AxB ≠ BxA IN TERMS OF POWER CONSUMPTION: SOME EXAMPLES ON FPGA

Eduardo Boemo and Gustavo Sutter

Escuela Politécnica Superior. Universidad Autónoma de Madrid
{eduardo.boemo, gustavo.sutter}@uam.es

ABSTRACT

This paper shows that, under certain conditions, digital arithmetical circuits do not meet the addition commutation property in terms of power consumption. That is, the power consumed by the operation A×B is different from B×A. As a consequence, it is possible to get a power saving simply permuting the circuit inputs, wherever any of the following three conditions are present: a) The data to be processed has a strong temporal correlation; b) The delays between the circuit paths are highly unequalized; c) One of the input data communication is broadcast type, meanwhile the other is local. In order to verify these hypotheses, several binary multipliers were constructed and measured. The power consumption reduction resulted between 12% and 28% in Virtex FPGAs.

1. INTRODUCTION

In arithmetic circuits like multipliers, an extra number of glitches can produce a significant variation in terms of power consumption. The effect should be visible for a single operation, but should be averaged to zero if the input data sequence is long and random enough. In that case, each pair of data should appear twice, sometimes as A×B, and sometimes as B×A. However, the hypothesis of random input data is rare at the input of the most of the signal processing hardware; usually a strong temporal correlation between successive data exists.

The consumption asymmetry between AxB and BxA can be reinforced by a combination of two additional facts that often are present in arithmetic circuits: a) The broadcasting of one of the operands through global lines, and b) One of the operands involved in the multiplication has a different variation rate, like occurs with the coefficients of a digital filter. In this case, average power will be lowered if the operands with the lower variation rate are introduced into the array using highly-loaded global lines.

Most of pipelined multipliers combine both global and local inputs. Fig. 1 shows an example: the Hatamian-Cash array [6]. If the circuit is pipelined (as is showed by the horizontal lines in the figure), the inputs corresponding to the operand *b* become locals, since they are captured by the flips-flops of the pipeline stages. As the length of the operand increases, the difference of capacitance between the two inputs A and B is increased. In this paper, several experiments have been carried out to validate the previous arguments for more than ten years of FPGA technology: from XC4K device to Virtex series. Section 3 presents the results in early FPGAs. In section 4 and 5, the experiment in Virtex are shown. Main results are summarized in section 6.

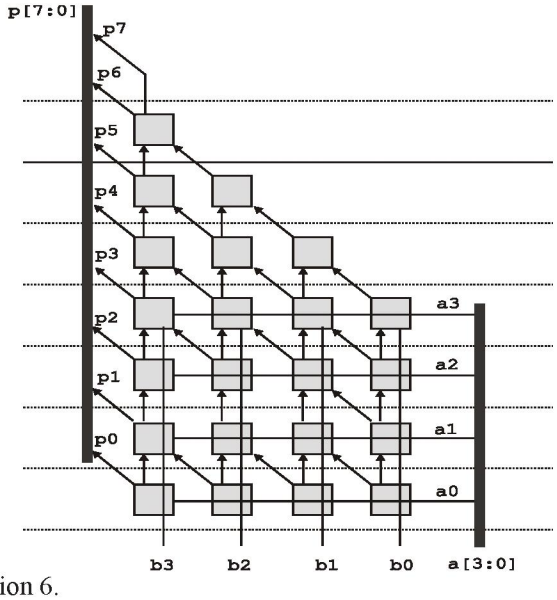

Fig. 1. Pipelining transform b inputs *b* in local, meanwhile *a* data data are globally inputted.

2. EXPERIMENTAL CIRCUITS: 4K SERIES

In order to show the effect of permutation of the operand, the average power of seven different multipliers were measured using 4 different input patters.

The principal characteristics of the benchmark circuits are summarized in table 1. The VHDL-12 and Xcore-9 prototypes have been obtained directly from the standard tools. The first one was synthesized departing from a VHDL behavioural description; meanwhile the second was produced using the Xilinx core generator. Additionally, a

1-4244-0606-4/07/$20.00 ©2007 IEEE.

set of five different array multipliers, pipelined with different logic depths, are included.

Table 1. XC4K-based Benchmark circuit features

Topology	Ref.	CLBs	FF	Logic Depth [LUT]	BW [MHz]
Guild-16	[3]	60	32	16	21.0
VHDL-12	[4]	56	32	12	32.1
Wallace-12	[5]	71	32	12	29.5
Hatamian-8	[6]	75	54	8	25.8
Hatamian-3	[6]	112	207	3	66.2
Hatamian-2	[6]	207	404	2	70.9
Xcore-9	[7]	52	96	9	78.1

3 EXPERIMENTAL RESULTS: 4K FAMILY

All prototypes were measured using the same board and input/output pin assignation. The outputs are loaded only with the logic analyzer probes, whose capacitance is lower that 3 pF [1]. As a consequence, all circuits have the same off-chip power, around 0.13 mW/MHz per output pin (for random numbers), and 0.18 mW/Hz for maximum activity sequences. All measurements were carried out at 2 MHz. The power reduction (PR) is calculated calculating the power saving respect to the worse case. That is:

$$PR = 100 * \frac{P_{AxB} - P_{BxA}}{\max(P_{AxB}; P_{BxA})} \quad (1)$$

Average power was measured using 4 different sets of input patters, whose main characteristics are summarized in table 2. The three power components: datapath, synchronization and off-chip, has been measured [2].

Table 2. Test vectors

Name	Description
Random-1	64 random vectors.
Random-8	512 random vectors with one of the operands presenting a variation 8-times lower.
Toggle-1	16 vectors to maximize activity. In each clock cycle, from 38 to 81% of the input bits, and from 50% to 100% of the output bits change.
Toggle-8	128 vectors. Operand A has a frequency 8-times lower.

As example, in Figure 2 is shown power reduction (PR) obtained for Hatamian-Cash multipliers. In all cases, the greatest power reductions correspond to vectors that maximize the input/output activity. The value of PR for random numbers is low but not zero, probably because of the finite length of the sequence.

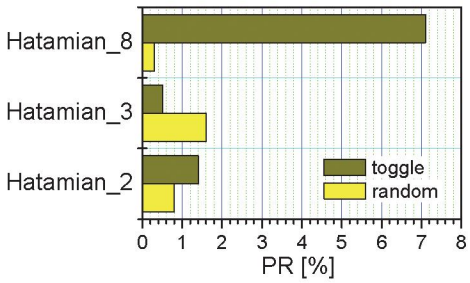

Fig. 2. Consumption reduction. Hatamian-Cash Arrays. XC4010PC84

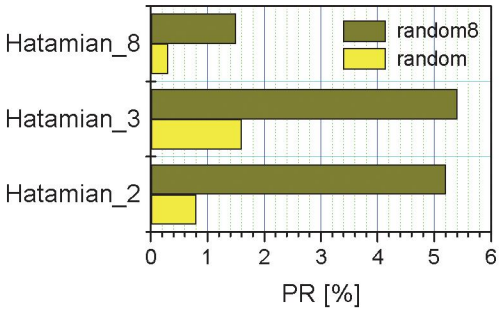

Fig. 3. Effect of global lines and data permutation XC4010PC84.

Vectors that maximize activity enlarge the difference in terms of power consumption. Greater activity in the inputs leads to magnify the glitch effect, the main cause of the asymmetry. The figure also shows that the PR is more significant as the logic depth grows, due to the avalanche effect of early glitches on the overall activity.

The set of Hatamian-Cash arrays also illustrates the effect of the global communication. That is, if the more loaded lines are utilized to input the lower activity data, an extra power reduction is obtained, as is shown in Figure 3. The effect in the Hatamian_8 version is smaller, since it has only two pipelined stages. Thus, the transformation from global to local lines is not significant.

Finally, Figure 4 shows the PR values obtained for the other circuits. With the only exception of the Wallace multiplier, in all cases the effect is magnified if more spurious activity is produced by the input data. Implementations were repeated using three different Xilinx 4K samples, obtaining similar results in all cases. For example, Figure 5 shows the consumption variations for the Hatamian-8 multiplier for two sequences; maximum activity and random (Toggle-8 and Random-1 respectively).

4 EXPERIMENTAL RESULTS: VIRTEX SERIES

In order to demonstrate the effect of permutation of the operands in Virtex devices, five 16-bit and 32-bit multipliers were constructed and measured. The circuits were tested using two sets of vectors. The first sequence, named *MaxTog* has maximum activity in one of the data input and practically no activity in the other input. The second test sequence, *AvgTog*, is random with different frequencies in each one of the operands

Table 3. Benchmark circuits. Virtex Family

32 bits Circuits		
Circuit	**Area** (Slices)	**Bandwidth** (MHz)
Core32	580	20.7
Exp32	561	26.1
Leo32	565	22..5
Syn32	571	21.1
Xst32	576	20.9
16 bits Circuits		
Core16	157	43.4
Exp16	149	50.0
Leo16	150	43.6
Syn16	152	45.2
Xst16	156	45.5

Four of the benchmark circuits depart from a behavioural VHDL code synthesized with different tools. The *Syn* acronym corresponds to Synplify Pro [8], *Xst* to Xilinx Synthesis Technologies [9], *Leo* to Leonardo Spectrum [10] and *Exp* to FPGA Express [11]. The last multiplier was obtained from the core generator CoreGen [12] included in the ISE tool. All multipliers are completely sequential, having registered input/output with slices flip-flops. Their area-time characteristics are shown in Table 3. A Virtex XCV800HQ240 was utilized to implement the circuits. In Tables 5 and 6, both the measured power consumption and PR figure can be observed.

In all cases, there is an important power saving if the order of the inputs is inverted. It is important to remark that, for each synthesizer, the sign of the difference in consumption is maintained for both the 16 bit and the 32 bit multipliers as well as for the sequences. Both Modelsim and the power estimation tool Xpower were also utilized to check the results. In both cases, the order in the inputs that minimizes consumption resulted the same, but not the exact value of power saving

Table 5. Dynamic Power consumption of A×B and B×A in mW/MHz. 16 multiplier set.

16 bits Circuits	MaxTog		
	P (A×B)	P(B×A)	PR
Core16	7,57	5,43	28,2%
Exp16	6,42	6,98	-8,1%
Leo16	7,69	6,01	21,8%
Syn16	5,82	7,63	-23,7%
Xst16	7,21	6,06	15,9%
	AvgTog		
Core16	2,45	2,20	10,5%
Exp16	2,41	2,53	-4,5%
Leo16	2,53	2,26	10,7%
Syn16	2,18	2,37	-8,2%
Xst16	2,40	2,30	4,0%

Table 6. Dynamic Power consumption of A×B and B×A in mW/MHz. 32-bits multiplier set.

32 bits Circuits	MaxTog		
	P(A×B)	P(B×A)	PR
Core32	34,12	27,77	16,7 %
Exp32	23,81	29,39	-6,5 %
Leo32	31,40	27,87	9,3 %
Syn32	32,31	35,12	-16,4 %
Xst32	32,29	29,45	9,3 %
	AvgTog		
Core32	11,92	9,94	18,6 %
Exp32	9,56	10,22	-19,0 %
Leo32	11,70	10,62	11,3 %
Syn32	10,04	12,00	-8,0 %
Xst32	11,71	10,62	8,8 %

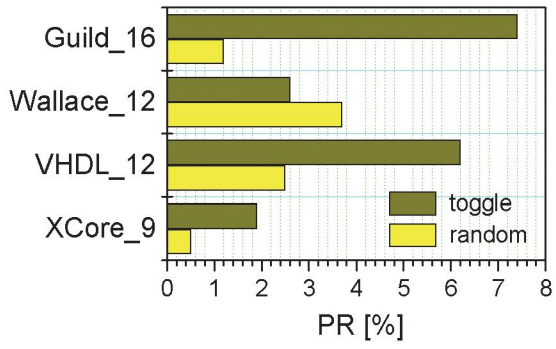

Fig. 4. Consumption reduction for different topologies. XC4010PC84.

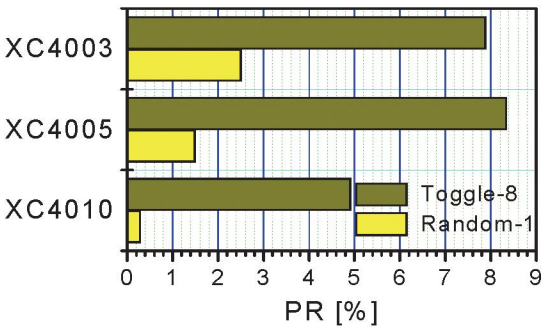

Fig. 5. Consumption reduction for different XC4K devices. Hatamian-8 multiplier.

5 CONCLUSIONS

The non-commutative property of power consumption in binary multipliers has been analyzed.

Under certain conditions, the permuting of the input data can lead to an important power reduction. In the 4K devices family 8 bit multipliers were used, obtaining a power saving of to 8%. In the Virtex devices family, using 16 and 32 bit multipliers the maximum reduction is up to 28%. It can be expected that other combinational blocks can behave in the same way. Finally, the use of a power estimation tools (or just a measurement of activity) can be help the designers to choose the best order of the input operand

6. REFERENCES

1. Tektronix Inc., *"TLA 700 Series Logic Analyzer User Manual"*, http://www.tektronix.com
2. E. Todorovich, G. Sutter, N. Acosta, E. Boemo and S. López-Buedo, "End-user low-power alternatives at topological and physical levels. Some examples on FPGAs", *Proc. DCIS'2000*, Montpellier, France, November 2000.
3. H. Guild, "Fully Iterative Fast Array for Binary Multiplication and Addition", *Electronic Letters,* pp.263, Vol.5, N°12, June 1969.
4 Xilinx corp, "Software Manual on Line: Synthesis and Simulation Design Guide 3.1i http://www.xilinx.com
5 C. Wallace, "A Suggestion for a Fast Multiplier", *IEEE Trans.on Electronic Computers,* pp.14-17, Feb 1964.
6. M. Hatamian and G. Cash, "A 70-MHz 8-bit x 8 bit Parallel Pipelined Multiplier in 2.5-um CMOS", *IEEE Journal of Solid-State Circuits*, August 1986.
7. Xilinx Inc, *"Software Manual on line: CORE generator Guide 3.1"*, http://support.xilinx.com
8. Synpicity Inc*; "Synplify Pro 7.1 On line Documentation"*, April 2002. www.synplicity.com
9. Xilinx Inc; "*Xilinx Synthesis Technology (XST) User Guide"*, http://www.xilinx.com, 2002.
10 Mentor Graphics, *"LeonardoSpectrum Bookcase v2002a"*, 2002. http://www.mentor.com
11. Synopsis Inc, *"FPGA Express 3.6.1 User Guide"*, Agosto 2001. http://www.synopsis.com/fpga/
12. Xilinx Inc; *"Core Generator Guide – ISE 5"*, available at http://www.xilinx.com, 2002.
13. Model Technologies, *"Modelsim 5.6 XE user Manual"*, 2003, http://www.model.com
14. Xpower, *"Xpower getting started"*, Available at http://support.xilinx.com.