

POWER OPTIMIZATION USING CODING METHODS ON ARITHMETIC OPERATORS

Eduardo Costa,
UFRGS Federal Univ. - Microelectronics Group
Caixa Postal 15064 - 91591-970 P. Alegre, Brazil
ecosta,bampi@inf.ufrgs.br

José Monteiro
IST/INESC - Algos Group
Rua Alves Redol, 1000-029 Lisboa, Portugal
jcm@algos.inesc.pt

This paper addresses the use of different coding methods for the arithmetic operators. Signal encoding is widely used to reduce the switching activity in buses. However, the signals need to be encoded and decoded since signal processing is executed in binary. To avoid this step, we investigate the viability of processing operators that use the same signal encoding as that used in the bus.

Gray and a Hybrid encoding for the arithmetic operators (adders and multipliers) are implemented and compared to binary operators. The power consumption of the new operators is computed for both uniform input probabilities and real input traces. The overall power consumption under different capacitive loads is also evaluated to indicate the conditions under which power can be saved using signal encoding.

I. INTRODUCTION

For most of the current IC designs in key high volume applications, meeting the targeted power dissipation budget is of paramount importance. Research in CMOS power modeling and optimization is thriving for more accurate predictions and power efficiency. Recent research [1], [2] has focused on the CMOS power consumption estimation. There is a need for precise estimation to guide the designers towards minimizing and meeting strict power budgets.

This paper focuses on power optimization techniques using coding as a method of decreasing the switching activity on arithmetic operators.

The main goal of the coding schemes is to reduce the overall switched capacitance in the system. In fact, in several applications, switching activity on high-capacitance buses accounts for a large fraction of the total power consumption. Recent research work have dealt with the encoding of the signals on the address buses in order to lower their switching activity [3], [4], [5], [6], [7]. A significant reduction in the switched capacitance is reported, with power savings between 33% and 75% using different coding schemes.

One of the most promising encodings that is used to reduce switching activity is the Gray code since only one bit changes between consecutive values. This is shown in [3] when the data being transmitted over a bus is sequential and highly correlated. In [8] it is observed that the use of Gray code to cache memory addresses can produce about 33% power reduction in the instruction cache.

In our work, we explore the small switching activity presented by Gray code scheme in order to implement Gray code operators. An arithmetic operator based on a variant of the Gray code approach, which we call the Hybrid arithmetic operator,

is also proposed in order to compensate for the non-regular aspect shown by the Gray code structure, as will be shown in the next sections. In this work, area and power consumption results obtained from the Gray and Hybrid operators are compared to Binary operators.

This paper is organized as follows. Section 2 discusses the main aspects related to coding for low power. Synthesized gray code operators are shown in Section 3. In Section 4, we present the Hybrid code definitions and the operators built with it. Section 5 presents the results of the Hybrid code operator on a sequential circuit and in Section 6 we discuss the main conclusions of this work.

II. CODING FOR LOW POWER

Coding has long been used in communication systems to control the statistics of the transmitted data symbols, or in other words, to control the spectrum of the transmitted signal [3].

Low-power techniques for global communication in CMOS VLSI using data encoding methods are overviewed in [4], where it is shown that such techniques can decrease the power consumed for transmitting information over heavy load communication paths (buses) by reducing the switching activity.

One technique that has been proposed in order to reduce the switching activity on buses is One-Hot Coding [3]. This technique is a redundant coding scheme with a one-to-one mapping between the n -bit data words to be sent and the m -bit data words that are transmitted. The main disadvantage of this technique is related to the wire quantity required, proportional to 2^n .

The Limited-Weight Codes is another technique proposed in order to obtain switching activity reduction on buses [5]. This technique requires transition signaling in order to reduce the switching activity, since with transition signaling only 1's generate transitions. According to [5], transition signaling is convenient for low-power as it offers a direct way of controlling the bus activity factor simply by reducing the number of logical 1's transmitted over the bus.

The Bus-Invert method as a means of encoding words for reducing I/O power, in which a word may be inverted and then transmitted if doing so reduces the number of transitions [6]. In this method an extra bus line, called *invert* is used. The method looks at two consecutive data words on the bus. If the Hamming distance between the next word and the current transmitted word is at most $k/2$, the next word is sent as it is, with *invert* set to 0. Otherwise, each bit of the next word is complemented and *invert* is set to 1, indicating that the word has been complemented. The Bus-Invert method is explored in [7] in order to sequence words under the Bus-Invert scheme for the minimum transitions, *i.e.*,

words can be complemented, reordered and then transmitted.

A Gray code sequence is a set of numbers in which adjacent numbers only have one bit difference. Therefore, for highly correlated signals the switching activity can be reduced significantly. This code has been applied to code address lines for both instruction access and data access to reduce the number of transitions [3]. Although there is no overhead in terms of the number of bit lines, translating between Gray and Binary requires complex, *i.e.*, power consuming, encoders/decoders.

III. GRAY CODE OPERATORS

As discussed above, the Gray code would be much more appealing if no encoders/decoders were needed. Therefore, we discuss in this work the implementation of adder and multiplier circuits operating directly with Gray encoded values. Besides avoiding the transcoders, the use of operands encoded in Gray may yield lower switching activity in the processing circuit itself due to the reduced number of transitions in the operands.

The Gray code adders and multipliers proposed were generated by describing them in Programmable Logic Array format and synthesizing them using `script.algebraic` in the SIS environment [9]. Thus, it is possible to obtain their power and area estimates and compare to the Binary code operators. Table I shows area estimates for 2 and 4 bits Binary and Gray adders and multipliers in terms of nodes and literal numbers.

TABLE I

NODES AND LITERAL NUMBERS FOR BINARY AND GRAY ADDERS AND MULTIPLIERS

Code	Area							
	Nodes				Literals			
	Adder		Multiplier		Adder		Multiplier	
	2b	4b	2b	4b	2b	4b	2b	4b
Bin	14	28	12	166	34	68	22	441
Gray	25	135	12	231	63	348	25	594

As shown in Table I, the Gray code adders and multipliers present more nodes and literals than Binary operators do. This occurs because for the Gray code operators the output bits depend on all of the input bits, so its logic circuit is more difficult to be minimized. For the Binary operators, the least significant bits of the result only depend on the least significant bits of the operands. This fact has a direct correlation to the power consumption, as shown in Table II.

In Table II, *uniform input* is the power consumption under an uniform input distribution, *input vectors* represent a sequence between all possible combinations between 0 and the maximum value for that number of bits, and *real trace* applied to 4-bit operators represents 100 points of 2 sinusoidal signals with 90 degree phase difference. Power results are obtained with the power estimate tool in SIS assuming Vdd=5V and freq=20MHz.

As can be observed from Table II, power consumption decreases when input vectors and real trace are applied to input operators. This fact occurs because of the temporal and spatial correlation effect [10]. As expected, the power decreases more in Gray code than Binary code operators. However, since the Gray code has a higher area as shown in Table I, the power con-

TABLE III
HYBRID CODE REPRESENTATION

Dec	Hyb	Dec	Hyb	Dec	Hyb	Dec	Hyb
0	0000	4	0100	8	1100	12	1000
1	0001	5	0101	9	1101	13	1001
2	0011	6	0111	10	1111	14	1011
3	0010	7	0110	11	1110	15	1010

TABLE IV
NUMBER OF TRANSITIONS FOR DIFFERENT CODES

Number of Bits	Number of Transitions			Diff	
	Binary	Gray	Hybrid ($m = 2$)	H→B	H→G
4 bits (0→15→0)	30	16	20	-33%	+25%
8 bits (0→255→0)	510	256	340	-33%	+33%
16 bits (0→65535→0)	131070	65536	87380	-33%	+33%

sumption in Binary code is lower than Gray code. As can also be seen in Table II, Gray multipliers present smaller power value than Binary operators. However, it is difficult to produce a regular structure in Gray code to apply in higher bit operators. It should be observed that it was impossible to synthesize from PLA multipliers with operands with a higher number of bits than those shown in Table II.

IV. HYBRID CODE

As was observed before, it is difficult to generate regular structures to synthesize Gray code. This is an important aspect because it is very difficult to minimize logic from PLA format for operands with a large number of bits. Thus, we propose a Hybrid code as an alternative way, which is a compromise between the Binary and the Gray.

A. Hybrid Code Definition

The idea of the Hybrid code is to split the operands in groups of m -bits, encode each group using the Gray code and use the Binary approach to propagate the carry between the groups. In this manner, the number of transitions inside each group is minimized and a regular structure can easily be built, where the least significant group of the result will also depend only on the least significant group of the operators. Table III exemplifies the Hybrid encoding for 4-bit numbers and groups of $m = 2$ bits.

Table IV presents the number of transitions for a counting sequence for the Binary, Gray and Hybrid codes. The Hybrid code presents a number of transitions right between the Gray and Binary codes, with 33% less number of transitions than the Binary code. Hence, for systems where the switched capacitance in the data buses is significant and where the data presents a high degree of correlation, up to a third of power can be saved.

B. Hybrid Adders

Using the Hybrid code it is possible to generate adders and multipliers with regular structures and then synthesize them for more bits operators than Gray code operators in PLA format.

TABLE II
POWER CONSUMPTION FOR BINARY AND GRAY ADDERS AND MULTIPLIERS

Code	Power(μW):uniform input				Power(μW):input vectors				Power(μW):real trace			
	Adder		Multiplier		Adder		Multiplier		Adder		Multiplier	
	2b	4b	2b	4b	2b	4b	2b	4b	2b	4b	2b	4b
Bin	129	257	59	1547	121	163	51	951	-	79	-	470
Gray	221	1158	73	1823	124	332	44	678	-	177	-	325

Table V shows power results for Binary and Hybrid adders synthesized in BLIF format in SIS environment.

Once again are applied the same input signals used before in Table II. Although it is possible to obtain a regular structure for the Hybrid adder, and synthesize higher adders structures than shown before in Gray structures, it is shown in Table V that Binary adders present smaller power consumption values than the Hybrid adders. This occurs because Hybrid adders present higher area than Binary adders. In BLIF format, the 2 bit Hybrid adder presents 24 nodes and 57 literal numbers against 10 nodes and 28 literal numbers from 2 bits adder structure. This difference is propagated to higher structures.

C. Hybrid Multipliers

For the Hybrid code multiplier, it is possible to generate a simple 2-bit structure using just 8 logic gates, as depicted in Figure 1. This simple structure presents an advantage in terms of area and power compared to the Binary structure as shown in Table VI from BLIF synthesized results. The 2-bit Hybrid multiplier is used in more complex multiplier circuits and the results are shown in Table VII.

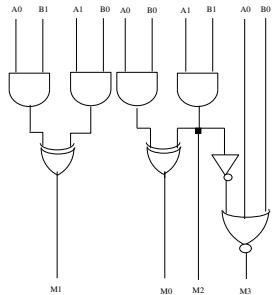


Fig. 1. 2-bit Hybrid multiplier circuit.

D. Hybrid Structure Using EXOR Gates

One of the most important aspects shown by the Hybrid code is related to how easy it is to change to and from the Binary code using EXOR gates in the input/output Binary circuits as shown

TABLE VI
AREA AND POWER FOR 2-BIT BINARY AND HYBRID MULTIPLIERS.

2-bit Multipliers	Area		Power (μW)	
	Nodes	Literals	Uniform	Vectors
Binary	15	36	64.1	56
Hybrid	8	20	55	36.7
Diff(Hyb→Bin)	-47%	-44%	-14%	-34%

in Figure 2 applied to a 2-bit Binary adder. This possibility is attractive due to Hybrid signals correlation aspect. Table VII shows area and power results for the Binary, Hybrid and Binary with EXOR gates multiplier structures synthesized in BLIF format.

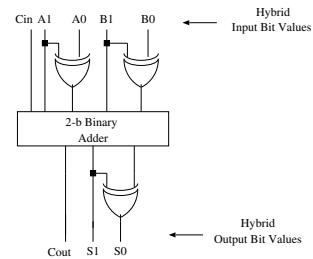


Fig. 2. 2-bit Hybrid Adder.

Table VII results show 4- and 8-bits Binary multipliers as better structures on power optimization than Hybrid structures. This occurs because it is difficult to get a regular Hybrid multiplier structure to be applied to more than 2-bits structure. For example the 4-bit Hybrid multiplier circuit presents 137 nodes and 322 literal against 92 nodes and 212 literal numbers for the Binary structure. Moreover, the use of Hybrid adders in more than 2-bit Hybrid multiplier causes an increase in area and thus power can not be optimized more than shown in Table VII. However, as shown in Table VII, there is a possibility to get power optimization using the Hybrid code with EXOR gates. In this case, it is possible to use a Hybrid signal with high correlation aspects with a more simple Binary structure. The correlation is related to number of points used in real trace, as shown in Table VII

V. ACCUMULATOR CIRCUIT

We have experimented our Hybrid code adder in an accumulator-type circuit, using different capacitive loads for the bus lines, and verified the impact on power consumption. The circuit structure is shown in Figure 3.

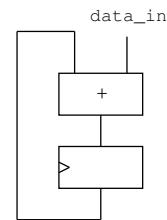


Fig. 3. Accumulator circuit.

We present power dissipation results using the Binary, Hybrid and Binary with EXOR adders, varied the load factor and the

TABLE V
POWER CONSUMPTION FOR BINARY AND HYBRID ADDERS IN BLIF FORMAT

Code	Power (μW) uniform input			Power (μW) input vectors			Power (μW) real trace input		
	2b	4b	8b	2b	4b	8b	2b	4b	8b
Bin	84	167	335	72	95	120	-	57	234
Hyb	170	340	680	131	167	179	-	84	448

TABLE VII
POWER CONSUMPTION FOR BINARY AND HYBRID MULTIPLIERS IN BLIF FORMAT

Code	Power (μW) uniform input			Power (μW) input vectors			Power (μW) real trace input			
	4b		8b	4b		8b	100 points		200 points	
	4b	8b	4b	8b	4b	8b	4b	8b	8b	
Bin	421		2332	266		759	148	1762	77	1574
Hyb	726		4104	385		1570	187	3151	96	2585
Bin with EXOR Gates	537		2554	292		1004	145	1897	75	1548

TABLE VIII

POWER RESULTS FROM 8-BIT SEQ. CIRCUIT (REAL TRACE INPUT)

Load Factor	Power (μW)-100 points			Power (μW)-200 points		
	Bin	Hyb	Bin/Hyb	Bin	Hyb	Bin/Hyb
10	386	606	474.5	386	576	466
50	1114	1308	1177	1111	1276	1165
100	2024	2186	2055	2018	2149	2039
150	2934	3063	2932	2924	3023	2913
500	9304	9206	9075	9268	9139	9029
1000	18104	17981	17849	18330	17877	17766

TABLE IX

POWER VALUES FROM 16-BIT SEQ. CIRCUIT (REAL TRACE INPUT)

Load Factor	Power (μW) 1000 points		Power (μW) 10000 points		Power (μW) 50000 points	
	Bin	Bin/Hyb	Bin	Bin/Hyb	Bin	Bin/Hyb
10	840	1032	797	966	768	920
50	2350	2538	2303	2458	2275	2520
100	4238	4419	4184	4323	4158	4483
150	6125	6301	6066	6188	6041	6137
500	19338	19474	19236	19242	19222	19179
1000	38213	38291	38050	37891	38052	37811

number of points of the real trace. Tables VIII and IX present results for 8- and 16-bit structures. In Tables VIII and IX, the structures were synthesized in BLIF format. The load factor is multiplied to the smallest load associated to the fan-in of a gate (0.01 pF). The load variation values are applied to buses at the adder and register outputs and real trace signals are applied to the other of input adder. As can be observed from Tables VIII and IX, it is possible to obtain power reduction in the accumulator circuit using more points in the real trace input signal. However, it is possible to obtain more power optimization using a Hybrid input signal with EXOR gates, as shown in Table VIII. In Bin/Hyb column it is shown a smaller power value. In Table IX it is shown the importance of the number of points in the trace input signal in a 16-bit structure. Using 50000 points for the input signal it is possible to reduce power with a load factor=500. In Binary, with EXOR gates, this aspect shows that more points are needed in complex structures to obtain a better signal correlation impact.

VI. CONCLUSIONS

The reduction of the number of transitions in order to minimize power consumption is the main goal of techniques that use different coding schemes on buses. In this work we presented a new approach for the use of different coding scheme, as a method of decreasing the power consumption of arithmetic operators. The results show that other possibilities can be used on power optimization than Binary code. However, this work also showed that it is difficult to obtain regular structures in multiplier circuits such as it is possible with Binary operators. In this work it has been presented the use of different coding adder operators in accumulator circuits and was possible to obtain power optimization related to load factors and real trace input signals. As future work we hope to explore regular structures with other different codes to obtain power optimization.

VII. ACKNOWLEDGMENT

This research was supported in part by the CAPES (Brazil) and by the portuguese FCT under program POCTI.

REFERENCES

- [1] F. Najm. A Survey of Power Estimation Techniques in VLSI Circuits. *IEEE Transactions on VLSI Systems*, 2(4):446–455, December 1994.
- [2] C-Y. Tsui, J. Monteiro, M. Pedram, S. Devadas, A. Despain, and B. Lin. Power Estimation Methods for Sequential Logic Circuits. *IEEE Transactions on VLSI Systems*, 3(3):404–416, September 1995.
- [3] A.P. Chandrakasan and R.W. Brodersen. *Low Power Digital CMOS Design*. Kluwer Academic Publishers, 1995.
- [4] M.R. Stan and W.P. Burleson. Low-Power Encodings for Global Communication in CMOS VLSI. *IEEE Transactions on VLSI Systems*, March 1997.
- [5] M.R. Stan and W.P. Burleson. Limited-weight Codes for Low-Power I/O. *IEEE International Workshop on Low Power Design*, April 1997.
- [6] M.R. Stan and W.P. Burleson. Bus-Invert Coding for Low-Power I/O. *IEEE Transactions on VLSI Systems*, March 1995.
- [7] R. Murgai, M. Fujita, and M. Oliveira. Using Complementation and Resequencing to Minimize Transitions. In 35th *Design Automation Conference*, June 1998.
- [8] C. Su and A. Despain. A Cache Design Tradeoffs for Power and Performance Optimization AQ Case Study. In *IEEE International Symposium on Low Power Design*, April 1995.
- [9] E.M. Sentovich and et al. SIS: A System for Sequential Circuit Synthesis. Technical report, May 1992.
- [10] P.E. Landman and J.M. Rabaey. Architectural Power Analysis: The Dual Bit Type Method. *IEEE Transactions on VLSI Systems*, 3(2):173–187, June 1995.