High Speed And Low Power Data Compressors

K. Sunitha Rani, P. Sri Kanth,
Electronics and Communication Department, JNT University, Hyderabad, Andhra Pradesh, India
sunitha.korra@gmail.com, psrikanth_bt@yahoo.co.in

ABSTRACT
The 3-2, 4-2 and 5-2 compressors are the basic components in many applications, in particular partial product summation in multipliers. In this paper novel architectures and designs of high speed, low power 3-2, 4-2 and 5-2 compressors capable of operating at ultra-low voltages are presented. The power consumption, delay and area of these new compressor architectures are compared with existing and recently proposed compressor architectures and are shown to perform better. The proposed architecture lays emphasis on the use of multiplexers in arithmetic circuits that result in high speed and efficient design. Also in all existing implementations of XOR gate and multiplexers, both output and its complement are available but current designs of compressors do not use these outputs efficiently. In the proposed architecture these outputs are efficiently utilized to improve the performance of compressors. The combination of low power, low transistor count and lesser delay makes the new compressors a viable option for efficient design.

Key words- compressor, MUX, ALU, CMOS

1. Introduction.
Multiplication is a basic arithmetic operation important in applications like digital signal processing which rely on efficient implementation of generic arithmetic logic units (ALU) and floating point units to execute dedicated operations like convolution and filtering. In the implementation of multipliers, the main phases are generation of partial products, reduction of partial products using CSA (carry-save architecture) [7-10] and a carry propagation adder for the computation of the final result. It is obvious that the second phase, that is, the reduction of the partial products contributes most to the overall delay, area and power. In most of these implementations, compressor lies directly within the critical path dictating the overall circuit, due to which the demand for high-speed and low-power compressors is continuously increasing [7-9]. This paper presents new compressor architectures that lay emphasis on the use of multiplexers in place of XOR gates to efficiently use the outputs from the previous stages and improve the overall performance. It is because the use of multiplexers improves the speed when placed in the critical path [2]. The rest of the paper is organized as follows: In Section 2 the efficiency of MUX and XOR-XNOR are compared and the possibility of replacing MUX with XOR-XNOR is discussed. In section 3, 4, 5 & 6 the proposed architectures of 3-2, 4-2 and 5-2 compressors are presented and compared with the existing architectures.

Implementations have been carried out in 0.18µm CMOS technology.

2. MUX Vs XOR-XNOR.
Existing CMOS designs of 2x1 multiplexer and 2-input XOR gate are shown in Fig.1 [2].

Fig.1. CMOS Implementations of (a) MUX (b) XORXNOR

In Fig.1(a), it can be seen that if both the select bit and its complement arrive before the inputs arrive then the output is generated with very less delay because switching of the transistors is already completed. Also if both the select bit and its complement are generated in the previous stage then the additional stage of the inverter is eliminated which reduces the overall delay in the critical path [2]. By using the output and its complement in every stage the total number of garbage outputs is reduced. By decreasing the number of transistors the overall power consumption and the area occupied is reduced considerably [1]. An alternative design of the multiplexer is shown in Fig.2.
This design of the multiplexer is faster than the CMOS design when buffers are not used at the output [10]. But these can only be used in the intermediate stages because of their limited driving capability. This design also consumes lesser power than the CMOS design [2]. In the proposed architectures the blocks where this design can be used are shown as MUX*.

3. 3-2 Compressor.
A 3-2 compressor takes 3 inputs X1, X2, X3 and generates 2 outputs, the sum bit S, and the carry bit C as shown in Fig. 3a.

The compressor is governed by the basic equation

\[ x_1 + x_2 + x_3 = \text{Sum} + 2\times\text{Carry} \]

![Fig.3. (a) A 3-2 Compressor (b) Conventional Implementation of the 3-2 compressor](image)

Fig. 3. (a) A 3-2 Compressor (b) Conventional Implementation of the 3-2 compressor

The 3-2 compressor can also be employed as a full adder cell when the third input is considered as the Carry input from the previous compressor block or \( X3 = \text{Cin} \).

Existing architectures shown in Fig. 3(b) employ two XOR gates in the critical path [3-6]. The equations governing the existing 3-2 compressor outputs are shown below

\[ \text{Sum} = x_1 \oplus x_2 \oplus x_3 \]
\[ \text{Carry} = (x_1 \oplus x_2) \cdot x_3 + (x_1 \oplus x_2) \cdot x_1 \]

In the proposed architecture shown in Fig. 4, the fact that both the XOR and XNOR values are computed is efficiently used to reduce the delay by replacing the second XOR with a MUX. This is due to the availability of the select bit at the MUX block before the inputs arrive. Thus the time taken for the switching of the transistors in the critical path is reduced.

![Fig.4. Proposed architecture of the 3-2 Compressor](image)

Fig. 4. Proposed architecture of the 3-2 Compressor

The equations governing the 3-2 compressor outputs are shown below

\[ \text{Sum} = (x_1 \oplus x_2) \cdot x_3 + (x_1 \oplus x_2) \cdot x_1 \]
\[ \text{Carry} = (x_1 \oplus x_2) \cdot x_3 + (x_1 \oplus x_2) \cdot x_1 \]

It can be seen that in this implementation the overall delay is \( \cdot\cdot\cdot\text{MUX} \) (where \( \cdot \) refers to delay).

4. 4-2 Compressor.
The 4-2 compressor has 4 inputs X1, X2, X3 and X4 and 2 outputs Sum and Carry along with a Carry-in (Cin) and a Carry-out (Cout) as shown in Fig. 5. The input Cin is the output from the previous lower significant compressor. The Cout is the output to the compressor in the next significant stage.

![Fig.5. A 4-2 Compressor Block](image)

Fig. 5. A 4-2 Compressor Block

Similar to the 3-2 compressor the 4-2 compressor is governed by the basic equation
The standard implementation [3-6] of the 4-2 compressor is done using 2 Full Adder cells as shown in Fig 6(a).

When the individual full Adders are broken into their constituent XOR blocks, it can be observed that the overall delay is equal to 4*-XOR. The block diagram in Fig. 6(b) shows the existing architecture for the implementation of the 4-2 compressor with a delay of 3*-XOR [3-6]. The equations governing the outputs in the existing architecture are shown below

$$\text{Sum} = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus C_{\text{in}}$$

$$C_{\text{out}} = \left( x_1 \oplus x_2 \right) \cdot x_3 + \left( x_1 \oplus x_3 \right) \cdot x_4$$

$$\text{Carry} = \left( x_1 \oplus x_2 \oplus x_3 \oplus x_4 \right) \cdot C_{\text{in}} + \left( x_1 \oplus x_2 \oplus x_3 \oplus x_4 \right) \cdot \left( C_{\text{out}} \oplus C_{\text{in}} \right)$$

However, like in the case of 3-2 compressor, the fact that both the output and its complement are available at every stage, is neglected [2]. Thus replacing some XOR blocks with multiplexers results in a significant improvement in delay.

The equations governing the outputs in the proposed architecture are shown below

$$\text{Sum} = \left( x_1 \oplus x_2 \right) \cdot x_3 + \left( x_1 \oplus x_2 \right) \cdot \left( x_3 \oplus x_4 \right) \cdot C_{\text{in}}$$

$$C_{\text{out}} = \left( x_1 \oplus x_2 \right) \cdot x_3 + \left( x_1 \oplus x_2 \right) \cdot \left( x_3 \oplus x_4 \right)$$

$$\text{Carry} = \left( x_1 \oplus x_2 \oplus x_3 \oplus x_4 \right) \cdot C_{\text{in}} + \left( x_1 \oplus x_2 \oplus x_3 \oplus x_4 \right) \cdot \left( x_3 \oplus x_4 \right)$$

The critical path delay of the proposed implementation is 4*-XOR + 2*-MUX.

**5. 5-2 Compressor.**

The 5-2 Compressor block has 5 inputs X1,X2,X3,X4,X5 and 2 outputs, Sum and Carry, along with 2 input carry bits (Cin1, Cin2) and 2 output carry bits (Cout1,Cout2) as shown in Fig.8a. The input carry bits are the outputs from the previous lesser significant compressor block and the output carry are passed on to the next higher significant compressor block.
The basic equation that governs the function of the 5-2 compressor block is given below:

$$x_1 + x_2 + x_3 + x_4 + x_5 + C_{in1} + C_{in2} = \text{Sum} + 2 \ast (\text{Carry} + C_{out1} + C_{out2})$$

(13)

The conventional implementation [3-6] of the compressor block is shown in Fig.8(b) where 3 cascaded full adder cells are used. When these full adders are replaced with their constituent blocks of XOR gates then it can be observed that the overall delay is equal to $6 \ast$-XOR for the sum or carry output. Many architectures have been proposed where the delay has been reduced to $5 \ast$-XOR (Fig.9a) and then further reduced to $4 \ast$-XOR. (Fig.9 b&c) [3-6].

In the proposed architecture changes have been made, to efficiently use the outputs generated at every stage, by replacing a few XOR blocks with MUX blocks. Also the select bits to the multiplexers in the critical path are made available much ahead than the inputs so that the critical path delay is minimized. For example the Cout2 output from the previous lesser significant compressor block is utilized as the select bit after a stage it is produced so that the MUX block is already switched and the output is produced as soon as the inputs arrive. Also if the output of the multiplexer is used as select bit for another multiplexer, then it can be used efficiently in similar manner because the negation of select bit is also required, as shown in Figure 1(a), in the design and an extra stage to compute the negation can be saved. Similarly replacing the XOR block in the second stage with a MUX block reduces the delay because the select bit X3 is already available and the time taken for the transistor switching to take place is done in parallel with the computation of the inputs of the block.
As mentioned before, in all the general implementations of the XOR or MUX block, in particular CMOS implementation, the output and its complement are generated. But in the existing architectures this advantage is not being utilized at all [3-6]. In the proposed architecture these outputs are utilized efficiently by using multiplexers at select stages in the circuit. Also additional inverter stages are eliminated. This in turn contributes to the reduction of delay, power consumption and transistor count (area). The equations governing the outputs are shown below:

\[ S_{\text{IN}} = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus C_{\text{in1}} \oplus C_{\text{in2}} \]
\[ C_{\text{out1}} = (x_1 + x_2) \cdot x_3 \oplus x_1 \cdot x_2 \]
\[ C_{\text{out2}} = (x_3 \oplus x_2) \cdot C_{\text{in}} + (x_1 \oplus x_5) \cdot x_4 \]
\[ \text{Carry} = ((x_3 \oplus x_2) \cdot x_3) \cdot (x_1 \oplus x_5 \oplus C_{\text{in}}) \cdot x_4 \oplus x_5 \cdot x_3 \cdot x_2 \]

The critical path delay of the proposed implementation is \(-XOR + 3^\text{rd}-MUX\). In the Carry generation module mentioned in Fig.10, we use the mathematical equation (15) to design a CMOS implementation of Cout1 as shown in Fig.11.

6. Simulation and results
   a. Simulation environment.
      All the simulations have been done using Cadence Tools. The calculation of power (including glitch power) and delay are carried out using the Virtual Analog Simulation tool already integrated into Cadence Tools. All the schematics and layouts (Fig 13, 15 & 17) are done using the CMOS 0.18-µm technology. Hence the circuits are optimized for this process technology.
      The simulations are performed under various voltages ranging from 0.9V to 3.3V. All the inputs are fcd at a frequency of 100MHz.
   B. Simulation results.

![Figure 12](image1.png)  
(a) Power consumption (nW), (b) Delay (ns), (c) Power Delay product for 5-2 compressors

![Figure 14](image2.png)  
(a) Power consumption (nW), (b) Delay (ns), (c) Power Delay product for 4-2 compressors
Figure 16(a) Power consumption(nW) (b) Delay(ns) (c) Power Delay product for 3-2 compressors

The figures 12, 14 & 16 show that the proposed architecture for the 5-2 compressor consumes 13.2% lesser power and is 26% faster than the existing architectures when operating at 1.8V. Because of the decrease in the number of transistors the overall area decreases by about 11.15% in the proposed 5-2 compressor. The 4-2 compressor architecture is 33.3% faster and consumes 15% lesser power than the existing architectures. Also the proposed 3-2 compressor is 7% faster and consumes 10.2% lesser power than the existing architectures. The improvement in the power-delay product is 36.4%, 27.8% and 24% in the proposed 5-2 compressor, 4-2 compressor and 3-2 compressor respectively.

7. Conclusions.

The architectures of the 3-2, 4-2 and 5-2 compressor are analyzed using CMOS and CMOS+ implementations of XOR and the MUX blocks. New 3-2, 4-2 and 5-2 compressor architectures have been proposed and compared with the existing architectures. Simulations have been performed over a range of voltages, from 0.9V to 3.3V. The proposed architectures perform better than the existing ones in every aspect i.e., area, power, delay and power-delay product over the complete voltage range simulated.

References.