Low Power 3-2 and 4-2 Adder Compressors Implemented Using ASTRAN

Jorge Tonfat, Ricardo Reis
jorgetonfat@ieee.org, reis@inf.ufrgs.br

Grupo de Microeletrônica (GME) – PGMICRO – UFRGS, Porto Alegre, RS, Brasil

Abstract

This paper presents two adder compressors architectures addressing high-speed and low power. Adder compressors are used to implement arithmetic circuits such as multipliers and digital signal processing units like the Fast Fourier Transform (FTT). To address the objective of high-speed and low power, it is well known that optimization efforts should be applied in all abstraction levels. In this paper are combined optimizations at logic, electrical and physical level. At the logic level, the circuit is optimized by using multiplexers instead of XOR gates to reduce delay, power and area. At the electrical level, this work presents an architecture that generate the XOR and XNOR signals simultaneously, this reduce internal glitches hence dynamic power as well. And finally at the physical level, and automatic layout generation tool (ASTRAN) is used to make the adder compressors layouts. This tool has proved to reduce power consumption and delay due to the smaller input capacitances of the complex gates generated compared to manual-designed layouts.

1. Introduction

Nowadays, it is advised an enormous development of mobile electronic gadgets ready for multimedia environments such as smartphones, portable video games and tablets. The main features of these devices are low power consumption and high performance. These two characteristics are contradictory. Hence we must find the best trade-off between power and delay to design such devices. Process technology scaling seems to complicate this situation. Thus design solutions are the main mechanism to keep the power consumption budget in control. The design optimization effort should be applied in all abstraction levels to obtain better results [1]. Traditional design flows as the standard cell approach limits the optimization effort because cell's library limits the number of logical functions as well as the limited sizing versions of these functions. In this context, the library-free approach appears as a possible solution [2]. The ASTRAN [3] (Automatic Synthesis of Transistor Networks) tool is used in this work to generate layouts of CMOS cells from its transistor level netlist description. Results from [4] show the effectiveness in terms of power and delay reduction compared to manual-designed layouts from a standard cell library.

This paper is organized as follows: In Section 2 is discussed previous works on adder compressors and is presented the architectures analyzed in this work. Subsequently in section 3 is presented the ASTRAN tool used in this work to automatic generate the layouts of the circuits. Simulation results for both compressors are shown in section 4 and finally the conclusions are presented in section 5.

2. Adder Compressor Architectures

Adder compressors have been used to implement arithmetic and digital signal processing (DSP) circuits for low power and high performance applications [5]. Compressors are also used in multiplier architectures. Multipliers are structured into three functions: partial-product generation, partial-product accumulation and final addition. The main source of power, delay and area came from the partial-product accumulation stage [6]. Compressors usually implement this stage because they contribute to the reduction of the partial products (reducing the number of adders at the final stage) and also contribute to reduce the critical path which is important to maintain the circuit's performance.

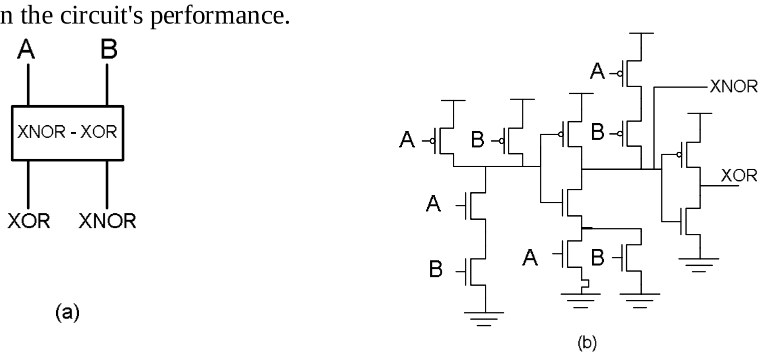


Fig. 1 – (a) XOR-XNOR module. (b)XOR-XNOR implemented with CMOS logic.

Compressors are composed basically of two types of modules: XOR-XNOR complex gates and multiplexers (MUX). A CMOS logic implementation of both modules is shown in Figure 1(b) and Figure 2(b).

A design space exploration for 4-2 and 5-2 compressors is presented in [6]. It shows the different possibilities to implement adder compressors based on different logic styles but with the same architecture. The results show some combinations of logic styles for the XOR-XNOR and MUX modules that achieve better results in terms of delay and power. For the XOR-XNOR module, the circuit from Figure 3(a) presents better features than others. And for the MUX module, a MUX based on transmission gates with an output buffer (Figure 3(b)) presents best results.

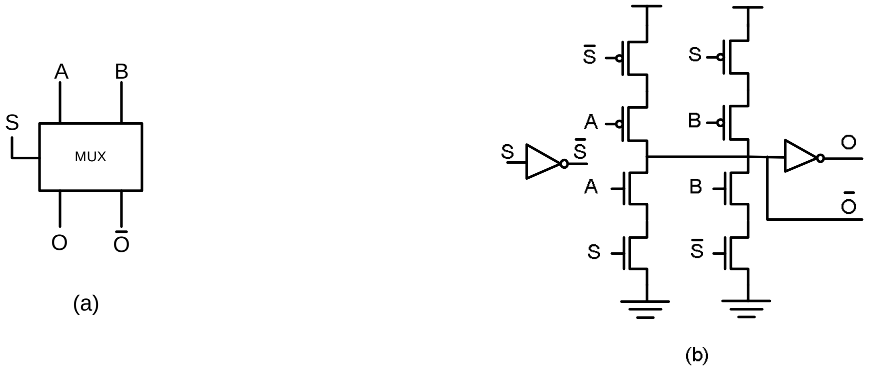


Fig. 2 – (a) Multiplexer (MUX) with two outputs. (b) Multiplexer (MUX) implemented with CMOS logic.

A new architecture is presented in [7]. In this architecture the emphasis lay on the use of multiplexers instead of XOR gates. This is because the use of multiplexers improves the speed when placed in the critical path [8]. For the XOR-XNOR module, this work uses the traditional CMOS logic style and for the MUX module they use a combination of a traditional CMOS logic style with a transmission gate logic style that is only used in the internal paths of the adder due to the limited driving capability.

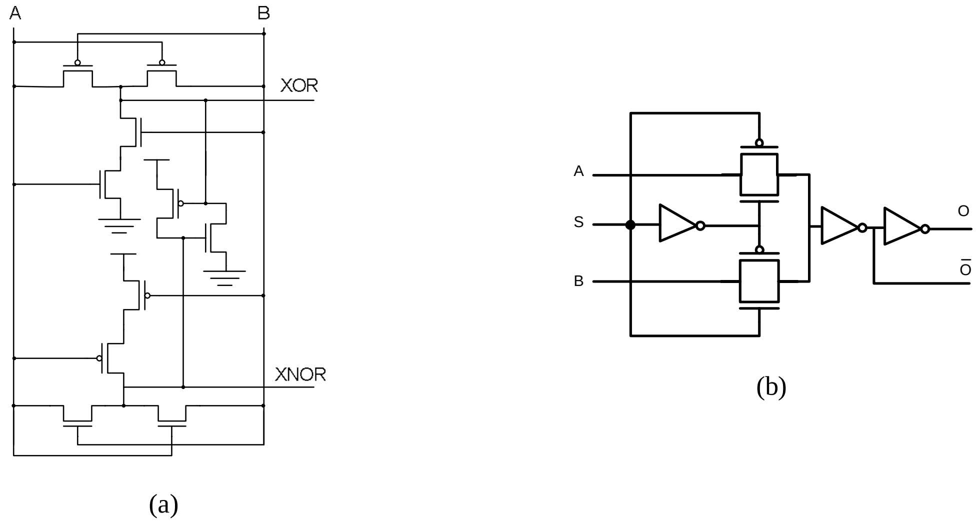
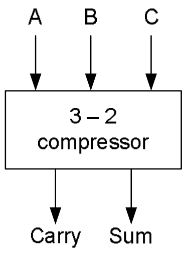


Fig. 3 – (a) XOR-XNOR module proposed in [6] (b) Multiplexer (MUX) implemented with transmission gates.

Based on previous works presented in [6] and [7], 3-2 and 4-2 Compressors are implemented in this work using ASTRAN.

2.1. 3-2 Compressor

In Figure 4 is presented the 3-2 adder compressor. As shown in the truth table, the operation is the same of the full adder. It takes 3 inputs A, B, C to generate 2 outputs, the sum and the carry bits.



(a)

A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(b)

Fig. 4 – (a) 3-2 adder compressor. (b) Truth table for the 3-2 adder compressor.

This compressor is governed by the following equation:

$$A + B + C = Sum + 2 * Carry \tag{1}$$

In Figure 5 are shown the two architectures implemented in this work. The traditional architecture is shown in Figure 5(a) and is implemented using the CMOS logic style of the XOR-XNOR and MUX modules. This architecture is governed by the following equations:

$$Sum = A \oplus B \oplus C \tag{2}$$

$$Carry = (A \oplus B) * C + \overline{(A \oplus B)} * A \tag{3}$$

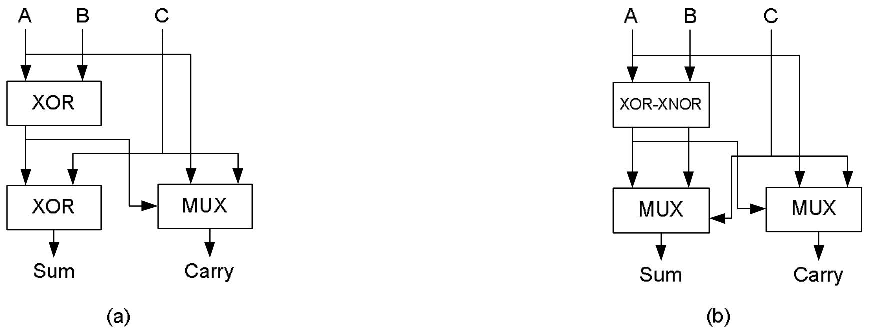


Fig. 5 – (a) Traditional architecture of 3-2 Compressors. (b) Enhanced architecture for a 3-2 Compressor.

In Figure 5(b) is shown the enhanced version of the 3-2 compressor architecture. This architecture uses the XOR-XNOR module presented in 3(a) and the transmission gate version of the MUX module presented in Figure 3(b). This architecture is governed by the following equations:

$$Sum = (A \oplus B) * \overline{C} + \overline{(A \oplus B)} * C \tag{4}$$

$$Carry = (A \oplus B) * C + \overline{(A \oplus B)} * A \tag{5}$$

2.2. 4-2 Compressor

The 4-2 Compressor has 5 inputs A, B, C, D and Cin to generate 3 outputs Sum, Carry and Cout as shown in Figure 6(a). The 4 inputs A, B, C and D and the output Sum have the same weight. The input Cin is the output from a previous lower significant compressor and the Cout output is for the compressor in the next significant stage. The conventional approach to implement 4-2 compressors is with 2 full adders connected serially as shown in Figure 6(b).

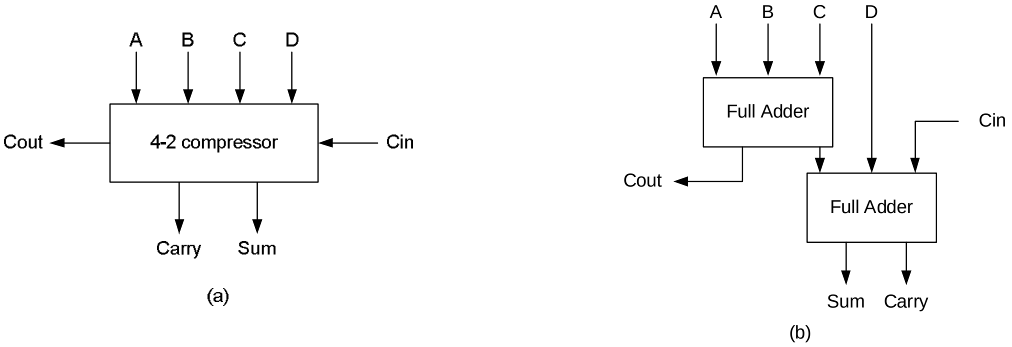


Fig. 6 – (a) 4-2 adder compressor. (b) 4-2 adder compressor implemented with full adders.

The 4-2 compressor is governed by the following equation:

$$A + B + C + D + Cin = SUM + 2 * (CARRY + Cout) \quad (6)$$

In Figure 7 are shown the two architectures implemented in this work. The traditional architecture is shown in Figure 7(a) and uses the CMOS logic style of the XOR-XNOR and MUX modules similarly to the 3-2 compressor. This architecture is governed by the following equations:

$$Sum = A \oplus B \oplus C \oplus D \oplus Cin \quad (7)$$

$$Cout = (A \oplus B) * C + \overline{(A \oplus B)} * A \quad (8)$$

$$Carry = (A \oplus B \oplus C \oplus D) * Cin + \overline{(A \oplus B \oplus C \oplus D)} * D \quad (9)$$

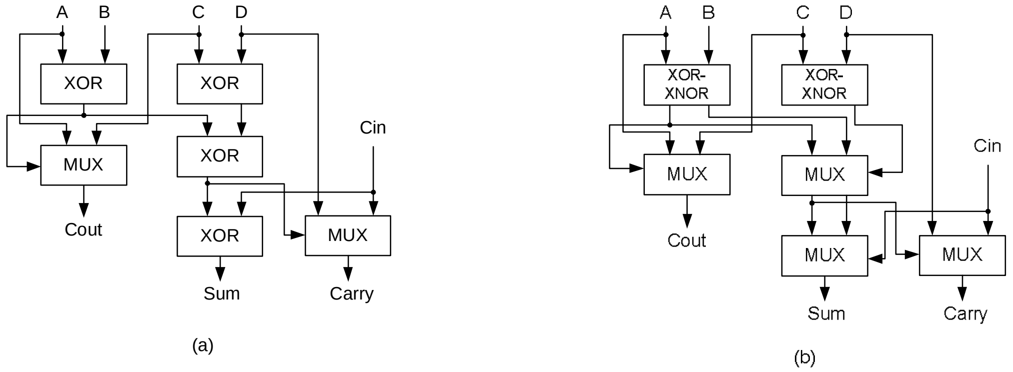


Fig. 7 – (a) Traditional architecture of 4-2 Compressors. (b) Enhanced architecture with the XOR-XNOR and MUX modules of a 4-2 Compressor.

In Figure 7(b) is shown the enhanced version of the 4-2 compressor architecture. This architecture uses the XOR-XNOR module presented in Figure 3(a) and the transmission gate version of the MUX module presented in Figure 3(b). This architecture is governed by the following equations:

$$Sum = \left((A \oplus B) * \overline{(C \oplus D)} + \overline{(A \oplus B)} * (C \oplus D)\right) * \overline{Cin} + \overline{\left((A \oplus B) * \overline{(C \oplus D)} + \overline{(A \oplus B)} * (C \oplus D)\right)} * Cin \quad (10)$$

$$Cout = (A \oplus B) * C + \overline{(A \oplus B)} * A \quad (11)$$

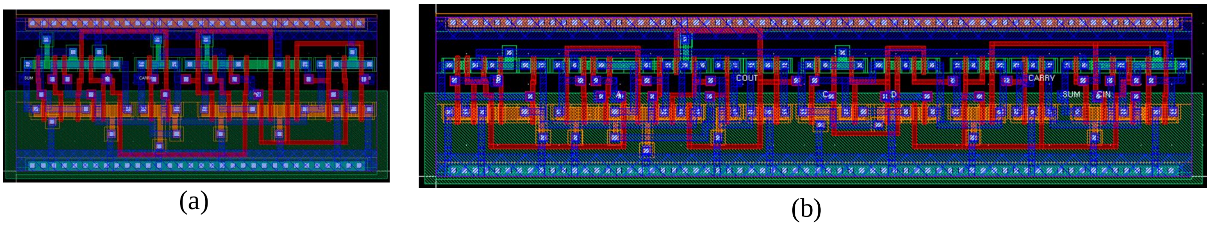
$$Carry = (A \oplus B \oplus C \oplus D) * Cin + \overline{(A \oplus B \oplus C \oplus D)} * D \quad (12)$$

3. Automatic Layout Generation using ASTRAN

ASTRAN [3] is a cell synthesis tool for automatic layout generation of CMOS cells from its transistor level netlist description, allowing different transistor sizes and no restrictions on the transistor network organization.

The tool receives as input a file in SPICE format with the netlist of the cells (with their respective and individually sized transistors), a configuration file (which defines the layout topology and control parameters to the generator), and a technology file (which contains a description of the design rules). For a given transistor network, the objective is to place and route the transistors using the proposed layout style in such a way that the cell width is minimized. At the end, the circuit is compacted to produce a layout in CIF, GDSII and LEF formats.

ASTRAN currently only supports designs in technology node equal or above 350nm, so in this work the AMS 350nm CMOS technology is used. In Figures 8(a) and 8(b) are shown the layouts generated. These layouts correspond to the enhanced architectures of the adder compressors.



(a) (b)

Fig. 8 – (a) 3-2 compressor layout. (b) 4-2 compressor layout.

4. Simulation Results

4.1. Simulation Environment

Simulations are performed by the HSPICE simulator. The simulation environments are shown in Figure 9 Each input is driven by buffers and each output is loaded by buffers to provide a realistic environment. For the 4-2 compressor environment, two compressors are placed in parallel because the critical path may cross adjacent compressors. The left most compressor is analyzed because it is most probable to have the longest delay. For both compressors, two configurations are proposed. The first is “Architecture 1” and is composed by the CMOS logic style of the XOR-XNOR and MUX modules and the traditional architecture for each compressor. The “Architecture 2” is composed by the modified XOR-XNOR module in Figure 3(a) and the transmission gate version of MUX with the enhanced architectures for each compressor.

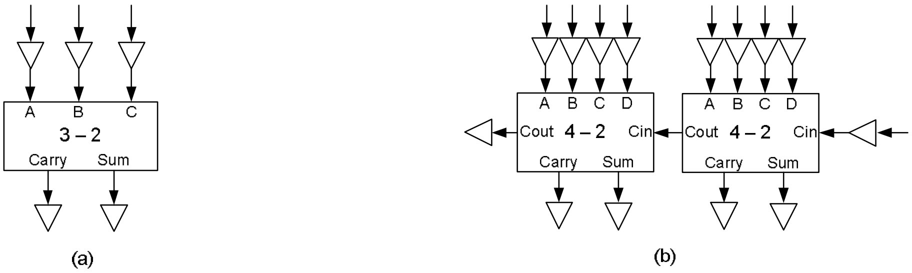


Fig. 9 – (a) 3-2 Compressor simulation environment. (b) 4-2 Compressor simulation environment.

4.2. Simulation Results for 3-2 Compressor

In Table 1 are shown the results for the 3-2 compressor. As expected, the changes in the logic style and architecture reduce the power while the delay is slightly affected. The power-delay product (PDP) at different supply voltages indicates that architecture 2 is more power efficient.

Tab.1 - Delay, Power and Power-delay product (PDP) comparison for the 3-2 compressor.

Delay (ns)	**1.2 V**	**1.8 V**	**2.5 V**	**3.3 V**
Architecture 1	6.80	2.62	1.61	1.21
Architecture 2	7.59	2.45	1.43	1.05

Power (uW)	**1.2 V**	**1.8 V**	**2.5 V**	**3.3 V**
Architecture 1	10	23.3	45.75	81.61
Architecture 2	8.15	19.45	38.42	71.35

PDP (fJ)	**1.2 V**	**1.8 V**	**2.5 V**	**3.3 V**
Architecture 1	68	61.05	73.66	98.75
Architecture 2	61.86	47.65	54.94	74.92

4.3. Simulation Results for 3-2 Compressor

In Table 2 are shown the results for the 4-2 compressor. Analyzing the power results, it seems that “architecture 1” is better in terms of power consumption, but the PDP shows that when the delay is considered, “architecture 2” offers better results.

Tab.2 - Delay, Power and Power-delay product (PDP) comparison for the 4-2 compressor.

Delay (ns)	**1.2 V**	**1.8 V**	**2.5 V**	**3.3 V**
Architecture 1	10.01	3.85	2.36	1.76
Architecture 2	9.97	3.26	1.88	1.37

Power (uW)	**1.2 V**	**1.8 V**	**2.5 V**	**3.3 V**
Architecture 1	15.78	38.68	76.23	136.1
Architecture 2	16.7	39.9	79.71	144.3

PDP (fJ)	**1.2 V**	**1.8 V**	**2.5 V**	**3.3 V**
Architecture 1	157.96	148.92	179.90	239.54
Architecture 2	166.50	130.07	149.85	197.69

Both circuits show that the better trade-off in terms of power and delay appears when the supply voltage is lower than the nominal (3.3V), around 1.8 V.

5. Conclusions and Future Work

The architectures of 3-2 and 4-2 compressors are analyzed using 2 logic styles for the XOR-XNOR and MUX modules. The layouts for these compressors are automatic generated with the ASTRAN tool. According to previous works [4], the layouts generated by ASTRAN contribute to reduce the power consumption of the circuits. Optimization techniques for low power should be applied in all abstraction levels, and this work the logic, electrical and physical levels are explored.

As future works we are already working on applying these optimizations on a 65 nm process, and analyze the circuit sensitivity to PVT (Process, Voltage and Temperature) variations.

6. Acknowledgment

The authors would like to thank CNPq and CAPES agencies for their financial support.

7. References

[1] J. M. Rabaey, *Low Power Design Essentials.* New York: Springer, 2009.

[2] R. Reis, “Design Automation of Transistor Networks, a New Challenge,” in *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on,* may 2011, pp. 2485 –2488.

[3] A. Ziesemer , C. Lazzari, R. Reis “Transistor Level Automatic Layout Generator for non-complementary CMOS Cells,” in *Very Large Scale Integration, 2007. VLSI - SoC 2007. IFIP International Conference on,* oct. 2007, pp. 116 –121.

[4] G. Posser, et.al., “A Study on Layout Quality of Automatic Generated Cells,” in *Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on,* dec. 2010, pp. 651 –654.

[5] M. Fonseca, et. al., “Design of Pipelined Butterflies from Radix-2 FFT with Decimation in Time Algorithm using Efficient Adder Compressors,” in *Circuits and Systems (LASCAS), 2011 IEEE Second Latin American Symposium on,* feb. 2011, pp. 1 –4.

[6] C.-H. Chang, et. al., “Ultra low-voltage Low-Power CMOS 4-2 and 5-2 Compressors for Fast Arithmetic Circuits,” *Circuits and Systems I: Regular Papers, IEEE Transactions on,* vol. 51, no. 10, pp. 1985 –1997, oct. 2004.

[7] S. Veeramachaneni, et. al., “Novel Architectures for High-Speed and Low-Power 3-2, 4-2 and 5-2 Compressors,” in *VLSI Design, 2007. Held jointly with 6th International Conference on Embedded Systems., 20th International Conference on,* jan. 2007, pp. 324–329.

[8] R. Zimmermann, W. Fichtner, “Low-Power Logic Styles: CMOS Versus Pass-Transistor Logic,” *Solid-State Circuits, IEEE Journal of,* vol. 32, no. 7, pp. 1079 –1090, jul 1997.