DESIGN OF COARSE-GRAINED POWER GATING FOR A FINE-GRAINED MANY-CORE PROCESSOR ARRAY

by
Shylesh Umapathy

A thesis
submitted in partial fulfillment of the requirements for the degree of
Master of Science in Engineering, Computer Engineering Option
in the Lyles College of Engineering
California State University, Fresno
May 2018
APPROVED
For the Department of Electrical and Computer Engineering

We, the undersigned, certify that the thesis of the following student meets the required standards of scholarship, format, and style of the university and the student's graduate degree program for the awarding of the master's degree.

_____________________________________
Shylesh Umapathy

Aaron Stillmaker (Chair) Electrical and Computer Engineering

Reza Raeisi Electrical and Computer Engineering

Nan Wang Electrical and Computer Engineering

For the University Graduate Committee:

_____________________________________
Dean, Division of Graduate Studies
AUTHORIZATION FOR REPRODUCTION
OF MASTER’S THESIS

Yes, ________ I grant permission for the reproduction of this thesis in part or in its entirety without further authorization from me, on the condition that the person or agency requesting reproduction absorbs the cost and provides proper acknowledgment of authorship.

__________ Permission to reproduce this thesis in part or in its entirety must be obtained from me.

Signature of thesis author: _____________________________________________
ABSTRACT

DESIGN OF COARSE-GRAINED POWER GATING FOR A FINE-GRAINED MANY-CORE PROCESSOR ARRAY

With the 53rd commemoration of Moore’s law and transistor sizing heading towards 4 nm, the number of transistors on an integrated circuit continue to double every year. However, there are many factors limiting this growth rate such as power consumption, which is a serious impediment for design of high-speed, low-power integrated circuits. In modern semiconductor manufacturing, leakage power in high-performance processors accounts for 20-30% of the total power. Power gating is one approach to reduce the power consumption of an integrated circuit by effectively disconnecting the power supply from blocks during idle mode and is commonly used in the design of commercial high-end processors and in SoC for portable applications.

In this thesis, coarse-grained power gating techniques are explored to reduce the power consumption of a fine-grained many-core processor array. This work provides a detailed analysis and comparison of the design tradeoffs between using the ring and grid methods of power gates placement. The RTL to GDSII flow was performed using Synopsys EDA tools and NanGate FreePDK45 standard open cell library. The results depict that, on a test 128-bit MAC unit, placing power gates using the grid methodology created an 8.52% less IR drop but an 8-9% congestion increase when compared to the ring methodology. The leakage power of the 128-bit MAC unit was reduced by 99.98% with both the grid and ring placement of power gates when compared to a design without power gates.

Shylesh Umapathy
May 2018
ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. Aaron Stillmaker for the continuous support during my thesis, for his patience, immense knowledge, and motivation. He has always been there for my help and without his guidance, this thesis and report wouldn’t have been possible. I was fortunate enough to work under his guidance and I would cherish this experience for a long time.

I am very thankful to my committee members Dr. Reza Raeisi, who has always inspired me with his leadership qualities and has always been there to help a student in every possible way. I wouldn’t have had a better understanding of Computer Architecture, if I hadn’t worked with Dr. Nan Wang, his seminars have helped me to sharpen my presentation skills. I am also very thankful to all the faculty members of Electrical and Computer Engineering for all the help.

I would like to thank my friends, especially Meet Shah, who has always helped me, whenever I ran into a trouble while installing and configuring the Synopsys tools. I would also like to thank Brian Cardwell, who had helped me to generate all the great plots in this report.

Finally, I must express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
<tr>
<td>CHAPTER 1: INTRODUCTION</td>
<td>11</td>
</tr>
<tr>
<td>CHAPTER 2: BACKGROUND</td>
<td>14</td>
</tr>
<tr>
<td>Low Power Architectures</td>
<td>16</td>
</tr>
<tr>
<td>Power Gating</td>
<td>21</td>
</tr>
<tr>
<td>Related Work</td>
<td>26</td>
</tr>
<tr>
<td>CHAPTER 3: TOOLS AND LIBRARIES</td>
<td>33</td>
</tr>
<tr>
<td>NanGate FreePDK45</td>
<td>34</td>
</tr>
<tr>
<td>CHAPTER 4: METHODOLOGY</td>
<td>36</td>
</tr>
<tr>
<td>Library preparation for NanGate FreePDK45</td>
<td>36</td>
</tr>
<tr>
<td>Designing 128-bit MAC unit</td>
<td>37</td>
</tr>
<tr>
<td>Synthesizing gate-level netlist</td>
<td>38</td>
</tr>
<tr>
<td>Place and Route Physical Design</td>
<td>40</td>
</tr>
<tr>
<td>Header design in Custom Compiler</td>
<td>50</td>
</tr>
<tr>
<td>HSPICE simulation of Power gates</td>
<td>52</td>
</tr>
<tr>
<td>CHAPTER 5: RESULTS AND ANALYSIS</td>
<td>53</td>
</tr>
<tr>
<td>ModelSim Simulation</td>
<td>53</td>
</tr>
<tr>
<td>Place and Route</td>
<td>53</td>
</tr>
<tr>
<td>CHAPTER 6: CONCLUSION AND FUTURE WORK</td>
<td>66</td>
</tr>
<tr>
<td>Conclusion</td>
<td>66</td>
</tr>
<tr>
<td>Future Work</td>
<td>67</td>
</tr>
<tr>
<td>APPENDICES</td>
<td>73</td>
</tr>
<tr>
<td>APPENDIX A: Library Compiler and Milkyway</td>
<td>.............................................................. 74</td>
</tr>
<tr>
<td>APPENDIX B: TLUplus File</td>
<td>........................................................................ 76</td>
</tr>
<tr>
<td>APPENDIX C: Verilog Code</td>
<td>........................................................................ 79</td>
</tr>
<tr>
<td>APPENDIX D: Design Compiler Script</td>
<td>........................................................................ 81</td>
</tr>
<tr>
<td>APPENDIX E: IC Compiler Script</td>
<td>........................................................................ 84</td>
</tr>
<tr>
<td>APPENDIX F: HSPICE Deck</td>
<td>........................................................................ 88</td>
</tr>
<tr>
<td>APPENDIX G: Detailed Timing Report</td>
<td>........................................................................ 90</td>
</tr>
<tr>
<td>APPENDIX H: Detailed Congestion Report</td>
<td>...................................................................... 100</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 1. Impact of scaling on ARM processor MIPS/mW [4] ........................ 11
Table 2. Fine-grained power gating techniques [18] ................................. 27
Table 3. Comparison between NEMS and CMOS header switch [24] .......... 32
Table 4. Leakage Power of Power gates .................................................. 52
Table 5. Impact of power gates placed in grid manner .............................. 58
Table 6. Impact of power gates placed in ring manner ............................... 60
Table 7. Impact of power gates on voltage and current ......................... 62
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Header and Footer power gates [17]</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>Grid of switches [19]</td>
<td>23</td>
</tr>
<tr>
<td>3</td>
<td>Ring of switches [19]</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>Leakage Power reduction [20]</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>Power reduction based on gating [20]</td>
<td>28</td>
</tr>
<tr>
<td>6</td>
<td>Power reduction per core in AMD x86-64 [12]</td>
<td>29</td>
</tr>
<tr>
<td>7</td>
<td>Noise due to power gating [21]</td>
<td>29</td>
</tr>
<tr>
<td>8</td>
<td>Incremental scheduling [21]</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>Energy saved based on temperature [22]</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>Impact of temperature on Leakage Power [23]</td>
<td>31</td>
</tr>
<tr>
<td>11</td>
<td>128-bit MAC unit</td>
<td>38</td>
</tr>
<tr>
<td>12</td>
<td>Block diagram of the 128-bit MAC unit in Design Compiler</td>
<td>39</td>
</tr>
<tr>
<td>13</td>
<td>Detailed circuit diagram in Design Compiler</td>
<td>39</td>
</tr>
<tr>
<td>14</td>
<td>Voltage domain</td>
<td>41</td>
</tr>
<tr>
<td>15</td>
<td>Voltage rails</td>
<td>41</td>
</tr>
<tr>
<td>16</td>
<td>Power gates placed in grid manner</td>
<td>42</td>
</tr>
<tr>
<td>17</td>
<td>Power gates placed in ring fashion</td>
<td>43</td>
</tr>
<tr>
<td>18</td>
<td>Synthesized Power Network</td>
<td>44</td>
</tr>
<tr>
<td>19</td>
<td>Committed Power Network</td>
<td>45</td>
</tr>
<tr>
<td>20</td>
<td>Standard cell placement</td>
<td>46</td>
</tr>
<tr>
<td>21</td>
<td>Routed power rails</td>
<td>47</td>
</tr>
<tr>
<td>22</td>
<td>Clock Tree Synthesis</td>
<td>48</td>
</tr>
<tr>
<td>23</td>
<td>Final routed design</td>
<td>49</td>
</tr>
</tbody>
</table>
Figure 24. Detailed routing ........................................................................................................... 49
Figure 25. Power gate HEADER_X1 ............................................................................................. 50
Figure 26. Power gate HEADER_X2 ............................................................................................. 51
Figure 27. Power gate HEADER_X4 ............................................................................................. 51
Figure 28. Power gate with terminals defined .............................................................................. 52
Figure 29. ModelSim simulation .................................................................................................. 53
Figure 30. 128-bit MAC unit cell count ....................................................................................... 54
Figure 31. 128-bit MAC unit total area ......................................................................................... 54
Figure 32. 128-bit MAC timing report ......................................................................................... 55
Figure 33. 128-bit MAC utilization report ..................................................................................... 56
Figure 34. 128-bit MAC power report ......................................................................................... 57
Figure 35. Impact on area and power with varied sized power gates placed in grid manner .......... 59
Figure 36. Impact on area and power with varied sized power gates placed in ring manner ........ 61
Figure 37. Leake power comparison during idle mode ................................................................. 62
Figure 38. 128-bit MAC congestion plot ...................................................................................... 63
Figure 39. Congestion report for ring pattern power gates placement ...................................... 64
Figure 40. Congestion report for grid pattern power gates placement ....................................... 65
CHAPTER 1: INTRODUCTION

In the current era, with technology scaling heading towards sub nanometer, there are more than billions of transistors on a chip, resulting in faster and smaller devices. However, high-performance chips have power consumption as a major obstacle. In the year 2014, 70 billion kWh of electricity was consumed by data centers, which accounts to 1.8% of the total U.S. electricity consumption [1]. Current study show energy consumption of data centers would continue to increase 4% from 2014 – 2020, estimating the data centers would consume 73 billion kWh in 2020 [1].

With technology scaling heading towards 4 nm [2], it is possible to accommodate more functionality per area and in turn the performance has improved in terms of throughput and latency. Along with improving the performance, technology scaling has also increased the power density, which is a one of the limiting factor for further reducing the feature size [3]. The table 1 depicts the impact of scaling on the power consumption of ARM processors [4].

<table>
<thead>
<tr>
<th>Processor</th>
<th>Technology (um)</th>
<th>Voltage</th>
<th>Power (mW)</th>
<th>MIPS/mW</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM710</td>
<td>0.60</td>
<td>5.0</td>
<td>424</td>
<td>0.08</td>
</tr>
<tr>
<td>ARM810</td>
<td>0.50</td>
<td>3.3</td>
<td>500</td>
<td>0.17</td>
</tr>
<tr>
<td>ARM910T</td>
<td>0.35</td>
<td>3.3</td>
<td>600</td>
<td>0.22</td>
</tr>
<tr>
<td>ARM920T</td>
<td>0.25</td>
<td>2.5</td>
<td>560</td>
<td>0.39</td>
</tr>
<tr>
<td>ARM1020E</td>
<td>0.18</td>
<td>1.5</td>
<td>400</td>
<td>1.25</td>
</tr>
<tr>
<td>ARM1020E</td>
<td>0.13</td>
<td>1.1</td>
<td>240</td>
<td>2.08</td>
</tr>
</tbody>
</table>
For transistor feature size of 90nm and above, the dynamic power consumption of an IC (Integrated Circuit) played a major role in determining the power requirements [3]. As transistors are getting smaller in size, there is reduction in the oxide thickness, gate length, and threshold voltage, leading to short-channel effects [3]. These lead to non-ideal behavior of a transistor in sub-threshold region [3, 5]. Leakage current is greatly affected by reduction in threshold voltage and gate oxide thickness, it is seen that, linear reduction in the threshold voltage results in an exponentially increase of the leakage current [5].

This thesis discusses various methodologies generally used to reduce the power consumption and focuses on the details related to coarse-grained power gating technique specifically for fine-grained processor array. This report also includes different approaches of using power gates, grid and ring style, using varied sized power gates and discusses the pros and cons of each design approach. A 128-bit Multiplication and Accumulation (MAC) unit was designed as the fine-grained processor. Synopsys EDA tools and IEEE 1801-2013 Unified Power Format (UPF) standard has been used to perform the coarse-grained power gating.

The report organization is as follows:

Chapter 1: Introduction – Introduces the thesis work and the goal of the work in brief.

Chapter 2: Background – This chapter gives details related to the various sources of power dissipation and the different techniques known to reduce the same.

Chapter 3: Tools and Library – In this chapter, details pertaining to the tools used for simulation and the standard cell library used for the work will be discussed.
Chapter 4: Methodology – This chapter details the approach and procedure used to implement the power gating technique.

Chapter 5: Results and Analysis – This chapter discusses the finding and the interpretation of the obtained results.

Chapter 6: Conclusion and Future work – This chapter summarizes the thesis work and provide details related to the future possible work.
CHAPTER 2: BACKGROUND

There are many sources of power dissipation in a transistor, and several known techniques to reduce them. The power dissipation in a transistor is composed of two components, dynamic and static power consumption. The factors contributing to the dynamic power are, the power lost during the charging and discharging of the load capacitance, usually represented as $P_{\text{switching}}$. And the short-circuit current, when both NMOS and PMOS transistors are ON, represented as $P_{\text{short circuit}}$ [3].

The key components of the static power consumption are, the sub-threshold leakage current, $I_{\text{sub}}$, gate leakage current, $I_{\text{gate}}$, and junction leakage current, $I_{\text{junction}}$. The total power consumption of a transistor is the sum of dynamic and static power [3],

$$P_{\text{total}} = P_{\text{dynamic}} + P_{\text{static}} \quad (2.1)$$

More details related to the power dissipation in a transistor are explained below:

1) **Dynamic Power Dissipation**: The dynamic power consumption of a circuit is mainly due to the switching activity in the transistors [3, 5]. The components of dynamic power dissipation are explained below:

   a. **Switching Power**: The main component of the dynamic power consumption in a transistor is the power lost during switching i.e., when the transistor operates [3, 5]. When a circuit operates, it must either charge or discharge the load capacitance. Even though the supply voltage for the circuit to operate is $V_{dd}$, but only half of the energy supplied by the source is stored in the load capacitance, the other half is dissipated by the PMOS logic [3]. This switching power mainly depends on the load capacitance ($C_L$), supply voltage ($V_{dd}$),
operating frequency \((f)\), and activity factor \((\alpha)\) [3, 5]. The switching power is written as follows [3]:

\[
P_{\text{switching}} = \alpha C_L V_{dd}^2 f \quad (2.2)
\]

b. Short-circuit current: The short-circuit power dissipation is encountered every time there is an activity in the circuit. The reason is that, for a very small duration, both the pull-up and pull-down circuit is partially ON during the switching. This leads to a direct path from the source \(V_{dd}\) to ground \(V_{ss}\), resulting in short circuit current. The short circuit current is said to be less than 10% of the total dynamic power dissipation [3, 6].

2) Static Power Dissipation: Static power dissipation was once negligible compared to dynamic power dissipation for technology above 180 nm, except in very low power applications [3]. But with reduction in transistor size, threshold voltage, and thinner gate oxides, leakage power is catching up and is now considered to be an important factor [3, 5]. Static power is consumed when the transistor is idle, and not switching. For low performance application, leakage can contribute for a large portion of power consumption, especially during long idle time [4]. The components of the static power are explained as follows:

a. Sub-threshold leakage: A transistor is in OFF state, when \(V_{gs} < V_t\), i.e., the voltage applied to the gate terminal isn’t sufficient to invert the channel and conduct \(I_d\). When the transistor is turned OFF, it is assumed that the current \(I_d\) is zero. But there is still a weak inversion in this situation, which leads to conduction between the drain and source, known as sub-threshold leakage [3, 4, 5].
The other factor adding to the sub-threshold leakage is DIBL (Drain Induced Barrier Leakage) effect. With reduced channel length, the threshold voltage $V_{t}$ also reduces for a given $V_{ds}$. In such scenario, drain has more potential to create the channel and hence reducing the gate effect or the threshold voltage. These factor in turn increase the sub-threshold leakage current [3, 5].

b. Gate leakage: This is the leakage current from the gate terminal to the body. A main factor controlling the gate leakage is the dielectric thickness [3, 5]. With technology scaling, even the thickness of the gate oxide is reducing, leading to an increase in gate leakage current [3, 4, 5]. The carriers in the gate terminal tunnel through the thin gate oxide accounting for the gate leakage current.

c. Junction leakage: Junction leakage current is mainly due to the p-n junction diode formed between the source-substrate and drain-substrate. Even though the substrate is usually tied to either $V_{dd}$ or $V_{ss}$ to avoid these diodes being forward biased, but with this configuration, the diodes are reverse biased and produce reverse leakage current [3].

**Low Power Architectures**

For the known sources of power dissipation in modern circuits, it is very essential to look for techniques that would help to reduce the same. The power optimization can be performed at various levels of the design phase. The below section provides details about the different techniques used for reducing the power consumption.
Power optimizing at Design time: Designers chose the architecture and algorithms keeping in mind the performance, area and power optimization of the circuit [5]. Below are few techniques that have proven to have a positive impact towards reducing the overall power consumption of circuits, they are:

1) Many-core architecture: Moving towards larger number of cores to handle tasks have proven to reduce the overall power consumption of the design without impacting the performance of the design. Smaller cores also have smaller memories, resulting in faster access and reduced power consumption due to less activity per core. This many-core architecture comes with programming complexity [3, 5, 7, 8, 9].

2) Parallelism: Parallelism/Concurrency is a way to implement the same design in multiple branches. Now these individual branches can operate at half of the speed, hence reducing the frequency and which in turn reduces the dynamic power [3, 5]. Since there are multiple branches performing the same function, there is an area overhead and the leakage power is further increased [5, 10].

3) Pipelining: Pipelining techniques improve the throughput of the design by overlapping the executing of multiple instructions by dividing them into stages. Pipelining also increases the area overhead by adding extra registers, but when compared to parallelism, the area overhead is reduced [5, 12].

Power optimizing at Standby: The techniques mentioned previously mostly focus on reducing the dynamic power consumption. But in the current technology trend, it is very essential to reduce the static power consumption as well. When all the transistors in the design are leaky, completely or partially turning off modules is a
better option to reduce the static power consumption [3, 4, 5, 6]. Below are few of the known techniques to reduce the power when the design is idle:

1) Clock gating: Clock gating is an effective way to turn off the dynamic power consumption of the registers during idle mode [3, 5]. In this technique, blocks can be turned off by stopping the clock that is fed to them. Using this method, the dynamic power dissipation of idle blocks can be reduced. This technique needs additional circuitry to control the clock and utmost care must be taken to avoid glitches. [5, 6, 12].

2) Power gating: The best-known way to reduce the leakage current is to disconnect the transistors from the supply ($V_{dd}$), which is achieved by power gating. In power gating, the transistors are connected to the power rails during normal operation and disconnected from the power rails during inactive periods. This technique helps in reducing both leakage and dynamic power consumption of a design [3, 5, 6, 12].

3) Dynamic body biasing: Body is the fourth terminal of a transistor which is usually tied to the source. It has been seen that, the body terminal can be used to control the performance and leakage of transistors [5]. Applying voltage to the body terminal, either increases/decreases the charge needed to invert the channel and this in turn has an impact on the threshold voltage $V_t$ [3]. Applying reverse body biasing to the transistors during idle mode has shown to increase the threshold voltage of the transistors, which in turn reduces the static power dissipation. This technique would need additional power rails and a circuitry to control the power supplied to the body terminal [3, 5, 13].

4) Supply voltage ramping: It is seen that, with ramping down the supply voltage, the leakage power also reduces. This technique is useful for reducing the leakage power consumption of the cache [5, 6]. Completely turning off
the power supply to memory results in the loss of the data stored. Studies indicated that, cache memories need a minimum voltage to retain the data stored in them, this is termed as DRV (Dynamic Retention Voltage) [5]. Using this technique, the supply voltage can be reduced, slightly above the DRV, so that along with reducing the static power dissipation of the memory, even the data is retained [5, 6].

**Power optimizing during runtime:** Along with reducing the power dissipation during idle or standby mode, it has been seen that, there are opportunities during runtime that can be used to reduce the runtime power dissipation as well. Periods of low activity is the best time to incorporate power optimization techniques [5, 6, 12]. During runtime, supply voltage and clock frequency are the parameters that can be adjusted to get the desired power reduction. Below listed are few of the techniques:

1) DVFS (Dynamic Voltage and Frequency Scaling): In this technique, the workload of a processor is monitored. Based on the current requirement/workload, the voltage and frequency applied to the processing block is controlled [3, 5]. During active periods, maximum voltage and frequency is applied to achieve better performance in terms of throughput and latency. Minimum or low voltage and frequency is applied during periods of low activity to reduce the dynamic and static power consumption. Having to adjust the supply voltage and frequency adaptively and continuously would be an overhead [5, 12, 14, 15].

2) Adaptive body biasing: As mentioned earlier, body is the fourth terminal of a transistor, that can be used to control the performance and power consumption of a transistor. During the period of low activity, reverse body bias can be applied to reduce the leakage power and forward body bias can
be applied during active periods to improve performance. This method needs additional power rails and having to adjust the voltage applied adaptively is an overhead [3, 5, 13].

3) Multiple voltage domains: In any integrated SoC (System on Chip), it is not common for all blocks to have same performance requirements. Based on the individual performance needs, the chip can be partitioned into multiple voltage domains. Now, the modules which are expected to be active most of the time can be supplied the full $V_{dd}$ to ensure the performance is not hindered. Whereas the supply voltage can be ramped down for modules which has lesser active periods, which reduces both dynamic and static power dissipation. The challenge in this approach is converting the voltage levels for signals that cross domains [3, 5, 12].

Power optimizing at Circuit level: Apart from reducing the power consumption at the architectural level, there are techniques that can be applied at the circuit level. Below are few of the known techniques:

1) Multiple threshold voltage cells: Usage of multiple threshold voltage cells is an effective tool in static power optimization. This approach is useful during the place and route stage. Reduction in the threshold voltage, $V_t$ of a transistor increases the performance and the leakage power. Increasing the $V_t$ of a transistor reduces the leakage power and the performance [3, 5]. It is a good design choice to use the low $V_t$ cells in the critical path of the design, to improve the performance and use high $V_t$ cells in non-critical path, to reduce the leakage power consumption. This approach is feasible only if there are multiple $V_t$ cells available in the technology library, and this increases manufacturing cost and needs level conversion [3, 5, 12].
2) Weak-inversion mode: A Complementary Metal Oxide Semiconductor (CMOS) transistor can operate in sub-threshold mode, known as weak-inversion mode. Cooling down the circuit close to absolute zero, using liquid helium can make it to operate in millivolts levels. A CMOS inverter can operate in sub-threshold mode at 36 mV at room temperature [5]. Operating circuits at such low voltage, drastically reduces the power consumption of a transistor. But there are few drawbacks with this approach, such as the noise margin is reduced to zero and distinguishing between a “0” and “1” would be difficult. Even the performance of the design is impacted because mobility of the carriers is degraded with reducing the temperature. Hence, this approach is valid only for low-performance, low-power applications [3, 5, 16].

Power Gating
As mentioned earlier in this chapter, power gating is one of the most convenient way to reduce both the dynamic and leakage power dissipation of a design during idle mode [3, 5]. This can be achieved by using a switch between the $V_{dd}$ and the design or between $V_{ss}$ and the design. If the supply voltage is gated, the switch/power gate is termed as a header, using PMOS transistor. If the ground is gated, the switch/power gate is called as a footer, using NMOS transistor. The figure 1 is a pictorial representation of the above mentioned.
In the figure 1, the configuration on the left-hand side is a header and the other is a footer configuration. The working of this design is as explained below:

1) **Active mode:** During active mode, the power switch would be ON, indicating the processor/core is connected to both $V_{dd}$ and $V_{ss}$, allowing for normal operation [3].

2) **Standby mode:** During inactive periods, the power switch would be turned OFF, so either the supply voltage or ground is disconnected from the core. In such situation, there is no connection between the source and the ground resulting in a very minimal power consumption [3, 5, 12].

In this thesis, the header configuration is used, to ensure the ground is stable, as this is a preferred configuration by most designers [3]. Based on the positioning of these power gates, there are two design configurations:

1) **Fine-grained power gating:** In this approach, there is a power gate added to every standard cell in the design [5, 17, 18]. This is not a design choice because there is a huge area overhead on the design to add power gated for every standard cell. Also, every time the cell is turned OFF, the output of the
cell should be isolated from the rest of the design, which increases complexity [4, 5].

2) **Coarse-grained power gating**: In this approach, power gates are added for a processing block/core. So, during inactive periods, the entire block is turned OFF. Compared to fine-grained, the area overhead is less in coarse-grained power gating and isolation at a block level is easier than isolating every cells output [5, 6, 17]. There are two different styles of coarse-grained power gating, they are as mentioned below:

a. Grid of switches: In this approach, power gates are inserted inside the design and placed in a grid manner as shown in the figure 2 [17, 19].

![Figure 2: Grid of switches](image)

This approach is mostly preferred whenever there is a need to retain the data stored in the gated block during sleep mode [5]. In this approach, since we have the power gates closer to the design blocks, the IR drop and routing distance to virtual $V_{dd}$ is shorter [4, 5, 6].

b. Ring of switches: In this approach, as the name indicates, power gated are placed around the perimeter of the module in a ring manner as shown in the figure 3 [17, 19].
This approach is mostly used when it is not expected to disturb the original design and there is no need to retain the data processed in the power gated block [5]. When compared with grid, ring of switches is less complex, but there is an increase in IR drop [5, 6].

There are a few factors that need to be considered when designing and placing the power gates, they are as follows:

1) **Leakage current**: Power gates are also transistors, so during sleep mode, the power is cut-off for the processing block/core, but the power gate is still connected to the power supply and it would continue to leak. Hence, the leakage of power gates should also be considered while designing and placing these power gates [5, 17].

2) **Rush current**: Power gates are used to cut-off power supply during inactive mode, but later the block should be powered-on again during active modes. The current drawn from the power gates when they are powered-on is called as rush current [5, 17]. This current is greater than the average current needed
for the normal operation because when the block is powered-off, all the capacitors in the design are discharged. So, when they are powered-on, all those capacitors try to pull maximum current from these switches to charge themselves. This rush current can damage the power switches and the power network as well. Hence it is advised to add more power gates to the design, so the rush current is divided between the parallel switches [17, 19].

3) **IR drop**: IR drop, or voltage droop is the voltage drop across the power switch when the circuit is powered-on and there is rush current. Adding multiple parallel power switch also reduces the impact of the IR drop [5, 17]. The placement of the power gates should ensure to minimize the IR drop across the entire core/design.

4) **Ramp-up time**: This is the time taken by the block to get functional after the sleep mode, once it is powered-on. It is expected to have a minimum ramp-up time because power gating techniques reduces the power consumption during inactive mode, but it should not impact the performance of the block in any sense [17, 19]. The power network should be designed in such a way that the ramp up time is minimal [5].

5) **State retention flipflops**: By using power gate technique, the power consumption during idle mode is reduced, but when the block is powered-on again, there could be a need for the data that was previously computed/stored to be retained. Powering-off all the transistors will make them to lose the data stored in them. To overcome this issue, State Retention flipflops are used [5, 19]. For the data that must be saved and reused in the next processing cycle is stored in these state retention registers before power-off [4]. The data is written back to the actual flipflops when it is powered on again. This technique is mainly used for power gating memories, especially cache [5].
6) **Isolation cells:** Isolation cells must be used when few of the processing blocks are powered-off, but the rest are still in active state. Powering down a block wouldn’t immediately discharge all the capacitances in the block. In such situation, the output of this power gated block would be in undetermined state [3, 5]. If the same output is fed to other processing blocks, it would lead to an error. Hence it is always preferred to include an isolation cell at the end of the power gated block, so the floating output is not fed to other blocks [3, 5, 19].

**Related Work**
With power being a serious impediment in low power applications, there have been various techniques proposed to reduce the impact. This section discusses a few recent research ideas proposed in this domain. In AMD x86-64 core, which is implemented using 32 nm technology, has employed various low power techniques, such as clock gating, multiple threshold voltage ($V_t$) cells, and power gating [12]. Moving from 45 nm to 32 nm, flops used in AMD x86-64 core resulted in an increase in clock switching power by 80% [5, 12], usage of clock gating as an effective technique reduced the impact.

Usage of multiple threshold voltage cells also has proven to improve the performance and reduce the leakage power [3, 5, 12]. Low $V_t$ cells would be placed in the critical path to improve the performance, whereas high $V_t$ cells would be used in the non-critical path to reduce the leakage power as used in AMD x86-64 [5, 12].

Power gating to reduce power consumption during idle mode has many design criteria, such as leakage power and performance of power gates. There have been
various fine-grained power gating techniques proposed such as Sleepy NAND, and Dual Sleep NAND power gates [18]. The reduction in the leakage power is as shown in the table 2:

<table>
<thead>
<tr>
<th>Technique</th>
<th>Leakage Power (uW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleepy NAND</td>
<td>1.844</td>
</tr>
<tr>
<td>Dual Sleep NAND</td>
<td>1.785</td>
</tr>
</tbody>
</table>

The major issue with fine-grained power gating is the impact on the area, as each of the standard cell will be power gated and isolation becomes complex. Power gates can be used to either control the voltage ($V_{dd}$) or the ground ($V_{ss}$) supplied to the core [3, 5]. There have been few research papers who have tried gating both $V_{dd}$ and $V_{ss}$ and compared the performance [20]. Dual gating is a way to gate both the $V_{dd}$ and $V_{ss}$. All the three methods achieve >70% reduction in short-circuit and leakage power reduction with little overhead of less than 15% on the delay [5, 20] as mentioned in the figure 4.
Figure 4: Leakage Power reduction [20]

Figure 5: Power reduction based on gating [20]
The figure 5 indicates the effectiveness of the different power gating techniques and it is clear, the PMOS power gating has a better power reduction compared to NMOS and Dual power gating [3, 20]. Power gating has also seen to reduce the total active power by almost 2% in AMD x86-64, when $V_{ss}$ was gated [12]. The figure 6 indicates the effectiveness of power gating in all cores of AMD x86-64.

![Figure 6: Power reduction per core in AMD x86-64 [12]](image)

Along with different approaches for the usage of power gating, research focus has also been towards the effects of power gating on the design. It has been seen that, when the power gated block is powered-on, the power gating noise has seen to affect the circuits of non-gated blocks [21]. The figure 7, shows the noise scenarios due to power gating.

![Figure 7: Noise due to power gating [21]](image)
To reduce the impact of the noise generated due to power gating, incremental scheduling has proven to be an effective approach [21]. Awakening the power gated module in an incremental fashion or daisy chain fashion is a known way to reduce the rush current and the noise on the non-gated blocks [3, 5, 21] and the results are as shown in the figure 8.

In most of the cases, the power gates are controlled based on a pre-defined logic or the current workload of the core [3, 5]. But it has also been seen that, the sleep signal of power gates can be controlled based on the temperature and not based on the idle-cycle counter or comparator, as temperature plays an important role in deciding the energy saved due to power gating [5, 22, 23]. The figure 9 indicates the power saving based on the temperature.
Based on the analysis in figure 9, power gating was applied to DRPA, \( \text{MuCCRA-2.32b} \) and the impact of temperature on leakage power was studied and the results is as mentioned in the figure 10 [23].

![Figure 10: Impact of temperature on Leakage Power [23]](image)

Along with using a CMOS transistor itself as a header or a footer switch, there have been studies to use Nanoelectromechanical-Systems (NEMS) switch. These switches have near-infinite off-resistance and low on-resistance [5, 24]. Hence, when the block is powered-off, there is nearly no leakage power during idle mode. It is also seen that; these switches consume 30 times less power than CMOS power switches and reduces the area overhead by 36-83% [24]. The table 3 gives the details of simulating NEMS and CMOS switches for different processing blocks.
Table 3. Comparison between NEMS and CMOS header switch [24]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Adder</th>
<th>Shifter</th>
<th>Multiplier</th>
<th>Divider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay NEMS (ns)</td>
<td>2.79</td>
<td>1.89</td>
<td>45.2</td>
<td>17.0</td>
</tr>
<tr>
<td>Delay CMOS (ns)</td>
<td>8.02</td>
<td>17.2</td>
<td>119</td>
<td>328</td>
</tr>
<tr>
<td>Header Width (um)</td>
<td>NEMS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>8.9</td>
<td>47.9</td>
<td>28.2</td>
</tr>
<tr>
<td></td>
<td>CMOS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>92</td>
<td>250</td>
<td>285</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Boot-up energy (fJ)</th>
<th>NEMS</th>
<th>CMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>159</td>
<td>157</td>
<td>2040</td>
</tr>
<tr>
<td>1671</td>
<td>24907</td>
<td>13770</td>
</tr>
</tbody>
</table>

From the table 3, it might seem using NEMS switches as header is a good approach, but the impact is seen mostly during fabrication. Using CMOS transistor as header switches reduces additional fabrication steps [3, 5].

There have been many research papers indicating the effectiveness of power gating and there have been many ideas for fine-grained power gating as well [18, 20]. Other research papers focus on the $Vdd$ or $Vss$ gating and its effectiveness [23, 25]. But there hasn’t been any clear literature to indicate which style is more effective, grid or ring of switches, and what are the tradeoffs of using them. This thesis focusses specifically on identifying the pros and cons of using ring and grid pattern of power gates placement, specifically for fine-grained processor array.
CHAPTER 3: TOOLS AND LIBRARIES

Synopsys EDA tools and NanGate FreePDK45 standard cell library have been used to perform the design flow. The details of all the tools used are as follows:

1) **Synopsys Library Compiler:** Library Compiler allows any standard cell libraries to be converted to the format compatible with Synopsys synthesis tools. The library compiler requires a library file (.lib) that describes all the logic gates necessary for synthesis operation. This file would be compiled into a database format (.db) which is suitable for Synopsys tool usage [26].

2) **Synopsys Design Compiler:** Design Compiler is an essential tool for Synopsys synthesis products. Design Compiler provides the smallest and fastest logical representation of any given function by optimizing the design. Design Compiler synthesizes the design into an optimized technology-dependent, gate-level design. This tool can be used to optimize both sequential and combination circuits for the area, power, and delay [27].

3) **Synopsys Milkyway Environment:** Milkyway is a Synopsys library format that stores all the circuit files from synthesis through place and route. This is a separate, standalone tool for preparing physical libraries from the layout data provided by an outside source [28].

4) **Synopsys IC Compiler:** Transforming a gate-level netlist into a placed and routed layout is performed using Synopsys IC (Integrated Circuit) compiler. IC Compiler is used for design planning, placement, clock tree synthesis, routing and optimization of designs [29].

IC compiler places standard cells in rows and routes all the nets between these cells automatically. It also routes the power and ground rails and
connects them to the power and ground pins of the standard cells. IC compiler also generates the clock tree and distributes the clock to all the sequential elements of the design [28]. Along with placing, routing, and clock tree synthesis, IC compiler also optimizes the design for effective placement and routing of the design to minimize the area, delay, and power [29].

5) **HSPICE**: HSPICE (Hailey Simulation Program with Integrated Circuit Emphasis) is used to check the integrity of circuits by performing the transient, frequency domain, and steady state analysis of the circuits to predict its behavior [30]. HSPICE is known for accurate circuit simulation, where it uses Monte Carlo, parametric sweep, worst-case, and data-table sweep analyses for analyzing the performance and yield [28, 30].

6) **Synopsys Custom Compiler**: Custom Compiler is Synopsys full-custom solution tool for custom design tasks, both analog and digital [31]. The tool can be used to draw the layout and schematic of any custom module. The tool also provides facilities to perform LVS (Layout versus Schematic), HSPICE simulation, parasitic extraction, and a means to export the designed cell to IC compiler, which can now be used as a standard cell [31].

**NanGate FreePDK45**

The NanGate FreePDK45 open cell library is an open-source, standard cell-library provided by a privately held US/Silicon Valley-based multinational corporation, for testing and exploring EDA flows [32]. NanGate FreePDK45 is a 45nm Open Cell Library (OCL) kit, that supplies technology files, design rules and scripts for a generic 45-nanometer process [32]. The 45nm OCL kit provides three different delay models, the details of which are as follows:
1) **NLDM (Non-Linear Delay Model):** This is a voltage-based model that relies on the input signal slope, and output capacitive loads. This model has been valid only for older technology nodes, as the resistive and inductive effects in interconnection wiring had been neglected [32].

2) **ECSM (Effective Current Source Model):** This is a current based model, which considers the non-linear transistor switching behaviors, allowing the accurate modeling of interconnections. In this model, the product of the supply voltage and the average current flowing out from $V_{dd}$ is calculated as the leakage power [32].

3) **CCS (Composite Current Source):** This is also a current source model, which is like the ECSM model, but this can be used for noise and power analysis, besides timing. The dynamic current for this model is calculated using transient analysis, and leakage power is the leakage current measured using simple DC analysis [32].

Each of these models has five different characterization corners as follows, the first three are based on performance [32].

- Slow
- Typical
- Fast
- Low temperature
- Worst case low temperature
CHAPTER 4: METHODOLOGY

Library preparation for NanGate FreePDK45

This section includes details regarding NanGate FreePDK45 libraries and the procedure followed to make it compatible with Synopsys EDA tools. The 45-nanometer standard cell library package received from the vendor included the below list of files [32]:

- NanGate Library files (.lib)
- NanGate Library Exchange Format (.lef) files
- NanGate Verilog (.v) file
- NanGate Graphic Data System II (GDSII) file
- NanGate Spice (.spi) files

To have these files compatible with Synopsys EDA tools, the following procedure was followed:

1) **Design Compiler:** Synopsys Design Compiler tool expected a database (.db) file of the NanGate FreePDK45, which included details regarding every standard cell [26]. The same information was available in the NanGate Library (.lib) file [32]. Synopsys Library Compiler was used to convert the library file to database file. The commands used for this conversion is mentioned in Appendix A.

2) **IC compiler:** Synopsys IC compiler required the developer to provide the details of the standard cells in a specific format known as Milkyway, along with the necessary technology (.tf) file [29]. Using Synopsys Milkyway environment, the NanGate lef files were used to generate the required files for Synopsys IC compiler [33]. The tcl script used for this conversion is mentioned in Appendix A.
Along with the above-mentioned files, Synopsys IC compiler also expected the developer to provide the RC coefficients in a specific format, known as Table Lookup (TLUPlus) files, which was essential for parasitic extraction [29]. The information needed for this file was manually created using the NanGate CAP tbl file as reference [32]. The created file and the commands used for conversion can be found in Appendix B.

**Designing 128-bit MAC unit**

A 128-bit MAC unit was designed to use as a fine-grained processor array. The Verilog code designed for the work is as mentioned in Appendix C. Along with the proposed design, Vedic multiplication technique was tested [34]. The Vedic multiplication technique did reduce the latency [34] when compared to the base design used, but Vedic multiplication uses parallel paths to compute the result, which requires more die area. Parallel paths in the design could improve the overall performance, but also increases the leakage power in a greater extent [3, 5]. Due to these reasons, the code mentioned in Appendix C has been used and tested using Altera MultiSim. The figure 11 shows the design of the 128-bit MAC unit.
Once the base design was simulated and tested to ensure proper functionality, the next step to complete the front-end flow was to synthesize the tested code. Synopsys Design Compiler tool was used to synthesize the 128-bit MAC unit using NanGate FreePDK45 as the standard cell library. The tool can be launched using the command `dc_shell -gui` [27]. The script used to load and test the design is mentioned in Appendix D. The final output of the tool would be a gate level netlist of the design. The figure 12 and 13 show the synthesized 128-bit MAC unit.
Figure 12. Block diagram of the 128-bit MAC unit in Design Compiler

Figure 13. Detailed circuit diagram in Design Compiler
Place and Route Physical Design

The next step in the ASIC process was to use the netlist generated using Design Compiler and then perform place and route, which is the back-end process [29]. Along with the netlist, the files which were mentioned in Chapter 3 should be given as inputs to Synopsys IC compiler to perform the physical design.

In this process, along with IC compiler, IEEE 1801-2013 Unified Power Format (UPF) has been used [35]. This is a standard way to define a low-power design intent in EDA tools. This format provides all the necessary supply network, switches, retention registers, and isolation aspects necessary for low power design methodology [35]. This IEEE standard specifies the relationship between the 128-bit MAC unit design (Verilog code) and the low-power design specification. The UPF and IC compiler tcl script is as mentioned in Appendix E.

Below mentioned are few important steps and corresponding results captured while performing the placement and routing of power gates in both ring and grid fashion, the steps are:

- The gate-level netlist generated from Design Compiler along with NanGate FreePDK45 related files necessary for Place and Route, the ones mentioned in Chapter 3, should be given as inputs to Synopsys IC compiler.
- Now the core area on which all the standard cells would be placed should be designed. The core utilization set for this work is 85%.
- All low power methodologies use UPF [35] for creating voltage domains, setting nets, mapping ports, and creating power switch.
• The next step in the process would be to create a voltage area, which would be used to place the power gates in either ring or grid fashion. The created voltage area along with the $V_{dd}$ and $V_{ss}$ rails is as shown in the figure 14. A closer look at the placed $V_{dd}$ and $V_{ss}$ rails can be seen in figure 15.

![Figure 14. Voltage domain](image1)

![Figure 15. Voltage rails](image2)
Now the power gates must be placed in the voltage domain. As mentioned earlier, this thesis uses both grid and ring fashion for placing the power gates. The figure 16 and figure 17 indicate the placement of the power gates in grid and ring manner respectively.

Figure 16. Power gates placed in grid manner
Once the power gates are placed in the required fashion, the next step would be to place the power grid network and synthesize it to understand the region with maximum IR drop. The figure 18 shows the synthesized power network for this thesis.
In the figure 18, the regions are highlighted with different colors based on the IR drop. The center of the network which has the maximum IR drop is highlighted in red color, indicating care should be taken while routing to this part of the network [29].

The synthesized power network must be committed and the figure 19 indicates the final power network grid used for routing the $V_{dd}$ and $V_{ss}$. 
The next step would be to place all the standard cells in an optimized way in the created voltage domain. The figure 20 shows the placement of the standard cells in the design.
• After the standard cells are placed. The power grid rails must be routed to the actual power rails $V_{dd}$ and $V_{ss}$. The figure 21, shows the routing of the power grid rails to Vdd and Vss.
After the standard cells are placed, the next step is to perform the Clock Tree Synthesis (CTS). CTS is performed to evenly distribute the clock signal to all the sequential components of the design [29]. It is important to ensure the skew and latency is minimal and there aren’t any setup and hold time violations during the CTS process [29]. The figure 22 shows the CTS result of the design.
• In the figure 22, the paths highlighted in green are the routed clock signals to the sequential elements.

• The final step of the physical design flow is to perform the routing. After the CTS and standard cell placement, IC compiler has all necessary information about the logic blocks, their pins, and the pins at the boundary. In this stage, the tool uses metals and vias to create electrical connection between the logic blocks as mentioned in the gate-level netlist [29]. The figure 23 is the final design with route optimized and figure 24 gives a closer view of the routing.
Figure 23. Final routed design

Figure 24. Detailed routing
Header design in Custom Compiler

The power gates used for the IC compiler place and route was provided by NanGate [32]. There were three version of the power gates, HEADER_X1, HEADER_X2, HEADER_X4 [32]. The width of the channel between the drain and source terminal was the factor that differentiated these three power gates. With the variation in the width of the transistor, the leakage power of these power gates also varied [32].

The power gate HEADER_X2 had the width of the channel twice, when compared to HEADER_X1 and HEADER_X4 had the width four times of HEADER_X1 [32]. The schematic of these power gates was viewed using Synopsys Custom Compiler. The figures 25, 26, and 27 are the layout view of the power gates HEADER_X1, HEADER_X2, and HEADER_X4 [32] respectively.

Figure 25. Power gate HEADER_X1
Figure 26. Power gate HEADER_X2

Figure 27. Power gate HEADER_X4
In the figures 25, 26, and 27, the width is indicated by the light blue color and it is increasing according to the width of the transistor. The power gates with the terminals defined is as shown in figure 28.

![Power gate with terminals defined](image)

Figure 28. Power gate with terminals defined

**HSPICE simulation of Power gates**

The characteristics of the power gates was tested using HSPICE [30]. Among all the characteristics of the power gates, leakage power was of primary concern for this thesis. The HSPICE deck used for testing is as mentioned in Appendix F [29]. The details of the leakage power of these power gates is as mentioned in the table 4. The cell area of all the three power gates was 0.532 um² [32].

<table>
<thead>
<tr>
<th>Power gate</th>
<th>Leakage Power (nW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEADER_X1</td>
<td>1.9472</td>
</tr>
<tr>
<td>HEADER_X2</td>
<td>3.4727</td>
</tr>
<tr>
<td>HEADER_X4</td>
<td>6.5240</td>
</tr>
</tbody>
</table>
CHAPTER 5: RESULTS AND ANALYSIS

ModelSim Simulation

The 128-bit MAC unit designed was tested using ModelSim to ensure the functionality was intact. The results in the figure 29 show the working of the 128-bit MAC unit.

![ModelSim simulation](image)

**Figure 29. ModelSim simulation**

From the figure 29, the 128-bit MAC unit accumulates the value on every rising edge of the clock as expected.

Place and Route

The simulated 128-bit MAC unit was synthesized, and the generated netlist was used in IC compiler for place and route. The total cell count of the 128-bit MAC unit is as mentioned in the figure 30.
Figure 30. 128-bit MAC unit cell count

The total area of the design is as mentioned in the figure 31.

Figure 31. 128-bit MAC unit total area

The clock period used for this 128-bit MAC unit was 14 ns, hence the frequency of operation is 71.42 MHz. The MAC unit was tested with clock period less than 14 ns, but it resulted in setup time violation [3], as a result the base clock frequency of this design was set to 71.42 MHz. The timing report of the design is as mentioned in the figure 32.
From the figure 32, the slack is 0.09 ns, which is a positive value, indicating the clock period was sufficient for the critical path to execute. The detailed timing report can be seen in Appendix G. The target utilization was set to 85%, with optimization, the utilization of the die area was 93.32%. The figure 33 in the utilization report for the 128-bit MAC unit.

Figure 32. 128-bit MAC timing report
Figure 33. 128-bit MAC utilization report

The dynamic and leakage power of the MAC unit is as mentioned in the figure 34. The IC compiler uses random activity factor to determine the dynamic power [29].
Along with the reports, it was also necessary to understand the routing congestion of the design. The congestion of the 128-bit MAC unit was found to be 0.37% and the detailed report is as mentioned in Appendix H.

With the above set of results, the next step was to reduce the power consumption during idle mode by placing the power gates in grid and ring manner. The main
intention of this thesis was to study the pros and cons of these styles and its impact on the power during idle mode. The table 5 gives the details obtained by placing the power gates in grid manner. It also includes the details for the three versions of power gates, i.e., HEADER_X1, HEADER_X2, and HEADER_X4 which was used to study the impact on using varied sized power gates.

Table 5. Impact of power gates placed in grid manner

<table>
<thead>
<tr>
<th>Parameter</th>
<th>HEADER_X1</th>
<th>HEADER_X2</th>
<th>HEADER_X4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of gates</td>
<td>182</td>
<td>95</td>
<td>52</td>
</tr>
<tr>
<td>Utilization</td>
<td>93.56%</td>
<td>93.60%</td>
<td>93.37%</td>
</tr>
<tr>
<td>Area</td>
<td>62442.968 um²</td>
<td>62313.176 um²</td>
<td>62284.698 um²</td>
</tr>
<tr>
<td>Clock Time Period</td>
<td>14 ns</td>
<td>14 ns</td>
<td>14 ns</td>
</tr>
<tr>
<td>Leakage power</td>
<td>354.39 nW</td>
<td>329.90 nW</td>
<td>339.24 nW</td>
</tr>
<tr>
<td>Routing congestion</td>
<td>3.71%</td>
<td>3.94%</td>
<td>3.78%</td>
</tr>
</tbody>
</table>

In the table 5, the leakage power was measured during idle mode, when the power gates had removed the supply voltage to the 128-bit MAC unit.

With increase in the width of the power gates, the total number of power gates used has been reduced by half. Since the area of the power gates HEADER_X1, HEADER_X2, and HEADER_X4 was 0.532 um² [32]. The overall impact on the area is also reduced with increase in the size of the power gate.

There was negligible impact on the critical path timing and utilization with the power gates. The small variation in the utilization is due to optimization randomness of the Synopsys IC compiler [29].
Even though the total number of gates have been reduced by half with increase in the width of the power gate, the leakage power during idle mode is almost the same. The reason being, with increase in the width of the power gates, through HSPICE simulation it was seen that, the leakage power of the individual gate also increased, the leakage power is as mentioned in table 4. Hence, it was an optimized approach to reduce the number of power gates with increase in the width.

By adding power gates to the design, the overall routing congestion has been increased from 0.37% to an average of 3.81% due to additional routing of the power gates and connecting the standard cells to the virtual power supply.

A pictorial representation of the above analysis is as shown in the figure 35.

Figure 35. Impact on area and power with varied sized power gates placed in grid manner
The same exercise was performed again, but the power gates were placed in a ring fashion, around the periphery of the core. The table 6 gives the details of the results obtained when the power gates HEADER_X1, HEADER_X2, and HEADER_X4 were placed in ring fashion.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>HEADER_X1</th>
<th>HEADER_X2</th>
<th>HEADER_X4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of gates</td>
<td>178</td>
<td>94</td>
<td>56</td>
</tr>
<tr>
<td>Utilization</td>
<td>93.35%</td>
<td>93.09%</td>
<td>93.42%</td>
</tr>
<tr>
<td>Area</td>
<td>62438.628 um²</td>
<td>62401.247 um²</td>
<td>62379.528 um²</td>
</tr>
<tr>
<td>Clock Time Period</td>
<td>14 ns</td>
<td>14 ns</td>
<td>14 ns</td>
</tr>
<tr>
<td>Leakage power</td>
<td>346.601 nW</td>
<td>326.438 nW</td>
<td>365.344 nW</td>
</tr>
<tr>
<td>Routing congestion</td>
<td>3.45%</td>
<td>3.61%</td>
<td>3.40%</td>
</tr>
</tbody>
</table>

From the table 6, it can be inferred that, the number of gates used depends on the size of the power gate, as the width of the power gate increases, the number of gates is reduced. The target utilization of the design was 85%, with the power gates placed in ring fashion, there is negligible impact on the utilization.

The total area has reduced with increase in the width of the power gates. The reason being, the area of the power gate is 0.532 um² [32] for all three versions, hence the impact on area is reduced with increase in the width of the power gates. Adding power gates to the design had negligible impact on the critical path of the design.

The power dissipation of the design during idle mode is due to the leakage power of the power gates. With increase in the width of the power gate, the leakage also
increases. But the overall leakage is almost the same for all the power gates because the total number of power gates have also reduced with increase in the width. The average congestion is 3.48% with the power gates placed in the ring fashion.

The data in table 6 is plotted in the figure 36.

![Power gates placed in the ring manner](image)

**Figure 36.** Impact on area and power with varied sized power gates placed in ring manner

Looking at the figures 35 and 36, it can be noted that, whether the power gates are placed in ring or grid pattern, the impact on the area, and leakage power is almost the same. The figure 37 is a plot of the idle mode leakage power for ring and grid for all the three version of power gates.
Along with the above-mentioned parameters, the voltage droop and the rush current were the other two parameters that was noted during the simulation, and they are as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Grid</th>
<th>Ring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage supplied</td>
<td>1.0839 V</td>
<td>0.9774 V</td>
</tr>
<tr>
<td>Current drawn</td>
<td>28.499 mA</td>
<td>28.002 mA</td>
</tr>
</tbody>
</table>

As mentioned in the Chapter 2, the two important parameters that should be considered while designing the power gates and the placement was the voltage droop and rush current. The voltage supplied to the power rails ($Vdd$) was 1.25 V.
The power gates would supply a virtual voltage, $VVdd$ to the design and as seen in the table 7, in grid style, the IR drop was found to be 0.1661 V, hence supplying a $VVdd$ of 1.0839 V to the design. Whereas in ring style, the IR drop was 0.2726 V, supplying a voltage of 0.9774 V to the design. This indicates that, when the power gates are placed around the perimeter of the design, there is more IR drop due to long distance power routing. To mitigate this issue and to supply higher voltage to the design, the number of power gates had to be increased, indicating more area and more leakage power during idle mode.

Taking a closer look at the congestion incurred due to the routing, when compared to the base design, both ring and grid increase the congestion of routing. The figure 38 is a plot of the Global Route Congestion (GRC) for every metal layer for the base 128-bit MAC unit.
The same report was generated and plotted while placing the power gates in ring and grid manner, the results are as shown in figure 39 and 40 respectively.

Figure 39. Congestion report for ring pattern power gates placement
From the figures 39 and 40, it is evident that, adding power gates to the design increases the routing congestion (Global Routing Congestion), but when compared to ring, grid pattern adds more congestion, almost by 8-9%. Congestion in routing indicates that the number of available tracks in an area for routing is less than the required number of tracks. Increase in routing congestion, could possibly lead to an increase in the area, power consumption and critical path delay [36]. Adding power gates to a design, adds the need for additional routing of the VVdd from the power gates to the standard cells, along with routing the power supply to the power gates. If the routing congestion is too severe, the design can even be un-routable.
CHAPTER 6: CONCLUSION AND FUTURE WORK

Conclusion

This thesis provides a detailed analysis and comparison of tradeoffs between using the ring and grid methods of power gates placement. Using power gates reduces the leakage power consumption during idle mode by 99.98%, as the power supply to the design is cut-off.

Based on the results obtained, it is evident that, with increase in the width of the power gate by 4×, the total number of power gates needed to turn-on a design reduces by 70%. This also reduces the die area by 0.12%, when compared with using power gates of default transistor width 1×.

The only leakage power during idle mode is due to the power gates, as they are still connected to the supply voltage, which accounts to 0.02% of the leakage power consumed by the design without power gates. With increase in the width of the power gate, the number of power gates needed is reduced. But reduction in the number of the power gates, doesn’t have a significant impact on the leakage power consumption during idle mode because wider the transistor, more is the leakage. If the width of the power gate is 4×, the leakage power per transistor increases by 29.85% when compared to 1× power gates. Adding power gates has negligible impact on the utilization and critical path timing of the design.

The above analogy holds true when the power gates are placed in ring and grid pattern. The two parameters that make the difference is the voltage droop and the congestion. From the results, it was seen that, placing the power gates in grid
methodology, created an 8.52% less IR drop than ring methodology. This is due to the long route taken by the power to reach the standard cell. In ring methodology, the required voltage can be supplied to the design but with more number of power gates when compared to grid. This in turn increases the area occupied and the leakage power during idle mode. Power gates placed in grid methodology created an 8-9% congestion increase when compared to the ring methodology.

Apart from the above analysis, it was also noted that, when power gates are placed in ring manner, it was not possible to retain the data computed/stored in the design during power-off mode [3, 5, 19]. But in grid, state retention registers can be used to retain the data before the idle mode [5].

**Future Work**

In this thesis, the power gates used for testing were standard headers by NanGate. It would be worth to see the impact of using wider transistors and verifying if the impact is still the same as mentioned in this report. With increasing the width of the transistor by 4×, the number of power gates used for this design was reduced by 70% and the impact on area was reduced by 0.12%. Along with reducing the number of power gates needed for a design, further increase in the width of the power gate could possibly reduce the routing congestion as well.

The low-power approach used in this work is power gating which can be used during standby mode. However, it would be interesting to explore the other low-power architectures such as Dynamic Voltage Frequency Scaling (DVFS), which would be an optimization technique during runtime [5, 14, 15]. And the body biasing technique, which can be applied during both runtime and standby [3, 5, 13].
Network-on-chip (NoC) architectures have dawned as a favorable solution for the forthcoming system-on-chip communication architecture designs. However, the routing algorithms are confronted with great challenges such as increased congestion in the center of the network. Encircle Routing (ER) algorithm has proven to reduce the congestion, but at the cost of increase in the power consumption due to the longer route [38, 39, 40]. Using the power gating technique to such system could help to combat the increase in power consumption and still make it an effective routing algorithm.
REFERENCES


[34] Y. Bansal, C. Madhu and P. Kaur, "High speed vedic multiplier designs-A review," 2014 Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, 2014, pp. 1-6


APPENDICES
APPENDIX A: LIBRARY COMPILER AND MILKYWAY
1) **Library Compiler Commands:** The below commands were used to create the required database file for Synopsys EDA tools.

```
#### Read the Vendor library file
read_lib NangateOpenCellLibrary.lib

#### Write the output file in .db format compatible for Synopsys tools
write_lib NangateOpenCellLibrary -format db -output NangateOpenCellLibrary.db
```

2) **Milkyway Commands:** The below tcl script [33] created the Milkyway database files for Synopsys EDA tools.

```
### Read the lef files of the vendor
read_lef -lib_name Nan45 -tech_lef_files NangateOpenCellLibrary.tech.lef -cell_lef_files "NangateOpenCellLibrary.macro.lef"

### Create the required Milkyway files
cmDumpTech
setFormField "Dump Technology File" "Library Name" "Nan45"
setFormField "Dump Technology File" "Technology File Name" "Nan45.tf"
formOK "Dump Technology File"
exit
```
APPENDIX B: TLUPLUS FILE
1) **Interconnect Technology File (ITF):** The below file was manually created and has all the information of the thickness and resistance of every metal layer, thickness and dielectric constant of all dielectrics.

```
TECHNOLOGY=Nangate45nm
DIELECTRIC metal10_diel_b {THICKNESS=2.000000 ER=2.500000}
DIELECTRIC metal10_diel_a {THICKNESS=2.000000 ER=2.500000}
CONDUCTOR M10 {THICKNESS=2.00000 WMIN=0.800000 SMIN=0.80000 RPSQ=0.030000}
DIELECTRIC metal9_diel_b {THICKNESS=2.000000 ER=2.500000}
DIELECTRIC metal9_diel_a {THICKNESS=2.000000 ER=2.500000}
CONDUCTOR M9 {THICKNESS=2.00000 WMIN=0.800000 SMIN=0.80000 RPSQ=0.030000}
DIELECTRIC metal8_diel_b {THICKNESS=0.820000 ER=2.500000}
DIELECTRIC metal8_diel_a {THICKNESS=0.800000 ER=2.500000}
CONDUCTOR M8 {THICKNESS=0.80000 WMIN=0.40000 SMIN=0.40000 RPSQ=0.075000}
DIELECTRIC metal7_diel_b {THICKNESS=0.820000 ER=2.500000}
DIELECTRIC metal7_diel_a {THICKNESS=0.800000 ER=2.500000}
CONDUCTOR M7 {THICKNESS=0.80000 WMIN=0.40000 SMIN=0.40000 RPSQ=0.075000}
DIELECTRIC metal6_diel_b {THICKNESS=0.290000 ER=2.500000}
DIELECTRIC metal6_diel_a {THICKNESS=0.280000 ER=2.500000}
CONDUCTOR M6 {THICKNESS=0.28000 WMIN=0.14000 SMIN=0.14000 RPSQ=0.210000}
DIELECTRIC metal5_diel_b {THICKNESS=0.290000 ER=2.500000}
DIELECTRIC metal5_diel_a {THICKNESS=0.280000 ER=2.500000}
CONDUCTOR M5 {THICKNESS=0.28000 WMIN=0.14000 SMIN=0.14000 RPSQ=0.210000}
DIELECTRIC metal4_diel_b {THICKNESS=0.290000 ER=2.500000}
DIELECTRIC metal4_diel_a {THICKNESS=0.280000 ER=2.500000}
CONDUCTOR M4 {THICKNESS=0.28000 WMIN=0.14000 SMIN=0.14000 RPSQ=0.210000}
DIELECTRIC metal3_diel_b {THICKNESS=0.120000 ER=2.500000}
DIELECTRIC metal3_diel_a {THICKNESS=0.140000 ER=2.500000}
CONDUCTOR M3 {THICKNESS=0.14000 WMIN=0.07000 SMIN=0.07000 RPSQ=0.250000}
DIELECTRIC metal2_diel_b {THICKNESS=0.120000 ER=2.500000}
DIELECTRIC metal2_diel_a {THICKNESS=0.140000 ER=2.500000}
```
CONDUCTOR M2 \{THICKNESS=0.14000 WMIN=0.07000 SMIN=0.07000 RPSQ=0.25000\}
DIELECTRIC metal1_diel_b \{THICKNESS=0.120000 ER=2.500000\}
DIELECTRIC metal1_diel_a \{THICKNESS=0.130000 ER=2.500000\}
CONDUCTOR M1 \{THICKNESS=0.13000 WMIN=0.07000 SMIN=0.06500 RPSQ=0.38000\}
DIELECTRIC poly_diel_b \{THICKNESS=0.085000 ER=2.500000\}
DIELECTRIC poly_diel_a \{THICKNESS=0.085000 ER=2.500000\}
CONDUCTOR POLY \{THICKNESS=0.085000 WMIN=0.05000 SMIN=0.07500 RPSQ=7.800000\}
DIELECTRIC field_active_diel \{THICKNESS=0.100000 ER=3.900000\}
DIELECTRIC field_base_diel \{THICKNESS=0.100000 ER=3.900000\}
VIA contact \{FROM=M1 TO=POLY RPV=25.00000\}
VIA via1 \{FROM=M2 TO=M1 RPV=5.00000\}
VIA via2 \{FROM=M3 TO=M2 RPV=5.00000\}
VIA via3 \{FROM=M4 TO=M3 RPV=5.00000\}
VIA via4 \{FROM=M5 TO=M4 RPV=3.00000\}
VIA via5 \{FROM=M6 TO=M5 RPV=3.00000\}
VIA via6 \{FROM=M7 TO=M6 RPV=3.00000\}
VIA via7 \{FROM=M8 TO=M7 RPV=1.00000\}
VIA via8 \{FROM=M9 TO=M8 RPV=1.00000\}
VIA via9 \{FROM=M10 TO=M9 RPV=0.50000\}

2) TLUplus file creation from ITF: The below command was used to create the file needed for RC extraction of IC compiler.

```
grdgenxo -itf2TLUPLUS -i Nangate_45.itf -o Nangate_max.tluplus
```
APPENDIX C: VERILOG CODE
The 128-bit MAC unit Verilog code is as follows:

```verilog
module mac_unit_128(data_a, data_b, data_out, clk, reset, cntl);
//Input variable declaration
input [127:0] data_a, data_b;
input clk, reset, cntl;
//Output variable declaration
output [255:0] data_out;
reg [255:0] data_out;
reg [255:0] mult_out;

always @(posedge clk)
begin
  if(!cntl)
    begin
      if (reset) //Reset the output
        begin
          data_out = 128'b0;
        end
      else
        begin
          mult_out = (data_a * data_b); //Product of the input variables
          data_out = data_out + mult_out; //Sum getting accumulated
        end
    end
endmodule
```
APPENDIX D: DESIGN COMPILER SCRIPT
The commands used for Design Compiler is as follows [37]:

###Setting NanGate file as the target and link library
set link_library [set target_library [concat [list NangateOpenCellLibrary.db LowPowerOpenCellLibrary.db] [list dw.Foundation.sldb]]]

###Reading the 128-bit MAC unit Verilog code
read_file -format verilog
{/home/fresnostate/Desktop/Nan45/ICComplier/Header_testing/low_power_lib/mac_unit_128.v
/home/fresnostate/Desktop/Nan45/ICComplier/Header_testing/low_power_lib/mac_unit_final.v}

###Using UPF to create the power domain, nets, ports, power switch and state table
create_power_domain top
create_supply_port VDD
create_supply_port VSS
create_supply_net VDD
create_supply_net VSS
connect_supply_net VDD -ports VDD
connect_supply_net VSS -ports VSS
create_power_domain power_gated -elements dut
create_supply_net VDD -domain power_gated -reuse
create_supply_net VSS -domain power_gated -reuse
create_supply_net VVDD -domain power_gated
create_power_switch power_gate -domain power_gated -input_supply_port {in VDD} -output_supply_port {out VVDD} -control_port {sleep cntl} -on_state [list on_state in {!sleep}] -off_state [list off_state {sleep}]
map_power_switch power_gate -domain power_gated -lib_cell HEADER_X1
add_port_state VDD -state {ON 1.25}
add_port_state VSS -state {GND 0}
add_port_state power_gate/out -state {ON 1.25} -state {OFF off}
create_pst power_top -supplies {VDD power_gate/out}
add_pst_state function1 -pst power_top -state {ON ON}
add_pst_state ssleep -pst power_top -state {ON OFF}

### Setting the voltage values
set_voltage 1.25 -obj {VDD}
set_voltage 1.25 -obj {top.primary.power}
set_voltage 1.25 -obj {power_gated.primary.power}
set_voltage 1.25 -obj {VVDD}
set_voltage 0.00 -obj {VSS}
set_voltage 0.00 -obj {top.primary.ground}
set_voltage 0.00 -obj {power_gated.primary.ground}

### Creating the clock signal
create_clock -period 16 -name design_clk clk
link
check_design
compile_ultra

### Writing out the netlist and UPF file after successful compilation
write -hierarchy -format verilog -output
/home/fresnostate/Desktop/Nan45/ICComplier/Header_testing/low_power_lib/mac_unit_128_rtl.v
save_upf
/home/fresnostate/Desktop/Nan45/ICComplier/Header_testing/low_power_lib/mac_unit_128.upf
The IC compiler commands used for the work is as follows [37]:

###Setting NanGate file as the target and link library
set link_library [set target_library [concat [list NangateOpenCellLibrary.db LowPowerOpenCellLibrary.db] [list dw_foundation.sldb]]]
set target_library "NangateOpenCellLibrary.db LowPowerOpenCellLibrary.db"

###Creating Milkyway library files for IC compiler
create_mw_lib -technology
/home/fresnostate/Desktop/Nan45/ICComplier/Header_testing/low_power_lib/Nan45.tf -mw_reference_library
{/home/fresnostate/Desktop/Nan45/ICComplier/Header_testing/low_power_lib/low45}
/home/fresnostate/Desktop/Nan45/ICComplier/Header_testing/low_power_lib/low45low} -bus_naming_style {{[%d]}}
/home/fresnostate/Desktop/Nan45/ICComplier/Header_testing/low_power_lib/head_test
open_mw_lib head_test
check_tlu_plus_files

###Providing the file created in Appendix B as an input
set_tlu_plus_files -max_tluplus
/home/fresnostate/Desktop/Nan45/ICComplier/Header_testing/low_power_lib/NanGate_max.tluplus -tech2itf_map
/home/fresnostate/Desktop/Nan45/ICComplier/Header_testing/low_power_lib/NanGate_45.map
open_mw_lib head_test
check_tlu_plus_files

###Reading the netlist generated by Design Compiler
read_verilog
{/home/fresnostate/Desktop/Nan45/ICComplier/Header_testing/low_power_lib/mac_unit_128_rtl.v}
load_upf
/home/fresnostate/Desktop/Nan45/ICComplier/Header_testing/low_power_lib/mac_unit_128.upf

###Setting the required values for UPF
set_domain_supply_net power_gated -primary_power_net VDD -
primary_ground_net VSS
set_domain_supply_net top -primary_power_net VDD -primary_ground_net VSS
derive_pg_connection -create_nets
derive_pg_connection -power_net VVDD -power_pin {VDD} -cells "dut" -verbose
derive_pg_connection -power_net VDD -power_pin {VDD}
derive_pg_connection -ground_net VSS -ground_pin {VSS}

###Creating the floor plan and voltage area for the design
create_floorplan -start_first_row -flip_first_row -left_io2core 10 -bottom_io2core 10 -right_io2core 10 -top_io2core 10 -core_utilization 0.85
create_voltage_area -coordinate { 11.9700 11.0050 267.7750 265.6900 } -power_domain power_gated -cycle_color

set_voltage 1.25 -obj {VDD}
set_voltage 0.00 -obj {VSS}
set_voltage 1.25 -obj {VVDD}

###Mapping the power switch to NanGate standard cell and creating the power rails
map_power_switch -domain power_gated -lib_cell HEADER_X4 power_gate
create_rectangular_rings -nets {VDD VSS} -left_offset 1 -left_segment_layer metal6 -left_segment_width 3 -right_offset 1 -right_segment_layer metal6 -right_segment_width 3 -bottom_offset 1 -bottom_segment_layer metal5 -bottom_segment_width 3 -top_offset 1 -top_segment_layer metal5 -top_segment_width 3

###Creating the clock and placing the standard cells
create_clock -name clk -period 14 -waveform {0 7} [get_ports {clk}]
create_fp_placement

###Creating the power network
set_fp_rail_constraints -add_layer -layer metal6 -direction vertical -max_strap 10 -min_strap 2 -min_width 3 -spacing minimum
set_fp_rail_constraints -add_layer -layer metal5 -direction horizontal -max_strap 10 -min_strap 2 -min_width 3 -spacing minimum
set_fp_rail_constraints -set_ring -horizontal_ring_layer { metal5 } -vertical_ring_layer { metal6 } -ring_width 3 -extend_strap region_boundary
set_fp_rail_constraints -set_global -no_routing_over_hard_macros -no_routing_over_soft_macros
set_fp_rail_region_constraints -voltage_area {power_gated}
synthesize_fp_rail -nets {VVDD VSS} -voltage_supply 1.25
synthesize_power_plan -power_budget 1000 -use_strap_ends_as_pads
commit_fp_rail

### Placing power switches in ring methodology
create_power_switch_ring -area_object power_gated -switch_lib_cell power_gate
-prefix POWERGATE -x_increment 19 -y_increment 19 -same_orientation

### Placing power switches in grid methodology
add_header_footer_cell_array -lib_cell "power_gate" -voltage_area power_gated
-x_increment 70 -y_increment 20 -snap_to_row_and_tile -prefix POWERGATE
-respect all -pattern normal
connect_power_switch -source dut/ctl -port_name enable_port -voltage_area
power_gated

place_fp_pins -block_level
derive PG_connection -verbose
preroute_standard_cells -nets {VVDD} -
extend_to_boundaries_and_generate_pins
preroute_standard_cells -nets {VDD} -extend_to_boundaries_and_generate_pins
preroute_standard_cells -nets {VSS} -extend_to_boundaries_and_generate_pins
remove_stdcell_filler -stdcell

### Performing place, clock and route optimization
place_opt
derive PG_connection
preroute_instances
remove_clock_uncertainty [all_clocks]
set_fix_hold [all_clocks]
clock_opt -only_cts -no_clock_route
route_zrt_group -all_clock_nets -reuse_existing_global_route true
route_opt
APPENDIX F: HSPICE DECK
The HSPICE deck used for testing the power gates is as follows [36]:

*Setting the supply voltage
.glob VDD VSS
vSupply VDD 0 1.0V
vGround VSS 0 0V
temp 25

*Including the parasitic file for the header X1 and technology file for NanGate
.inc 'HEADER_X1.sp'
.include 'nangate_45nm.txt'

Xtst VDD VSS SLEEP VVDD SLEEPOUT HEADER_X1
.tran 1p 60ns
.options post=2

*Performing transient analysis
VSLEEP SLEEP VSS pulse (1 1 7.99ns 0.01ns 0.01ns 7.99ns 16ns)

.print TRAN V(SLEEP)
.print TRAN V(VVDD)
.print TRAN V(SLEEPOUT)

.measure dynpower avg power from 0n to 10n
.print dynpower

.measure tran leakage0 rms I(vSupply) from=1n to=60n
.print leakage0

.end
APPENDIX G: DETAILED TIMING REPORT
Report: timing
-path full
-delay max
-max_paths 1
Design: mac_unit_128
Version: M-2016.12
Date: Thu Apr 19 23:41:04 2018

Information: Some cells in the design are using inferred operating conditions.

* Some/all delay information is back-annotated.

Parasitic source: LPE
Parasitic mode: RealRC
Extraction mode: MAX
Extraction derating: 0/0/0

Information: Percent of Arnoldi-based delays = 99.65%

Information: Percent of CCS-based delays = 97.86%

Startpoint: dut/data_out_reg[25]
(rising edge-triggered flip-flop clocked by clk)
Endpoint: dut/data_out_reg[255]
(rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

<table>
<thead>
<tr>
<th>Point</th>
<th>Incr</th>
<th>Path</th>
<th>Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>clock clk (rise edge)</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>clock network delay (propagated)</td>
<td>0.57</td>
<td>0.57</td>
<td></td>
</tr>
<tr>
<td>dut/data_out_reg[25]/CK (DFF_X1)</td>
<td>0.00</td>
<td>0.57 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/data_out_reg[25]/Q (DFF_X1)</td>
<td>0.06 c</td>
<td>0.63 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U107/CO (FA_X1)</td>
<td>0.16 c</td>
<td>0.79 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U106/CO (FA_X1)</td>
<td>0.04 c</td>
<td>0.84 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U105/CO (FA_X1)</td>
<td>0.05 c</td>
<td>0.89 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U104/CO (FA_X1)</td>
<td>0.06 c</td>
<td>0.94 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U103/CO (FA_X1)</td>
<td>0.05 c</td>
<td>0.99 f</td>
<td>1.25</td>
</tr>
<tr>
<td>Description</td>
<td>Value1</td>
<td>Value2</td>
<td>Value3</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>dut/intadd_10/U102/CO (FA_X1)</td>
<td>0.05 c</td>
<td>1.04 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U101/CO (FA_X1)</td>
<td>0.04 c</td>
<td>1.08 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U100/CO (FA_X1)</td>
<td>0.05 c</td>
<td>1.13 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U99/CO (FA_X1)</td>
<td>0.04 c</td>
<td>1.17 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U98/CO (FA_X1)</td>
<td>0.05 c</td>
<td>1.22 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U97/CO (FA_X1)</td>
<td>0.05 c</td>
<td>1.27 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U96/CO (FA_X1)</td>
<td>0.05 c</td>
<td>1.32 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U95/CO (FA_X1)</td>
<td>0.05 c</td>
<td>1.37 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U94/CO (FA_X1)</td>
<td>0.06 c</td>
<td>1.43 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U93/CO (FA_X1)</td>
<td>0.05 c</td>
<td>1.48 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U92/CO (FA_X1)</td>
<td>0.05 c</td>
<td>1.52 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U91/CO (FA_X1)</td>
<td>0.05 c</td>
<td>1.57 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U90/CO (FA_X1)</td>
<td>0.05 c</td>
<td>1.62 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U89/CO (FA_X1)</td>
<td>0.05 c</td>
<td>1.67 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U88/CO (FA_X1)</td>
<td>0.06 c</td>
<td>1.73 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U87/CO (FA_X1)</td>
<td>0.06 c</td>
<td>1.78 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U86/CO (FA_X1)</td>
<td>0.06 c</td>
<td>1.84 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U85/CO (FA_X1)</td>
<td>0.05 c</td>
<td>1.89 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U84/CO (FA_X1)</td>
<td>0.05 c</td>
<td>1.94 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U83/CO (FA_X1)</td>
<td>0.04 c</td>
<td>1.98 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U82/CO (FA_X1)</td>
<td>0.06 c</td>
<td>2.04 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_10/U81/S (FA_X1)</td>
<td>0.05 c</td>
<td>2.09 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U82/CO (FA_X1)</td>
<td>0.18 c</td>
<td>2.27 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U81/CO (FA_X1)</td>
<td>0.06 c</td>
<td>2.33 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U80/CO (FA_X1)</td>
<td>0.06 c</td>
<td>2.38 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U79/CO (FA_X1)</td>
<td>0.05 c</td>
<td>2.43 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U78/CO (FA_X1)</td>
<td>0.05 c</td>
<td>2.48 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U77/CO (FA_X1)</td>
<td>0.05 c</td>
<td>2.52 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U76/CO (FA_X1)</td>
<td>0.06 c</td>
<td>2.58 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U75/CO (FA_X1)</td>
<td>0.04 c</td>
<td>2.62 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U74/CO (FA_X1)</td>
<td>0.04 c</td>
<td>2.66 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U73/CO (FA_X1)</td>
<td>0.05 c</td>
<td>2.71 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U72/CO (FA_X1)</td>
<td>0.05 c</td>
<td>2.76 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U71/CO (FA_X1)</td>
<td>0.05 c</td>
<td>2.81 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U70/CO (FA_X1)</td>
<td>0.06 c</td>
<td>2.86 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U69/CO (FA_X1)</td>
<td>0.04 c</td>
<td>2.90 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U68/CO (FA_X1)</td>
<td>0.06 c</td>
<td>2.96 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U67/CO (FA_X1)</td>
<td>0.04 c</td>
<td>3.00 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U66/CO (FA_X1)</td>
<td>0.06 c</td>
<td>3.06 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U65/CO (FA_X1)</td>
<td>0.04 c</td>
<td>3.10 f</td>
<td>1.25</td>
</tr>
<tr>
<td>Component</td>
<td>Delay (c)</td>
<td>f (f)</td>
<td>Margin (f)</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------</td>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>dut/intadd_7/U64/CO (FA_X1)</td>
<td>0.06</td>
<td>3.15</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U63/CO (FA_X1)</td>
<td>0.04</td>
<td>3.19</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U62/CO (FA_X1)</td>
<td>0.06</td>
<td>3.25</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U61/CO (FA_X1)</td>
<td>0.04</td>
<td>3.29</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U60/CO (FA_X1)</td>
<td>0.05</td>
<td>3.34</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U59/CO (FA_X1)</td>
<td>0.04</td>
<td>3.38</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U58/CO (FA_X1)</td>
<td>0.04</td>
<td>3.43</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U57/CO (FA_X1)</td>
<td>0.04</td>
<td>3.47</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U56/CO (FA_X1)</td>
<td>0.06</td>
<td>3.52</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U55/CO (FA_X1)</td>
<td>0.05</td>
<td>3.57</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U54/CO (FA_X1)</td>
<td>0.05</td>
<td>3.62</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U53/CO (FA_X1)</td>
<td>0.05</td>
<td>3.67</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U52/CO (FA_X1)</td>
<td>0.05</td>
<td>3.72</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U51/CO (FA_X1)</td>
<td>0.05</td>
<td>3.77</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U50/CO (FA_X1)</td>
<td>0.05</td>
<td>3.82</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U49/CO (FA_X1)</td>
<td>0.05</td>
<td>3.87</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_7/U48/S (FA_X1)</td>
<td>0.06</td>
<td>3.93</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U47/CO (FA_X1)</td>
<td>0.08</td>
<td>4.00</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U46/CO (FA_X1)</td>
<td>0.05</td>
<td>4.05</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U45/CO (FA_X1)</td>
<td>0.05</td>
<td>4.10</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U44/CO (FA_X1)</td>
<td>0.04</td>
<td>4.14</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U43/CO (FA_X1)</td>
<td>0.06</td>
<td>4.20</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U42/CO (FA_X1)</td>
<td>0.05</td>
<td>4.25</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U41/CO (FA_X1)</td>
<td>0.06</td>
<td>4.31</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U40/CO (FA_X1)</td>
<td>0.05</td>
<td>4.36</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U39/CO (FA_X1)</td>
<td>0.05</td>
<td>4.41</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U38/CO (FA_X1)</td>
<td>0.05</td>
<td>4.46</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U37/CO (FA_X1)</td>
<td>0.04</td>
<td>4.50</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U36/CO (FA_X1)</td>
<td>0.05</td>
<td>4.55</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U35/CO (FA_X1)</td>
<td>0.04</td>
<td>4.59</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_6/U34/S (FA_X1)</td>
<td>0.07</td>
<td>4.66</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_5/U35/S (FA_X1)</td>
<td>0.09</td>
<td>4.76</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_4/U34/CO (FA_X1)</td>
<td>0.07</td>
<td>4.83</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_4/U33/S (FA_X1)</td>
<td>0.07</td>
<td>4.91</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U1521/ZN (XNOR2_X1)</td>
<td>0.10</td>
<td>5.00</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U1522/ZN (XNOR2_X1)</td>
<td>0.03</td>
<td>5.03</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U34/CO (FA_X1)</td>
<td>0.18</td>
<td>5.21</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2181/ZN (AOI21_X1)</td>
<td>0.03</td>
<td>5.24</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2188/ZN (OAI21_X1)</td>
<td>0.02</td>
<td>5.26</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U31/CO (FA_X1)</td>
<td>0.05</td>
<td>5.31</td>
<td>1.25</td>
</tr>
<tr>
<td>Component</td>
<td>Current (µA)</td>
<td>Voltage (V)</td>
<td>Resistance (Ω)</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------</td>
<td>-------------</td>
<td>----------------</td>
</tr>
<tr>
<td>dut/U523/ZN (AOI21_X1)</td>
<td>0.02</td>
<td>5.33</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U1504/ZN (OAI21_X1)</td>
<td>0.10</td>
<td>5.43</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U28/CO (FA_X1)</td>
<td>0.06</td>
<td>5.48</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U27/CO (FA_X1)</td>
<td>0.07</td>
<td>5.56</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U26/CO (FA_X1)</td>
<td>0.06</td>
<td>5.61</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U25/CO (FA_X1)</td>
<td>0.05</td>
<td>5.66</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U24/CO (FA_X1)</td>
<td>0.04</td>
<td>5.71</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U23/CO (FA_X1)</td>
<td>0.04</td>
<td>5.75</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U22/CO (FA_X1)</td>
<td>0.05</td>
<td>5.79</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U21/CO (FA_X1)</td>
<td>0.05</td>
<td>5.84</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U20/CO (FA_X1)</td>
<td>0.05</td>
<td>5.89</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U19/CO (FA_X1)</td>
<td>0.05</td>
<td>5.94</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U18/CO (FA_X1)</td>
<td>0.04</td>
<td>5.98</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U17/CO (FA_X1)</td>
<td>0.05</td>
<td>6.03</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U16/CO (FA_X1)</td>
<td>0.04</td>
<td>6.07</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U15/CO (FA_X1)</td>
<td>0.05</td>
<td>6.12</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U14/CO (FA_X1)</td>
<td>0.05</td>
<td>6.17</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U13/CO (FA_X1)</td>
<td>0.05</td>
<td>6.22</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U12/CO (FA_X1)</td>
<td>0.05</td>
<td>6.27</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U11/CO (FA_X1)</td>
<td>0.05</td>
<td>6.31</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U10/CO (FA_X1)</td>
<td>0.05</td>
<td>6.36</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U9/CO (FA_X1)</td>
<td>0.05</td>
<td>6.41</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U8/CO (FA_X1)</td>
<td>0.04</td>
<td>6.45</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U7/CO (FA_X1)</td>
<td>0.05</td>
<td>6.50</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U6/CO (FA_X1)</td>
<td>0.07</td>
<td>6.57</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U5/S (FA_X1)</td>
<td>0.14</td>
<td>6.71</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_0/U3/S (FA_X1)</td>
<td>0.09</td>
<td>6.79</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26437/ZN (INV_X1)</td>
<td>0.02</td>
<td>6.81</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26438/ZN (AOI222_X1)</td>
<td>0.07</td>
<td>6.88</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_0/U2/CO (FA_X1)</td>
<td>0.05</td>
<td>6.94</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26441/ZN (NAND2_X1)</td>
<td>0.06</td>
<td>7.00</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26444/ZN (AOI21_X1)</td>
<td>0.03</td>
<td>7.03</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_1/U2/CO (FA_X1)</td>
<td>0.05</td>
<td>7.08</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2866/ZN (NOR2_X1)</td>
<td>0.02</td>
<td>7.10</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2867/ZN (AOI21_X1)</td>
<td>0.04</td>
<td>7.14</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2871/ZN (NAND2_X1)</td>
<td>0.04</td>
<td>7.18</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2872/ZN (OAI21_X1)</td>
<td>0.04</td>
<td>7.23</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2874/ZN (NAND2_X1)</td>
<td>0.05</td>
<td>7.28</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2875/ZN (OAI21_X1)</td>
<td>0.02</td>
<td>7.30</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2876/ZN (INV_X1)</td>
<td>0.01</td>
<td>7.31</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2883/ZN (OAI222_X1)</td>
<td>0.04 c</td>
<td>7.35 r</td>
<td>1.25</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>dut/U2887/ZN (OAI22_X1)</td>
<td>0.02 c</td>
<td>7.37 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_153/U2/CO (FA_X1)</td>
<td>0.06 c</td>
<td>7.44 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U717/ZN (AND2_X1)</td>
<td>0.03 c</td>
<td>7.46 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U744/ZN (OAI22_X1)</td>
<td>0.04 c</td>
<td>7.50 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U747/ZN (NAND2_X1)</td>
<td>0.02 c</td>
<td>7.52 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U749/ZN (AOI21_X1)</td>
<td>0.03 c</td>
<td>7.55 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U750/ZN (AOI22_X1)</td>
<td>0.04 c</td>
<td>7.59 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_189/U2/CO (FA_X1)</td>
<td>0.05 c</td>
<td>7.65 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2905/ZN (AND2_X1)</td>
<td>0.03 c</td>
<td>7.68 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2906/ZN (OAI22_X1)</td>
<td>0.04 c</td>
<td>7.71 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2907/ZN (NAND2_X1)</td>
<td>0.02 c</td>
<td>7.73 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2908/ZN (NAND2_X1)</td>
<td>0.01 c</td>
<td>7.75 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2909/ZN (NAND2_X1)</td>
<td>0.01 c</td>
<td>7.76 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2927/ZN (OAI22_X1)</td>
<td>0.03 c</td>
<td>7.79 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2931/ZN (OAI22_X1)</td>
<td>0.04 c</td>
<td>7.82 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_128/U3/CO (FA_X1)</td>
<td>0.06 c</td>
<td>7.88 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3318/ZN (NAND2_X1)</td>
<td>0.02 c</td>
<td>7.90 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3320/ZN (NAND2_X1)</td>
<td>0.01 c</td>
<td>7.91 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3322/ZN (NOR2_X1)</td>
<td>0.03 c</td>
<td>7.94 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3325/ZN (OAI21_X1)</td>
<td>0.02 c</td>
<td>7.96 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_119/U4/CO (FA_X1)</td>
<td>0.05 c</td>
<td>8.01 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U649/ZN (AOI21_X1)</td>
<td>0.04 c</td>
<td>8.05 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U778/ZN (OAI21_X1)</td>
<td>0.03 c</td>
<td>8.08 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_71/U2/CO (FA_X1)</td>
<td>0.06 c</td>
<td>8.14 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3344/ZN (NAND2_X1)</td>
<td>0.03 c</td>
<td>8.17 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3345/ZN (OAI21_X1)</td>
<td>0.03 c</td>
<td>8.20 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26911/CO (FA_X1)</td>
<td>0.04 c</td>
<td>8.24 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3347/ZN (NAND2_X1)</td>
<td>0.04 c</td>
<td>8.29 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3349/ZN (NAND3_X1)</td>
<td>0.02 c</td>
<td>8.31 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_72/U2/CO (FA_X1)</td>
<td>0.06 c</td>
<td>8.37 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2977/ZN (AND2_X1)</td>
<td>0.05 c</td>
<td>8.42 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2978/ZN (OAI22_X1)</td>
<td>0.04 c</td>
<td>8.46 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2981/ZN (AOI222_X1)</td>
<td>0.04 c</td>
<td>8.50 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_188/U2/CO (FA_X1)</td>
<td>0.05 c</td>
<td>8.55 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3290/ZN (OAI21_X1)</td>
<td>0.07 c</td>
<td>8.62 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3292/ZN (NAND2_X1)</td>
<td>0.02 c</td>
<td>8.64 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3295/ZN (OAI21_X1)</td>
<td>0.03 c</td>
<td>8.67 r</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3299/ZN (AOI21_X1)</td>
<td>0.03 c</td>
<td>8.70 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3301/ZN (OR2_X1)</td>
<td>0.03 c</td>
<td>8.73 f</td>
<td>1.25</td>
</tr>
<tr>
<td>dut</td>
<td>Value</td>
<td>8.75 r</td>
<td>1.25</td>
</tr>
<tr>
<td>----------------</td>
<td>-------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>U3304/ZN (NAND2_X1)</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3305/ZN (OAI21_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>intadd_47/U4/CO (FA_X1)</td>
<td>0.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U650/ZN (AOI21_X1)</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3410/ZN (OAI21_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3411/ZN (NAND2_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3414/ZN (AOI21_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3425/ZN (NAND2_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3427/ZN (NAND3_X1)</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>intadd_158/U2/CO (FA_X1)</td>
<td>0.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3255/ZN (NOR2_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3256/ZN (AOI21_X1)</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>intadd_186/U2/CO (FA_X1)</td>
<td>0.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U1146/ZN (NAND2_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U1148/ZN (AOI21_X1)</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U651/ZN (OAI22_X1)</td>
<td>0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U8/ZN (AOI21_X1)</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U605/ZN (OR2_X1)</td>
<td>0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U1169/ZN (OR2_X1)</td>
<td>0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U425/ZN (AOI22_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U1172/ZN (AOI22_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>intadd_184/U2/CO (FA_X1)</td>
<td>0.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U418/ZN (NOR2_X1)</td>
<td>0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3228/ZN (OAI21_X1)</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U652/ZN (OAI21_X1)</td>
<td>0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3243/ZN (NOR2_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U312/ZN (NOR2_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3248/ZN (OAI21_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3251/ZN (NOR2_X1)</td>
<td>0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3252/ZN (AOI21_X1)</td>
<td>0.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>intadd_183/U2/CO (FA_X1)</td>
<td>0.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3195/ZN (NAND2_X1)</td>
<td>0.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U634/ZN (AOI21_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3198/ZN (OAI21_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3202/ZN (AOI21_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3204/ZN (OR2_X1)</td>
<td>0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3207/ZN (NAND2_X1)</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U3208/ZN (OAI21_X1)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>intadd_129/U3/CO (FA_X1)</td>
<td>0.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>intadd_129/U2/CO (FA_X1)</td>
<td>0.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component</td>
<td>Delay (ns)</td>
<td>Power (μW)</td>
<td>Voltage (V)</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>dut/U421/ZN (OR2_X1)</td>
<td>0.04</td>
<td>10.14</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U221/ZN (AOI21_X1)</td>
<td>0.04</td>
<td>10.18</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U828/ZN (AOI21_X1)</td>
<td>0.03</td>
<td>10.21</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/ntadd_181/U2/CO (FA_X1)</td>
<td>0.05</td>
<td>10.26</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U422/ZN (NOR2_X1)</td>
<td>0.05</td>
<td>10.31</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3370/ZN (AOI21_X1)</td>
<td>0.03</td>
<td>10.33</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U315/ZN (AOI21_X1)</td>
<td>0.03</td>
<td>10.36</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3385/ZN (OR2_X1)</td>
<td>0.04</td>
<td>10.40</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3387/ZN (NAND3_X1)</td>
<td>0.02</td>
<td>10.42</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/ntadd_160/U2/CO (FA_X1)</td>
<td>0.05</td>
<td>10.48</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3158/ZN (NOR2_X1)</td>
<td>0.04</td>
<td>10.52</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3159/ZN (AOI21_X1)</td>
<td>0.02</td>
<td>10.54</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/ntadd_180/U2/CO (FA_X1)</td>
<td>0.07</td>
<td>10.62</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2948/ZN (AND2_X1)</td>
<td>0.05</td>
<td>10.67</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2949/ZN (AOI22_X1)</td>
<td>0.04</td>
<td>10.70</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U2952/ZN (AOI222_X1)</td>
<td>0.07</td>
<td>10.77</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/ntadd_179/U2/CO (FA_X1)</td>
<td>0.12</td>
<td>10.89</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U1118/ZN (OR2_X1)</td>
<td>0.05</td>
<td>10.94</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U333/ZN (AOI22_X1)</td>
<td>0.03</td>
<td>10.97</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U1121/ZN (AOI22_X1)</td>
<td>0.03</td>
<td>11.00</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/ntadd_178/U2/CO (FA_X1)</td>
<td>0.06</td>
<td>11.06</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3154/ZN (NAND2_X1)</td>
<td>0.03</td>
<td>11.09</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3155/ZN (AOI21_X1)</td>
<td>0.03</td>
<td>11.12</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/ntadd_42/U3/CO (FA_X1)</td>
<td>0.06</td>
<td>11.18</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/ntadd_42/U2/CO (FA_X1)</td>
<td>0.05</td>
<td>11.23</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U608/ZN (NOR2_X1)</td>
<td>0.03</td>
<td>11.26</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3106/ZN (AOI21_X1)</td>
<td>0.02</td>
<td>11.29</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U653/ZN (AOI22_X1)</td>
<td>0.04</td>
<td>11.33</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U7/ZN (AOI21_X1)</td>
<td>0.04</td>
<td>11.37</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U316/ZN (OR2_X1)</td>
<td>0.04</td>
<td>11.40</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3111/ZN (AND2_X1)</td>
<td>0.03</td>
<td>11.43</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3112/ZN (AOI22_X1)</td>
<td>0.03</td>
<td>11.45</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3113/ZN (INV_X1)</td>
<td>0.02</td>
<td>11.47</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3114/ZN (AOI222_X1)</td>
<td>0.04</td>
<td>11.52</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U6/ZN (AOI21_X1)</td>
<td>0.04</td>
<td>11.56</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3118/ZN (OR2_X1)</td>
<td>0.04</td>
<td>11.60</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3119/ZN (AND2_X1)</td>
<td>0.03</td>
<td>11.62</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3120/ZN (AOI22_X1)</td>
<td>0.03</td>
<td>11.65</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3121/ZN (INV_X1)</td>
<td>0.02</td>
<td>11.66</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U3122/ZN (AOI222_X1)</td>
<td>0.05</td>
<td>11.72</td>
<td>1.25</td>
</tr>
</tbody>
</table>
dut/U3134/ZN (AOI21_X1) 0.03 c 11.75 f 1.25
dut/U3137/ZN (OAI21_X1) 0.03 c 11.80 f 1.25
dut/U3138/ZN (NAND2_X1) 0.04 c 11.84 r 1.25
dut/U3141/ZN (AOI21_X1) 0.03 c 11.87 f 1.25
dut/intadd_175/U2/CO (FA_X1) 0.05 c 11.92 f 1.25
dut/U3040/ZN (NAND2_X1) 0.04 c 11.96 r 1.25
dut/U3041/ZN (OAI21_X1) 0.03 c 11.99 f 1.25
dut/intadd_54/U4/CO (FA_X1) 0.08 c 12.07 f 1.25
dut/U654/ZN (AOI21_X1) 0.05 c 12.12 r 1.25
dut/U3013/ZN (OAI21_X1) 0.02 c 12.14 f 1.25
dut/U3015/ZN (NOR2_X1) 0.07 c 12.21 r 1.25
dut/U3018/ZN (OAI21_X1) 0.03 c 12.24 f 1.25
dut/intadd_164/U3/CO (FA_X1) 0.07 c 12.31 f 1.25
dut/U26544/ZN (INV_X1) 0.04 c 12.35 r 1.25
dut/U26548/ZN (OAI21_X1) 0.02 c 12.37 f 1.25
dut/U26549/ZN (INV_X1) 0.05 c 12.42 r 1.25
dut/intadd_74/U2/CO (FA_X1) 0.04 c 12.45 r 1.25
dut/U26550/ZN (INV_X1) 0.02 c 12.47 f 1.25
dut/intadd_190/U2/CO (FA_X1) 0.08 c 12.56 f 1.25
dut/U26560/ZN (NAND2_X1) 0.04 c 12.60 r 1.25
dut/U26561/ZN (OAI21_X1) 0.03 c 12.63 f 1.25
dut/intadd_146/U4/CO (FA_X1) 0.07 c 12.70 f 1.25
dut/U655/ZN (AOI22_X1) 0.06 c 12.76 r 1.25
dut/U26572/ZN (OAI21_X1) 0.06 c 12.81 f 1.25
dut/U26573/ZN (NAND2_X1) 0.04 c 12.85 r 1.25
dut/U26576/ZN (AOI21_X1) 0.03 c 12.88 f 1.25
dut/intadd_168/U3/CO (FA_X1) 0.06 c 12.94 f 1.25
dut/intadd_168/U2/CO (FA_X1) 0.07 c 13.01 f 1.25
dut/U26577/ZN (NOR2_X1) 0.04 c 13.05 r 1.25
dut/U26580/ZN (OAI21_X1) 0.02 c 13.06 f 1.25
dut/intadd_174/U2/CO (FA_X1) 0.06 c 13.12 f 1.25
dut/U26588/ZN (NOR2_X1) 0.03 c 13.15 r 1.25
dut/U26591/ZN (OAI21_X1) 0.02 c 13.17 f 1.25
dut/intadd_130/U3/CO (FA_X1) 0.07 c 13.24 f 1.25
dut/intadd_130/U2/CO (FA_X1) 0.05 c 13.29 f 1.25
dut/U796/ZN (OR2_X1) 0.04 c 13.34 f 1.25
dut/U222/ZN (AOI22_X1) 0.03 c 13.36 r 1.25
dut/U799/ZN (AOI21_X1) 0.02 c 13.39 f 1.25
dut/intadd_173/U2/CO (FA_X1) 0.06 c 13.45 f 1.25
<table>
<thead>
<tr>
<th>Component</th>
<th>Delay (c)</th>
<th>Voltage (r)</th>
<th>Power (f)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dut/U691/ZN (NOR2_X1)</td>
<td>0.03</td>
<td>13.47</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U693/ZN (OAI21_X1)</td>
<td>0.03</td>
<td>13.51</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U695/ZN (OAI222_X1)</td>
<td>0.07</td>
<td>13.57</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U5/ZN (AOI21_X1)</td>
<td>0.05</td>
<td>13.63</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U656/ZN (OR2_X1)</td>
<td>0.03</td>
<td>13.66</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_169/U2/CO (FA_X1)</td>
<td>0.06</td>
<td>13.71</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26604/ZN (NOR2_X1)</td>
<td>0.07</td>
<td>13.78</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26607/ZN (OAI21_X1)</td>
<td>0.02</td>
<td>13.80</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_172/U2/CO (FA_X1)</td>
<td>0.08</td>
<td>13.88</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26617/ZN (NAND2_X1)</td>
<td>0.03</td>
<td>13.91</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26618/ZN (OAI21_X1)</td>
<td>0.02</td>
<td>13.94</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_140/U2/CO (FA_X1)</td>
<td>0.07</td>
<td>14.01</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26619/ZN (INV_X1)</td>
<td>0.02</td>
<td>14.03</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/intadd_191/U2/CO (FA_X1)</td>
<td>0.04</td>
<td>14.07</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26623/ZN (AND2_X1)</td>
<td>0.03</td>
<td>14.09</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26626/ZN (OAI21_X1)</td>
<td>0.02</td>
<td>14.12</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26628/ZN (NAND2_X1)</td>
<td>0.03</td>
<td>14.15</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U638/ZN (OR2_X1)</td>
<td>0.04</td>
<td>14.18</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26633/ZN (OAI211_X1)</td>
<td>0.03</td>
<td>14.21</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26639/ZN (XNOR2_X1)</td>
<td>0.05</td>
<td>14.26</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/U26640/ZN (OAI22_X1)</td>
<td>0.03</td>
<td>14.28</td>
<td>1.25</td>
</tr>
<tr>
<td>dut/data_out_reg[255]/D (DFF_X1)</td>
<td>0.00</td>
<td>14.28</td>
<td>1.25</td>
</tr>
</tbody>
</table>

Clock clk (rise edge) 14.00 14.00
Clock network delay (propagated) 0.41 14.41
Dut/data_out_reg[255]/CK (DFF_X1) 0.00 14.41
Library setup time -0.03 14.39
Data required time 14.39

Data required time 14.39
Data arrival time -14.28

Slack (MET) 0.10
APPENDIX H: DETAILED CONGESTION REPORT
The detailed congestion report is as follows:

Initializing Data Structure ...
INFO: legalizer_via_spacing_check_mode 0
Reading technology information ...
  Technology table contains 10 routable metal layers
  This is considered as a 10-metal-layer design
Reading library information from DB ...
Reading misc information ...
  array <unit> has 0 vertical and 184 horizontal rows
  48 pre-routes for placement blockage/checking
  467 pre-routes for map congestion calculation
Checking information read in ...
  design style = Horizontal masters, Horizontal rows

Preprocessing design ...
  splitting rows by natural obstacles ...
  ... design style 0
  ... number of base array 1 0
INFO:... use original rows...
[end initializing data for legality checker]
Running global router for congestion map ...
Start Global Route ...
[Init] Elapsed real time: 0:00:00
[Init] Elapsed cpu time: sys=0:00:00 usr=0:00:00 total=0:00:00
[Init] Stage (MB): Used 0 Allocr 0 Proc 0
[Init] Total (MB): Used 20 Allocr 29 Proc 2352
Printing options for 'set_route_zrt_common_options'
  -dbin_list_of_nets : {}
  -extraction_mode : parallel
  -verbose_level : 0

Printing options for 'set_route_zrt_global_options'

Begin global routing.
Warning: Cell contains tie connections which are not connected to real PG. (MW-349)
Turn off antenna since no rule is specified
Cell Min-Routing-Layer = metal1
Cell Max-Routing-Layer = metal10
Information: Multiple default contact via1_0 found for layer via1. (ZRT-021)
Information: Multiple default contact via1_1 found for layer via1. (ZRT-021)
Information: Multiple default contact via1_3 found for layer via1. (ZRT-021)
Information: Multiple default contact via1_5 found for layer via1. (ZRT-021)
Information: Multiple default contact via2_5 found for layer via2. (ZRT-021)
Information: Multiple default contact via2_6 found for layer via2. (ZRT-021)
Information: Multiple default contact via2_0 found for layer via2. (ZRT-021)
Information: Multiple default contact via2_1 found for layer via2. (ZRT-021)
Information: Multiple default contact via3_0 found for layer via3. (ZRT-021)
Warning: Ignore 2 top cell ports with no pins. (ZRT-027)
Found 1 voltage-areas.
Via on layer (via1) needs more than one tracks
Warning: Layer metal1 pitch 0.140 may be too small: wire/via-down 0.140, wire/via-up 0.170. (ZRT-026)
Current Stage stats:
[End of Read DB] Elapsed real time: 0:00:03
[End of Read DB] Elapsed cpu time: sys=0:00:00 usr=0:00:03 total=0:00:03
[End of Read DB] Stage (MB): Used 83 Allocr 76 Proc 0
[End of Read DB] Total (MB): Used 104 Allocr 105 Proc 2352
Constructing data structure ...
Design statistics:
Design Bounding Box (0.00,0.00,278.97,277.60)
Number of routing layers = 10
layer metal1, dir Hor, min width = 0.07, min space = 0.06 pitch = 0.14
layer metal2, dir Ver, min width = 0.07, min space = 0.07 pitch = 0.19
layer metal3, dir Hor, min width = 0.07, min space = 0.07 pitch = 0.14
layer metal4, dir Ver, min width = 0.14, min space = 0.14 pitch = 0.28
layer metal5, dir Hor, min width = 0.14, min space = 0.14 pitch = 0.28
layer metal6, dir Ver, min width = 0.14, min space = 0.14 pitch = 0.28
layer metal7, dir Hor, min width = 0.40, min space = 0.40 pitch = 0.80
layer metal8, dir Ver, min width = 0.40, min space = 0.40 pitch = 0.80
layer metal9, dir Hor, min width = 0.80, min space = 0.80 pitch = 1.60
layer metal10, dir Ver, min width = 0.80, min space = 0.80 pitch = 1.60
Current Stage stats:
[End of Build Tech Data] Elapsed real time: 0:00:00
[End of Build Tech Data] Elapsed cpu time: sys=0:00:00 usr=0:00:00 total=0:00:00
[End of Build Tech Data] Stage (MB): Used 3 Allocr 3 Proc 0
[End of Build Tech Data] Total (MB): Used 108 Allocr 109 Proc 2352
Net statistics:
Total number of nets = 44240
Number of nets to route = 0
Number of single or zero port nets = 53
Number of nets with min-layer-mode soft = 249
Number of nets with min-layer-mode soft-cost-medium = 249
Number of nets with max-layer-mode hard = 249
44187 nets are fully connected,
of which 44187 are detail routed and 0 are global routed.
Current Stage stats:
[End of Build All Nets] Elapsed real time: 0:00:00
[End of Build All Nets] Elapsed cpu time: sys=0:00:00 usr=0:00:00 total=0:00:00
[End of Build All Nets] Stage (MB): Used 14 Allocr 14 Proc 0
[End of Build All Nets] Total (MB): Used 122 Allocr 124 Proc 2352
Average gCell capacity 1.72 on layer (1) metal1
Average gCell capacity 6.84 on layer (2) metal2
Average gCell capacity 9.70 on layer (3) metal3
Average gCell capacity 4.69 on layer (4) metal4
Average gCell capacity 2.90 on layer (5) metal5
Average gCell capacity 3.09 on layer (6) metal6
Average gCell capacity 1.74 on layer (7) metal7
Average gCell capacity 1.74 on layer (8) metal8
Average gCell capacity 0.86 on layer (9) metal9
Average gCell capacity 0.86 on layer (10) metal10
Average number of tracks per gCell 9.96 on layer (1) metal1
Average number of tracks per gCell 7.38 on layer (2) metal2
Average number of tracks per gCell 9.96 on layer (3) metal3
Average number of tracks per gCell 5.01 on layer (4) metal4
Average number of tracks per gCell 4.98 on layer (5) metal5
Average number of tracks per gCell 5.01 on layer (6) metal6
Average number of tracks per gCell 1.75 on layer (7) metal7
Average number of tracks per gCell 1.75 on layer (8) metal8
Average number of tracks per gCell 0.87 on layer (9) metal9
Average number of tracks per gCell 0.88 on layer (10) metal10
Number of gCells = 396010
Current Stage stats:
[End of Build Congestion map] Elapsed real time: 0:00:00
[End of Build Congestion map] Elapsed cpu time: sys=0:00:00 usr=0:00:00
total=0:00:00
[End of Build Congestion map] Stage (MB): Used 0 Alloctr 0 Proc 0
[End of Build Congestion map] Total (MB): Used 123 Alloctr 124 Proc 2352
Total stats:
[End of Build Data] Elapsed real time: 0:00:00
[End of Build Data] Elapsed cpu time: sys=0:00:00 usr=0:00:00 total=0:00:00
[End of Build Data] Stage (MB): Used 19 Alloctr 19 Proc 0
[End of Build Data] Total (MB): Used 124 Alloctr 125 Proc 2352
Current Stage stats:
[End of Blocked Pin Detection] Elapsed real time: 0:00:00
[End of Blocked Pin Detection] Elapsed cpu time: sys=0:00:00 usr=0:00:00
total=0:00:00
[End of Blocked Pin Detection] Stage (MB): Used 0 Alloctr 0 Proc 0
Information: Using 1 threads for routing. (ZRT-444)
multi gcell levels ON

Start GR phase 0
Current Stage stats:
[End of Initial Routing] Elapsed real time: 0:00:00
[End of Initial Routing] Elapsed cpu time: sys=0:00:00 usr=0:00:00 total=0:00:00
[End of Initial Routing] Stage (MB): Used 0 Alloctr 0 Proc 0
Initial. Routing result:
Initial. BothDirs: Overflow = 19235 Max = 11 GRCs = 15965 (20.16%)
Initial. H routing: Overflow = 5144 Max = 11 (GRCs = 2) GRCs = 4436
(11.20%)
Initial. V routing: Overflow = 14091 Max = 9 (GRCs = 2) GRCs = 11529
(29.11%)
Initial. metal1 Overflow = 3 Max = 1 (GRCs = 3) GRCs = 3 (0.01%)
Initial. metal2 Overflow = 6485 Max = 9 (GRCs = 2) GRCs = 4793 (12.10%)
Initial. metal3 Overflow = 3505 Max = 11 (GRCs = 2) GRCs = 2792 (7.05%)
Initial. metal4 Overflow = 6741 Max = 5 (GRCs = 2) GRCs = 5918 (14.94%)
Initial. metal5 Overflow = 1570 Max = 3 (GRCs = 4) GRCs = 1571 (3.97%)
Initial. metal6     Overflow = 817 Max = 3 (GRCs = 1) GRCs = 768 (1.94%)
Initial. metal7     Overflow = 66 Max = 1 (GRCs = 63) GRCs = 70 (0.18%)
Initial. metal8     Overflow = 47 Max = 1 (GRCs = 44) GRCs = 50 (0.13%)
Initial. metal9     Overflow = 0 Max = 0 (GRCs = 0) GRCs = 0 (0.00%)
Initial. metal10    Overflow = 0 Max = 0 (GRCs = 0) GRCs = 0 (0.00%)

Initial. Total Wire Length = 0.00
Initial. Layer metal1 wire length = 0.00
Initial. Layer metal2 wire length = 0.00
Initial. Layer metal3 wire length = 0.00
Initial. Layer metal4 wire length = 0.00
Initial. Layer metal5 wire length = 0.00
Initial. Layer metal6 wire length = 0.00
Initial. Layer metal7 wire length = 0.00
Initial. Layer metal8 wire length = 0.00
Initial. Layer metal9 wire length = 0.00
Initial. Layer metal10 wire length = 0.00
Initial. Total Number of Contacts = 0
Initial. Via via1_4 count = 0
Initial. Via via2_8 count = 0
Initial. Via via3_2 count = 0
Initial. Via via4_0 count = 0
Initial. Via via5_0 count = 0
Initial. Via via6_0 count = 0
Initial. Via via7_0 count = 0
Initial. Via via8_0 count = 0
Initial. Via via9_0 count = 0
Initial. completed.

Start GR phase 1
Current Stage stats:
[End of Phase1 Routing] Elapsed real time: 0:00:00
[End of Phase1 Routing] Elapsed cpu time: sys=0:00:00 usr=0:00:00
total=0:00:00
[End of Phase1 Routing] Stage (MB): Used 0 Alloctr 0 Proc 0
[End of Phase1 Routing] Total (MB): Used 124 Alloctr 125 Proc 2352
phase1. Routing result:
phase1. Both Dirs: Overflow = 19235 Max = 11 GRCs = 15965 (20.16%)
phase1. H routing: Overflow = 5144 Max = 11 (GRCs = 2) GRCs = 4436 (11.20%)
phase1. V routing: Overflow = 14091 Max = 9 (GRCs = 2) GRCs = 11529 (29.11%)
phase1. metal1: Overflow = 3 Max = 1 (GRCs = 3) GRCs = 3 (0.01%)
phase1. metal2: Overflow = 6485 Max = 9 (GRCs = 2) GRCs = 4793 (12.10%)
phase1. metal3: Overflow = 3505 Max = 11 (GRCs = 2) GRCs = 2792 (7.05%)
phase1. metal4: Overflow = 6741 Max = 5 (GRCs = 2) GRCs = 5918 (14.94%)
phase1. metal5: Overflow = 1570 Max = 3 (GRCs = 4) GRCs = 1571 (3.97%)
phase1. metal6: Overflow = 817 Max = 3 (GRCs = 1) GRCs = 768 (1.94%)
phase1. metal7: Overflow = 66 Max = 1 (GRCs = 63) GRCs = 70 (0.18%)
phase1. metal8: Overflow = 47 Max = 1 (GRCs = 44) GRCs = 50 (0.13%)
phase1. metal9: Overflow = 0 Max = 0 (GRCs = 0) GRCs = 0 (0.00%)
phase1. metal10: Overflow = 0 Max = 0 (GRCs = 0) GRCs = 0 (0.00%)

phase1. Total Wire Length = 0.00
phase1. Layer metal1 wire length = 0.00
phase1. Layer metal2 wire length = 0.00
phase1. Layer metal3 wire length = 0.00
phase1. Layer metal4 wire length = 0.00
phase1. Layer metal5 wire length = 0.00
phase1. Layer metal6 wire length = 0.00
phase1. Layer metal7 wire length = 0.00
phase1. Layer metal8 wire length = 0.00
phase1. Layer metal9 wire length = 0.00
phase1. Layer metal10 wire length = 0.00
phase1. Total Number of Contacts = 0
phase1. Via via1_4 count = 0
phase1. Via via2_8 count = 0
phase1. Via via3_2 count = 0
phase1. Via via4_0 count = 0
phase1. Via via5_0 count = 0
phase1. Via via6_0 count = 0
phase1. Via via7_0 count = 0
phase1. Via via8_0 count = 0
phase1. Via via9_0 count = 0
phase1. completed.

Start GR phase 2
Current Stage stats:
[End of Phase2 Routing] Elapsed real time: 0:00:00
[End of Phase2 Routing] Elapsed cpu time: sys=0:00:00 usr=0:00:00
  total=0:00:00
[End of Phase2 Routing] Stage (MB): Used 0  Alloctr 0  Proc 0
[End of Phase2 Routing] Total (MB): Used 124  Alloctr 125  Proc 2352
phase2. Routing result:
  phase2. Both Dirs: Overflow = 19235 Max = 11 GRCs = 15965 (20.16%)
  phase2. H routing: Overflow = 5144 Max = 11 (GRCs = 2) GRCs = 4436 (11.20%)
  phase2. V routing: Overflow = 14091 Max = 9 (GRCs = 2) GRCs = 11529 (29.11%)
  phase2. metal1  Overflow = 3 Max = 1 (GRCs = 3) GRCs = 3 (0.01%)
  phase2. metal2  Overflow = 6485 Max = 9 (GRCs = 2) GRCs = 4793 (12.10%)
  phase2. metal3  Overflow = 3505 Max = 11 (GRCs = 2) GRCs = 2792 (7.05%)
  phase2. metal4  Overflow = 6741 Max = 5 (GRCs = 2) GRCs = 5918 (14.94%)
  phase2. metal5  Overflow = 1570 Max = 3 (GRCs = 4) GRCs = 1571 (3.97%)
  phase2. metal6  Overflow = 817 Max = 3 (GRCs = 1) GRCs = 768 (1.94%)
  phase2. metal7  Overflow = 66 Max = 1 (GRCs = 63) GRCs = 70 (0.18%)
  phase2. metal8  Overflow = 47 Max = 1 (GRCs = 44) GRCs = 50 (0.13%)
  phase2. metal9  Overflow = 0 Max = 0 (GRCs = 0) GRCs = 0 (0.00%)
  phase2. metal10 Overflow = 0 Max = 0 (GRCs = 0) GRCs = 0 (0.00%)

phase2. Total Wire Length = 0.00
phase2. Layer metal1 wire length = 0.00
phase2. Layer metal2 wire length = 0.00
phase2. Layer metal3 wire length = 0.00
phase2. Layer metal4 wire length = 0.00
phase2. Layer metal5 wire length = 0.00
phase2. Layer metal6 wire length = 0.00
phase2. Layer metal7 wire length = 0.00
phase2. Layer metal8 wire length = 0.00
phase2. Layer metal9 wire length = 0.00
phase 2. Layer metal10 wire length = 0.00
phase 2. Total Number of Contacts = 0
phase 2. Via via 1_4 count = 0
phase 2. Via via 2_8 count = 0
phase 2. Via via 3_2 count = 0
phase 2. Via via 4_0 count = 0
phase 2. Via via 5_0 count = 0
phase 2. Via via 6_0 count = 0
phase 2. Via via 7_0 count = 0
phase 2. Via via 8_0 count = 0
phase 2. Via via 9_0 count = 0
phase 2. completed.
[End of Whole Chip Routing] Elapsed real time: 0:00:00
[End of Whole Chip Routing] Elapsed cpu time: sys=0:00:00 usr=0:00:00 total=0:00:00
[End of Whole Chip Routing] Stage (MB): Used 19 Alloctr 19 Proc 0
Congestion utilization per direction:
  Average vertical track utilization = 59.76 %
  Peak vertical track utilization = 200.00 %
  Average horizontal track utilization = 57.14 %
  Peak horizontal track utilization = 141.67 %
Current Stage stats:
  [GR: Done] Elapsed real time: 0:00:00
  [GR: Done] Elapsed cpu time: sys=0:00:00 usr=0:00:00 total=0:00:00
  [GR: Done] Stage (MB): Used -4 Alloctr -4 Proc 0
  [GR: Done] Total (MB): Used 121 Alloctr 122 Proc 2352
GR Total stats:
  [GR: Done] Elapsed real time: 0:00:04
  [GR: Done] Elapsed cpu time: sys=0:00:04 usr=0:00:04 total=0:00:04
  [GR: Done] Stage (MB): Used 100 Alloctr 93 Proc 0
  [GR: Done] Total (MB): Used 121 Alloctr 122 Proc 2352
Writing out congestion map...
Updating congestion ...
[DBOUT] Elapsed real time: 0:00:00
[DBOUT] Elapsed cpu time: sys=0:00:00 usr=0:00:00 total=0:00:00
[DBOUT] Stage (MB): Used  -83  Alloctr  -76  Proc  0
[DBOUT] Total (MB): Used  22  Alloctr  30  Proc 2352
Final total stats:
[End of Global Routing] Elapsed real time: 0:00:04
[End of Global Routing] Elapsed cpu time: sys=0:00:00 usr=0:00:04 total=0:00:04
[End of Global Routing] Stage (MB): Used  1  Alloctr  1  Proc  0

Information: Reporting global route congestion data from Milkyway...

Both Dirs: Overflow = 4840 Max = 8 (2 GRCs) GRCs = 2995 (3.78%)
H routing: Overflow = 1573 Max = 5 (3 GRCs) GRCs = 1176 (1.48%)
V routing: Overflow = 3267 Max = 8 (2 GRCs) GRCs = 1819 (2.30%)