Power Estimation Approach for SRAM-based FPGAs

Karlheinz Weiß, Carsten Oetker, Igor Katchan, Thorsten Steckstor, Prof. Dr. Wolfgang Rosenstiel

Forschungszentrum Informatik (FZI), Embedded System Design Group, Haid-und-Neu-Straße 10, D-76131 Karlsruhe
and the University of Tübingen, Technical Computer Science, Sand 13, D-72076 Tübingen, Germany

weiss@fzi.de, oekter@fzi.de, katchan@fzi.de stecki@fzi.de, rosen@informatik.uni-tuebingen.de

Abstract

This paper presents the power consumption estimation for the novel Virtex architecture. Due to the fact that the XC4000 and the Virtex core architecture are very similar, we used the basic approaches for the XC4000-FPGAs power consumption estimation and extended that method for the new Virtex family. We determined an appropriate technology-dependent power factor K_p to calculate the power consumption on Virtex-chips, and developed a special benchmark test design to conduct our investigations. Additionally, the derived formulas are evaluated on two typical industrial designs. Our own emulation environments called SPYDER-ASIC-X1 and SPYDER-VIRTEX-X2 were used, which are best suited for the emulation of hardware designs for embedded systems.

1 Introduction

In the last few years, the significant increase in on-chip gate capacity and the simultaneous decrease in the run-times through programmable logic cells, provided by novel FPGA architectures, has led to a dramatic increase in the power consumption of FPGA devices. That fact depends basically on two physical factors. The first factor is the circuit speed (in terms of average clock frequency) and the second is the utilized chip area (in terms of routing length, utilized programmable logic cells and occupied *IOs*). Under appropriate operating conditions, these devices can consume several amperes of current.

In order to guarantee proper system operation, this FPGA power consumption must be estimated in an early design step to develop an appropriate board design, a power supply unit, and provide sufficient ambient conditions to carry off the heat during the hardware design cycle.

This paper addresses that problem and focuses mainly on estimating the power consumption for the novel *Virtex*[1]-FPGA architecture [1]. Chapter two gives an overview of the previous work, especially the state-of-the-art estimation method applied to the widely introduced *XC4000* architecture, which shows a large similarity to the *Virtex* architecture. This published estimation approach used for the *XC4000*-FPGAs will be extended in chapter three to derive formulas to calculate power consumption for the *Virtex* architecture. Chapter four describes the tool set of our own emulation environment SPYDER[2], which was developed to emulate large hardware designs on *Virtex* or *XC4000* FPGA target architectures. These tools were used to get the experimental results in chapter five. Additionally, chapter five draws a clear dividing line for which kind of design structures the derived formulas can be successfully applied or not. Chapter six summarizes the work and chapter seven gives a brief future outlook.

2 Previous Work

A general introduction into the area of power estimation for FPGAs can be found in [2] and [3]. The total estimated power consumption P_{EST} consists of three parts, as shown in formula 1:

$$P_{EST} = P_{STAT} + P_{IO} + P_{INT} \qquad (1)$$

The static power consumption P_{STAT} is caused by the on-chip leakage current, which can be found in the appropriate device data sheet. That current is typically in the range of a few micro-amperes and must be multiplied with the device core voltage V_{Core}. The computed P_{STAT} value resides in the range of a few micro-watts and can be ignored in most power estimations, except for extreme low power applications, e.g. for handheld devices.

The second part, $P_{IO}=P_{DCOUT}+P_{ACOUT}$, consists of two additional sub-parts, which must be calculated with the following formulas 2 and 3:

$$P_{DCOUT} = \sum_{n=1}^{d} P_{DCn} \qquad (2)$$

[1] *Virtex* is a trademark of Xilinx Inc, San Jose

[2] This work was supported in part with funds from the Deutsche Forschungsgemeinschaft under reference number 3221040 within the priority program “Design and Design Methodology of Embedded Systems”.

$$P_{ACOUT} = \sum_{n=1}^{a} C_n \cdot V_n \cdot f_n \cdot V_{CCIO} \qquad (3)$$

Where: d = number of DC outputs
P_{DCn} = DC output power of output n
a = number of AC outputs
C_n = capacitive load on output n
V_n = voltage swing of output n
f_n = switching frequency of output n

The P_{DCOUT} value depends on the number of steady-state outputs. The P_{DCn} values can be found in the appropriate device data sheet. The P_{ACOUT} value depends on the capacitive loads and the switching frequency. The C_n and V_n value can also be found in the data sheet, but the f_n must be extracted from the application running on the FPGA and depends on the average ratio of toggling output pins. A detailed description for computing f_n and P_{ACOUT} can be found in [2].

For FPGAs with a small on-chip gate count and many *IOs*, the P_{IO} plays the major role in total power consumption. In this paper, we work with the novel *Virtex* chips, which provide a significant increase in on-chip gates and clock speed. That means the third part P_{INT} increases and plays the major role in the estimation of power consumption. This will be gone into in more detail in chapter 2.1.

2.1 Internal Power Estimation for *XC4000*

The internal power consumption is caused by the charging and discharging of the capacitance on each internal node that is switched. As described in [4], the internal power consumption P_{INT} can be computed with the following formula, which also plays an important role in our approach described in chapter 3:

$$P_{INT} = V_{Core} \cdot K_p \cdot f_{Max} \cdot N_{LC} \cdot Tog_{LC} \qquad (4)$$

Where: V_{Core} = core Voltage
K_p = technology dependent constant
f_{max} = maximum clock speed
N_{LC} = number of logic cells used
Tog_{LC} = average of logic cells toggling at each clock cycle

device family	K_p power factor (x 10^{-12})	V_{Core}
XC4000E	72	5.0 V
XC4000EX	47	5.0 V
XC4000XL	28	3.3 V
XC4000XLA	17	3.3 V
XC4000XV	13	2.5 V

Table 1. Technology power factors for the *XC4000*-family

As shown in table 1, the power factors published by Xilinx for the *XC4000* devices decreases from $72 x 10^{-12}$ to $13 x 10^{-12}$ mainly due to the fact that the chip technology decreases to the deep sub-micron range, which forces also the core voltages to decrease from 5 V to 2.5 V. The N_{LC} value describes the used logic cells in the design. Typically, the designer gets only the value of used *CLBs* (n_{CLB}) from the design software tool. Therefore, the relationship between N_{LC} and *XC4000-CLBs* is given as $N_{LC} = 2.375 \text{ x } n_{CLB}$.

The Tog_{LC}-value must also be discussed. This value describes the average number of logic cell nodes toggling at each clock cycle. In [4], some recommended average values are published (e.g. 20% in most designs, 12.5% for 16-bit counters). Additionally, as mentioned in [3] and [4], a detailed knowledge about the internal design behavior related to the average toggling rate can only be reached through performing simulation, which is limited by the percentage of real-time stimulus that can be applied, especially for very large designs.

3 Estimating Power Consumption for *Virtex*

If we take a look at the internal *CLB* structure (*XC4000*) and *Slices* structure (*Virtex*), we can find a significant structural similarity. Thus it makes sense to extend the approach used to estimate internal power consumption for *XC4000*-chips to the new *Virtex*-FPGAs.

We used the formula given in equation 4 for calculation of the internal power for *Virtex* chips. The first problem is to determine an appropriate technology power factor $K_{P(Virtex)}$ as shown in the following formula, which has not been previously published:

$$K_{P(Virtex)} = \frac{P_{INT(measured)}}{V_{Core} \cdot f_{Max} \cdot N_{LC} \cdot Tog_{LC}} \qquad (5)$$

In order to compute $K_{P(Virtex)}$, we used a special benchmark design. This design operates on a *Virtex* chip provided by our own emulation platform called SPYDER-VIRTEX-X2, which will be introduced in the next chapter. This tool enables us to separately measure the *Virtex* core current (I_{Core}) and *IO* current (I_{IO}).

If using a *Virtex* FPGA, the core voltage V_{Core}=2,5V must be applied, and f_{Max} is indicated in the appropriate report file provided by the design software tool. Additionally, in another report file, the designer can get the used *Slices* (n_{Slice}) on the *Virtex* chip. Based on the *Virtex* data sheet in [1], the relationship between the number of used logic cells N_{LC} and the *Slices* can be determined as: $N_{LC} = 2.25 \text{ x } n_{Slice}$.

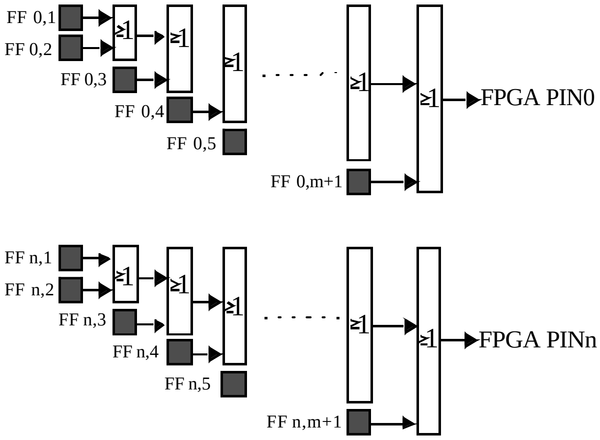


Figure 1. Structure of the benchmark test design

The last important value in equation 5 is the average toggling ratio Tog_{LC} at each clock cycle. As mentioned above, this value greatly depends on the investigated design and can only be exactly determined by performing a detailed simulation.

In order to avoid such a simulation step, we used a special benchmark test design, whose basic idea is to implement as many *Flip-Flops* (*FF*) as possible in combination with a reasonable amount of random logic in order to get a high chip utilization. The random logic will be generated through a disjunction of all internal generated *FF*-nodes. The result is one signal, which will be connected to a dedicated output pin. The structure of this design is depicted in figure 1, followed by the appropriate *VHDL* code in figure 2. This method forces the synthesis tool to implement all *FF* without deleting the *FF*-nodes during optimization steps. Additionally, each *FF* will be switched in each clock cycle in order to get as many changing *FF*-nodes as possible to force a worst case scenario on the chip. A side effect using this approach is the easy determination of Tog_{LC}, which can be directly set to 100% without simulation.

```
OR_TEMP: PROCESS(FlipFlops, TEMP)
BEGIN
   FOR J IN 0 TO (MAX_IO-1) LOOP
      TEMP(J)(1)<=FlipFlops(J*MAX_OR)OR FlipFlops(J*MAX_OR+1);
      FOR I IN 2 TO (MAX_OR-1) LOOP
         TEMP(J)(I) <= TEMP(J)(I-1) OR FlipFlops(J*MAX_OR+I);
      END LOOP;
   END LOOP;
END PROCESS;
```

Figure 2. *VHDL* code of the benchmark design

After reset, all *FF* will be cleared to zero. During operation, all *FF* toggle between one and zero simultaneously in each clock cycle. Due to the fact, that all FF-nodes have the same value at a certain point in time, all inputs to the OR-gates carrys the same value. That means the intermediate values between the OR-gates simultaneously toggles also in each clock cycle, which forces additionally maximum activity in the random logic and on the routing lines as well.

The *VHDL* code consists of two nested loops. The first loop is controlled through the loop value MAX_IO, which defines the number of toggling IO-pins for different *IO*-current measurements.

The second value MAX_OR controls the inner loop, which can be used to change the number of *FF* and the number of sequential OR-stages. With this parameter, the chip utilization can be scaled for different core current measurements. This benchmark test design was used for a couple of experiments conducted on the Spyder-System, which will be introduced in the next chapter. In chapter five, the experimental results based on this benchmark test design will be further described.

4 Emulation Environment Spyder

The basic idea of the Spyder-System is to get a detailed view into the internal system behavior of complex embedded systems based on real-time emulation. Currently, the Spyder-System consists of a tool set of four compatible platforms. Spyder-Core-P1/PPC403 is an emulation environment based on the embedded PowerPC PPC403, which is closely connected to different FPGAs to emulate typical embedded target systems in the area of industrial automation, automotive and telecommunication. In the past, the tools were used in different research projects published in [5] [6] [7]. The next generation is currently under development and is called Spyder-Core-P2/SH3. It is based on the novel compatible Hitachi embedded controllers *7709A* and *7729-DSP*.

The Spyder-Asic-X1 is a low cost evaluation platform for *XC4000*-based FPGA designs. That platform will be further described in chapter 4.2.

Additionally, a high-end platform for the emulation of large FPGA-designs, e.g. IP-Cores, under real-time conditions, was developed. It is the fourth tool of the Spyder-System and is called Spyder-Virtex-X2, which will be used for the experimental work described in this paper. For more information about the Spyder-System, visit the web page at http://www.fzi.de/sim/spyder.html.

4.1 Emulation Platform Spyder-Virtex-X2

The basic component is a *Virtex* FPGA with the package type BG432. Therefore, FPGA chips in the range from *XCV300* up to *XCV800* can be implemented. The architecture of Spyder-Virtex-X2 is depicted in figure 3.

The *Virtex* FPGA is closely coupled via a dedicated PCI-Interface-chip (*PLX9080*) to a development PC. This feature enables both the easy download of bit images into the *Virtex* chip and communication between the PC and the application operating in the *Virtex* FPGA via the PCI-bus, which provides a high performance bandwidth. The communication is a means to evaluate a running application, e.g. a dedicated IP-Core, before its integration into an embedded system. Using the PC with its entire periphery (e.g. display, hard disk, keyboard) instead of a specialized microcontroller makes that evaluation process much easier.

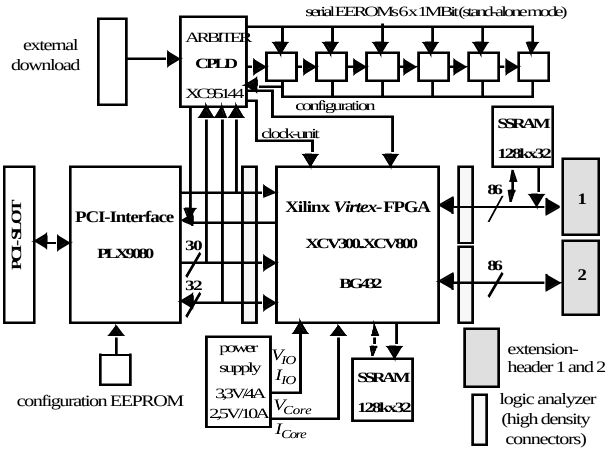


Figure 3. Architecture of Spyder-Virtex-X2

Two powerful extension headers make it possible to connect the *Virtex* FPGA with further application- specific hardware units, e.g. a microcontroller and its core environment, to assemble a complete embedded system architecture for emulation purposes. These ports are compatible via a backplane with the other tools of the Spyder-System mentioned above, which provides different microcontroller types.

A further significant feature is the ability to connect all on-board signals via up to nine high density connectors to a logic analyzer. These connectors provide a powerful support during the debugging process.

A power supply unit provides the *Virtex* FPGA with the necessary voltages: V_{Core} = 2.5V with a current of up to 10 A and V_{IO} with a current of up to 4 A. Two current measurement instruments can be connected inside the different current path systems for I_{Core} and I_{IO} to measure the power consumption. This feature was used for the experiments described in chapter 5.

An arbiter controls the local side of the PCI-bus between PLX9080 and three different download modes, which can be summarized as follows:

- download via PCI-bus, set *Virtex* in slave mode
- download via external master, e.g. a microcontroller-unit, set *Virtex* in slave mode
- download via serial EEPROMs, set *Virtex* in master mode, used for stand-alone mode

Additionally, two on board 128kx32 SSRAM-devices enable the emulation of applications, which need a large extension memory for, e.g. graphic or large filter applications

For more information, refer to the corresponding user manual and data sheet in [8].

4.2 Emulation Platform SPYDER-ASIC-X1

This tool is a low cost platform based on the *XC4000* architecture. The implemented package type enables applications on different *XC4000*-FPGAs in the range from *XC4010* up to *XC4085* chips. Different on board voltages provide the power supply for the E, EX, XL and XLA technologies of the *XC4000* family.

The basic idea and the features of this tool are similar to the high-end version of SPYDER-VIRTEX-X2, except for the limitation in bandwidth to the PC via the AT-ISA-Bus and a smaller gate capacity of the used *XC4000*-FPGAs. The appropriate architecture is shown in figure 4.

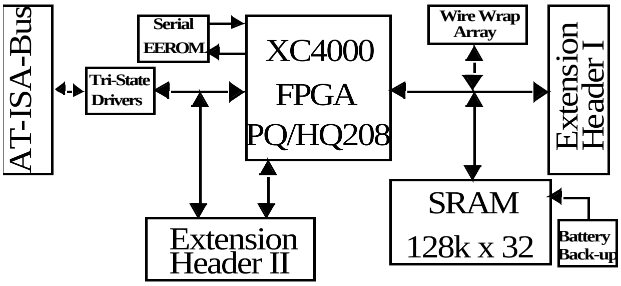


Figure 4. Architecture of SPYDER-ASIC-X1

This platform was also used for the power consumption experiments, which will be described in the next chapter. For more detailed information about SPYDER-ASIC-X1, refer to the corresponding user manual and data sheet in [9].

5 Experimental Results

After the introduction of the SPYDER tool set, which was developed by us, we will give the experimental results.

First, we performed some measurements to determine the mentioned K_P technology power factor for *Virtex* as introduced in chapter three. Then we evaluated the formulas 1, 2 and 3 on *Virtex* chips based on our benchmark test design, which is also described in chapter three. Additionally, we compared our results with the „*Virtex Power Estimator*" tool provided by Xilinx via the internet (see web page: http://www.xilinx.com/support/techsup/powerest/) and the description in [10].

We used a SPYDER-VIRTEX-X2 board with a XCV300BG432 device to determine the $K_{p(Virtex)}$ technology power factor. We applied the benchmark test design depicted in figures 1 and 2 to utilize the *Slices* of the *Virtex* chip with 4%, 43% and 97%, as shown in table 2.

util.	4%	Slice	43%	Slice	97%	Slice
f_{Max}	P_{INT} measured	computed K_P	P_{INT} (measured)	computed K_P	P_{INT} (measured)	computed K_P
MHz	W	10^{-12}	W	10^{-12}	W	10^{-12}
10	0,050	6,584	0,320	4,214	0,678	4,021
20	0,090	5,926	0,620	4,082	1,258	3,727
30	0,135	5,926	0,930	4,082	1,975	4,335
40	0,175	5,761	1,230	4,049	2,378	3,523
50	0,218	5,741	1,518	3,998	3,180	3,769
66	0,280	5,586	1,953	3,897	4,013	3,603

Table 2. Measured Core Power and computed K_P

In the case of using a *XCV300* chip, 4% utilization equals 135 *Slices*, 43% equals 1350 *Slices* and 97% equals 3000 *Slices*. If we insert the measured P_{INT} (see table 2), V_{Core} = 2.5V, f_{Max} according to table 2, N_{LC}=2.25 x n_{Slices} and Tog_{LC}=1 into equation 5, we can compute the appropriate K_P factors, which are also depicted in table 2. Additionally, we can compute an average technology power factor as follows:

$$K_{P(average)} = \frac{1}{n} \cdot \sum_n K_p = 4,6 \times 10^{-12} \cdot As \qquad (6)$$

The K_p values for the *XC4000* family in table 1 shows a decreasing K_P factor from 72 x 10^{-12} down to 13 x 10^{-12}, because of the improved chip technology from the *XC4000E* chips to the *XC4000XV* chips. Now, the *Virtex* chips show an additional technology power factor dip from 13 x 10^{-12} to 4.6 x 10^{-12}, which reflects the further technology improvement between the *XC4000XV* and the *Virtex* FPGAs.

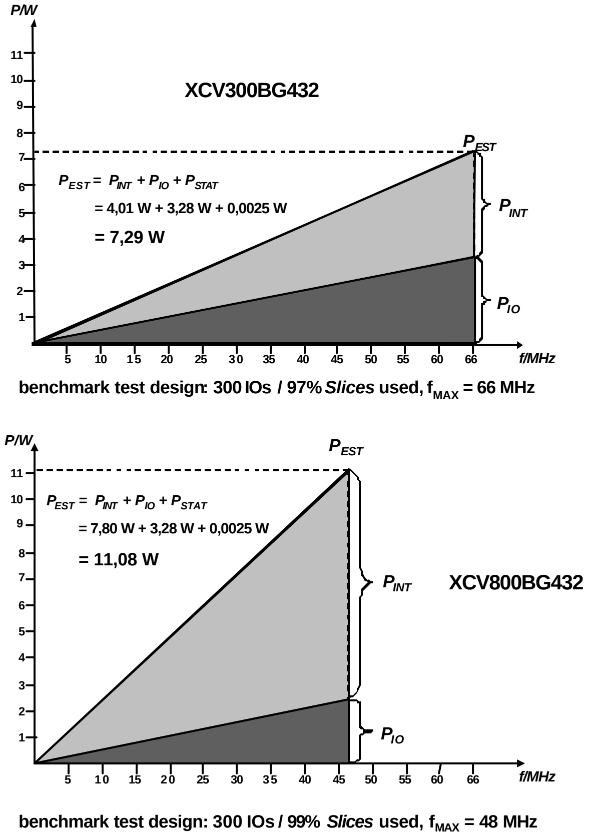
To determine the *IO* power consumption, we used our benchmark test design to toggle the outputs. Each output has a capacitive load of 10pF and the output voltage of all outputs were set to V_{IO} = 3.3 V. As shown in table 3, under the worst case conditions (if all *IO* toggle in each clock cycle at 66 MHz), up to 3.277 W can be consumed through the *IO*-buffers.

f_{togIO}	10 IO P_{IO}	50 IO P_{IO}	100 IO P_{IO}	200 IO P_{IO}	300 IO P_{IO}
MHz	W	W	W	W	W
10	0.018	0.090	0.172	0.353	0.554
20	0.036	0.177	0.343	0.685	1.082
30	0.054	0.262	0.508	1.011	1.620
40	0.074	0.356	0.677	1.335	2.122
50	0.093	0.446	0.858	1.656	2.624
66	0.1195	0.578	1.069	2.250	3.277

Table 3. Measured *IO* power consumption (V_{IO} = 3.3 V)

If we accumulate all three parts for the total estimated power consumption P_{EST} as given in equation 1, we get the diagram depicted in figure 5. Using the benchmark test design, up to a total power consumption of 7.29 W can be consumed by an entirely utilized *XCV300* chip with 300 used *IO* pins at 66MHz.

Using a *XCV800* chip, the benchmark test design consumes up to 11.1 W at f_{Max} = 48 MHz and utilizes 99% of the *Slices*. This increase in power consumption is mainly caused by the larger core current, which increases to I_{Core} = 3.12 A. The leakage current and the *IO* current is still in the same range as the smaller *XCV300* chip. This means, if using FPGA chips, which have a large on-chip gate capacity, the power consumption of the core will become the major factor. Therefore, we focus on estimating this factor in our investigations.



benchmark test design: 300 IOs / 97% *Slices* used, f_{MAX} = 66 MHz

benchmark test design: 300 IOs / 99% *Slices* used, f_{MAX} = 48 MHz

Figure 5. Power consumption on *Virtex* FPGAs

The investigations above are based on the benchmark test design in figure 2, which implements a worst case scenario on the FPGA chip. This was done to analyze the behavior of the different parts, which plays a own role in the total estimated power consumption. In the next two chapters 5.1 and 5.2, we use two typical industrial designs to evaluate the experimental results.

5.1 Internal Power Estimation for the *Actuator Sensor Interface (ASI)*

In the last few years, we worked closely with different companies developing an *Actuator Sensor Interface* called the *ASI*-Master unit. The main feature of the *ASI*-Master is to connect up to 128 binary actuator and sensor devices to a control unit via one bifilar cable, which is used for both data exchange and power supply for the actuators and sensors. This *ASI*-Master has been introduced in many applications in the area of industrial automation. If we connect more than one cable to a *ASI*-Master, it is called a multi-master channel solution. We developed an architecture with up to four channels, which is shown in figure 6 and named *ASI* Quad Master. It is capable of communicating with up to 4 x 128 actuator and sensor devices in order to generate a process data image every 5 ms. The architecture in figure 6 is written in *VHDL* and can be synthesized on different FPGA target architectures. The hardware provides four *ASI*-UARTs, which perform a bidirectional parallel to serial data exchange and a manchester coded *ASI*-specific protocol.

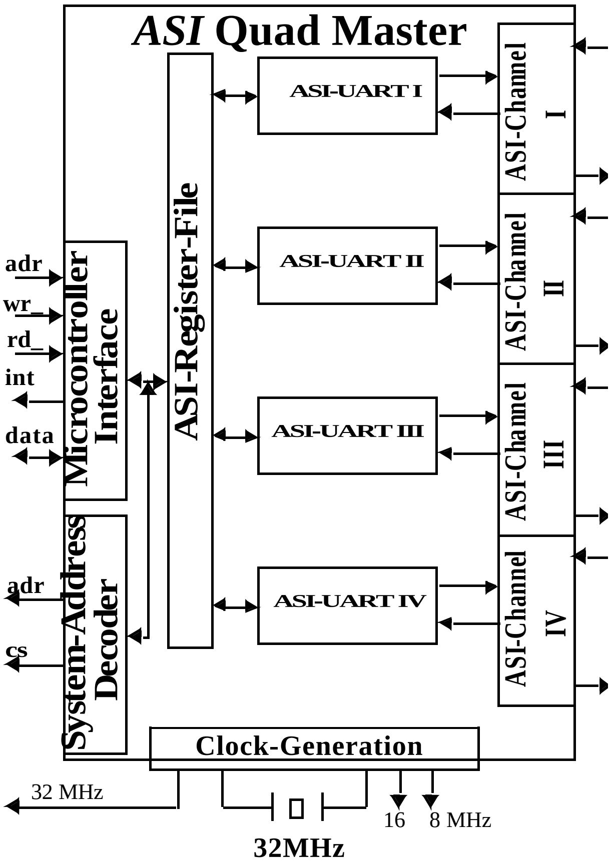


Figure 6. Architecture of the ASI Quad Master

The *ASI*-Master functionality description is beyond the scope of this paper. In the past, that application was also used as an evaluation benchmark design in a couple of research projects. For more information about the ASI-Master, the reader is referred to [6][7].

At this point we would like to mention that the structure of that *ASI*-Master design is composed of many finite state machines and counters inside the *ASI*-UARTs. This information will be important for the subsequent analysis. Additionally, this design needs only a few *IO* signals. That means, the *IO*-part P_{IO} and the static part P_{Stat} in the total power estimation (see equation 1) can be omitted. Only P_{INT} must be taken into account.

In order to estimate P_{INT} given in equation 4, which is based on our technology-dependent power factor K_p=4.6x10^{-12} for *Virtex* FPGAs, we implement the *VHDL*-code for the introduced *ASI* Quad Master shown in figure 6 on a *XCV300BG432*-FPGA chip. In this case, 711 *Slices* are occupied. If the design operates with the maximum clock speed (f_{Max} = 32 MHz), a core current of I_{Core} = 36.7mA can be measured. As shown in the second column in table 4, the chip requires a core voltage of

2.5 V, and the design then has an internal power consumption of P_{INT}=91.7mW.

Chip: XCV300 BG432	*ASI*-Quad Master design figure 6	benchmark test design figure 2	*ASI* internal power estimation based on equat. 4	Xilinx Estimator
utilized Slices	711	711	711	711
f_{Max}	32MHz	32MHz	32MHz	32MHz
I_{Core}	36.7mAmeasured	224mA measured	----	---
V_{Core}	2.5V	2.5V	2.5V	2.5V
Tog_{LC}	---	1.0	0.15	0.15
P_{INT}	91.7mW measured	560mW measured	91.2mW computed	92.4mW computed

Table 4. *ASI* measurement and computation results

This paper focuses on estimation. If we use equation 4 to determine the internal power consumption, all parameters are available except the corresponding toggle rate Tog_{LC}-value. As mentioned above, a precise value can only be determined using simulation. In order to quickly achieve an estimated value, some papers [4][3] recommend a Tog_{LC}-range between 0.1 and 0.2. With this recommendation, the estimated range for the internal power is between $P_{INT}(0.1)$ = 60.8mW and $P_{INT}(0.2)$ = 121.6mW. If we transform equation 4 to compute the toggle rate Tog_{LC} and use the real measured P_{INT}=91.7mW as an input parameter, the precise value can be determined with Tog_{LC} = 0.15, which resides in the middle of the recommended range for this specific industrial design. It shows that the recommended range makes sense for a fast estimation of power consumption in an early design step.

In the next step we prove whether the applied benchmark test design in figure 1 and 2 is able to model typical designs used in the daily work of designers. This question is very important, because the experimental determination of the technology power factor K_p=4.6x10^{-12} for *Virtex* is mainly based on that model. Therefore, the benchmark test design was used to model the ASI Quad Master design. We set the appropriate control values (MAX_IO and MAX_OR) to get the same utilization as used in the *ASI* design. If the benchmark test design is operated at the same frequency, a core current of 224 mA is measured, which results in an internal power consumption of P_{INT}=560mW as shown in the third column of table 4. In contrast, if we compute the internal power based on equation 4, we estimate 588mW, which is only 5% higher than the real value.

Note that this design is developed to toggle all FF in each cycle (Tog_{LC} = 1.0). This is the worst case scenario. If we reduce Tog_{LC} to a typical value of e.g. 0.15, we estimate 91.2mW (see column four in table 4). That value is about 1% below the real value. This step shows that the benchmark test design is able to model the internal behavior of practical industrial designs like *ASI.*

In the last step, we compared the *Virtex Power Estimator* [10] provided by Xilinx via the web (see the fifth column, table 3) with our approach. This tool, whose underlying model and equations have not been previously published, estimate a power consumption of 92.4mW, which is slightly above that value which we get when we use equation 4 based on our K_P-factor. The result indicates, that it seems to be that Xilinx uses a similar approach in the background, as our approach presented in this paper.

5.2 Restrictions of the Power Estimation Approach

In order to draw a clear dividing line for which kind of design structures this approach can be successfully applied, we used a second practical industrial design. In the past, we developed a double precision Floating Point Unit (FPU), whose core is based on a 52x52 bit parallel multiplier, which occupies the major part of the chip resources.

The architecture in figure 7 was written in *VHDL* and implemented on a *Virtex XCV800BG432* chip, which is provided on SPYDER-VIRTEX-X2.

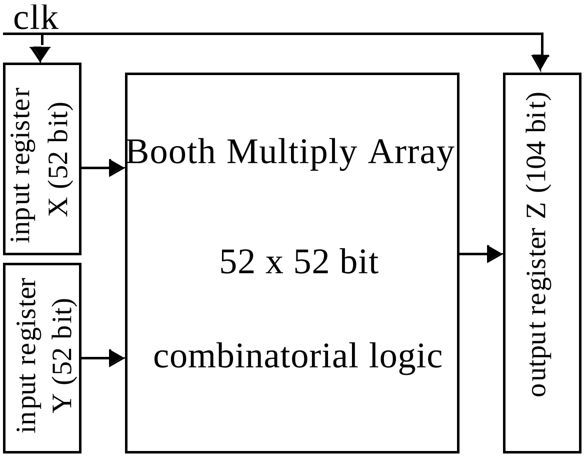

clk
input register X (52 bit)
input register Y (52 bit)
Booth Multiply Array
52 x 52 bit
combinatorial logic
output register Z (104 bit)

Figure 7. Multiply-Core of the FPU

The major part of the total power consumption is determined by the internal power. The static and *IO* power parts can be ignored. During chip operation, the following measurements were made:

	measured design values	computed with equation 4	*Virtex* power estimator
util. Slices	2812 (23%)	2812	2812
FF	229	---	---
function generators	5584	---	---
V_{Core}	2.5 V	2.5 V	2.5 V
f_{Max} = 4 MHz	I_{Core} = 0.717 A measured P_{INT} = 1.793 W	P_{INT} = 0.291W	P_{INT} = 0.253 W
3 x f_{Max} 12MHz	I_{Core} = 2.056 A measured P_{INT} = 5. 140 W	P_{INT} = 0.873 W	P_{INT} = 0.759W

Table 5. Multiplier measurement and computation results

The second column shows the measurements. The multiplier utilizes 2812 *Slices*. Only 229 FF were used, but 5584 function generators. This indicates, that mainly combinatorial logic was implemented. That fact is important for further analysis.

If the multiplier operates at the maximum frequency of 4 MHz, the core current is measured as 0.717 A, which leads to a internal power consumption of 1.793 W.

In the third column of table 5, the shown parameters will be inserted in the equation 4 to compute P_{INT}. The result for P_{INT} = 0.291 W shows that the estimated value differs significantly from the measured value of 1.793 W. The same observation could be made, if we used the *Virtex power estimator* provided by Xilinx (see column four in table 5).

This result shows the power estimation approach provided in this paper, and also the Xilinx approach provided in the web, cannot be used for design structures which are dominated by large combinatorial logic blocks like the multiplier investigated here. If the design structure is dominated by finite state machines and counters, as investigated in chapter 5.1 using the *ASI* design, this approach is able to estimate very good values for the internal power consumption.

6 Summary

We started with the introduction of a power estimation method for the widely used *XC4000*-FPGA architecture. This approach was extended to the novel *Virtex*-FPGA chips. In order to apply this method, it is necessary to determine a *Virtex*-dependent specific technology power factor K_P to estimate the internal power consumption. Therefore, we developed a benchmark test design, which is characterized by the fact, that all *FF* toggle in each clock cycle. This design represents a worst case scenario on the chip and the toggle rate can be set to one.

Two major factors influence the power consumption. Both the speed $f(f_{Max})$ and utilization *f(Slices)* can be scaled with the benchmark test design. We performed different investigations and determined the technology power factor:

- $K_{P(Virtex)}$ = 4.6 x 10^{-12} As.

In order to conduct the experiments, we used our own emulation environment SPYDER-VIRTEX-X2.

Two typical designs developed for an industrial application were used for the verification of the derived formula for the internal power estimation on *Virtex*-FPGAs. The result of the verification shows:

- If the structure of a design is dominated by finite state machines and/or counters, the presented approach can be successfully applied to estimate internal power consumption in an early design step.
- If the structure of a design is dominated by combinatorial logic, the approach cannot be used.

7 Future Work

In the future, a further increase in the on-chip gate capacity up to 2 million gates can be expected, e.g. *Virtex-E* architecture, which enables the implementation of large sophisticated designs, which are dominated by the internal power consumption. Therefore, a dependable method to estimate that internal power consumption in an early design step will become important to avoid thermal design risks. As presented in this paper, large combinatorial logic blocks can not be sufficiently estimated. Therefore, we focus our future work in developing an appropriate estimation method, which addresses this problem.

References:

[1] Xilinx Inc.: *The programmable Logic Data Book,* 1999.

[2] Altera Corp.: *Evaluating Power for Altera Devices,* Application Note 74, version 3, May 1999.

[3] Actel Corp.: *Predicting the Power Dissipation of Actel FPGAs,* Application Note, April 1996.

[4] Xilinx Inc.: *A Simple Method of Estimating Power in XC4000XL/EX/E FPGAs,* Application Brief X014, June 30, 1997.

[5] K. Weiß, R. Kistner, W. Rosenstiel: *Analysis of the XC6000 Architecture for Embedded System Design.* Field-Progammable Custom Computing Machines (FCCM), Napa Valley CA, April 1998.

[6] K. Weiß, T. Steckstor, C. Nitsch, W. Rosenstiel: *Performance Analysis of Real-Time-Operation Systems by Emulation of an Embedded System.* 10th IEEE International Workshop on Rapid System Prototyping (RSP), Clearwater, Florida, USA, 1999.

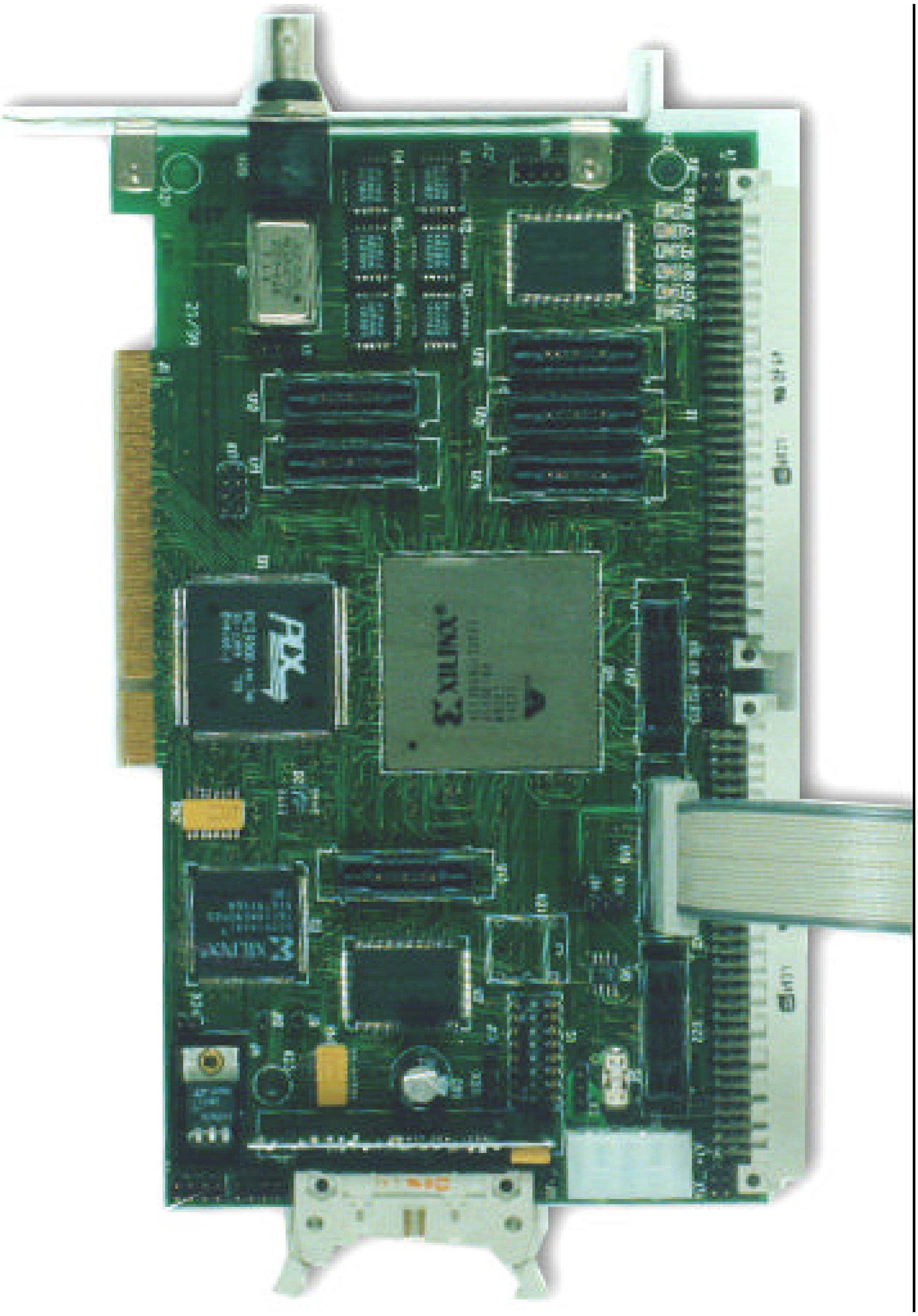
[7] K. Weiss, T. Steckstor, W. Rosenstiel: *Exploiting FPGA-Features during the Emulation of a Fast Reactive Embedded System.* ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA), Monterey, CA, USA, 1999

[8] I. Katchan, C. Oetker, T. Steckstor, K. Weiß: SPYDER-VIRTEX-X2 user manual, version 1.0, http://www.fzi.de/sim/spyder.html, september 1999.

[9] K. Weiss, T. Steckstor, C. Oetker, R. Kistner: *Data Sheet and User Manuel SPYDER-ASIC-X1.* http://www.fzi.de/sim/spyder.html, FZI & University Tübingen 1998.

[10] Xilinx Inc: *Virtex Power Estimator User Guide,* Application Note 152, version 1.0, April 28, 1999.

Appendix A



Picture of SPYDER-VIRTEX-X2