Moving a Processing Element from Hot to Cool Spots: Is This an Efficient Method to Decrease Leakage Power Consumption in FPGAs?

Amor Nafkha
SCEE-SUPELEC, France

Pierre Leray
SCEE-SUPELEC, France

Yves Louet
SCEE-SUPELEC, France

Jacques Palicot
SCEE-SUPELEC, France

CONTENTS

8.1 Introduction .................................................. 198
8.2 Power Consumption Sources ................................. 200
  8.2.1 Static Power ........................................... 200
  8.2.2 Short-Circuit Power ................................... 201
  8.2.3 Dynamic Power ......................................... 201
  8.2.4 Dynamic Versus Static Power ........................... 202
8.3 Digital Thermal Sensor ...................................... 202
  8.3.1 Ring Oscillator vs. System Monitor ................... 205
  8.3.2 Ring Oscillator vs. FPGA $V_{CCINT}$ Voltage Supply .... 206
8.4 Experimental Setup ........................................ 207
  8.4.1 Processing Element Number 1: Random Number Generator ........................................ 208
  8.4.2 Processing Element Number 2: MicroBlaze ............ 209
8.5 Experiments and Results ................................... 209
  8.5.1 Temperature vs. Leakage Current Consumption ...... 209
  8.5.2 FPGA Temperature Analysis ............................. 211
8.6 Conclusion .................................................. 213
8.1 Introduction

Even recently, Sustainable Development (SD) was only the concern of green
groups. However, today, SD has become a paramount issue and an aspiration
of long-term civilization development of human beings since “Resolution
42/187 of the United Nations General Assembly” was passed in December
1987. The Brundtland Commission of the United Nations defined SD as the
development that “meets the needs of the present without compromising the
ability of future generations to meet their own needs” [1]. From then on, several
United Nations’ Conferences (from Rio de Janeiro-1992 to Durban-2011)
confirmed this important issue. One of the most obvious aspects and chal-
lenges of SD is the Earth climate change and the ever-growing CO2 emission.
Currently, 3% of the world-wide energy is consumed by the ICT (Information
and Communications Technology) infrastructure, which causes about 2% of
the world-wide CO2 emissions and surprisingly is comparable to the world-
wide CO2 emissions due to all commercial airplanes. The ICT sector’s carbon
footprint is expected to quickly grow to 1.4 Giga ton CO2 equivalents by
2020, nearly 2.7% of the overall carbon footprint from all human activities [2].
These values of carbon footprint are extremely impressive. They have been
confirmed by a lot of scientific studies and reported in many relevant inter-
national conferences and workshops, such as the ”Next Generation Wireless
Green Networks Workshop” held in SUPELEC in November 2009 [3]. Basi-
cally, one should deal with the fundamental challenges with a twofold aim, in
order to attempt to solve these problems:

- Decrease the ICT footprint itself,
- Use ICT so as to decrease the “human beings” activities footprint.

This chapter deals with electronics power consumption in nanoscale CMOS
technology. Traditionally the Integrated Circuit (IC)’s power consumption was
due to dynamic power consumption. Consequently, in order to decrease fre-
quency while increasing ICs throughput, the solution was to parallelize pro-
cesses. Nevertheless one of the most significant power related subjects that
arose recently is static power leakage due to the leakage current. It was stated
in [4] that the leakage power consumption is strongly correlated to the circuit
area. Therefore a new trade-off between parallelism and power consumption
should be found. This was discussed, for example, in [5]. Leakage power rep-
resents a significant share of the total power dissipation especially for 65nm
technology and below. A lot of papers are dealing with leakage power con-
sumption itself within the framework of nanoscale technology [4,6]. If, in the
Moving a Processing Element from Hot to Cool Spots: Is This an Efficient ...

past, FPGAs were mainly concerned with dynamic power consumption [7], it remains no more true with new technologies (65nm and below) [8] and with the large number of transistors inside a FPGA circuit: the leakage power consumption becomes more and more important both in the used and unused part of the component. Section 2 will give more details on power consumption origins. A lot of works carried out in the recent years studied ways to reduce leakage power consumption in CMOS [9–11] and particularly for FPGA [12,13].

Real measurements in [14,15] show that dynamic power consumption may be decreased with the dynamic and partial reconfiguration (DPR), and this saved power consumption depends on the application and the design’s architecture. Dynamic activation and deactivation of the FPGA’s fabric clock network can be implemented to decrease dynamic power consumption. However, neglecting the reconfiguration process power consumption makes only sense if the reconfiguration time is very low compared to the application execution time’s constraints. The study in [12] indicates that the polarity of the inputs in FPGA hardware structures may significantly impact leakage power consumption (average reduction of roughly 25%). In [16,17], it has been shown that the system power consumption can be decreased by exploiting the dynamic partial reconfiguration capability of FPGAs using power reduction techniques such as clock gating, hardware deactivation and removal. To the best of our knowledge, the benefit of dynamic partial reconfiguration to reduce the die temperature and then the static power consumption has not been yet evaluated.

In this work, starting with the high correlation between chip temperature and static power consumption, associated with the possibility to move a Processing Element (PE) from a hot spot to a cooler one (already published in the literature for throughput improvement [18], and for dynamic power consumption management using DPR, see previous paragraph), we study this moving possibility so as to decrease leakage power consumption. This idea was first proposed in [19]. The aim of this chapter is to conclude on the feasibility of this idea.

To summarize, we have to answer to the question: “Is there, in a FPGA, a sufficient temperature difference between hot and colder areas to decrease leakage power consumption if a PE is moved between these two zones?” If the answer is yes, then there are subsequent questions:

- Is the temperature gradient sufficiently slow to avoid having a moving process which is repeated too often?
- Is this temperature gradient’s associated time greater or smaller than the reconfiguration time?
- Whatever the moving process is (DPR is good candidate, an high speed DPR technique has been proposed in [20]), it will introduce some power consumption overhead, Is the total power consumption budget positive?
The chapter is focused on temperature difference matter (the fundamental question) and is organized as follows. The next section gives a brief overview of the different power consumption sources and some definitions. Section 8.3 describes the Digital Thermal Sensor used to measure spots temperature. Section 8.4 gives both the hardware and software’s set-up. In this section, the two different PEs used to increase the temperature are described. Experimental results concerning the temperature difference between hot and cool spots inside the FPGA are provided and analyzed in section 8.5. Finally, some discussion and conclusions are given in the last section.

8.2 Power Consumption Sources

The total power consumption in the FPGA can be defined as: 
\[ P_t = P_{st} + P_{sc} + P_{dy}, \]
where \( P_{st} \) defines the static power that a circuit consumes even when it is in the standby mode, \( P_{sc} \) denotes the short-circuit power, which is due to the short-circuit current flowing from power supply to ground when both \( p \)-network and \( n \)-network are ON, and \( P_{dy} \) defines the dynamic power consumption that occurs when the output signal of a CMOS logic cell makes a transition. Each component of the power consumption is discussed in more detail in the following subsections.

8.2.1 Static Power

The static power is the power consumed by the device due to leakage currents when there is no activity or switching in the design. Using finer semiconductor technologies has caused an increase in static power consumption in FPGAs. In fact, as transistor size decreases and lower voltages are used, a greater leakage current occurs in the transistor channel when the transistor is in the “OFF” state. The three dominant sources of leakage current in a CMOS circuit are: sub-threshold leakage, gate leakage and drain junction leakage. Those three major leakage current mechanisms are illustrated in Fig.8.1.

The value of the sub-threshold leakage current is significantly increased with technology scaling. This component increases exponentially with temperature. However, the rate of leakage reduction with temperature is diminished with technology scaling. Note that the sub-threshold leakage current is the dominant leakage mechanism. The static power of a logic circuit can be expressed as follows [21]:

\[ P_{st} = \sum I_{leak} V_{dd}, \]  

(8.1)

where \( I_{leak} \) is the current that flows between the supply rails in the absence of switching activity, \( V_{dd} \) is the supply voltage. Reducing static power dissipation is ultimately the key to maximizing stand-by time and battery life.
8.2.2 Short-Circuit Power

During the switching operation, a conducting path exists through the pull up and pull down network of a gate and, as a consequence, a short-circuit current is occurring. In [22], the short-circuit power consumption of a simple inverter is defined as:

$$P_{sc} = \frac{\beta}{12} (V_{dd} - V_{tn} - V_{tp})^3 \frac{\tau}{T},$$  \hspace{1cm} (8.2)

where $\beta$ denotes the gain factor of an MOS transistor, $V_{tn}$, $V_{tp}$ are the threshold voltages of nMOS and pMOS transistors respectively. $\tau$ denotes the input rise and fall times, and $T$ is the period-time of a signal ($= 1/f$).

8.2.3 Dynamic Power

Dynamic power is the power consumed by the device during switching events in the core or in the input/output pins. The main variables affecting this power are the capacitance’s charging and discharging, frequency toggling, and voltage supplying. Transistors are used for logic and programmable interconnections, so in general a large portion of the dynamic power is due to the routing fabric. The formula for dynamic power is given by:

$$P_{dy} = \frac{\alpha}{2} CV_{dd}^2 f,$$

where $\alpha$ denotes the percent of circuit that switches each cycle, $C$ is the capacitance, $V_{dd}$ and $f$ are the voltage supply and the toggle frequency respectively. The core FPGA voltage supply and capacitance are reduced with each new process node technology. Note that dynamic power is not a function of transistor size, but rather a function of switching activity and load capacitance. Thus, it is data dependent. A very large number of papers have considered the distribution of dynamic power consumption in FPGAs (among them, [7]).
8.2.4 Dynamic Versus Static Power

Until recently, dynamic power has been the dominant source of power consumption. Figure 8.2 shows the cross-over point, where static power overtakes dynamic power, to be at 65 nm. Leakage power has increased drastically in the scaled technologies because of the exponential relationship of sub-threshold current to threshold voltage. Also, since the voltage supply is not scaled down as much as the feature’s size in order to have high-performance transistors, gate leakage has increased because of low gate oxide thickness. Therefore, with smaller process geometries, worsening leakage current causes static power to dominate the power consumption.

Figure 8.3 shows the total power consumption of a 15mm die fabricated, as a function of the die temperature, made of a 100 nm technology with a voltage supply of 0.7 V. Although the leakage power is about 9% of the total power at 40°C, it increases to 49% of the total power at 100°C. This clearly shows the necessity of using techniques for temperature reduction. In the rest of this chapter, we will perform a deep analysis of the temperature’s behavior inside the FPGA. Of course, for that purpose, we have to measure this temperature. The next section will describe an efficient way to perform this measurement.

8.3 Digital Thermal Sensor

The self-heating of the FPGA is a very important factor that impacts the total power consumption. On-chip measurements of local temperature present an opportunity to incorporate temperature management techniques and performance optimization into the FPGA fabric. This section presents the design of
a sensors system that continually measures the temperature values at various locations on the FPGA. The main idea behind this work is to perform a smart power management when an high local die temperature is detected. The sensors system comprises of an array of digital thermal sensor implemented on the FPGA device.

The chip’s temperature can be measured using the junction forward voltage of the clamping diodes located in the FPGA pads as suggested by [25]. This method is based on the voltage current relationship in an ideal diode. Even though of this method has the advantage of being immune to the power supply variations, it requires some additional external analog circuits. Recently, Xilinx FPGA fabric series included an analog voltage and temperature sensor connected to an A/D converter called System Monitor [26]. This sensor is located at the center of the die, and provides an average temperature. Recently infrared imaging used a powerful tool for thermal sensing [27]. Using the thermal images, authors investigated the magnitude of thermal gradients and hotspots in die. This method is able to reduce the thermal tracking errors by 40%. However, this latter technique lack from the fact that it needs some additional hardware equipment.

An original work in [28] proposes a simple way to measure chip’s heating using a ring oscillator associated to a frequency counter. This work has been completed in [29,30]. A ring oscillator consists of a feedback loop that includes an odd number of inverters needed to produce the phase shifting that maintains the oscillation. In [31], Lopez-Buedo et al. proposed the minimization of the operation frequency of a ring oscillator in order to minimize the problems related to self-heating, power consumption, and counter size. Due to the lack of knowledge of hot spot locations inside the die, all thermal sensors nodes are spread in a uniform distribution way across the FPGA according to the works published in [30,32].

![FIGURE 8.3 Power consumption of a die as a function of temperature](image-url)
Exhibiting a linear dependence of oscillation frequency on chip’s temperature, at a fixed value of $V_{CCINT}$ voltage supply, the ring oscillator circuit can be embedded inside any FPGA fabric in conjunction with a counter. It paves the way for efficient and highly accurate temperature measurement. The ring oscillator is characterised by its reduced area and power overhead which allows it to be replicated as many times as needed, so as to create a FPGA thermal map. The resulting period of the ring oscillator is defined by equation 8.4, and it is twice the sum of the delays of all inverter elements that make the loop chain.

$$T = 2Nt_p,$$  \hspace{1cm} (8.4)

where $N$ denotes the number of odd inverters and $t_p$ is the propagation delay of a single inverter that is part of the loop.

Figure 8.4 shows the RTL schematic of the proposed temperature sensor based on a ring oscillator. The thermal sensor can be divided into three major modules. The first module of the sensor is made of a ring oscillator controlled by an external enabled signal, the second module is 12-bit cyclic counter and the last module is a 14-bit reference time counter which indicates the number of rising edge reference 50 MHz clock between two events.

During the process, the first 12-bit counter is clocked by the output of the ring oscillator module, and generates a boolean event to the second 14-bit reference time counter. This boolean is equal to “true” if the 12-bit counter value is equal to $2^{12}$, otherwise the signal’s value equals “false”. The reference 14-bit counter determines the number of rising edge reference clocks that have been counted between two “true” events. The reference counter value is sent back to the PC via the RS232 port (runs at 19.6kHz). The FPGA temperature is gathered from an on-chip hardware sensor called System Monitor which is located at the center of the FPGA’s core and developed around a 10 bits analog-to-digital converter.
FIGURE 8.5 Measured ring oscillator’s frequency in MHz (million-Hertz) versus FPGA’s temperature in degree Celsius

8.3.1 Ring Oscillator vs. System Monitor

In order to analyse the relationship between the FPGA’s temperature and the ring oscillator’s frequency, we have used a Peltier element. To provide a large and stable FPGA’s temperature range, one side of a Peltier cell is placed on the FPGA and the other side is connected to an aluminum heatsink. To keep the system flexible and responding fast when cooling down is desired, it was useful to keep the heatsink always attached. The Peltier cell is able to provide a stable temperature within our desired range (25-85°C). We can catch the desired temperature by increasing or decreasing the Peltier current/voltage supply. The ring oscillator’s inverters are implemented in LUT (Look Up Table), and their RLOC (Relative location Constraints) were used to equalize the physical distances between inverters.

To have a good estimation of the ring oscillator frequency as a function of an FPGA’s temperature, we have placed four sensors inside the FPGA at the following locations: Northeast (NE), Southeast (SE), Southwest (SW) and Northwest (NW) (Figure 8.6). In Figure 8.5, the y-axis represents the ring oscillators’ frequencies and the x-axis represents the FPGA’s temperature given by the System Monitor. We noticed that the experimental results are given when the FPGA $V_{CCINT}$ voltage supply is equal to 0.998V. The variation of frequency between the different locations is due to the propagation delay’s
variation which exists between the ring oscillator’s inverters and the various locations, and which is a Xilinx place-and-route related phase. Experimental results show that there is indeed a high correlation and proportionality between the propagation delay inside a ring oscillator and the FPGA’s temperature. Linear approximation seems to be a near fit of the recorded curve. Below is the equation that expresses the mean ring oscillator’s frequency value for FPGA’s temperature conversion, by means of the traditional linear first-order fitting equation:

\[ f = -0.0094 \times T + K, \]  

(8.5)

Where \( f \) is the ring oscillator’s frequency in MHz, \( T \) is the temperature in degree Celsius(°C), and \( K \) is a calibration constant which can be easily calculated for a given initial temperature and ring oscillator’s frequency. The frequency value varies linearly with temperature. The slope of the line, equal to 9.4KHz/°C, gives us the effective change in ring oscillator frequency corresponding to an increase or decrease of temperature.

8.3.2 Ring Oscillator vs. FPGA V\textsubscript{CCINT} Voltage Supply

One of the problems faced, in the experimental measurement of the ring oscillator’s frequency, is a significant variation of the gates’ propagation delay, which is due to the FPGA’s V\textsubscript{CCINT} voltage supply’s change. The impact of the different voltage supply scaling techniques on the propagation delay of the CMOS gates has been widely investigated in [33–36]. It is well known that the voltage supply’s scaling is one of the most used technique to reduce the power consumption. However, this approach has not been used in the present chapter. Figure 8.7 shows the mean ring oscillator’s frequencies when the FPGA’s V\textsubscript{CCINT} supply voltage varies from 0.97V to 1.016V, for a given chip’s temperature. Scaling down V\textsubscript{CCINT} alleviates the ring-oscillator’s propagation delay,
whose increase is an issue. As a consequence, the frequency is decreased. In order to provide various on-board $V_{CCINT}$ voltage values, the used Xilinx ML550 Networking Interfaces platform implements a margin control circuitry providing $\pm 2.5\%$, $\pm 5\%$, $\pm 7.5\%$, and $\pm 10\%$ voltage supply’s adjustments. In order to overcome the problem of the supply sensitivity of our ring oscillator, we will be using a fixed value of $V_{CCINT} = 0.998V$, voltage supply, for all results given in the rest of this chapter.

8.4 Experimental Setup

We have performed experiments on the Virtex-5 FPGA to study the FPGA power consumption’s behaviors. The ML550 development board, (see Figure 8.8, from Xilinx) would be well suited to power consumption’s measurement for several reasons [37]. The FPGA on this board is a XC5VLX50T which is a Virtex5 LX series. This board provides us with 5 power rails (core, IOs, peripherals) and current sense resistors which could simplify the experimental measurement. The Virtex5 FPGAs generally requires at least two different voltages. The recommended Virtex5 core voltage, designated $V_{CCINT}$, is $1.0 \pm 10\%V$. Depending on the I/O standard being implemented, the Virtex5 I/O voltage supply, designated $V_{CCO}$, can vary from 1.2V to 3.3V. Moreover, Xilinx defines an auxiliary voltage, $V_{CCAux}$, which is recommended to operate at $2.5 \pm 10\%V$ to supply the FPGA’s clock resources. The Virtex5 FPGA also contains a System Monitor which features on-die temperature and voltage measurement capabilities that provide valuable information for the development, evaluation and quality assurance process of power designs. The board...
hosts a two header connector which provides test points for the ML550 power regulators. Moreover, to measure different currents drained by the FPGA, the ML550 board contains a series shunt 10\,m\Omega \pm 1\% (named 3W Kelvin current sense resistors) on each voltage’s regulator lines. Thus, the current will be the voltage across the shunt resistor divided by the resistance value itself. Since the sensitivity of the $V_{CCINT}$, $V_{CCO}$, and $V_{CCAux}$ sensors are pretty low ($0.5 – 2\,mV$), voltage amplifiers are needed. To this end, we have used integrated instrumentation amplifiers AD620, from Analog Devices. The AD620 has a feature to increase gain between $1 – 10,000$ times with an external resistor. Those amplifiers permit to regulate the gain, thanks to the sole change of a single resistance, called $R_G$. We wanted to reach, at the output of the voltage amplifier, a voltage value ranging between $100 – 400\,mV$, then, Depending on the component’s availability in the laboratory, gains of about 100, 275, and 2750 respectively for the $V_{CCINT}$, $V_{CCO}$, and $V_{CCAux}$ were established using resistances of 449, 180, and 18\,\Omega.

Figure 8.8 shows the schematic of connections between ML550 sensors and amplifiers. The output voltage pins from the board are the differential outputs from the sensors. In order to observe the behaviour of the ring oscillator across a wide temperature range, the Peltier module is installed above the virtex5 chip. In the rest of this section, we will introduce different hardware/software modules used for different experiments. Since we would like to implement two types of PEs: a combinatorial based one and another one which is software based we define the two following PEs.
8.4.1 Processing Element Number 1: Random Number Generator

In order to increase the temperature at one of the thermal sensor location inside the FPGA region, we have designed a parametric processing unit based on the random number generator. A 512-bit random number is being generated with the help of the exclusive OR (XOR) gate. The most significant digit and the least significant digit are both processed through the XOR gate and give the first bit of the random number and then this equation runs inside the loop 512 times so as to give a 512 bits random number. So the generated number contains 512 random bits per clock cycle. To significantly increase the temperature, the used clock’s frequency can be doubled by using the Digital Content Manager (DCM)’s components inside the FPGA.

This PE is a combinatorial type one whereas the next PE described in the following section is software based.

8.4.2 Processing Element Number 2: MicroBlaze

In order to increase the temperature of a predefined place in the FPGA, a MicroBlaze embedded soft-core processor has been implemented. The advantage of using this type of soft processors is that they are highly reconfigurable and can be customized according to our needs. Here, the Microblaze runs at 125 MHz, the floating point unit being enabled, and we chose a 64 × 64 standard matrix multiplication as an application target. MicroBlaze uses both the instruction and data sides, and its local memory buses connect it to a dual-ported primarily on-chip block random access memory, via separate interface controllers.

8.5 Experiments and Results

Power consumption and performance measurement were done for a Xilinx Virtex5 FPGA. As said in section 8.3.2 the results are based on the measurements made at a nominal VCCINT level of 0.998V. The clock frequency was kept fixed at 100 MMz. The relationship between temperature, leakage power consumption and the on-chip temperature has been explored and the results are presented in the following subsections.

8.5.1 Temperature vs. Leakage Current Consumption

In order to measure leakage current for different die-temperatures, experimental and analytical results are carried out using Agilent 34401A multimeter and Xilinx XPower analyzer tool (XPA) [38], respectively. The XPA is part of the
Xilinx ISE design suite and provides an accurate way to analyze the power profile of post Place & Route designs. Table 8.1 shows the measured leakage current for the various die-temperature. At that level we are considering current leakage rather than power leakage (bearing in mind that flicking through both notions is obvious) in order to be in accordance with the Xilinx general curve from [38] given in Figure 8.9.

**TABLE 8.1**

<table>
<thead>
<tr>
<th>FPGA</th>
<th>Leakage Current consumption using experimental and XPA tool.</th>
</tr>
</thead>
<tbody>
<tr>
<td>T(°C)</td>
<td>29</td>
</tr>
<tr>
<td>XPA (mA)</td>
<td>298</td>
</tr>
<tr>
<td>Exp(mA)</td>
<td>314</td>
</tr>
</tbody>
</table>

Measuring voltages across different sense resistors enables the computation of the FPGA $I_{CCINT}$ leakage current consumption at any time. Figure 8.10 shows the leakage current consumption measurements corresponding to different FPGA’s fabric temperature. Data points were obtained via the running of each experiment until temperature stabilization. This takes several minutes.

In Figure 8.10, the continuous line presents the best-fitting quadratic func-
Moving a Processing Element from Hot to Cool Spots: Is This an Efficient ...

211

FIGURE 8.10 Leakage current’s consumption versus FPGA’s fabric temperature. As regard the temperature $T$ parameter, and the data of the present experiment. It can be defined as:

$$I_{CCINT} = 0.0899T^2 - 2.889T + 321.724,$$

where $T$ is the junction temperature in degree Celsius ($^\circ$C), and $I_{CCINT}$ is the leakage current in milliAmper (mA). An important conclusion that can be drawn here is that for low-temperature range (30-40$^\circ$C), ten-degree temperature variation can cause a small increase of the leakage current (roughly 33 mA). By contrast, the same temperature range (60-70$^\circ$C) in a hot FPGA chip significantly affects the leakage current (roughly 90 mA).

8.5.2 FPGA Temperature Analysis

Figure 8.11 and figure 8.12 show the sensors and processing elements location and the experimental temperature map of the Xilinx Virtex5 FPGA. In order to perform this measurement, four temperature sensors are inserted into the four corners of the device named respectively NE, NW, SE, and SW. Data-intensive processing elements (as defined in section 8.4) have been inserted into the NW location. The core idea here is that we are able to increase locally the temperature of the NW position by running the processing elements. Moreover, the four sensors give us the temperature’s evolution of the FPGA
at each corner when the processing elements are running. We assume that a linear temperature gradient exists inside the FPGA. Consequently, we are able to draw the thermographs presented in the following figures, thanks to the four sensors. Furthermore, the temperature values are those obtained after a long time (several seconds) in the steady state behavior, this means we have not considered in this analysis the temperature peaks, which have very short duration of less than one ms. For each experiment, the temperature’s steady state before the PE running is homogeneous in all the locations of the FPGA.

Figure 8.12 presents the result obtained with the PE number one (the Random Number Generator). The steady state’s temperature prior to the running of the PE is equal to 32°C. The maximum temperature’s increase is 5°C. The difference between the hottest and coolest hotspots after several seconds is roughly 2°C.

Figure 8.11 presents the same type of results obtained with PE number 2 (MicroBlaze). The steady state’s temperature prior to the running of the PE is equal to 40°C. The maximum temperature’s increase is 6.5°C. The difference between the hottest and coolest hotspots after several seconds is roughly 2.5°C.

Based on the FPGA’s thermal imaging result, the main result we obtained here is that for any given location of the hot Processing element (MicroBlaze runs at 125 MHz/RNG runs at 100 MHz), the temperature difference between the hot and colder FPGA regions don’t exceed 3°C once temperature has stabilized.

To get a better information of the impact of the filling rate of the FPGA
Moving a Processing Element from Hot to Cool Spots: Is This an Efficient ...

FIGURE 8.12 Thermograph obtained by an uniform interpolation of sensors responses: RNG case

on the temperature’s difference between the hot and cooler FPGA regions, figures 8.13 and 8.14 show the thermal map after FPGA heating with 8 and 18 PEs number one running at 100 MHz. In the case of a filling rate of 50%, the temperature’s difference increases and reaches 16°C. This latter difference is very optimistic and the mean temperature of the FPGA can not be reduced by this value. Moreover, the leakage current’s consumption results given in figure 8.9 are produced for a mean FPGA’s temperature. So, we can expect lower leakage current’s consumption reduction, when we reduce locally the temperature, as compared to Xilinx results.

Figure 8.15 illustrates the temperature’s difference between hottest and coldest spots as a function of the FPGA’s occupancy. As previously, the occupancy is performed with several PEs number one running at 100 MHz. As expected the difference increases vs the occupancy rate to a maximum when the rate is equal to 50%. Then, as this can be easily understood, the difference decreases, since the temperature of the coolest spot in this zone increases, when PEs are implemented in the free zone. From Figures 8.14 and 8.15 we conclude that the maximum temperature’s difference is obtained with a filling rate of 50%. In this situation if we use this gain value of 16°C on figure 8.10 the associated power gain value obtained is lower than 100 mW. As already explained, this value is very optimistic because leakage power consumption gains in figure 8.10 are given for mean temperature of the FPGA.
FIGURE 8.13 Thermograph obtained by an uniform interpolation of sensors responses: 8 PEs N1.

FIGURE 8.14 Thermograph obtained by an uniform interpolation of sensors responses: 18 PEs N1.
FIGURE 8.15 Temperature’s difference between hottest and coolest spots vs. FPGA occupancy.

8.6 Conclusion

As stated above, the static power is mostly due to sub-threshold leakage current. This current, although is small for low FPGA’s temperature, increases exponentially with temperature. In order to reduce power consumption by means of reducing hotspot’ temperature, we have studied in depth the temperature’s difference between hot and cooler spots inside the FPGA. On the basis of the results given in the section 8.5 and the original work published by S.Liu et al in [39] a detailed analysis has brought us to the following conclusions:

- The temperature’s differences between the hotter and colder FPGA regions do not exceed 16°C, so moving hotspots to colder regions can not save more than 100 mW when the die’s temperature range is (35-50°C). This optimistic gain is made under the assumption than moving a PE will result in a decreasing temperature on the spot, which is not so obvious as we experimented it.

- On the basis of the original work of S.Liu et al in [39] dealing with partial reconfiguration to reduce Virtex4 power consumption, we are able to have a good estimation of the chip’s power consumption during the partial reconfiguration’s processing time (≈600 mW). This power consumption overhead is larger than the saved one given in the above item. Therefore,
with current available technologies (mainly DPR), there is no gain in static power consumption by moving PEs from a hot to a cool spot.

The analysis performed in this chapter shows that two possibilities exist so as to decrease leakage power consumption:

- Decreasing the overall area,
- Decreasing the temperature.

The previous conclusions said that decreasing the temperature inside one FPGA is not efficient. This study provides answers to the questions raised in the title of the chapter. Taking into account these conclusions, we are currently investigating two ways to decrease leakage power consumption.

The first one meets the objective of decreasing the overall area. In fact for a given application (Radio access technology, for example), traditionally, designers used a sufficiently big FPGA to run all the PEs needed by the application. Our proposal, consists in using a small FPGA, thanks to a DPR to fill it, on the fly, with the required PEs needed by the application. To fill the required PEs at the right time will, of course, need some additional scheduling which should also be taken into account in the global power budget.

The second method we are investigating, meets the objective of decreasing the temperature. We are investigating power saving in the case of a multi-FPGA architecture. The main idea behind this study is that instead of using one high density FPGA to implement the application, it should be more efficient to use multiple low density FPGAs. In this case, we are able to maintain a high temperature’s difference between adjacent FPGAs. Again we will use DPR to move PE from one FPGA to another one.

Bibliography


Moving a Processing Element from Hot to Cool Spots: Is This an Efficient ...


[18] A. Gupte and P. Jones, “Hotspot mitigation using dynamic partial recon-
configuration for improved performance,” in Proceedings of the 2009 Inter-
national Conference on Reconfigurable Computing and FPGAs, ser. RE-

communications: power consumption consideration,” in Proc. ISRSSP
2010 - 2nd International Symposium on Radio Systems and Space

face for realtime partial reconfiguration of FPGA.” in Proc. International
conference on ReConFigurable computing and FPGAs, 2009, pp. 386–391.


[22] H. Veendrick, “Short-circuit dissipation of static CMOS circuitry and
its impact on the design of buffer circuits,” IEEE Journal of Solid-State

[23] “Power consumption in 65 nm FPGAs,” Xilinx, Inc.

and minimization in CMOS VLSI circuits,” IEICE Trans. Electronics,
Special Section on Low-Power LSI and Low-Power IP, vol. E88-C, no. 4,


[27] A. N. Nowroz and S. Reda, “Thermal and power characterization of field-
programmable gate arrays,” in Proceedings of the 19th ACM/SIGDA in-
ternational symposium on Field programmable gate arrays, ser. FPGA

[28] N. P. G. Quenot and B. Zavidovique, “A temperature and voltage mea-

[29] K. M. Zick and J. P. Hayes, “On-line sensing for healthier FPGA sys-
tems,” in Proceedings of the ACM/SIGDA 18th International Symposium
on Field Programmable Gate Arrays, FPGA 2010, Monterey, California,

FPGAs using,” in Proc. FPL 1997 Workshop, Lecture Notes in Computer
Moving a Processing Element from Hot to Cool Spots: Is This an Efficient ... 219


[37] “Virtex-5 FPGA ML550 networking interfaces platform user guide,” Xilinx Inc.

[38] “Virtex-5 FPGA system power design considerations: White paper,” Xilinx Inc.


Author Contact Information

Amor Nafkha, Pierre Leray, Yves Louet, and Jacques Palicot, SCEE-SUPELEC, France, Email: anafkha@rennes.supelec.fr, pierre.leray@supelec.fr, yves.louet@supelec.fr, jacques.palicot@supelec.fr.