between the edges. Euler module produced the possible output from the common edge modules. The fragment assembly functions to assemble the possible output while sequence generator combined the possible outputs and compare which are the best outputs. Next, the output selector produced the selected output according to the algorithm. There are four outputs from these SGM.

The clock gating module has been inserted in the block diagram to analyze how far clock gating reduced power consumption in SGM. The clock gating modules use input *clock* and *en*(enable) to produce *clk_gate* output. *Clk_gate* output acts as *clk_gate* input to all seven modules. The data only can be transfer from clock to clock when enable input is in high condition.

The interaction between blocks can be described in using a DNA strand as the input. A DNA strand, AATAATC is converted to 6-bit will be AAT, ATA, TAA, AAT and ATC. The long DNA strand entered the first and second module. The first module assigned edge to the path while second module break the long strand into short DNA fragments called vertices. Then, third block find and match the fractions. Since there are same vertices, the module considered only one vertice. It reduced number of vertices since there are common coding. The output from the module entered Euler module. The Euler module applied Euler path algorithm theory to the fragments. Then, fragment assembly module assembled the pieces of DNA fragments. It started to assembled and find the shortest path of the original DNA strand based on Euler path algorithm. The sequence generator produced the possible outputs which are the original DNA strand, AATAATC and the shortest path, AATC. Output selector produced the most selected output. In this example, the right output is AATC.

C. FPGA design flow

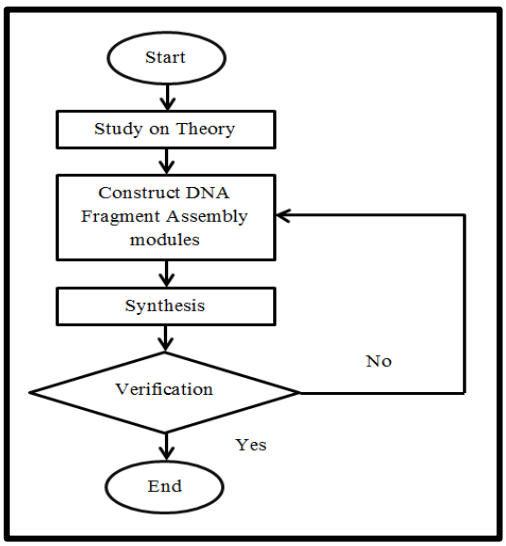


Figure 6 FPGA design flow

Figure 6 shows the FPGA design flow. The FPGA design flow started with study on theory as well as literature review. The scope of study covered information on DNA, DNA fragment assembly, Euler and de Bruijn algorithm, and low power techniques. Then, process of constructing the DNA fragment assembly modules can be proceeds. The process started with draw a draft of block diagram which is show the input and the output of SGM. After build the Verilog codes using Xilinx software, synthesis step took over to convert the Verilog code into Register Transfer Logic(RTL) schematic view. Verification process was done since the output must be verified before proceed to ASIC design flow. The error occurred in this step must be corrected. The simulation was simulated with a testbench.

D. ASIC design flow

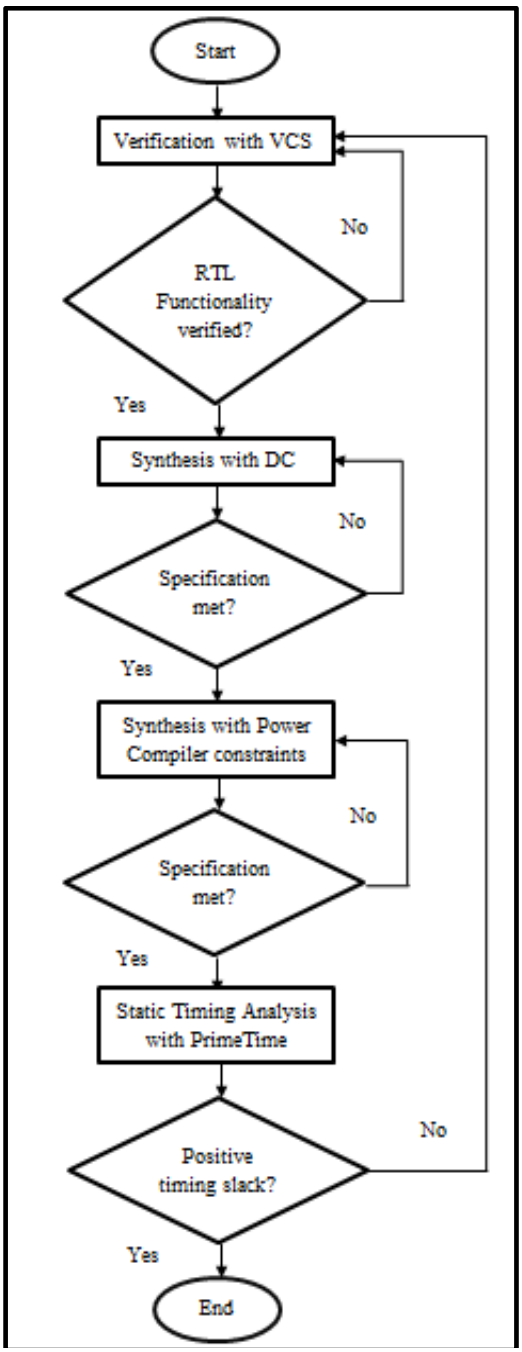


Figure 7 ASIC design flow

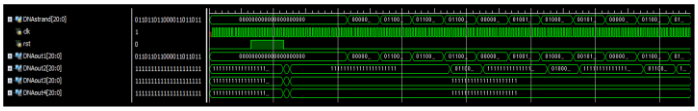
Figure 7 above shows the ASIC design flow. The process started with Verification Compiler Synopsys(VCS). The function of VCS is to verify the Verilog code. A waveform of the verification tools can be generated. From the waveform, the input and output can be analyzed and the RTL functionality was checked. The error can be detected and need to be corrected before proceed to the next step. If there is no

error and the specification was met, the SGM can proceed to the next step. The SGM was synthesized using Design Compile (DC). The libraries were setup. Constraints were applied. Various clock periods were used to determine the range of best period for SGM. Then, if the specification is met and there is no error, the synthesis process continued with Power Compiler. Analysis had been done. All the data values were recorded. The process continued with Static Timing Analysis (STA). This analysis was used PrimeTime. This advance analysis was done to determine the best timing slack for the SGM.

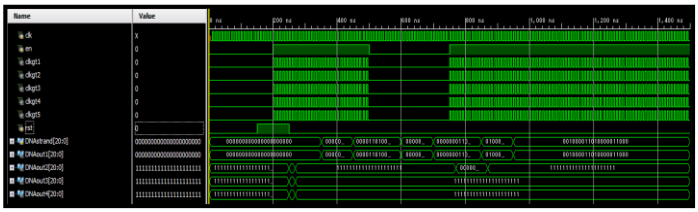
V. RESULTS AND DISCUSSIONS

A. Xilinx FPGA Simulation Results

The simulations were done to verify the functionality of the SGM. The output waveforms were shown at Figure 8. Figure 8(a) was SGM waveform while Figure 8(b) was SGM with clock gating. The output was corrected as assigned in the testbench. The clock gating for each module will only be ON when the enable is ON. The data also transmit when the clock gating output is high.



(a)



(b)

Figure 8 Simulation result (a) SGM (b) SGM with clock gating

Table 1 shows the DNA coding represented with binary numbers for the SGM.

TABLE I. BINARY REPRESENTATION OF DNA CODES

DNA Code	***Binary***
A	000
C	001
G	010
T	011
Ignore	111

Synthesizing process from Verilog code to RTL was done using Xilinx software. Schematics were displayed to view the interactions between the modules and identified unconnected modules. The synthesized schematics also were generated. Both schematics and synthesized schematics were shown in Figure 15 and Figure 16 respectively in Appendix.

B. VCS Simulation Results

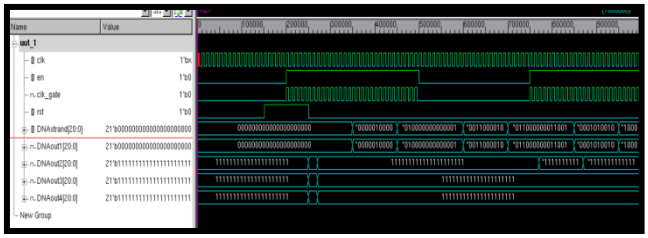


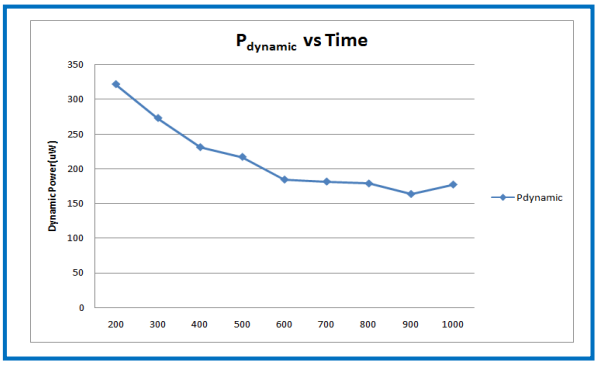
Figure 9 Verified waveform

In VCS, the functionality had been verified once again using the testbench. The testbench and all the Verilog files were loaded in VCS to be verified. In Figure 9 above, the clock gating functioned correctly.

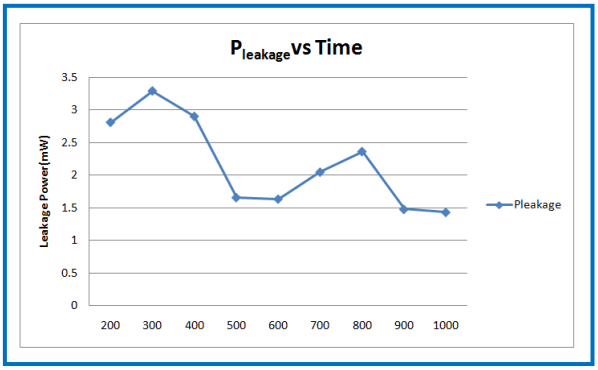
D. DC Simulation Results

There were four differences type of compile. Firstly, the SGM without any power optimization constraints were compiled. Second were the SGM with clock gating techniques. While the third is combination of clock gating technique and power optimization constraints. The fourth was the third SGM which were compiled with high effort constraints.

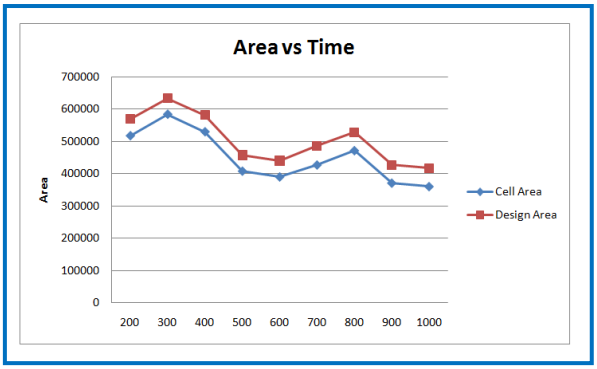
1) SGM



(a)



(b)

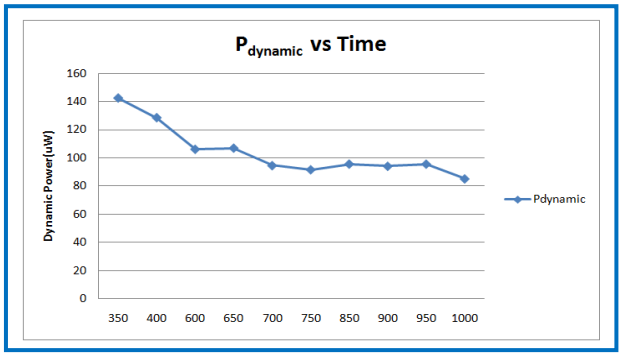


(c)

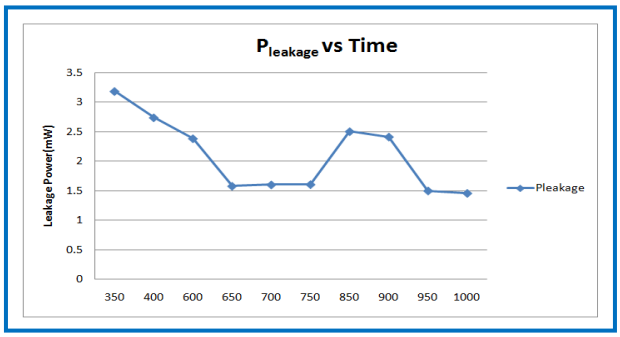
Figure 10 SGM (a) Dynamic Power vs Time (b) Leakage Power vs Time (c) Area vs Time

The graphs in Figure 10 represented SGM results after compiled with DC. The SGM have higher dynamic power at period of 200ns. The graph of $P_{dynamic}$ vs time was shown in Figure 10(a). The period ranges from 200ns until 1000ns. While the power ranges from 150uW to 330uW.Although at lower period the $P_{dynamic}$ is higher, the power decreased towards period increasing. $P_{dynamic}$ data was recorded in uW while $P_{leakage}$ was recorded in mW. $P_{leakage}$ has bigger power dissipation than $P_{dynamic.}$ The graph of $P_{leakage}$ versus time was shown in Figure 10(b). The highest value of $P_{leakage}$ is at 300ns. Figure 10(c) shows both cell and design area versus time. The value of area became smaller when the period is higher. The ranges of area are from 350,000 to 640,000.

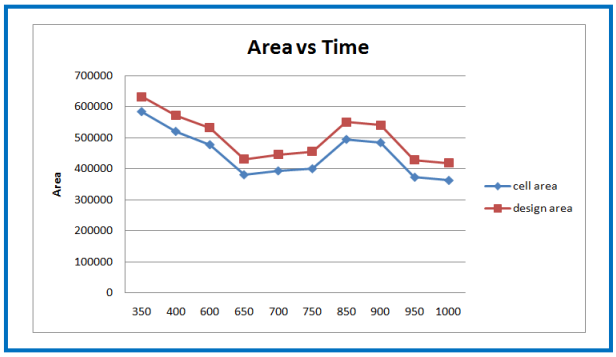
2) SGM with Clock Gating



(a)



(b)



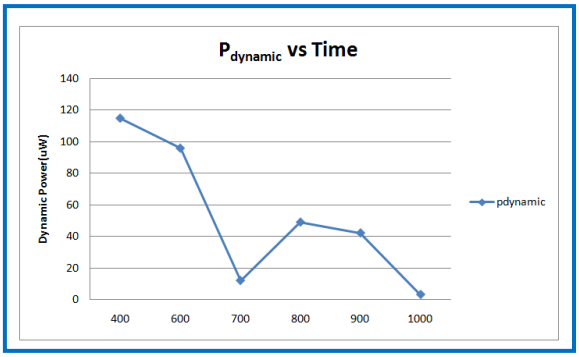
(c)

Figure 11 SGM with Clock Gating (a) Dynamic Power vs Time (b) Leakage Power vs Time (c) Area vs Time

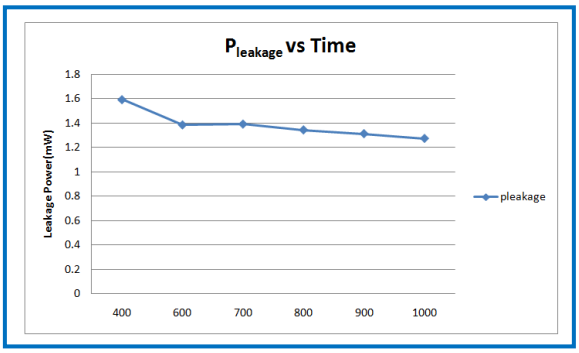
Figure 11 shows three graphs that represented SGM and clock gating technique. The manually inserted of clock gating successfully reduced the power consumption since at

Figure 11(a), the $P_{dynamic}$ ranges only from 80uW to 140uW. It is means that the highest $P_{dynamic}$ is 140uW at 350ns. Without clock gating implementation, the SGM $P_{dynamic}$ is 250uW at 350ns. This clock gating technique reduced the power consumption more than 40%. The $P_{leakage}$ at Figure 11(b) did not have many differences since the leakage power cannot control by designer. The area results decreased when higher period. In Figure 11(c), the ranges of area are from 350,000 to 640,000. This is proven that how effective clock gating technique in order to reduce dynamic power since this technique managed to reduce the power consumption and gives very minimal affect to area.

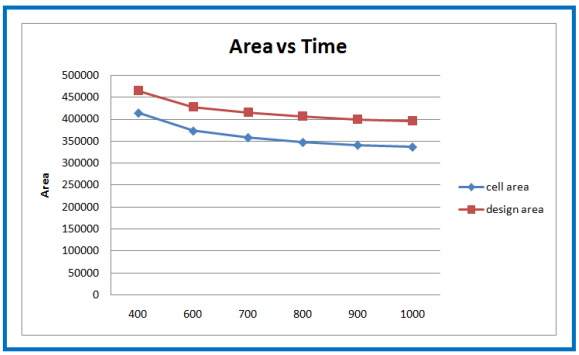
3) SGM with Clock Gating and Power Compiler



(a)



(b)



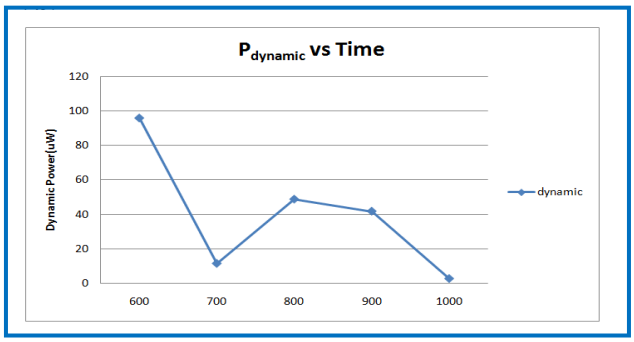
(c)

Figure 12 SGM with Clock Gating and Power Compiler (a) Dynamic Power vs Time (b) Leakage Power vs Time (c) Area vs Time

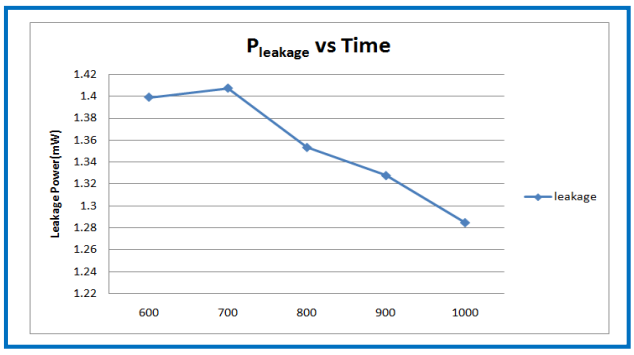
This section shows the results data from combination both low power techniques. They are clock gating and power compiler. This combination was a successful method since both $P_{dynamic}$ and $P_{leakage}$ gives a positive result. This method gives excellent ranges of $P_{dynamic}$ at Figure 12(a). The ranges are from 3uW to 120uW. At 1000ns, this method gives reading 3uW and tends to reduce more than 90% power consumption when compared to SGM with clock gating which gives reading 80uW at 1000ns.

The constraints were set to minimum value managed to optimize both $P_{dynamic}$ and $P_{leakage.}$ These constraints try to get the smallest value of leakage and dynamic power when the value is set to 0 mW. The $P_{leakage}$ decreased to 1.6mW compared to 3.5mW in method without power compiler. The ranges for $P_{leakage}$ are 1.2mW to 1.6mW. The area size also decreased. The new ranges are from 340,000 to 460,000. All the four methods that were used to analyze SGM did not use any constraints to get the minimum area. So, this method successfully reduces the area size and ranges.

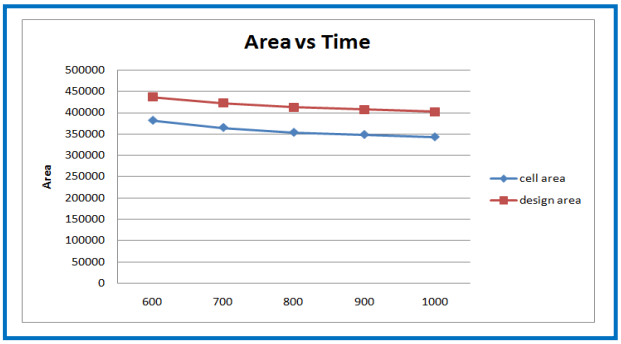
4) SGM with Clock Gating and Power Compiler – Compile with High Effort



(a)



(b)



(c)

Figure 13 SGM with Clock Gating and Power Compiler - High Effort (a) Dynamic Power vs Time (b) Leakage Power vs Time (c) Area vs Time

The last method compiled the previous SGM clock gating and power compiler method with high effort compiler. This high effort compiler more efficient and advanced compared to medium and low effort. Although there is not much differences between the results from previous method, this method also reduced the $P_{dynamic}$ to the lowest value in this analysis which is 2.8uW at 1000ns. This last method also decreased the value of $P_{leakage.}$ Both dynamic and leakage power graphs can be seen at Figure 13(a) and (b) respectively. The area size ranges from 350,000 to 450,000.

All the fourth versions showed that when the period is higher, the power consumption became lower. In terms of frequency, increasing of time means decreased in frequency. Indirectly, the methods implemented in the SGM produced low power consumption but in terms of speed, it resulted in low frequency. The lower power consumption can be reduced, the higher frequency that can be achieved is lower.

The power consumption at 1000ns in normal version of SGM was 177.5uW while SGM with clock gating and power compiler produced 3.2uW. The percentage that clock gating and power compiler managed to reduce was 98%.

D. PT Results Simulation

Static Timing Analysis is an advanced method to apply timing optimization. This analysis was performed by using PrimeTime.in the PrimeTime, designers can generate and interpret the timing reports. A clean and optimized run script also can be created. The timing violations also can be debugged using PrimeTime. STA functions to verify the timing. STA did not verify the functions of the modules. STA is fast in generating reports.

1) SGM

The timing slack for T_{MAX} and T_{MIN} were shown in Table II and III. Table II shows the result from DC while Table III shows result from STA. There is negative slack on DC timing results. However, STA results give a larger positive value of $T_{MAX.}$

TABLE II. DC TIMING RESULTS

Time(ns)	*T_{MAX} (T_{SETUP})*	*T_{MIN} (T_{HOLD})*
600	0.00	3.58
700	-458.30	3.94
800	-125.23	4.31
900	0.01	4.67
1000	0.00	5.04

TABLE III. PT TIMING RESULTS

Time(ns)	*T_{MAX} (T_{SETUP})*	*T_{MIN} (T_{HOLD})*
600	472.3636	3.5424
700	564.1144	3.9078
800	738.8798	4.2733
900	822.8370	4.6388
1000	928.1415	5.0043

2) SGM with Clock Gating

The value of T_{MIN} gives positive value for both timing slack. There is not much difference between T_{MIN} for both DC and STA timing results. However, the STA results are better than DC timing analysis. The timing slack for T_{MAX} and T_{MIN} were shown in Table II and III. Table II shows the result from DC while Table III shows result from STA. There is negative slack on DC timing results. While STA results give a positive value of $T_{MAX.}$

TABLE IV. DC TIMING RESULTS

Time(ns)	*T_{MAX} (T_{SETUP})*	*T_{MIN} (T_{HOLD})*
600	-208.03	3.58
700	0.00	3.94
800	0.00	4.31
900	0.03	4.67
1000	0.00	5.04

TABLE V. PT TIMING RESULTS

Time(ns)	*T_{MAX} (T_{SETUP})*	*T_{MIN} (T_{HOLD})*
600	474.6583	3.5424
700	567.5256	3.9078
800	667.3383	4.2733
900	774.1036	4.6388
1000	873.9178	5.0043

3) SGM with Clock Gating and Power Compiler

STA used formal mathematical techniques instead of vectors. This type of analysis did not use dynamic logic simulation. That is why results in timing of STA are better than DC. The T_{MAX} gives zero timing slack when compile with DC. However, STA results give a bigger number of positive value.

TABLE VI. DC TIMING RESULTS

Time(ns)	*T_{MAX} (T_{SETUP})*	*T_{MIN} (T_{HOLD})*
600	0.00	3.58
700	0.00	5.70
800	0.00	4.31
900	0.01	4.52
1000	0.04	7.46

TABLE VII. PT TIMING RESULTS

Time(ns)	*T_{MAX} (T_{SETUP})*	*T_{MIN} (T_{HOLD})*
600	556.6838	3.5424
700	636.1858	5.6757
800	735.0558	4.2733
900	833.5012	4.4855
1000	932.9248	7.4289

4) SGM with Clock Gating and Power Compiler – Compile with High Effort

This STA result shows that least of period contributed to least positive value. When the period is higher, the timing slacks also became higher. Both DC and STA timing results for T_{MIN} similar but for T_{MAX}, they are totally different except for the positive value of slack.

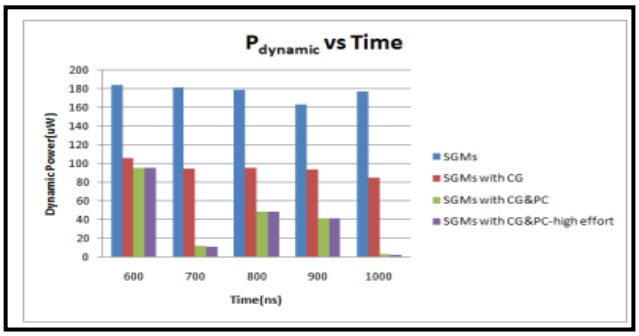
TABLE VIII. DC TIMING RESULTS

Time(ns)	*T_{MAX} (T_{SETUP})*	*T_{MIN} (T_{HOLD})*
600	0.01	3.58
700	0.01	5.70
800	0.00	4.31
900	0.00	4.52
1000	0.02	7.46

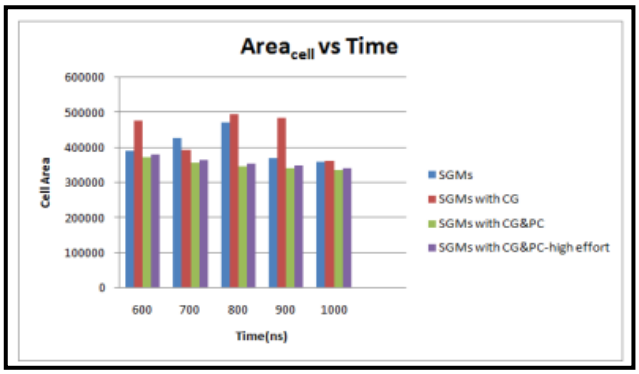
TABLE IX. PT TIMING RESULTS

Time(ns)	*T_{MAX} (T_{SETUP})*	*T_{MIN} (T_{HOLD})*
600	387.4489	3.5424
700	390.4757	5.6757
800	579.3708	4.2733
900	501.1537	4.4855
1000	749.9245	7.4289

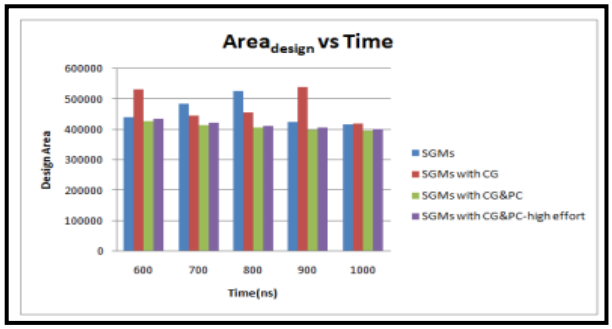
E. Comparison Chart



(a)



(b)



(c)

Figure 14 Comparison chart (a) Dynamic Power vs Time (b) Cell Area vs Time (c) Design Area vs Time

There are many differences between method number one, two and three. Method number one is SGM without additional constraints. While methods number two is SGM plus clock gating technique. Number three is SGM with clock gating and power compiler. Since this analysis focus on dynamic power consumption, the method that mostly can reduce power consumption is the third method. The method shows how the combination of both low power techniques effectively reduced the power consumption. The switching activity and clock frequency were reduced with clock gating technique while the power compiler constraints help to minimize the power consumption. The method also reduced the size of area. The design area increased a little bit with fourth method.

F. Calculation and Comparison on Power Consumption

This part displayed the calculation of power consumption between two methods. The normal SGM acted as the benchmark to compare with another three different techniques and method. These power consumptions, $P_{dynamic}$ and $P_{leakage}$ data were taken on period of 1000ns. In Table X, the $P_{dynamic}$ reduced 51.98% when SGM applied with manual clock gating. While $P_{leakage}$ increased 2.09% due to additional clock gating module in front of each sub-modules in SGM. However, combinational of clock gating and power compiler reduced both $P_{dynamic}$ and $P_{leakage.}$ In Table 4.20, it shows that $P_{dynamic}$ reduced 98.20% while $P_{leakage}$ reduced as much as 15.57%. Last comparison is the clock gating and power compiler techniques compile with high effort. The amount reduced of $P_{dynamic}$ is higher than the second comparison. It is 98.43%. However, the $P_{leakage}$ amount reduced less than the second comparison. It is due to remap and reconstructing the design in synthesis stage. The best method identified is SGM with clock gating and power compiler.

TABLE X. COMPARISON BETWEEN SGM AND SGM WITH CG

	SGM	SGM with CG	Amount reduced
$P_{dynamic(uW)}$	177.4853	85.2343	51.98%
$P_{leakage(mW)}$	1.4291	1.4589	-2.09%

TABLE XI. COMPARISON BETWEEN SGM AND SGM WITH CG&PC

	SGM	SGM with CG&PC	Amount reduced
$P_{dynamic(uW)}$	177.4853	3.1994	98.20%
$P_{leakage(mW)}$	1.4291	1.2734	15.57%

TABLE XII. COMPARISON BETWEEN SGM AND SGM WITH CG&PC(HE)

	SGM	SGM with CG&PC(HE)	Amount reduced
$P_{dynamic(uW)}$	177.4853	2.7906	98.43%
$P_{leakage(mW)}$	1.4291	1.2850	10.08%

VI. CONCLUSION

In conclusion, this design managed to find the most low power consumption for SGM. The clock gating technique successfully be inserted in the design and it function properly as it gives the best result. The implementation of clock gating also not affect any wrong results. The combination of clock gating and power compiler reduced 98% of power consumption compared to power consumption of normal SGM. Low power consumption also produced less area of the circuit. Therefore, the low power techniques implementation did not merely reduced power consumption but also reduced both cell and design area. Future analysis will be using advanced low power technique such as power gating.

ACKNOWLEDGEMENT

The author would like to thank Mr. Abdul Karimi Halim for supervising this project. With his guidances and technical feedback, this project managed to come out with the results. Author would also like to appreciate the Faculty of Electronic Engineering in Universiti Teknologi MARA Shah Alam for providing suitable workplace for the author to conduct the project and finish it successfully.

REFERENCES

[1] L. Li and S. Khuri, "A Comparison Of DNA Fragment Assembly Algorithm", In Proceeding of 2004 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences, Las Vegas, pp. 329-335, 2004.

[2] Y. Zhang* and M. S. Waterman, "DNA Sequence Assembly and Multiple Sequence Allignment by an Eulerian Path Approach" Department of Mathematics, University of Southern California, Los Angeles, May 2003

[3] P.A. Pevzner*, H. Tang†, and M. S. Waterman†‡§, An Eulerian path approach to DNA fragment assembly, PNAS, vol.98, no.17, 9748–9753, 2001.

[4] D. MacLean, J. D. G. Jones and D. J. Studholme, Application of 'next-generation' sequencing technologies to microbial genetics, Nature Reviews ,Vol.7, 287- 296, April 2009

[5] D. R. Zerbino, Genome assembly and comparison using de Bruijn graphs, September 2009

[6] P. E. C.Compeau, P. A Pevzner& G.Tesler, How to apply de Bruijn graphs to genome assembly, Nature biotechnology, vol.29, no.11, 987-991, 2011

[7] J. Ravindra and T. Anuradha, "Design of Low Power Risc Processor by Applying Clock Gating Technique", International Journal of Computer Applications (0975-8887), vol. 43, April 2012.

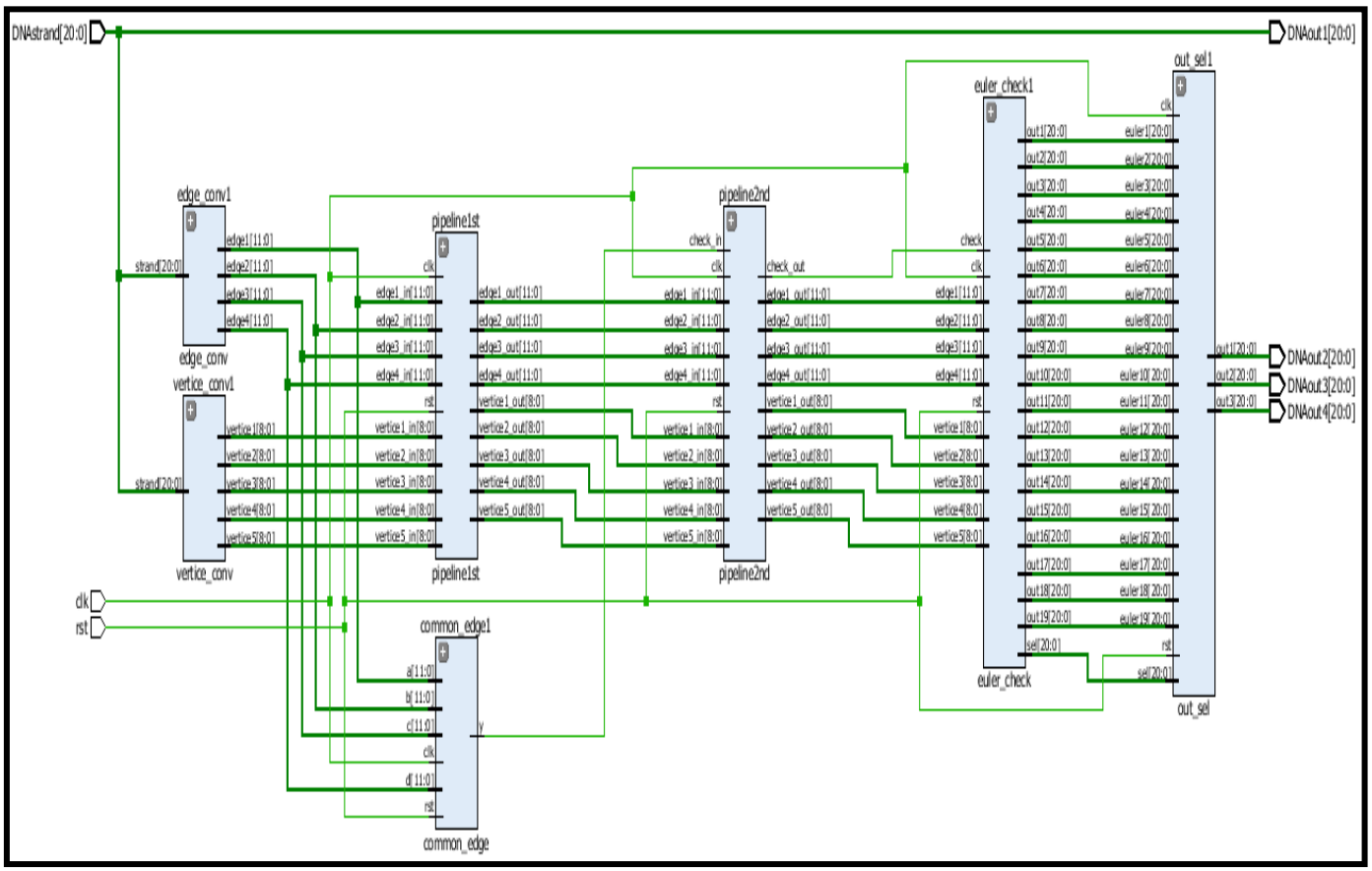
[8] J. Shinde and Dr. S.S. Salankar, "Clock Gating – A Power Optimizing Technique for VLSI Circuits", Department of Electronics and Telecommunication Engineering, Nagpur

[9] "Synopsys Low-Power Flow User Guide"Synopsys Inc.,Version D-2010.03, March 2010

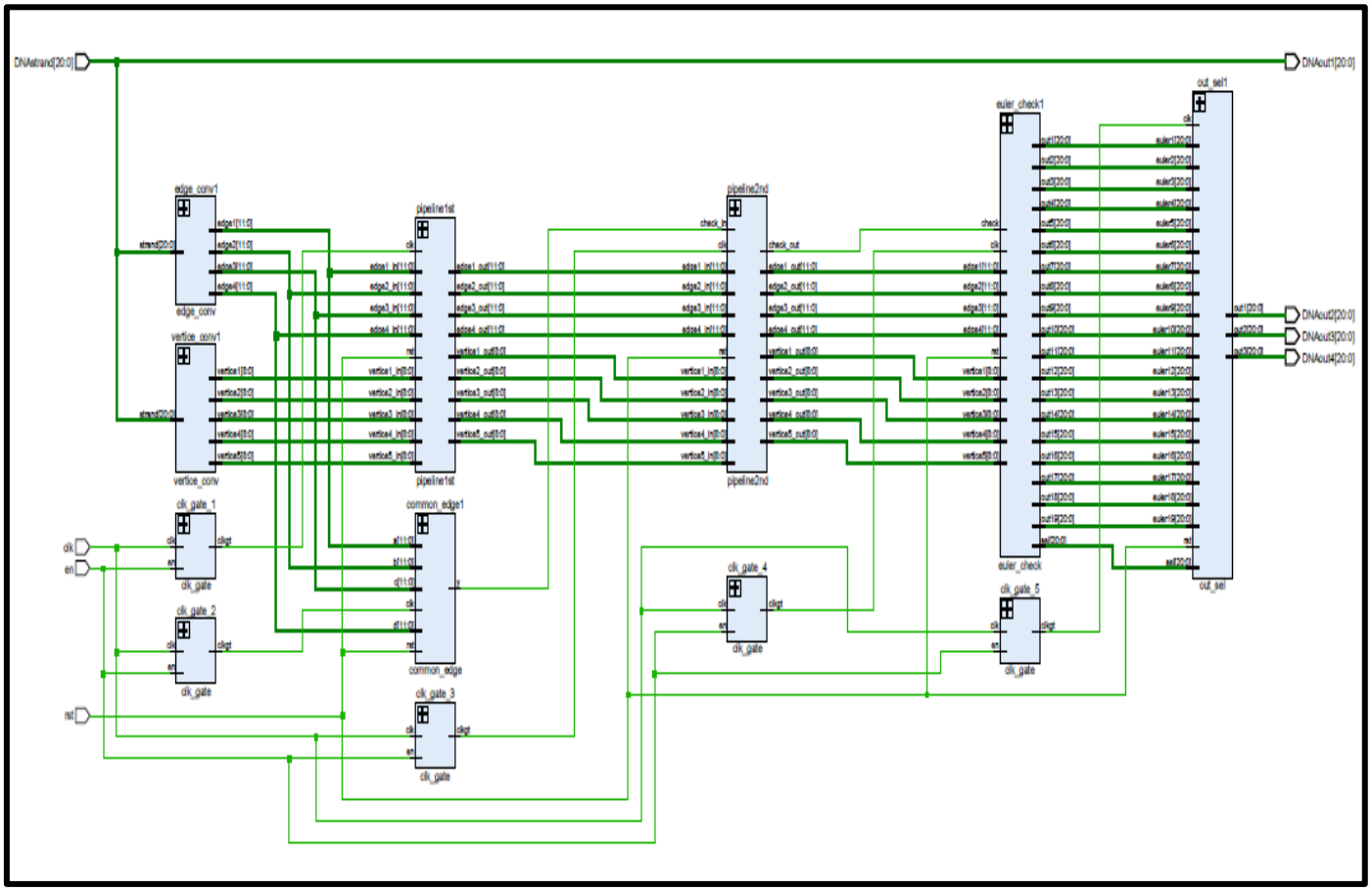
[10] "Low Power Flow-Logic Implementation", Low Power Tutorial, Synopsys Inc.

[11] D. Chinnery and K. Keutzer, "Overview of the Factors Affecting the Power Consumption", Department of Electrical Engineering and Computer Sciences, University of California, USA

APPENDIX

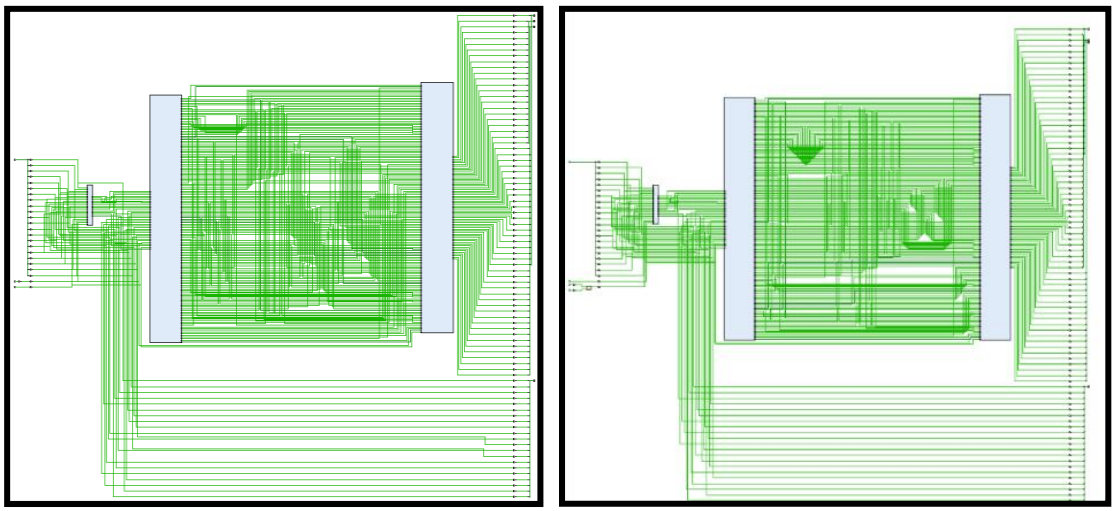


(a)



(b)

Figure 15 Schematic (a) SGM (b) SGM with clock gating



(a) (b)

Figure 16 Synthesized schematics (a) SGM (b) SGM with clock gating