Leakage Power Analysis of Asynchronous Pipeline Templates

Behnam Ghavami, Mohammad Salehi, Hossein Pedram
Asynchronous LAB, Computer Engineering Department, Amirkabir University of Technology
E-mail: ghavamib@aut.ac.ir, {m_salehi, pedram}@ce.aut.ac.ir

Abstract

In this paper we studied accurate static power consumption in five well-known templates used in implementation of asynchronous circuits. The studied templates comprising PCHB, PCFB, STFB, HC and MOUSTRAP have been used to model a 5-stage 2-bit pipeline by HSPICE in 0.18um CMOS technology. In order to analyze the static power consumption, several behavioral studies were conducted and the results demonstrated dependence of the amount of static power to the number of tokens in the pipeline stages. Also it have been proven that the MOUSTRAP template consumes less static power.

1. Introduction

Reduction in the size and the growth in number of the transistors in contemporary circuits signify the problem of global synchronization. One solution is to eliminate the global clock signal and take advantage of asynchronous design methods. In such circuits there is no global clock, hence stages communicate utilizing local handshaking [6]. The main advantages of these circuits are low power dissipation, high performance and better noise properties [7]. The disadvantages of this style are complexity of design flow and less availability of CAD tools.

Among the numerous asynchronous design styles being developed, template-based fine-grained pipelines have decreased the complexity of design effort [1][6]. Template-based design is somewhat similar to standard cell design in synchronous logic. Templates can be either pre-designed to implement simple logic functions, with handshaking, or can be synthesized to create more complex ones. Template based approaches also have the advantage of removing the need for generating, optimizing, and verifying specifications for complex distributed controllers, which is both difficult and error-prone [2]. In this approach the specification of the circuit is broken to a number of parts which are mappable to predefined templates.

Figure 1. Bundled-Data and Data-Driven Pipeline template

Up to now a number of pipeline templates have been proposed for template based asynchronous circuit design each having its own benefits and drawbacks. Based on the type of data validity indication between stages, these circuits may be classified into two groups: Bundled-Data (BD) and Data-Driven (DD) (Figure 1) [1].

Today, increasing the chip density and reducing the feature size in VLSI circuits have caused static power consumption to become one of the main challenges in the digital design. Furthermore, as transistors shrink in size, the level of doping has to be increased, thereby causing leakage current to become dominant [3]. Also, in asynchronous circuits as one class of VLSI circuits, static power increases with the scaling of CMOS manufacturing technology into deep sub-micron era. In this paper we have studied the static power consumption of three well-known dual-rail asynchronous pipe-
line templates. The paper is organized as follows. Section 2 and 3 describe BD and DD based micro-pipelined designs along with the five templates which will be analyzed. Section 4 elaborates the proposed methodology for analysis of static power. Finally, the results of HSPICE simulations are presented in section 5 and the paper is concluded.

2. BD based micro-pipelines

In these design styles, the Request and Acknowledge signals are used as handshaking signal and the data is transmitted using the single-rail channel, i.e. one signal per data bit. The associated channel is called a bundled-data channel [1]. In this design style, there is a control circuits which generates the handshaking signal based on the time required for the data to be processed. The data validity is indicated by the Request signal which is asserted when a stage has valid data. Hence, if the delay of control circuit is less than the functional block delay, a matched delay circuit should be added for increasing the delay. These templates have very aggressive timing assumptions that introduce stringent transistor sizing requirements and high demands on post-layout verification. In following, two well-known micro-pipeline templates based on this style are described.

2.1. HC template

The structure of one stage of the HC pipeline [9] is shown in Figure 2. A key feature of this pipeline style is that it uses decoupled control of evaluation and precharge: separate Eval and Pc signals are generated by each stage’s control. Precharge occurs when Pc is asserted and Eval is de-asserted. Evaluation occurs when Pc is de-asserted and Eval is asserted. When both signals are de-asserted, the gate output is effectively isolated from the gate inputs; this is the isolate phase. To avoid short circuit, Pc and Eval are never simultaneously asserted.

An asymmetric C-element is used as a completion detector. The C-element output is fed through a matched delay, which (combined with the completion detector) matches the worst-case path through the function block. The HC pipeline stage cycles through three phases. After it completes the evaluate phase, it enters the isolate phase (where both Eval and Pc are de-asserted) and subsequently the precharge phase, after which it re-enters the evaluate phase, completing the cycle.

2.2. MOUSTRAP template

Another method that can be implemented using BD is MOUSTRAP [8]. MOUSTRAP is the acronym for Minimal-Overhead Ultra-high-Speed Transition signaling Asynchronous Pipeline. The circuit for one stage of this method, which utilizes an XNOR gate, is shown in Figure 3. Operation of the circuit can be followed using throughput equations for all methods. Hence we omit the description due to space limitation. All details are given in [8].

MOUSTRAP uses two-phase protocol, and hence each low to high or high to low transition represents the arrival of a new data item, and, hence, the throughput is equal to the time between two transitions. Since the pulse duration of high and low states are different, hence, two slightly different values can be calculated for the throughput.
3. DD based micro-pipelines

Alternatively, in DD templates, the data is sent using two wires for each bit of information. Dual-rail encoding allows for data validity to be indicated by the data itself. This latter approach requires careful timing analysis but allows the reuse of synchronous single-rail logic. In dual-rail encoding a data bit is represented using two signals. For example, 00 indicates data is not ready, 01 and 10 indicate bit 0 and 1, respectively and 11 is invalid case. The encoded data indicates both data validity and data value, so the Request and Acknowledge signals can be generated using the data. Consequently unlike BD structure, in most methods in this category, no matched delay circuit is required for the proper functioning. In this section, three well known micro-pipelined templates based on DD architecture are studied under equal conditions.

3.1. PCFB template

A Pre-Charged Full Buffer (PCFB) [1] is an asynchronous circuit that reads some inputs in each cycle of operation, performs a particular calculation, and then writes the results to one or more of its output ports. All input/output operations are done based on the four-phase handshaking [6]. Figure 4 shows the template for one bit PCFB buffer.

![Figure 4. One bit PCFB Buffer](image)

The following sub-circuits can be numbered for the circuit: 1. Output generator, 2. Input validity checker, 3. Output validity checker, 4. Input acknowledge generator, 5. A sub-circuit that generates en signal. In this structure, after validation of inputs, outputs are calculated and In_Ack signal becomes high then the followings are done simultaneously: 1. output becomes neutral when Out_Ack becomes high, 2. In_Ack becomes low when input becomes neutral. In this style, a high en indicates an empty buffer whilst a low en is an indication of a full buffer. PCFB has a slack of 1 which means, is a chain of N buffers that can comprise N valid data simultaneously [1].

3.2. PCHB template

A Pre-Charged Half Buffer (PCHB) [1] functionality is very much like a PCFB. This template also uses the four-phase handshaking. Figure 5 shows the internal implementation of a simple PCHB buffer. As in this template there exist a maximum of N/2 tokens in an N-buffer chain, it is called to have a slack of 1/2.

![Figure 5. One bit PCHB Buffer](image)

3.3. STFB template

The last Dual-rail template is Single Track Full Buffer (STFB) [4]. In contrast with the previous templates in which only request signal was embedded in the data, both request and acknowledge signals are embedded in the data. An STFB buffer is depicted in Figure 6. This style is comprised of three components: 1. output generator, 2. next stage state accomplishment detector, 3. internal state accomplishment detector. When an input (Lx) is driven high by the previous stage, the corresponding NAND gate will drive Sx low, thereby driving both the corresponding Rx and Ack high. Ack going high causes Lx to reset low,
enabling the previous stage to send a new token. Meanwhile, Rx going high causes Busy to become low, restoring Sx high and preventing the NANDs to re-fire even if a new token arrives. The restoring of Sx, in turn, resets Ack. The cycle completes when the next stage lowers Rx, resetting Busy low, and allowing a new data token to be processed. This style has a slack of 1 similar to PCFB.

![Figure 6. One bit STFB Buffer](image)

### 4. Static power analysis

Static power is caused by leakage current while the gates are idle, that is, when there is no signal transition. Leakage current of the whole template is highly dependent on the input of transistors in idle states which in turn is determined by the behavior of templates. Hence, firstly we should derive idle states of each of the previously introduced styles by accurate examination of each template's behavior. To do so, an approach is to view the system as communicating blocks. Then for each template, all idle states (according to the corresponding input vector) will be determined. For each template, there are some idle states which last more than other idle states. We call these states dominant ones. Total static power dissipation is mostly dependent on these ones. Therefore to find dominant states of each template, we modeled behavior of a 5-stage 2-bit pipeline in Verilog with delay information back annotated from layout of each template (Figure 7). First, we implemented a module of token generator and a module of token sink. Token Generator generates tokens with different tunable frequencies while Token Sink waits on its input port for a new token to be received and consumed. After determining all idle states for each template, leakage current is measured for each of the idle states in TSMS 0.18um running HSPICE simulation. According to simulation results, the time duration of each idle state is dependant to the number of tokens (valid data) that exist in pipeline. Note that the slack of a template has a key role in behavior of the pipeline. One of the most influential factors of leakage current is the transistors' size. On the other hand, for correct functionality of asynchronous templates, the timing limitation must be considered, which in turn, requires proper transistor sizing. So, in order to justify our way of analysis, we determined transistor sizes of all templates equally while considering minimum area and correct functionality. In the following we explore the leakage current of different templates in detail.

![Figure 7. Pipeline under simulation](image)

#### 4.1. Analysis of HC template

The slack of HC template is 1. Our results prove that, this template goes to idle state according to the conditions shown in Figure 8.

![Figure 8. Idle states of HC](image)

Our behavioral simulation results show when the number of valid tokens is more than 1/2 number of buffers, the state with Req_L=1, T_R=0 is dominant. Having tokens less than 1/2 number of buffers, the state with Req_L=0, T_R=1 will be dominated.

Leakage current for each of the idle states are shown in figure 9. Differences in each of the states are due to the effect of transistor stacking.
4.2. Analysis of MOUSTRAP template

The slack of MOUSTRAP template is 1/2. When the number of valid tokens is more than 1/4 number of pipeline stages, the state with Rin=1 & Aout=0 is dominant. Having tokens less than 1/4 number of pipeline stages, the state with Rin=0 & Aout=1 will be dominated (Figure 10).

![Figure 10. Idle states of MOUSTRAP](image)

According to the two values of high and low inputs in an input, each of idle states are transformed to two states. Leakage current for each of the idle states are shown in Figure 11.

![Figure 11. Leakage current of MOUSTRAP](image)

4.3. Analysis of PCFB template

The slack of PCFB template is 1. Our results prove that, this template goes to idle state according to the conditions shown in Figure 12. Our behavioral simulation results show when the number of valid tokens is more than 1/2 number of buffers, the state with input-valid=1, out-ack=0 is dominant. Having tokens less than 1/2 number of buffers, the state with input-valid=0, out-ack=1 will be dominated.

![Figure 12. Idle states of PCFB](image)

According to the two values of high and low inputs in an input valid data, each of idle states are transformed to two states. Leakage current for each of the idle states are shown in figure 13. Differences in each of the states are due to the effect of transistor stacking.

![Figure 13. Leakage current of PCFB](image)

4.4. Analysis of PCHB template

The slack of PCHB template is 1/2. When the number of valid tokens is more than 1/4 number of pipeline stages, the state with Out_Ack=0 is dominant. Having tokens less than 1/4 number of pipeline stages, the state with input-valid=0, out-ack=1 will be dominated (Figure 14).

![Figure 14. Leakage current of PCHB](image)
Figure 14. Idle states of PCHB

Leakage currents of idle states are shown in Figure 15.

Figure 15. Leakage current of PCHB in TSMC 018 um

4.5. Analysis of STFB template

Idle states of this template are shown in Figure 16.

Figure 16. Idle states of STFB template

The slack of STFB template is 1 like PCFB, so pipeline behavior of this template, pertinent to the valid tokens, is equal to the PCFB template. However, state transition of each cycle in STFB is less; hence, idle states last shorter in this template.

Figure 17 shows the leakage current of each of idle states.

Figure 17. Leakage current of STFB in TSMC 018 um

5. Results and Conclusions

In this paper we demonstrated pertinence of static power dissipation of three different asynchronous pipeline templates to the behavior of these templates. Then leakage currents of these templates (as the main contributing factor to the static power) are compared with each other. The results show that the BD templates consume less leakage power compare with DD templates.

The STFB template consumes less leakage power by a factor of 3 to 4 among DD templates. Although, PCHB is composed of less logic compared to PCFB and has less leakage current in idle states, in one of its idles states it has more leakage current due to its special transistor stacking effect. Table 1 compares static power of the five templates in the conditions of high and low data traffic. STFB, HC and PCFB templates are under high traffic while the number of valid tokens is more than half number of pipeline stages. In case of fewer tokens they are under low traffic. PCHB and MOUTRAP templates are under high traffic while the number of valid tokens is more than 1/4 number of pipeline stages and under low traffic otherwise.

Table 1. Static power of different templates

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HC</td>
<td>7.55E-07</td>
<td>2.54E-07</td>
<td>5.13E-07</td>
<td>5.61E-08</td>
</tr>
<tr>
<td>MOUSTRAP</td>
<td>1.90E-07</td>
<td>1.31E-08</td>
<td>1.88E-07</td>
<td>1.57E-09</td>
</tr>
<tr>
<td>PCFB</td>
<td>1.35E-06</td>
<td>1.99E-06</td>
<td>2.49E-06</td>
<td>2.10E-07</td>
</tr>
<tr>
<td>PCHB</td>
<td>2E-06</td>
<td>1.68E-06</td>
<td>2.25E-06</td>
<td>1.55E-06</td>
</tr>
<tr>
<td>STFB</td>
<td>4.98E-07</td>
<td>5.13E-07</td>
<td>5.13E-07</td>
<td>3.32E-07</td>
</tr>
</tbody>
</table>
6. References


