RAPID ESTIMATION OF POWER CONSUMPTION FOR HYBRID FPGAS

Chun Hok Ho[1], *Philip H.W. Leong*[2], *Wayne Luk*[1], *Steven J.E. Wilton*[3]

[1]Department of Computing
Imperial College London
London, England
{cho,wl}@doc.ic.ac.uk

[2]Dept. of Computer
Science and Engineering
Chinese University of Hong Kong
Hong Kong
phwl@cse.cuhk.edu.hk

[3]Dept. of Electrical
and Computer Engineering
University of British Columbia
Vancouver, B.C., Canada
stevew@ece.ubc.ca

ABSTRACT

A hybrid FPGA consists of island-style fine-grained units and domain-specific coarse-grained units. This paper describes an approach to estimate the power consumption of a set of hybrid FPGA architectures. The dynamic power consumption of the fine-grained units is obtained using standard FPGA tools, and the coarse-grained units using standard ASIC tools. Based on this approach, the dynamic power consumption of different hybrid FPGA architectures can be studied and we report on results over a set of floating point benchmark circuits.

1. INTRODUCTION

Since total power consumption of FPGAs increases with each reduction of integrated circuit feature size, power reduction has become one of the primary concerns in FPGA architecture. Power consumption can be divided into two parts, static and dynamic power consumption. Static power dissipation is due to leakage while dynamic power dissipation is due to switching activity of the circuits. It is reported that a commercial FPGA device with 90nm technology process consumes 62% of its total power as dynamic power [1].

In previous work we presented a hybrid FPGA architecture which can significantly increase speed and reducing area for floating point applications [2]. This paper focuses on methods to estimate the dynamic power estimation of hybrid FPGAs with different architectures.

The core of a hybrid FPGA is domain-specific coarse-grained units in which the most frequently used logic is placed to reduce area and delay. In the design phase of the coarse-grained unit and hybrid FPGA architecture, one critical issue is that how to rapidly estimate the dynamic power consumption across a number of candidate designs. Traditional power measurement involving retrieval of switching activity based on post-place and route simulation may not be practical as for every new architecture, the application circuit has to be manually mapped and the configuration of the coarse-grained unit may be different each time. In addition, simulation test vectors have to be adjusted according to the mapped design. Such estimation is too tedious and is usually not necessary for initial design exploration.

To address this issue, this paper proposes a high level dynamic power consumption estimation technique for hybrid FPGAs. We assume constant activity rate on all the nets in a design. The basic idea is to measure the power consumption of the coarse-grained unit with ASIC tools and the power consumption of the fine-grained unit with FPGA design tools under the same conditions. The total power estimate can be given by the sum of these two measurements. In addition, a technology mapper is introduced to allow automatic generation of the benchmark circuit from a dataflow graph. This eliminates the tedious manually mapping procedure for each benchmark circuit. Once a first level estimation is determined across candidate architectures, more detailed investigations can be made using other tools.

The key contributions of this paper include:

1. A high level power estimation flow which can determine the power and energy used in the hybrid FPGA design (Section 3 and 4).
2. A technology mapper which can produce hybrid FPGA circuit according to the architecture and topology of a coarse-grained unit (Section 5).
3. Comparison of the relative benefits of hybrid FPGAs with dedicated coarse-grained units compared with conventional island-style FPGAs consisting purely of fine-grained units over a set of floating point benchmark circuits (Section 6).

This approach is based on a methodology present in reference [3]. While previous work has not considered power consumption, allowing the performance of hybrid FPGAs to be evaluated according to multiple metrics including area, speed and dynamic power consumption.

978-1-4244-1961-6/08/$25.00 ©2008 IEEE.

2. RELATED WORK

Previous studies have been made on the power estimation of FPGA devices. Kuan and Rose suggested a methodology and presented results comparing dynamic power consumption between FPGA and ASIC devices [4]. Poon et. al. proposed a power model for variety of FPGA architectures based on VPR [5]. Tuan et. al. investigated the breakdown of power consumption in a $90nm$ FPGA device [1].

Several architectural modification has been suggested in order to reduce the power consumption in FPGA device. Lamoureux et.al. have proposed a technique to reduce the dynamic power consumption but inserting programmable delay in the look-up table (LUT). Tuan et. al. have demonstrated a low-power FPGA feature with voltage scaling, low-power configuration memory, power gating and standby mode [1].

We are not aware of any reported studies concerning the power reduction achieved by employing embedded domain-specific coarse-grained blocks, nor of methodologies to facilitate such studies.

3. OVERVIEW OF THE POWER ESTIMATION

Let P_{fgu} be the power dissipation of fine-grained unit and let P_{cgu} be the power dissipation of coarse-grained unit. The dynamic power dissipation (P_{all}) of a hybrid FPGA can be represented by the following equation:

$$P_{all} = P_{fgu} + P_{cgu} + P_r \tag{1}$$

In the proposed flow, P_{fgu} is estimated using a spreadsheet approach, and P_{cgu} is determined using an ASIC power estimation tool. However, neither P_{cgu} nor P_{fgu} account for the power dissipation in routing resources between coarse-grained and fine-grained units. We introduce P_r which represents this power which can be obtained by modelling the output loading of the coarse-grained unit.

In order to employ the proposed power estimation flow, the following assumptions are made and justified as follows:

1. The architecture of the fine-grained unit should be similar to the commercial one. This is because the fine-grained fabric of commercial FPGAs are mature technology and the associated tool chain is well developed. We further assume that the area, timing and power is modelled accurately.
2. The process technology used in building the coarse-grained unit should be similar to that of the fine-grained unit. With similar transistor size, the area and delay of the units can be directly compared. Moreover, transistors with similar size have similar capacitance which is a critical factor when estimating power consumption.
3. Constant activity rates are assumed on all the nets in a design. This allows us to rapidly estimate the power consumption without estimating activity rate using post-place and route simulation.
4. Apart from logic cells, registers and embedded blocks, there are several components which may affect the dynamic power consumption in an FPGA. Such components include I/O cells and clock management units. In this study, we assume all hybrid FPGAs share the same architecture so the dynamic power consumption of these components are the same. Only the dynamic power dissipated in computation cores are considered in the estimation.

The estimation begins with a benchmark application. It is described as a dataflow graph and a technology mapper produces a mapped circuit for the hybrid FPGA architecture. Prior to the power measurement on the hybrid FPGA, the characteristics of the circuit have to be obtained. These include the configuration and connection of both fine-grained and coarse-grained units, the frequency of the circuit and the area of the circuit. The configuration of these units can be generated automatically from a dataflow graph using a technology mapper as described in Section 5. In addition, the architecture of the coarse-grained unit is described in HDL. It is then synthesised using standard cell design flow to obtain the area of a coarse-grained unit. The delay of the coarse-grained unit can be determined by loading corresponding configuration.

Once the area and delay of the coarse-grained units are estimated, a virtual embedded block (VEB) methodology can be applied to obtain the area and delay [2] of the whole hybrid FPGA circuit.

Given the area of fine-grained units and the operation frequency, the power consumption of the fine-grained units can be determined by a commercial FPGA power estimation tool as the fine-grained unit is the same architecture as the commercial one. A high level power estimation, which assumes a constant toggle rate of all nets is employed to obtained the dynamic power dissipation of fine-grained units. Similar to estimating the power consumption of the fine-grained unit, the power consumption of coarse-grained unit can be obtained from an ASIC power estimation tool, assuming a fixed operation frequency and toggle rate for all the nets.

The dynamic consumption of the routing resources between the coarse-grained and fine-grained unit can be modelled by setting the appropriate output loading of the coarse-grained unit. A calibration scheme is proposed to determine the output loading. The dynamic consumption of an existing embedded block in a commercial FPGA is first measured. An equivalent embedded block is then implemented with standard cell design flow and the dynamic power consumption is assessed assuming no loading at the outputs. Then the output loading of the embedded block is increased until the dynamic power consumption matches the commercial FPGA. This value represents the average loading of the routing resources and is applied to the output of the coarse-grained unit. The dynamic power consumption of the routing resources can be obtained from the ASIC power estimation tool. The high level estimation of the dynamic consumption of the hybrid FPGA can be obtained by combining both numbers.

The same approach can be applied to energy consumption or the power-delay product which are often better metrics for FPGA architectures than power consumption.

The major issue concerning the proposed approach is accuracy. There are several contributing factors. (1) separate, unmatched power models for the fine-grained and coarse-grained units, (2) assumption of constant switching activity rate and (3) the uncertainty of the power dissipation in routing resources. To address (1), similar technology process is used between coarse-grained unit and fine-grained unit. To address (2), we estimate the power consumption under different switching activity rates to measure the lower and upper bounds for the power consumption. We will study this problem in future research. Issue (3) is addressed by proposing a calibration scheme which adjusts the output loading of the coarse-grained unit.

4. EXAMPLE ESTIMATION FLOW

This section illustrates a detail power estimation flow of one particular hybrid FPGA architecture as a case study. We assume the fine-grained units of the hybrid FPGA have the same architecture as those of the Xilinx Virtex II. The Xilinx tool chain, including the power estimation tool and the associated CAD tools such as Xilinx ISE 9.2i are used in this flow.

A web-based power estimation tool [6] is provided by the vendor which employs a spreadsheet. This tool requires users to specify the frequency, number of registers used, number of look-up table (LUT) used, number of embedded multiplier used, number of block memory used, amount of routing used and the average toggle rate of the design.

We assume the same hybrid FPGA as [2] for this example. The architecture of this hybrid FPGA and the coarse-grained unit is illustrated in Figure and 1. The coarse-grained unit has floating point adders and floating point multipliers and the hybrid FPGA is designed for implementing floating point applications.

Applications are first described as a dataflow graph and mapped to hybrid FPGA using a technology mapper as described in Section 5. Then a VEB flow is used to capture the area and delay of the user application implemented on the FPGA as described in [2]. The coarse-grained unit is described in a hardware description language (HDL) and a

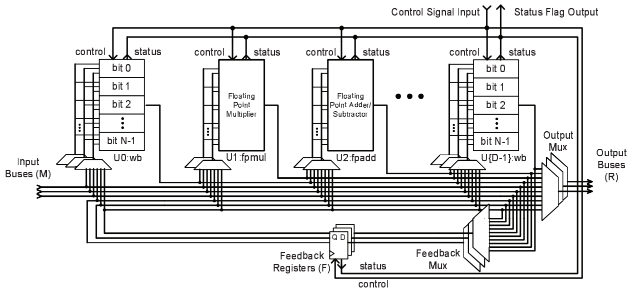


Fig. 1. Coarse-grained unit architecture.

standard cell design flow is used to synthesise the circuit as an ASIC. In addition, a UMC $0.13\mu m$ technology process and its associated standard cell library is used in the construction of the coarse-grained unit. We believe this technology process is similar to the one used in Xilinx Virtex II device.

For different circuits, there are different configurations in each coarse-grained unit and the timing and power consumption values are different as well. The bitstreams generated by technology mapper mentioneded in 5 affect the timing and power consumption of configured coarse-grained units. Timing can be determined by setting case analysis constraints according to the bitstream so that the tool can recognise false paths in the design. The timing representing the frequency of the coarse-grained unit can be used in the VEB flow to model hybrid FPGA as described below.

To obtain the power consumption of the fine-grained units, we need to determine the number of LUTs and registers used in the design. A VEB methodology is employed for this purpose [3]. This method allows us to instantiate dummy logic cells with the same area and delay as coarse-grained unit. Therefore, we can use the vendor tool chain to extract the area and timing of the hybrid FPGA [2]. By subtracting the LUT logic cells and registers used by dummy blocks, the LUTs and registers used by fine-grained unit can be determined. The frequency of the circuit can also be obtained from the VEB flow.

When estimating the power consumption of the fine-grained unit using the web-tool, we choose a medium amount of routing resources because most circuits involve floating point operators which are dominated by random logic. According to the vendor's suggestion [6], medium routing should be selected for random logic. There are other information can be specified in the power estimation tool such as the I/O cells and clock configurations. As the power estimation on I/O cells and clock management units are not included in the comparison, these information can be ignored in power estimation. Once all the required data is specified in the power estimation tool, the dynamic power consumption of the circuit can be obtained.

It would appear at first that dynamic power consumption of a coarse-grained unit can be determined by setting a constant toggle rate on all the nets in that unit. However, this is usually not the case as there may be some unused wordblocks where the input is always a constant and therefore the activity rate of these blocks is zero. Instead, we assume constant toggle rates on all *used* wordblocks and floating point operators, and zero toggle rate on all unused wordblocks. As a result, all unused wordblocks have zero dynamic power consumption. In an analogous manner, we also assume unused logic cells have zero dynamic power consumption in the commercial FPGA. Unused wordblocks can be identified from the routing configuration bits in the bitstream. Similar to the fine-grained fabric, a 12.5% toggle rate is applied to all the nets in used wordblocks and floating point operators.

The output loading of the coarse-grained unit has to be considered when estimating the dynamic power consumption. Based on the calibration experiment as explained in Section 6, the output loading of each pin is found to be 4.5pF. This allows the ASIC tool chain to consider the dynamic power consumption of the coarse-grained unit logic with the associated routing resources to the fine-grained unit.

The total dynamic power consumption of hybrid FPGA can be obtained by the sum of dynamic power consumption from coarse-grained and fine-grained unit.

5. TECHNOLOGY MAPPER

To study how different combinations of coarse-grained units affect the overall power and energy consumption in hybrid FPGA applications, a technology mapper is developed to translate dataflow graph to hybrid FPGA device.

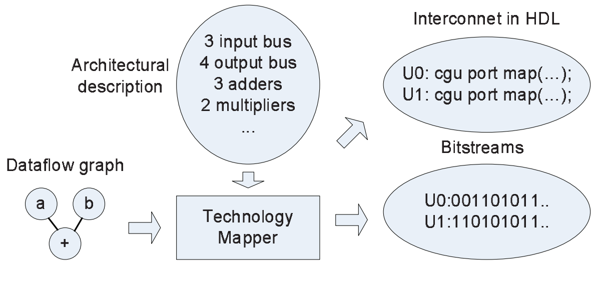
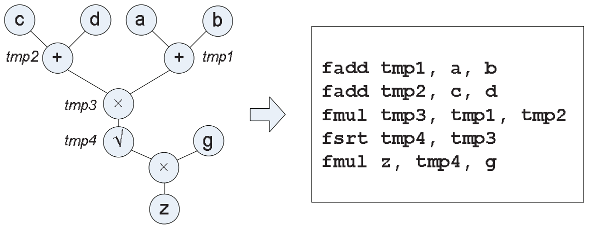


Fig. 2. Logic flow of technology mapper.

Figure 2 illustrates the logic flow. The mapper requires a specification of the coarse-grained unit to map floating point applications onto it. As the coarse-grained unit is parametrisable, the technology mapper has to consider the architectural description of the coarse-grained unit before mapping the application circuits. The architectural description includes type and number of floating point operators, number of input and output buses, number of feedback registers and the number of generic wordblocks. The descriptions also include the placement sequence of the generic wordblocks and the floating point operators as they may affect the routing configuration during mapping. Once the architecture of the coarse-grained unit is defined in the mapper, the number of the coarse-grained units in the hybrid FPGA has to be specified as this is a resource constraint.

After the architecture of the hybrid FPGA is specified, the mapper can take a dataflow graph as an input and produce a mapped design on the hybrid FPGA. The mapped design consists of placed and routed configuration in every instantiated coarse-grained unit, configuration of the computation units implemented by fine-grained units, and the interconnection between them. The outputs of the mapper are a VHDL description which describes how the coarse-grained units and soft-cores are connected and bitstreams for every coarse-grained unit and how they should be configured.

Each node on the dataflow graph represents a floating point operation in an application. The functionality of the node is not limited by the architecture of the coarse-grained unit. For example, a node can represents square root operation but the coarse-grained units are not necessary to support this. The dataflow graph for the mapper is represented as an assembly-language-like format. For example, the formula $z = \sqrt{a + b \times c + d} \times g$ can be expressed as Figure 3.



```
fadd tmp1, a, b
fadd tmp2, c, d
fmul tmp3, tmp1, tmp2
fsrt tmp4, tmp3
fmul z, tmp4, g
```

Fig. 3. Sample dataflow graph and its representation.

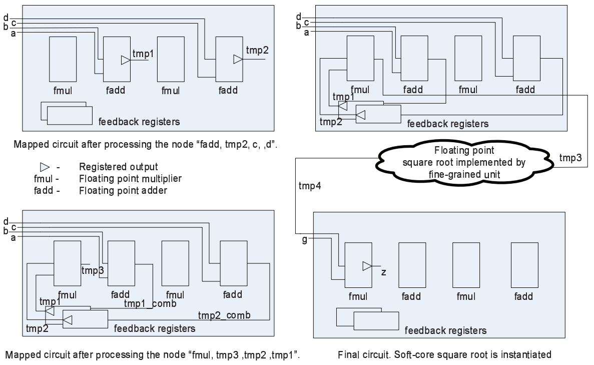
The mapping algorithm of the mapper is based on a greedy search. While this may not produce optimal solution, we find the quality of mapping results is acceptable. It is elaborated in detail in Section 6

The mapper is configured for area-saving and optimises the density of nodes mapped to a coarse-grained unit. The mapper processes nodes sequentially from the text input. When mapping the node to coarse-grained unit, the following condition is considered:

1. Location of input – the mapper tries to map the node to the same coarse-grained unit of the input block.
2. Functionality of the coarse-grained units – the mapper may instantiate soft-cores outside coarse-grained unit if none of them support the operation
3. Availability of the coarse-grained units – the mapper may instantiate soft-core outside coarse-grained unit

if all the coarse-grained units are used up.

4. Number of input bus of the coarse-grained unit – if there is no input available for the coarse-grained unit, the mapper may instantiate new coarse-grained unit for new operation.
5. The location of the floating point operators – feedback registers in the coarse-grained unit may be inferred if the operation requires feedback path.
6. Number of output bus of the coarse-grained unit – if the output bus is used up, some nets cannot be routed to other coarse-grained unit. This may result in unroutable design and backtracking is required to resolve this issue.


d
c
b
a
tmp1
tmp2
fmul
fadd
fmul
fadd
feedback registers
▷ - Registered output
fmul - Floating point multiplier
fadd - Floating point adder
tmp3
tmp1_comb
tmp2_comb
Floating point square root implemented by fine-grained unit
tmp4
g
z

Mapped circuit after processing the node "fadd, tmp2, c, ,d".

Mapped circuit after processing the node "fmul, tmp3 ,tmp2 ,tmp1".

Final circuit. Soft-core square root is instantiated

Fig. 4. Mapping of equation $z = \sqrt{a + b \times c + d} \times g$.

As an example, Figure 4 presents how the formula $z = \sqrt{a + b \times c + d} \times g$ is mapped to the hybrid FPGA.

While this algorithm can be generally applied to most scenarios, there may be a case that a signal, which is in the coarse-grained unit, cannot be routed because all the output bus are utilised. If this happens, backtracking can be applied on that net to trace how the signal is formulated and certain operations are duplicated on a new coarse-grained unit to reproduce the signal.

Using this mapper, the same circuit can be mapped and implemented on different hybrid FPGAs with different configuration of coarse-grained units. Therefore, we can study how the coarse-grained unit affects the power and energy consumption of that circuit.

6. RESULTS

There are several parameters to be specified in the construction of coarse-grained unit. In this paper, the following properties of the coarse-grained unit are assumed: 3 input-buses, 4 output-buses, 4 feedback registers, 5 generic wordblocks, 2 floating point multipliers located in the second and fifth column, 2 floating point adders located in the third and sixth column. The bus widths are all 32-bit and the floating point operators are single precision. In addition Xilinx Virtex II 3000 FPGA is chosen to be host fine-grained fabric for the hybrid FPGA as the technology process is similar to the coarse-grained unit.

Eight benchmark circuits are used in this study. Five of them are computational kernels, one is a Monte Carlo simulation datapath and two of them are synthetic circuits. The *bfly* benchmark performs the computation $z = y + x * w$ where the inputs and output are complex numbers; this is commonly used within a Fast Fourier Transform computation. The *dscg* circuit is the datapath of a digital sine-cosine generator. The *fir4* circuit is a 4-tap finite impulse response filter. The *mm3* circuit performs a 3-by-3 matrix multiplication. The *ode* circuit solves an ordinary differential equation. The *bgm* circuit computes Monte Carlo simulations of interest rate model derivatives priced under the Brace, Gątarek and Musiela (BGM) framework [7]. In addition, a synthetic benchmark circuit generator, based on [8] is used. The generator can produce floating point circuits from a characterisation file describing circuit and cluster statistics. Two synthetic benchmark circuits are produced. Circuit *syn2* contains 5 floating point adders and 4 floating point multipliers. Circuit *syn7* contains 25 floating point adder and 25 floating point multipliers.

The technology mapper can map the applications effectively in terms of the number of coarse-grained units instantiated. For most design, the mapper can instantiate the least possible coarse-grained units, which is determined by the number of floating point operators required divided by the number of floating point operators in a coarse-grained unit. Then the result is rounded up to the nearest integer. Table 1 summarises the performance of the technology mapper. It shows that only circuit *syn7* and *bgm* requires more coarse-grained units than the minimum requirements. This is because some outputs of coarse-grained units are used up so some of the signals cannot be routed. Consequently, the mapper has to replicate some of the operations which increases the number of coarse-grained unit required. It is expected that the utilisation rate of the coarse-grained unit can be improved by increasing the number of output buses.

Circuit	number of adder	number of multiplier	Minimum CGU	Mapped CGU	Relative Difference
bfly	4	4	2	2	0%
dscg	2	4	2	2	0%
fir	3	4	2	2	0%
mm3	2	3	2	2	0%
ode	2	2	2	2	0%
bgm	9	11	6	7	16%
syn2	5	4	3	3	0%
syn7	25	25	13	16	23%

Table 1. Performance of the technology mapper. Most circuits require minimum number of coarse-grained unit (CGU).

The power performance of a hybrid FPGA is compared to the Xilinx Virtex II 3000 which has the same architecture as the hybrid FPGA but no coarse-grained unit. In addition, Xilinx ISE 9.2i is used in FPGA implementation flow. All ASIC designs are synthesised in Synopsys Design Compiler V2006.06 and the dynamic power dissipation is obtained from Synopsys Power Compiler V2006.06.

In order to determine a suitable output loading of the coarse-grained unit, an embedded multiplier is implemented using standard cell flow which has similar architecture as the embedded multiplier used in Virtex II FPGA. The output loading of each pin on the embedded multiplier is adjusted to match the same dynamic power consumption as the one in Virtex II FPGA. We find that the output loading is 4.5pF. Therefore this value is used as the output loading of the coarse-grained unit.

	Hybrid FPGA			XC2V3000-6		Ratio
Circuit	# of CGU	Power (mW)	Freq. (MHz)	Power (mW)	Freq. (MHz)	
bfly	2	204	343	791	86	3.9
dscg	2	185	343	609	88	3.3
fir	2	130	310	674	89	5.2
mm3	2	137	259	544	85	4.0
ode	2	135	309	458	90	3.4
bgm	7	398	221	1,806	79	4.5
syn2	3	204	341	781	88	3.8
syn7∗	16	1,084	342	3,441	76	3.2
					Geometric Mean:	3.9

Table 2. Power estimations. ∗Circuit *syn7* cannot fit in a XC2V3000-6 FPGA so the power number of FPGA implementation is obtained from a XC2V8000-5 FPGA.

Table 2 summarises the dynamic power consumption of hybrid FPGA and Xilinx Virtex II FPGA. Our finding agrees with [4] which suggests the dynamic power consumption ratio of FPGA to ASIC is around 12. The dynamic power consumption of the hybrid FPGA architecture stands between these ratio. Based on this observation, we are more confident in the proposed power estimation flow.

Since the hybrid FPGA can run higher frequency than the Virtex II FPGA, it is expected that for the same operation, the hybrid FPGA can complete faster. Their energy consumption is different as the elapsed time is different. Figure 5 illustrates the ratio in dynamic energy consumption between the hybrid FPGA and the Virtex II FPGA. The energy consumption for is determined by dynamic power consumption divided by the operating frequency. On average, floating point applications implemented on hybrid FPGA can reduce dynamic energy consumption by a factor of 14 compared to the Virtex II FPGA.

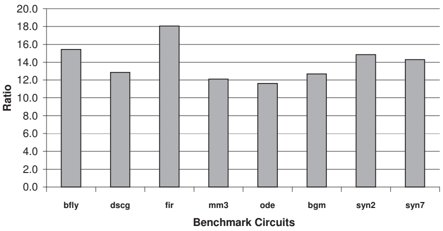


Fig. 5. Dynamic energy consumption ratio.

7. CONCLUSION

This paper introduces a rapid approach to estimate dynamic power consumption in hybrid FPGAs. Both FPGA and ASIC tool chains are used in the estimation. In addition, a technology mapper is proposed to translate benchmark applications to different hybrid FPGA architecture to eliminate the manual mapping process. Using this method, the area, delay and power consumption of a hybrid FPGA can be estimated. Future research includes verification of this approach by comparing with other power estimation flows such as VPR [5], studies of different hybrid FPGA and coarse-grained unit architectures and extending the approach to estimate static power consumption.

Acknowledgements

The support of the UK EPSRC (grants EP/C549481/1 and EP/D060567/1), the ORS Award scheme and Xilinx, Inc. is gratefully acknowledged.

References

[1] T. Tuan, S. Kao, A. Rahman, S. Das, and S. Trimberger, "A 90nm low-power FPGA for battery-powered applications," in *Proc. FPGA*, 2006, pp. 3–11.

[2] C. Ho, C. Yu, P. Leong, W. Luk, and S. Wilton, "Domain-Specific FPGA: Architecture and Floating Point Applications," in *Proc. FPL*, 2007, pp. 196–201.

[3] C. Ho, P. Leong, W. Luk, S. Wilton, and S. Lopez-Buedo, "Virtual Embedded Blocks: A Methodology for Evaluating Embedded Elements in FPGAs," in *Proc. FCCM*, 2006, pp. 35–44.

[4] I. Kuon and J. Rose, "Measuring the Gap Between FPGAs and ASICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 203–215, Feb. 2007.

[5] K. K. W. Poon, S. J. E. Wilton, and A. Yan, "A detailed power model for field-programmable gate arrays," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 10, no. 2, pp. 279–302, 2005.

[6] Xilinx Inc., *Web Power Tool User Guide*. http://www.xilinx.com/ise/power_tools/wpt_help/app_docs/ web_power_tool_help.htm.

[7] G. Zhang, P. Leong, C. H. Ho, K. H. Tsoi, C. Cheung, D.-U. Lee, R. Cheung, and W. Luk, "Reconfigurable acceleration for Monte Carlo based financial simulation," in *Proc. ICFPT*, 2005, pp. 215–222.

[8] P. D. Kundarewich. and J. Rose, "Synthetic circuit generation using clustering and iteration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 6, pp. 869–887, June 2004.