Design of Low Power FPGA using Autonomous Power Gating and LEDR Encoding

Sandhya Rani.P, Venkata Satish.N
Department of Electronics and Communication Engineering,
Sri Sai Aditya Institute of Science and Technology, Surampalem, India
sandhyarani.in@gmail.com, satish.nitw13@gmail.com

ABSTRACT: The most important key challenge in the IC scaling era is to deliver high performance solutions in the process of minimizing power, area and cost. The main objective of this paper is to reduce power consumption by considering power optimization techniques using Low Power FPGA design. Due to the dynamic properties of power consumptions this paper focuses on some of the power gating techniques that is standby power and dynamic power can be reduced by LEDR (LEVEL ENCODING DUAL RAIL) Encoding. The proposed autonomous fine grain power gating method consists of LUTs. Each LUT has its own sleep transistor & sleep controller. So when any one of LUTs are inactive, they can be set to sleep mode immediately. Another proposed method is LEDR encoding is used to data flow at I/P and O/P of FPGAs. Hence power reduction is achieved by selectively setting the functional units into a low leakage mode when they are inactive.

KEYWORDS: Asynchronous FPGA, standby power, dynamic power, Power gating, LEDR (Level Encoding DUAL RAIL) encoding, look up table, sleep transistors, sleep controller.

INTRODUCTION:

Due to the spectacular raise in use of portable and battery-operated appliances, lower power consumption has become inevitability in order to prolong the battery life. Power consumption is the main part of the equation determining the end product's size, weight, and efficiency. Field programmable gate arrays (FPGA) are becoming more attractive for these applications due to their shorter product life cycle. FPGAs are programmable, so they allow product differentiation. Selecting an appropriate FPGA architecture is critical in achieving the best static and dynamic power consumption. Flash-based FPGAs by Micro semi are the low-power leaders in the industry. In addition to utilizing the low-power attributes of flash-based FPGAs, user can deploy several design techniques to further reduce overall power. The important FPGA power components to consider in the following sections:

Power-up (inrush power): Inrush power is the amount of power drawn by the device during power-up.

Configuration power: Configuration power is the amount of power required during the loading of the FPGA upon power-up (specific to SRAM-based programmable logic devices).

Static (standby) power: Static power is the amount of power the device consumes when it is powered-up but not actively performing any operation.

Dynamic (active) power: Dynamic power is the amount of power the device consumes when it is actively operating.

Sleep power (low-power mode): Some FPGA devices offer low-power or sleep modes. In some cases, this may be different from static power.

This application note focuses on reducing the dynamic power. In general, the dynamic power is calculated using the formula shown in eq.1:

\[ P = \alpha \cdot C \cdot V^2 \cdot F \]  

---
Where $\alpha$ is the switching activity, $C$ is the capacitive load, $V$ is the supply voltage, and $F$ is the frequency.

In flash FPGAs, the components that consume dynamic power are clock networks, logic blocks, routing resources (nets), I/Os, memory, PLLs, etc. These components have different $D$, $C$, $V$, and $F$ values. For example, the dynamic power of a net depends on the average switching ($D$), the total capacitive loading of the net ($C$), the net's voltage swing ($V$), and the frequency ($F$).

The total dynamic power consumption ($PDYN$) in flash FPGAs is approximated in eq.2:

$$PDYN = PCLOCK + PS_{CELL} + PC_{CELL} + PNET + PINPUTS + POUTPUTS + PMEMORY + PLL$$

--- eq.2

Where, $PCLOCK = $ Global Clock Contribution

$PS_{CELL} = $ Sequential Cells Contribution

$PC_{CELL} = $ Combinatorial Cells Contribution

$PNET = $ Routing Net Contribution

$PINPUTS = $ I/O Input Buffer Contribution

$POUTPUTS = $ I/O Output Buffer Contribution

$PMEMORY = $ RAM Contribution

$PLL = $ PLL Contribution

In FPGA design, the clock gating and power gating is important work. To implement clock gating, circulation is employed. The idea of circulation is to retain the contents of the flip-flop in the sleep state. Circulation can reduce the dynamic power consumption of registers and the gates in the fan-out of the registers. However, the standby power consumption of the clock network cannot be reduced. The standby power is a serious problem because it has an enormously large number of transistors to achieve its programmability. Low-cost FPGAs consume up to hundreds of milliwatts power. Power gating has emerged as the most effective design technique to achieve low standby power. Power gating techniques are based on selectively setting the functional units into a low leakage mode when they are inactive.

**FPGA (Field Programmable Gate Array):**

A Field-Programmable Gate Array (FPGA) is a semiconductor device containing programmable logic components called "logic blocks", and programmable interconnects. Logic blocks can be programmed to perform the function of basic logic gates such as AND, and XOR, or more complex combinatorial functions such as decoders or mathematical functions. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

**FPGA (Field Programmable Gate Array):**

A Field-Programmable Gate Array (FPGA) is a semiconductor device containing programmable logic components called "logic blocks", and programmable interconnects. Logic blocks can be programmed to perform the function of basic logic gates such as AND, and XOR, or more complex combinatorial functions such as decoders or mathematical functions. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

In an FPGA logic blocks are implemented using multiple level low fan-in gates, which gives it a more compact design compared to an implementation with two-level AND-OR logic. FPGA provides its user a way to configure:

1. The intersection between the logic blocks.

2. The function of each logic block.

Logic block of an FPGA can be configured in such a way that it can provide functionality as simple as that of transistor or as complex as that of a
microprocessor. It can be used to implement different combinations of combinational and sequential logic functions. Logic blocks of an FPGA can be implemented by any of the following:

1. Transistor pairs
2. Combinational gates like basic NAND gates or XOR gates
3. n-input Lookup tables
4. Multiplexers & Wide fan-in And-OR structure

Routing in FPGAs consists of wire segments of varying lengths which can be interconnected via electrically programmable switches. Density of logic block used in an FPGA depends on length and number of wire segments used for routing. Number of segments used for interconnection typically is a trade-off between density of logic blocks used and amount of area used up for routing.

Asynchronous FPGA Architecture:

Design: The asynchronous architecture it detects the activity of a power gated domain.

The activities are:
1) To determine when logic block is standby state, when sleep state & when active state.
2) It compares the phase of the input data and output data.
3) It determines the function of lookup table. Dynamic power reducing purpose introduce dual rail encoding (existing) and level encoding dual rail (proposed) architecture. Standby power reducing purpose introduced autonomous fine grain power gating technique. The registers store the data value and produce the output to switch block. Sleep controller monitor wake up the successive block when it gets data. The switch block consists of pass transistor switches. In a switch block, a wire-set consists of four wires: two for data lines (Vout and Rout). The circuit can be implemented by using below the architecture.

AUTONOMOUS FINE GRAIN POWER GATING:

The logic block can be connected to the switch block. Each switch block connected to the other switch block. This logic block and switch block described below the architecture diagram:

One for the acknowledge signal and one for the wake-up signal. A pass-switch block consists of four pass switches and a single memory bit. This FPGA architecture logic block can be connected to the switch block.
In the switch block there are four signals:

1) Data signal (first bit)  
2) data signal (second bit)  
3) acknowledgement signal  
4) data arrival signal.

The above four signal acknowledgement signal and data arrival signal connected to previous logic block. The two pass switches are used for the four wires of the wire-set, one Va, Ra, ACK and wakeup signal wires respectively. The pass switches are controlled by the same memory bits. No need to include external clocks or oscillations.

Fig4: logic diagram representing data and selectors of D latch of level encoding

This architecture operation based on data’s and phase signal. The D latch signal given to selectors and selectors produce the output of level encoding signal. In LEDR encoding, no spacer is required. In LEDR encoding, each data value has two types of code words with different phases. Above example shows the data values “0” and “1” are transferred. The main feature is that the sender sends data values alternately in phase 0 and phase 1. Because no spacer is required, the number of signal transitions is half of four-phase dual-rail encoding. As a result, the throughput is high and the power consumption is small. Based on this observation, in the proposed FPGA, LEDR encoding is employed for implementing the asynchronous architecture to reduce the dynamic power.

The simulation result has been shown in Fig-5, which indicates the total power consumption at various levels and compared with the existing method. It is also observed that the proposed method has improved significantly in terms of power consumption. Also total estimation of power consumption based on inputs, outputs, signals along with Vcc has shown in terms of current (mA) and power ratings (mW) in a & b.

Fig-5: Simulation result of Total estimated Power consumption

An estimated power results are as follows:

a) EXISTING:

<table>
<thead>
<tr>
<th>Power summary</th>
<th>I(mA)</th>
<th>P(mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total estimated power consumption</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>Vcc1 1.80V:</td>
<td>26</td>
<td>37</td>
</tr>
<tr>
<td>Vcc3.3 3.30V:</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Inputs:</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Logic:</td>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>Outputs:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Vcc033</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Signals:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Quiescent Vcc1 1.80V:</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>Quiescent Vcc3.3 3.30V:</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

b) PROPOSED:
CONCLUSION:

This paper focused on a low-power asynchronous FPGA architecture by adopting several power gating and encoding techniques to reduce power consumption. Design of Level encoding Dual Rail technique can effectively eliminate the excessive data transition without increasing loading on the global clock signal.

The implementation of the autonomous fine-grain power gating has been done efficiently using the standby state to wake up the Logic block before the data arrives and power OFF the Logic block only when the data does not come for quite a while.

REFERENCES:


