A New Reconfigurable Clock-Gating Technique for Low Power SRAM-based FPGAs

L. Sterpone
Dipartimento di Automatica e Informatica
Politecnico di Torino
Torino, Italy
luca.sterpone@polito.it

L. Carro, D. Matos
Departemento de Informatica Aplicada
UFRGS
Porto Alegre, Brazil

S. Wong, F. Fakhar
Computer Engineering Laboratory
TU Delft
Delft, The Netherlands

Abstract—**Power consumption is dramatically increasing for Static Random Access Memory Field Programmable Gate Arrays (SRAM-FPGAs), therefore lower power FPGA circuitry and new CAD tools are needed. Clock-gating methodologies have been applied in low power FPGA designs with only minor success in reducing the total average power consumption. In this paper, we developed a new structural clock-gating technique based on internal partial reconfiguration and topological modifications. The solution is based on the dynamic partial reconfiguration of the configuration memory frames related to the clock routing resources. For a set of design cases, figures of static and dynamic power consumption were obtained. The analyses have been performed on a synchronous FIFO and on a r-VEX VLIW processor. The experimental results shown that the efficiency in the total average power consumptions ranges from about 28% to 39% with respect to standard clock-gating approaches. Besides, the proposed method is not intrusive, and presents a very limited cost in term of area overhead.**

I. Introduction

Power consumption is one of the key concerns for portable and mobile applications [1]. The increasing need of low-power Integrated Circuits (ICs) is currently driving the development of new solutions and techniques to reduce their power consumption. Recently, among the several technologies available, reconfigurable devices such as Static Random Access Memory Field Programmable Gate Array (SRAM-based FPGA) have been the goal of a plethora of research activities oriented to the reduction of their power consumption. This has been due to the strong appealing that mobile applications have for SRAM-based FPGAs, since they combine reconfigurable characteristics, increasing design density, high speed and low overall design cost [2]. However, power consumption is exceptionally critical in these devices, since unlike ASICs, SRAM-based FPGAs are composed of several resources such as logic blocks, interconnections and I/O pads, all programmed by a configuration memory of SRAM-cells and characterized by significant power consumption [3].

Recently, researchers investigated the average power consumptions of FPGAs, showing that on the average for a given circuit, power is 10 times greater in FPGAs than in customized ASICs [4]. Dynamic power counts as a large percentage of the power dissipation of current SRAM-based FPGAs, which is mainly related to the charging, and discharging of parasitic capacitance when logic transitions occur [5]. Several techniques have been developed in order to reduce the power consumption of reconfigurable FPGAs. They rely on four different methods: clock-gating, placement-based, supply voltage reduction and reconfiguration-based.

Clock-gating is a method of reducing the switching activity on circuit signals, and it is based on the temporarily disabling of the clock signal in specific registers, whenever their outputs are not relevant [6]. This technique, albeit having been successfully used in ASICs, is not that effective in SRAM-based FPGAs, as it does not achieve a drastic power reduction since a large component of power consumption is due to the switching activities of the clock signals along the routing switches [7]. For this reason researchers investigated the possibility of modifying the way a circuit is mapped on the FPGA array by acting on the synthesis, technology mapping or placement and routing algorithms [5][6][8][9][10][11].

Since the propagation of the clock is performed through the global FPGA routing network, the placement of clock loads has a considerable impact on clock wire usage. Clock load placement can be done such that the clock capacitance is minimized, reducing the dynamic power [10]. Although being able to optimize the leakage power, in general these solutions are able to partially reduce the power consumption, since they are limited by the intrinsic routing congestion and the complexity of a design [5].

A field-programmable dual-Vdd configuration for FPGA power reduction has been proposed in [12]. Although effective, this approach requires the design of novel FPGA logic and interconnection fabric that includes Vdd programmability; therefore it is not applicable to Commercial-of-the-Shelf (COTS) FPGAs.

Reconfiguration of the entire FPGA has been proposed in [13] [14]. These approaches rely on the off-line generation of multiple bitstreams, each one related to a single configuration of the circuit where parts of the circuit have low power consumption. The generated bitstreams are downloaded into the FPGA by an external controller, which configures the FPGA according to the portion of the circuit that has low power consumption constraints. This technique presents a main drawback that is the interruption of the circuit

978-3-9810801-7-9/DATE11/©2011 EDA

functionalities during the modification of the bitstream, since during the full reconfiguration of the FPGA its operation is blocked.

In this paper we propose a novel low power technique by applying structural clock-gating to designs mapped on SRAM-based FPGAs. The method is based on the dynamic reconfiguration of the clock routing resources by directly acting on the configuration memory contents. The proposed approach is general and it is platform independent since it can be used on whatever SRAM-based FPGAs adopting internal partial reconfiguration facilities. The proposed method is based on a controller directly mapped on the logic resources available on the FPGA performs the access to the configuration memory content by means of the Internal Configuration Access Port (ICAP) available in all the most recent SRAM-based FPGA devices [14]. The controller is able to perform the basic reading and writing operations for modifying the configuration memory frames related to the clock interconnections, in order to enable or disable the propagation of the clock signal to the next FFs or circuit modules.

The developed technique has several advantages with respect to the existing power-savings techniques for SRAM-based FPGAs. First of all, the application of topological clock gating is not intrusive, since it is implemented without inserting logic resources at the clock input signals of the specific registers of circuit modules. Enabling or disabling the routing segment controlling the correspondent configuration memory bit performs the enabling or disabling of the clock. Secondly, the application of the proposed approach is independent from the user design since all the configuration memory bit locations of each routing segments are available, and the reconfiguration of the routing segments can be performed dynamically without interrupting the FPGA's circuit functionalities. The configuration memory bit locations within the bitstream has been extracted thanks to an experimental characterization of all the FPGA's configuration memory locations [15]. The cost of the configuration memory controller is extremely limited, since it requires less than 1% of the sequential and logic resources available on a medium size SRAM-based FPGA, while the power required for writing a single clock configuration memory frame is negligible with respect to the obtained power saving.

We applied the proposed technique on a benchmark applications consisting of a synchronous First-In First-Out (FIFO) circuit implemented on a Virtex-V SRAM-based FPGA and an r-VEX core implemented on Xilinx Virtex-4. The proposed technique demonstrated a drastic reduction of the total power consumption. In details, we obtained an average power-savings of more that 63% with respect to the FIFO without application of any power-savings techniques, while we obtained an improvement of more than 28% with respect to the standard clock-gating approach.

The paper is organized as follows. Section II describes related works on power reduction in SRAM-base FPGAs, focusing on clock-gating techniques and gives a background on the FPGA configuration memory and the internal routing clock structures. Section III presents the developed Reconfigurable-based Clock Gating (R-CG) technique. Experimental results are presented in Section IV and final conclusions and future works are presented in Section V.

II. FPGA Architecture and Low Power Techniques

SRAM-based FPGAs consist of a configuration memory with several SRAM cells that allow one to program almost any combinational and sequential circuit by configuring the logic functions implemented on Look-Up-Tables (LUTs) and FFs and to program the interconnections between them. LUTs, FFs and routing resources are organized in an island-style matrix divided in reconfigurable regions, where Configurable Logic Blocks (CLBs) are surrounded by hardwired interconnections divided in horizontal and vertical channels. Each CLB contains programmable LUTs and FFs and a switch matrix of programmable interconnection points (PIPs) as illustrated in Figure 1.a, b and c. Specific configuration memory bits configure the global programmable points, the local region programmable points and PIPs of each switch matrix.

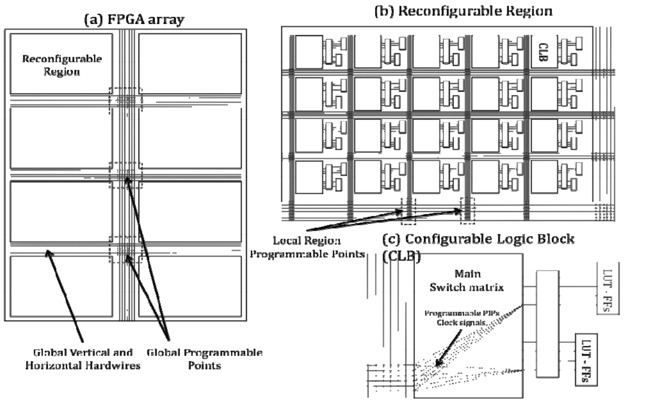
(a) FPGA array
Reconfigurable Region
(b) Reconfigurable Region
CLB
Local Region Programmable Points
(c) Configurable Logic Block (CLB)
Main Switch matrix
Programmable PIPs Clock signals
LUT - FFs
LUT - FFs
Global Vertical and Horizontal Hardwires
Global Programmable Points

Fig. 1. The FPGA architecture hierarchical scheme consisting of the FPGA array (a) of reconfigurable regions (b) containing the matrix of Configurable Logic Blocks (CLBs) (c).

Clock signals are distributed through the global routing and its propagation is divided accordingly to the reconfigurable regions in order to have the minor number of clock signals routed to each CLB. Three kinds of configuration memory bits control the clock distribution:

1. The global programmable points: enable the link between the propagation of the clock on the vertical global hardwires to the horizontal ones. The correspondent configuration memory bits control the access to the reconfigurable regions where the clock tree should be propagated.
2. The local region programmable points: enable the link between the global horizontal hardwires and the local vertical hardwires, which propagates the clock to a specific column of CLBs.
3. The Programmable Interconnection Points (PIPs): activate the link between vertical hardwires and the specific FFs

input.

The clock signals will be exclusively routed into reconfigurable regions containing the specific registers or modules.

Clock-gating, also known as dynamic power management, is a technique to reduce clock induced power consumption. The technique consists in adding a gate-control signal that disables the clock of a circuit whenever it is not used, avoiding power dissipation due to unnecessary charging and discharging of unused circuits, as illustrated in Figure 2 [16]. This methodology can be applied in many circuits, because on any clock cycle, some circuits in the whole system will be idle. When microprocessors are considered, sometimes a subset of functions implemented will be inactive, depending on the type of instruction being executed.

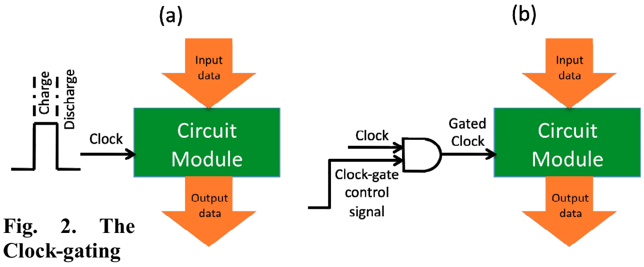

Fig. 2. The Clock-gating approach applied on a circuit module. (a) Circuit module without clock-gating. (b) Circuit module with clock-gating.

The clock-gating technique has been developed for ASICs several years ago [6]. The basic idea is to selectively disable the clock of a specific part of a circuit when outputs or state transitions happen. The transitions and the propagations of clock signal among logic gates and Flip-Flops (FFs) consume a large quantity of energy. Clock-gating is able to reduce the switching activity on those circuit signals and thus to present power-savings by preventing unnecessary activity of specific circuit modules when the outputs of those registers are inconsequential to the circuit outputs, introducing a global decrease of the performances and small area overhead [6].

This technique has been recently applied to SRAM-based FPGAs [17]. Since the FPGA architecture presents a fixed island style geometric structure of the clock trees, the idea is to incorporate, at the design level, the gating capabilities by inserting a feedback multiplexer in front of the input of the FFs, and combining it with the separated clock trees existing on the FPGA global clock interconnections. More recently, FPGA manufacturers embedded clock-gating characteristics directly onto FPGA devices. Modern Xilinx-based FPGAs such as Virtex-5 and Virtex-6 have clock enable pins on the FFs. However, this clock-gating strategy can be applied only on specific regions of the FPGA, and they cannot be easily adapted to the mapped circuit. Besides, the usage of such pins does not provide any power reduction on the clock tree itself [18] [19].

As SRAM-based FPGAs have several hardwired and programmable connections that are predefined, the effectiveness of traditional power reduction techniques like Vdd reduction and clock gating is compromised. In details, one cannot just reduce the Vdd of non-critical path, and even while using clock gating the clock is still routed to several other routing switches on the chip. By using the dynamic reconfiguration-based topological modification of the clock routing here proposed, one could achieve a substantial power savings at a very small cost, as it will be shown.

III. The Porposed Technique: Reconfigurable-Based Clock-Gating

The proposed approach is based on the modification of the clock routing resources programmed on SRAM-based FPGAs. As described in Section II, the clock signal is distributed among different routing segments that are controlled by specific configuration memory bits. The access to the configuration memory bits correspondent to the clock routing resource is possible thanks to the Internal Configuration Access Port (ICAP) available in the most recent SRAM-based FPGA devices [14]. The modifications of the configuration memory bits allow opening or closing the correspondent routing segment thus enabling or disabling the propagation of the clock signal to the next circuit modules. The structural modification of the clock routing segments allows inserting clock-gating characteristics without requiring the addition of a control-gate (typically an AND gate). Thanks to the decoding of the configuration memory bits we performed in [15], we implemented a configuration memory controller able to configure the clock routing resources in order to disable the clock propagation to the circuit modules when the outputs and the state of those registers are not relevant. Configuration may be performed dynamically, without interrupting the FPGA functionalities with basic reading and writing operations on the FPGA's configuration memory.

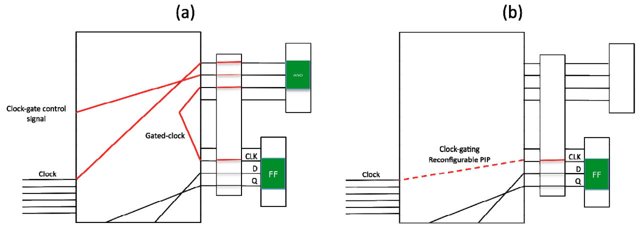

Fig. 3. The basic reconfigurable clock-gating approach. (a) The standard implementation of the clock-gating structure on FPGA. (b) The proposed clock-gating reconfigurable PIP.

The application of clock-gating techniques requires the addition of extra logic in order to implement control gates. Generally, when standard clock-gating technique is applied to SRAM-based FPGAs that leads to the usage of a LUT for each inserted control gate and the addition of extra routing. In details, considering the placement and routing of a single FF on the FPGA architecture, in order to clock-gate a unique FF, it is necessary to add a LUT programmed as an AND gate and to program at least 5 PIPs as illustrated in Figure 3.a. The proposed approach does not require any extra logic and routing resources, since we directly modify the configuration memory bits in order to connect and disconnect properly the Clock-gating Reconfigurable PIP, as illustrated in Figure 3.b.

In details, when the configuration memory bits related to the clock-gating PIP are configured, the clock signal is connected to the FF's clock input. Vice versa, nullifying the PIP's configuration memory bits interrupts the propagation of the clock to the FF's clock input. Therefore, the clock-gating reconfigurable PIP is disconnected, and a ground logic value is propagated to the FF's clock input, blocking the clock-induced power dissipation on the FF.

A. Clock routing segments hierarchical identification

Generally a circuit mapped on SRAM-based FPGAs may consist of several modules hierarchically organized. Besides, each circuit's module may use several FFs, and their position may vary over the FPGA's array depending on the place and route optimization method performed. In order to apply the proposed technique it is necessary to identify the hierarchical distribution of the clock routing segments, as this allows one to identify the most suitable clock-gating configuration points.

Starting from the Native Circuit Description (NCD) of the circuit mapped on the SRAM-based FPGA, we extracted all the physical information related to the place and route description creating a routing graph consisting of vertices and edges correspondent to the logic elements, either combinational than sequential, and routing resources. Secondly, we extracted from the routing graph the clock routing tree identifying each routing bifurcation and the destination sequential points. It is important to observe that it is impossible to extract this information from the circuit netlist, since that format does not contain the physical information related to the hardwired or PIP routing resources. The identified clock routing resources are combined with the bitstream database location producing the bit coordinates of the configuration memory cells controlling the clock routing.

The final result of this flow is a database containing the configuration bits coordinates and the respective hierarchical position within the mapped circuit. The configuration memory bits are stored as frame and bit number according to the FPGA's configuration memory physical organization [18][19], where the frame and bit number allow the recognizing of the bit column and the bit row respectively.

Two kinds of configuration memory bits are identified: those related to the clock routing resources correspondent to each circuit's module clock input and the ones related to the clock routing resources of each FF's input. The bit coordinates will be stored into the clock-gating configuration controller depending on the clock-gating strategy: modular or FF-based.

B. The clock-gating configuration memory controller

The main core of the proposed approach is based on the clock-gating configuration memory controller. It consists of a hardware module mapped on the SRAM-based FPGA that directly accesses to the ICAP port.

The controller consists of four modules: clock-gating control, bit coordinates storage, configurator and the ICAP controller. The *clock-gating control* receives the control signals typically connected to the circuit's enable signals and manages the configurator and the bit coordinates storage. The clock-gating control unit can be modified and adapted to the specific characteristics of the circuit under power-optimization. The definition of the clock-gating characteristics is strictly related to the granularity level where it is applied. In the present paper, we applied the proposed method to a modular level. The approach is not directly applied to each FF, because in this case, the power consumption spent in reconfiguring all the clock-interconnections at the FF's input will be more power consuming than standard clock-gating technique. Besides, we parameterize the penalty of the re-configuration in order to detect the clock idleness. For each tested circuit, we defined a different clock idleness profile, in order to avoid continuous reconfiguration of a design portion when the clock enable toggles every clock cycle.

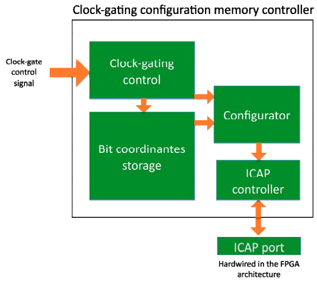
Clock-gating configuration memory controller
Clock-gate control signal
Clock-gating control
Configurator
Bit coordinantes storage
ICAP controller
ICAP port
Hardwired in the FPGA architecture

Fig. 4. The scheme of the clock-gating configuration controller.

When a clock-gate control signal is activated the module performs the following operations: it identifies the bit coordinates associated to the clock-routing segment related to the enable signal, it activates the bit coordinates storage and the configurator module.

The *bit coordinates storage* contains the bit coordinates organized in data words, each one containing the frame code and the bit number addressed by the clock-gate signal. It implements a Content Addressable Memory (CAM) that returns to the configurator module the coordinates of the configuration memory bits related to the activated clock-gate signal.

The *Configurator* is the main module of the controller. It executes all the configuration operations by acting on the ICAP controller. The configuration consists in two steps: the clock-gating selection and the configuration process. In the clock-gating selection the controller writes the logic value of the selected configuration memory bits depending from the kind of configuration. Two configurations are possible: *enabling* and *disabling*. When the enabling configuration is activated the logic value is '1', vice versa the configuration controller will write the logic value '0'. Once the temporary value of the clock-gating configuration memory bits is defined, the configurator performs the configuration process.

The configuration process is performed according to the ICAP protocol that has been implemented by the ICAP controller. The FPGA's clock configuration memory frames are divided according to the reconfigurable regions, as illustrated in Figure 1. The configurator progressively transmits to the *ICAP controller* the 8 bits word for the reconfiguration process. Once the configuration is performed, the controller interrupts its operations.

IV. Experimental Results

The effectiveness of the proposed approach has been demonstrated using two benchmarks. One consisting of a synchronous First-In First-Out (FIFO) circuit implemented on a XC5VLX30 Virtex-5 SRAM-based FPGA having 30,720 Logic Cells and 4,800 Slices and secondly, we evaluated the performances of the Delft Reconfigurable VLIW processor (ρ-VEX) implemented on a Virtex-4 XC4VFX60 having 59,904 Logic Cells and 26,624 Slices.

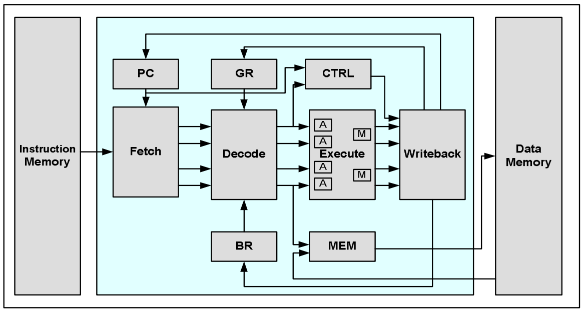

Fig. 5. 4-issue ρ-VEX VLIW Processor.

The Delft Reconfigurable VLIW processor (ρ-VEX) is a reconfigurable softcore VLIW. The ISA of the processor is based on the VEX ISA [21]. Different parameters of the processor, such as the number and type of functional units (FUs), number of multiported registers (size of register file), number and type of accessible FUs per syllable, width of memory buses, and different latencies can be changed. Figure 5 depicts the organization of a 32-bit, 4-issue ρ-VEX VLIW processor. The ρ-VEX processor consists of fetch, decode, execute, and writeback stages. The fetch unit fetches a VLIW instruction from the attached instruction memory, and splits it into syllables that are passed on to the decode unit. In this stage, register contents used as operands are fetched from the register file. The actual operations take place in either the execute unit, or in one of the parallel branch or control (CTRL) or load/store or memory (MEM) units. Arithmetic logic unit (ALU) and multiplier (MUL) operations are performed in the execute stage. All jump and the CTRL unit handles branch operations, and the MEM unit handles all data memory load and store operations. All write activities are performed in the writeback unit to ensure that all targets are written back at the same time. The different write targets could be the general register (GR) file, branch register (BR) file, data memory or the program counter (PC).

A development tool chain including a parameterized C compiler and a cycle-accurate simulator is freely available from HP [22]. The ρ-VEX processor implements most of the operations of the VEX operations set.

Circuit characteristics

In this section, we describe the implementation characteristics of the circuits on the selected FPGAs.

We mapped three different FIFOs with 256 data words of 16, 32 and 64-bits and a ρ-VEX core; the circuits' characteristics are depicted in Table 1:

1. circuit without the application of any low power technique.
2. CG: circuit with the application of the clock-gating technique.
3. CG-Z: circuit with the application of the clock-gating technique with high-impedance connection (i.e. when the clock is disabled the inputs of the circuit are configured as high-impedance).
4. R-CG: circuit with the application of the proposed Reconfigurable-Clock Gating.

Table 1. The characteristics of the implemented circuits.

Circuit Technique	**LUTs [#]**	**FFs [#]**
FIFO 16bits	**6,628**	**2,382**
FIFO 16bits - CG	**7,214**	**2,382**
FIFO 16bits - CG-Z	**7,628**	**2,382**
FIFO 16bits R-CG	**6,758**	**2,603**
FIFO 32bits	**8,874**	**4,722**
FIFO 32bits - CG	**10,004**	**4,722**
FIFO 32bits - CG-Z	**10,634**	**4,722**
FIFO 32bits - R-CG	**9,004**	**4,973**
FIFO 64bits	**11,248**	**9,368**
FIFO 64bits - CG	**14,284**	**9,368**
FIFO 64bits - CG-Z	**14,832**	**9,368**
FIFO 64bits - R-CG	**11,378**	**9,581**
r-VEX	**12,568**	**10,822**
r-VEX - CG	**14,230**	**10,822**
r-VEX – CG-Z	**14,704**	**10,822**
r-VEX R-CG	**12,978**	**11,026**

In this case, the clock-gating configuration controller has been directly connected to the FIFO, WE and RE signals.

The logic resources required by the proposed R-CG technique are used to implement the clock-gating configuration memory controller. It requires 221 FFs and 130 LUTs where most of the resources are needed to store the configuration data related to configuration memory bits coordinates. Nevertheless the standard clock-gating approaches do not require any FFs overhead, LUTs required are directly proportional with the number of data word length in order to implement the clock-gating. Regarding circuit speed performance, the proposed approach does not introduce any path delay increase since it does not require any intrusive logic resources. Vice versa we observed critical path delay increased by 8-10% on the average across the different circuits implemented using standard CG and CG-Z approaches.

As second real case benchmark, a 4-issue ρ-VEX processor has been implemented on a Xilinx Virtex-4 XC4VFX60 FPGA. It has four ALUs, two multipliers, one load/store unit and one branch unit. It has a 64-element 32-bit 4-write 8-read ports multi-ported register file implemented using BRAMs. For the purpose of dynamic power calculation in this 4-issue processor, we executed a small CPU intensive application on it. The application calculates the 45th element of the Fibonacci series. This processor utilizes 14,740 4-input LUTs and 35 BRAMs for its implementation on the Virtex-4 XC4VFX60 FPGA. The processor can run at a maximum frequency of 80 MHz.

A. Power consumption results

We compared the power consumption characteristics of the proposed technique with standard CG and CG-Z approaches considering a working frequency of 50 MHz in the case of the FIFO and 80 MHz for the r-VEX processor, measuring the power consumption with a *LeCroy WaveRunner 44Xi* model equipped with high-impedance calibrated probes and able to compute the average power consumption. While the ρ-VEX has been configured to execute the specific Fibonacci series program, five functional scenarios have been executed for the FIFO: hold operations, reading and writing with minimum and maximum switching activities. The average results we obtained are illustrated in Table 2.

The results, illustrated in Figure 6, clearly demonstrated that the proposed low power technique (R-CG) obtains an average power savings of more that 62% with respect to the FIFO without any application of low power approach and 40% with respect to the ρ-VEX processor without any low power approach, while we obtained an improvement of 28% and 39% with respect to the standard clock-gating approaches on the FIFO and ρ-VEX processor.

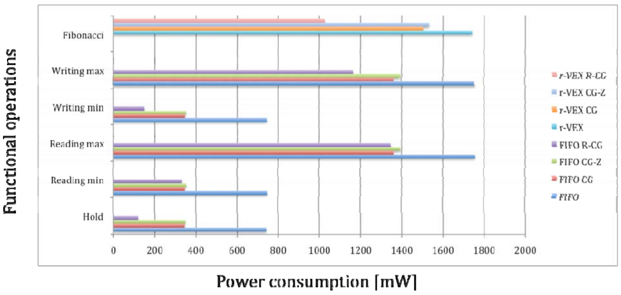

Fig. 6. Power consumptions data comparing the proposed Reconfigurable-based Clock-Gating technique (R-CG) applied to the synchronous FIFO and to the r-VEX VLIW processor, considering circuits without any power-savings technique, with standard clock-gating technique (CG) and with high-impedance clock-gating (CG-Z).

We estimated that the minimum power consumption required for the reconfiguration of clock routing resources is about 15.67 mW.

V. Conclusions and Future Works

In this paper we present a new low power technique for SRAM-base FPGAs based on the structural modification of the clock routing resources. The approach dynamically reconfigures the configuration memory bits related to the clock routing segments. The developed technique has several advantages versus previously existing power-savings techniques for SRAM-based FPGAs. It is not intrusive, therefore it does not degrade the circuit performances and it does not require a large availability of FPGA logic resources.

Future works will involve the analysis of the frequency impact on the developed low power technique in order to characterize the maximum working frequency affordable with the reconfigurable approach. Another future direction involves the evaluation of more complex circuits with a heterogeneous architecture in order to understand the impact of the reconfigurable clock gating. Besides, we are currently investigating the trend of the power consumption with respect of the placement position of logic resources. Since the position of logic elements influences the clock routing three, we expect that an optimized placement will guarantee a greater power savings.

References

[1] P. Jamieson, T. Becker, W. Luk, P. Cheung and T. Rissa, “Benchmarking Reconfigurable Architectures in the Mobile Domain”, IEEE Symposium on Field-Programmable Custom Computing, 2009.

[2] F. Li, D. Chen, L. He and J. Cong, “Architecture evaluation for power efficient FPGAs”, in Proceedings of the ACM International Symposium on Field-Programmable Gate Arrays, Feb. 2003, pp. 175 – 184.

[3] 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, February 2007.

[4] T. Tuan and B. Lai, “Leakage power analysis of a 90nm FPGA”, in Proceedings of IEEE Custom Integrated Circuits Conference, 2003, pp. 57 – 60.

[5] J. H. Anderson, F. N. Najm, “Active Leakage Power Optimization for FPGAs”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 25, No. 3, March 2006.

[6] L. Benini, P. Siegel and G. D. Micheli, “Saving power by synthesizing gated clocks for sequential circuits”, IEEE Design and Test of Computers, Vol. 11, pp. 32 – 41, 1994.

[7] S. Huda, M. Mallick, J. H. Anderson, “Clock gating architecture for FPGA power reduction”, IEEE International Conference on Field Programmable Logic, 2009, pp. 112 – 118.

[8] J. Chen, X. Wei, Q. Zhou, Y. Cai, “Power Optimization through Edge Reduction in LUT-based Technology Mapping”, IEEE International Computer Society Workshop on VLSI, pp. 1087 – 1091.

[9] J. Lamoureux, J. E. Wilton, “On the interaction between power-aware FPGA CAD Algorithm”, IEEE International Conference con CAD, ICCD, 2003, pp. 701 – 708.

[10] J. Lamoureux and S. Wilton, “Clock-aware placement for FPGAs”, IEEE International Conference on Field-Programmable Logic and Applications, 2007, pp. 124 – 131.

[11] K. Vorwerk, M. Rahman, J. Dunoyer, Y. C. Hsu, A. Kundu and A. Kennings, “A technique for minimizing power during FPGA placement”, IEEE International Conference on Field-Programmable Logic and Applications, 2008, pp. 233 – 238.

[12] F. Li, Y. Lin, L. He, “Field Programmability of Supply Voltages for FPGA Power Reduction”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 26, No. 4, April 2007, pp. 752 – 764.

[13] J. Becker, M. Hubner, G. Hettich, R. Constapel, J. Eisenmann, and J. Luka, “Dynamic and partial FPGA exploitation”, Proceedings of the IEEE, 95 (2): 438 – 452, July 2007.

[14] B. Blodget, C. Bobda, M. Huebner and A. Niyonkuru, “Partial and Dynamically Reconfiguration of Xilinx Virtex-II FPGAs”, Lecture Notes in Computer Science, Springer Berlin, Vol. 3203/2004, 2004, pp. 801 – 810.

[15] L. Sterpone, M. Violante, “A new reliability-oriented place and route algorithm for SRAM-based FPGAs”, IEEE Transactions on Computers, Vol. 55, No. 6, June 2006, pp.732 – 744.

[16] Y. Kiat-Seng, R. Kaushik, “Low-voltage, low-power VLSI subsystems”, New York, MCGraw-Hill Professional Engineering, 2005.

[17] Q. Wang, S. Gupta and J. H. Anderson, “Clock power reduction for Virtex-5 FPGAs”, in ACM/SIGDA International Symposium on FPGAs, 2009, pp. 13 – 22.

[18] Virtex-5 FPGA Datasheet, Xilinx Datasheet document, 2007.

[19] Virtex-6 FPGA Datasheet, Xilinx Datasheet document, 2009.

[20] J.A. Fisher, P. Faraboschi, and C. Young, “Embedded Computing: A VLIW Approach to Architecture, Compilers and Tools”, Morgan Kaufmann, 2004.

[22] Hewlett-Packard Laboratories. VEX Toolchain. [Online]. Available: http://www.hpl.hp.com/downloads/vex/