A Probabilistic Power Prediction Tool for the Xilinx 4000-Series FPGA

Timothy Osmulski, Jeffrey T. Muehring, Brian Veale, Jack M. West, Hongping Li,
Sirirut Vanichayobon, Seok-Hyun Ko, John K. Antonio, and Sudarshan K. Dhall

School of Computer Science
University of Oklahoma
200 Felgar Street
Norman, OK 73019
Phone: (405) 325-7859
antonio@ou.edu

Abstract. The work described here introduces a practical and accurate tool for predicting power consumption for FPGA circuits. The utility of the tool is that it enables FPGA circuit designers to evaluate the power consumption of their designs without resorting to the laborious and expensive empirical approach of instrumenting an FPGA board/chip and taking actual power consumption measurements. Preliminary results of the tool presented here indicate that an error of less than 5% is usually achieved when compared with actual physical measurements of power consumption.

1   Introduction and Background

Reconfigurable computing devices, such as field programmable gate arrays (FPGAs), have become a popular choice for the implementation of custom computing systems. For special purpose computing environments, reconfigurable devices can offer a cost-effective and more flexible alternative than the use of application specific integrated circuits (ASICs). They are especially cost-effective compared to ASICs when only a few copies of the chip(s) are needed [1]. Another major advantage of FPGAs over ASICs is that they can be reconfigured to change their functionality while still resident in the system, which allows hardware designs to be changed as easily as software and dynamically reconfigured to perform different functions at different times [6].

Often a device's performance (i.e., speed) is a main design consideration; however, power consumption is of growing concern as the logic density and speed of ICs increase. Some research has been undertaken in the area of power consumption in CMOS (complimentary metal-oxide semiconductor) devices, e.g., see [4, 5]. However, most of this past work assumes design and implementation based on the use of standard (basic cell) VLSI techniques, which is typically not a valid assumption for application circuits designed for implementation on an FPGA.

2   Overview of the Tool

A probabilistic power prediction tool for the Xilinx 4000-series FPGA is overviewed in this section. The tool, which is implemented in Java, takes as input two files: (1) a *configuration file* associated with an FPGA design and (2) a *pin file* that characterizes the signal activities of the input data pins to the FPGA. The configuration file defines how each CLB (configurable logic block) is programmed and defines signal connections among the programmed CLBs. The configuration file is an ASCII file that is generated using a Xilinx M1 Foundation Series utility called *ncdread*. The pin file is also an ASCII file, but is generated by the user. It contains a listing of pins that are associated with the input data for the configured FPGA circuit. For each pin number listed, probabilistic parameters are provided which characterize the signal activity for that pin.

Based on the two input files, the tool propagates the probabilistic information associated with the pins through a model of the FPGA configuration and calculates the activity of every internal signal associated with the configuration [1]. The activity of an internal signal s, denoted a_s, is a value between zero and one and represents the signal's relative frequency with respect to the frequency of the system clock, f. Thus, the average frequency of signal s is given by $a_s f$.

Computing the activities of the internal signals represents the bulk of computations performed by the tool [1]. Given the probabilistic parameters for all input signals of a configured CLB, the probabilistic parameters of that CLB's output signals are determined using a well-defined mathematical transformation [2]. Thus, the probabilistic information for the pin signals is transformed as it passes through the configured logic defined by the configuration file. However, the probabilistic parameters of some CLB inputs may not be initially known because they are not directly connected to pin signals, but instead are connected to the output of another CLB for which the output probabilistic parameters have not yet been computed (i.e., there is a feedback loop). For this reason, the tool applies an iterative approach to update the values for unknown signal parameters. The iteration process continues until convergence is reached, which means that the determined signal parameters are consistent based on the mathematical transformation that relates input and output signal parameter values, for every CLB.

The average power dissipation due to a signal s is modeled by $\frac{1}{2} C_{d(s)} V^2 a_s f$, where $d(s)$ is the Manhattan distance the signal s spans across the array of CLBs, $C_{d(s)}$ is the equivalent capacitance seen by the signal s, and V is the voltage level of the FPGA device. The overall power consumption of the configured device is the sum of the power dissipated by all signals. For an N x N array of CLBs, Manhattan signal distances can range from 0 to $2N$. Therefore, the values of $2N + 1$ equivalent capacitance values must be known, in general, to calculate the overall power consumption. Letting S denote the set of all internal signals for a given configuration, the overall power consumption of the FPGA is given by:

$$P_{\text{avg}} = \sum_{s \in S} \frac{1}{2} C_{d(s)} V^2 a_s f$$

$$= \frac{1}{2} V^2 f \sum_{s \in S} C_{d(s)} a_s . \tag{1}$$

The values of the activities (i.e., the a_s's) are dependent upon the parameter values of the pin signals defined in the pin file. Thus, although a given configuration file defines the set S of internal signals present, the parameter values in the pin file impact the activity values of these internal signals.

3 Calibration of the Tool

Let S_i denote the set of signals of length i, i.e., $S_i = \{s \in S \mid d(s) = i\}$. So, the set of signals S can be partitioned into $2N + 1$ subsets based on the length associated with each signal. Using this partitioning, Eq. 1 can be expressed as follows:

$$P_{\text{avg}} = \frac{1}{2}V^2 f \left(C_0 \sum_{s \in S_0} a_s + C_1 \sum_{s \in S_1} a_s + \cdots + C_{2N} \sum_{s \in S_{2N}} a_s \right) \tag{2}$$

To determine the values of the tool's capacitance parameters, actual power consumption measurements are taken from an instrumented FPGA using different configuration files and pin input parameters. Specifically, $2N + 1$ distinct measurements are made and equated to the above equation using the activity values (i.e., the a_s's) computed by the tool. For the j-th design/data set combination, let P_j denote the measured power and let $A_{j,k}$ denote the aggregate activity of all signals of length k. The resulting set of equations is then solved to determine the $2N + 1$ unknown capacitance parameter values:

$$\frac{1}{2}V^2 f \begin{pmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,2N} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,2N} \\ \vdots & & \ddots & \vdots \\ A_{2N,0} & A_{2N,1} & & A_{2N,2N} \end{pmatrix} \begin{pmatrix} C_0 \\ C_1 \\ \vdots \\ C_{2N} \end{pmatrix} = \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_{2N} \end{pmatrix} \tag{3}$$

Solving the above equation for the vector of unknown capacitance values is how the tool is calibrated.

4 Power Measurements

For this study, a total of 70 power measurements were made using 5 different configuration files and 14 different data sets. Descriptions of these configuration files and data sets are given in Tables 1 and 2, respectively. All of the configuration files listed in Table 1 each take a total of 32-bits of data as input. The first three configurations (fp_mult, fp_add, int_mult) each take two 16-bit operands on each clock cycle, and the last two (serial_fir and parallel_fir) each take one 32-bit complex operand on each clock cycle. The 32 bits of input data are numbered as 0 through 31 in Table 2, and two key parameters are used to characterize these bits: an *activity factor, a* and a *probability factor, p*. The activity factor of an input bit is a value

between zero and one and represents the signal's relative frequency with respect to the frequency of the system clock, f. The probability factor of a bit represents the fraction of time that the bit has a value of one.

Fig. 1 shows plots of the measured power for all combinations of the configuration files and data sets described in Tables 1 and 2. For all cases, the clock was run at $f = 30$ MHz. With the exception of the fp_mult configuration file, the most active data set file (number 6) is associated with the highest power consumption. Also, the least active data set file (number 5) is associated with the lowest power consumption across all configuration files. There is somewhat of a correlation between the number of components utilized by each configuration and the power consumption; however, note that even though the serial_fir implementation is slightly larger than parallel_fir, it consumes less power. This is likely due to the fact that the parallel_fir design requires a high fan-out (and thus high routing capacitance) to drive the parallel multipliers.

Table 1. Characteristics of the configuration files.

Configuration File Name	Description	Component Utilization of Xilinx 4036xla
fp_mult	Custom 16-bit floating point multiplier with 11-bit mantissa, 4-bit exponent, and a sign bit [3].	368
fp_add	Custom 16-bit floating point adder with 11-bit mantissa, 4-bit exponent, and a sign bit [3].	339
int_mult	16-bit integer array multiplier; produces 32-bit product [3].	509
serial_fir	FIR filter implementation using a serial-multiply with a parallel reduction add tree. Input data is 32-bit integer complex. Constant coefficient multipliers and adders from core generator.	1060
parallel_fir	FIR filter implementation using a parallel-multiply with a series of delayed adders. Input data is 32-bit integer complex. Constant coefficient multipliers and adders from core generator.	1055

Table 2. Characteristics of the data sets.

Data Set Number	Description
1	Pins 0 through 15 $\Rightarrow p = 0.0$ and $a = 0.0$. Pins 16 through 31 $\Rightarrow p = 0.5$ and $a = 1.0$
2	Pins 0 through 15 $\Rightarrow p = 0.0$ and $a = 0.0$ Pins 16 through 31 $\Rightarrow p = 0.75$ and $a = 0.4$
3	Pins 0 through 15 $\Rightarrow p = 0.25$ and $a = 0.45$ Pins 16 through 31 $\Rightarrow p = 0.0$ and $a = 0.0$
4	Pins 0 through 15 $\Rightarrow p = 0.5$ and $a = 1.0$ Pins 16 through 31 $\Rightarrow p = 0.0$ and $a = 0.0$
5	Pins 0 through 31 $\Rightarrow p = 0.0$ and $a = 0.0$
6	Pins 0 through 31 $\Rightarrow p = 0.5$ and $a = 1.0$
7	Even numbered pins $\Rightarrow p = 0.0$ and $a = 0.0$ Odd numbered pins $\Rightarrow p = 0.5$ and $a = 1.0$
8	Even numbered pins $\Rightarrow p = 0.3$ and $a = 0.5$ Odd numbered pins $\Rightarrow p = 0.7$ and $a = 0.5$
9	Even numbered pins $\Rightarrow p = 0.5$ and $a = 1.0$ Odd numbered pins $\Rightarrow p = 0.0$ and $a = 0.0$
10	Even numbered pins $\Rightarrow p = 0.8$ and $a = 0.1$ Odd numbered pins $\Rightarrow p = 0.2$ and $a = 0.15$
11	For all pins, p and a selected at random (different from data set 12).
12	For all pins, p and a selected at random (different from data set 11).
13	Pins 0 through 2, $p = 0.1$ and $a = 0.1$ Pins 3 through 5, $p = 0.2$ and $a = 0.2$, etc., p's continue to increase in steps of 0.1 and a's increase to 0.5 in steps of 0.1 and then decrease back down to 0.0.
14	Pin 0, $p = 0.1$ and $a = 0.2$ Pin 1, $p = 0.2$ and $a = 0.4$ Pin 2, $p = 0.3$ and $a = 0.6$, etc., p's continue to increase to 1.0 in steps of 0.1 (and then decrease) and a's increase to 1.0 in steps of 0.2 (and then decrease).
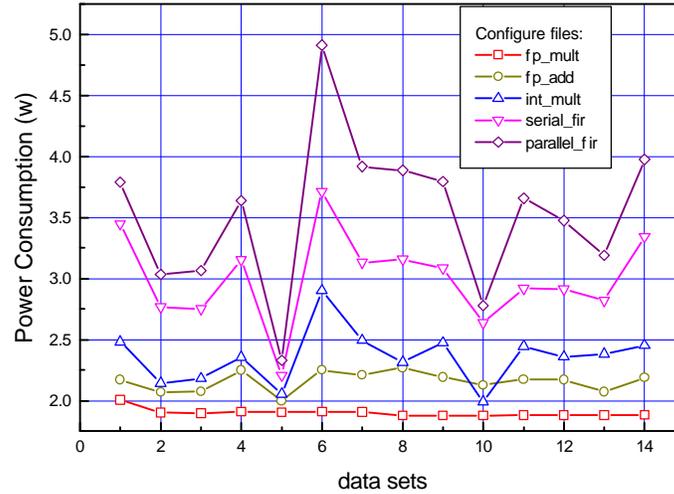
Fig. 1. Measured power consumption for the configuration files and data sets described in Tables 1 and 2.

5 Experimental Evaluation of the Tool

Because 73 values are used to model all of the internal capacitances of the device used in this study, at least three more measurement scenarios are required to calibrate all capacitance values (by solving the complete set of linear equations defined by Eq. 3). Fortunately, however, we were able to calibrate a subset of capacitance values by considering the power consumption of the two FIR filters (serial_fir and parallel_fir). This was because there turned out to be a total of only 28 non-zero entries for the rows of the matrix of Eq. 3, corresponding to aggregate activities for the two FIR filter designs.

   Fig. 2 shows the measured power consumption curve along with 29 different prediction curves generated by the tool for the serial FIR filter design. One of the prediction curves corresponds to predicted values based on using all 28 measured values to calibrate the tool's capacitance values (this curve is labeled "all" in the legend of the figure). This curve naturally has excellent accuracy; predicted power consumption values match measured values nearly perfectly.[1] The remaining 28 prediction curves are associated with capacitance values determined by using all but one of the measured data values to calibrate the tool (the data set not used is indicated in the legend of the figure). For each of these curves, the data set not used in the

---

[1] The reason the predicted values do not match measured values exactly is because the equations used to determine capacitance values did not have full rank, and thus a least-squares solution was determined.
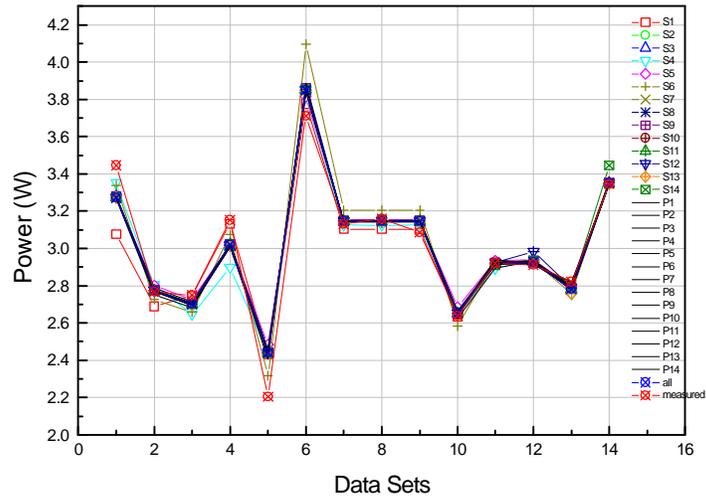
Fig. 2. Measured and predicted power consumption curves using various calibration scenarios for the serial FIR filter implementation.
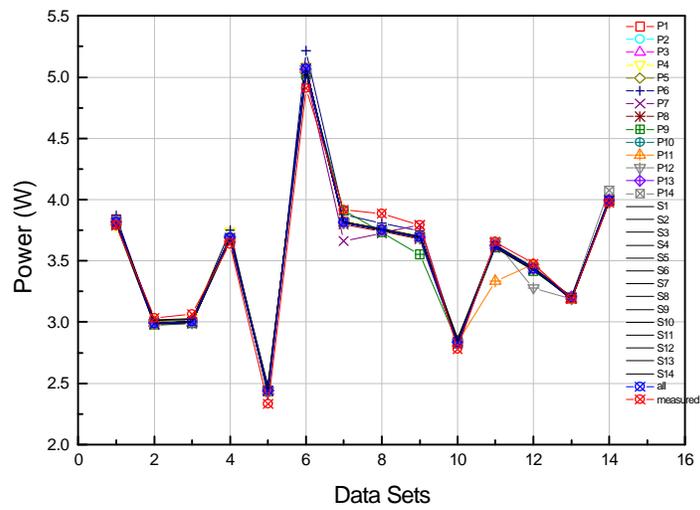
Fig. 3. Measured and predicted power consumption curves using various calibration scenarios for the parallel FIR filter implementation.

calibration of the tool's capacitance values generally associates with the highest error in the predicted value for that data point. For example, note that when data set number six for the serial FIR (labeled S6 in the figure's legend) was not used in the calibration process, the resulting prediction for that value was highest (around 10% error). When data sets associated with the parallel FIR design were not included, the prediction curves did not change, thus those curves are all drawn as solid lines with no symbols. Fig. 3 shows the same type of results as Fig. 2, except for the parallel FIR instead of the serial FIR.

6   Summary

To summarize the results for both filter designs, when all 28 sets of measurements are used to calibrate the tool, the maximum error in predicted versus measured power is typically less than about 5%. With one data set removed, the maximum error increases to about 10%, and the predicted value with this highest error is typically associated with the data set not used in calibrating the tool. This level of error is acceptable for most design environments, and represents a considerable accomplishment in the area of power prediction for FPGA circuits. Thus, these preliminary results indicate that the tool is able to adequately predict power consumption (i.e., for data sets not used in calibrating the tool). By using more data sets to calibrate the tool in the future, it is expected that even greater prediction accuracy and robustness will be achieved.

Acknowledgements

References

1. T. Osmulski, *Implementation and Evaluation of a Power Prediction Model for Field Programmable Gate Array*, Master's Thesis, Computer Science, Texas Tech University, 1998.
2. K. P. Parker and E. J. McClusky, "Probabilistic Treatment of General Combinatorial Networks," *IEEE Trans. Computers*, vol. C-24, pp. 668-670, June 1975.
3. B. Veale, *Study of Power Consumption for High-Performance Reconfigurable Computing Architectures*, Master's Thesis, Computer Science, Texas Tech University, 1999.
4. T. L. Chou, K. Roy, and S. Prasad, "Estimation of Circuit Activity Considering Signal Correlations and Simultaneous Switching," *Proc. IEEE Int'l Conf. Comput. Aided Design*, pp. 300-303, Nov. 1994.
5. A. Nannarelli and T. Yang, "Low-Power Divider," *IEEE Trans. Computers*, Vol. 48, No. 1, Jan. 1999, pp. 2-14.
6. *Xilinx XC4000E and XC4000X Series Field Programmable Gate arrays, Product Specification*, Xilinx Inc., v1.5, http://www.xilinx.com/partinfo/databook.htm#xc4000, 1999.