High performance 5 : 2 compressor architectures

R. Menon and D. Radhakrishnan

Abstract: Fast arithmetic circuits are key elements of high performance computers and data processing systems. In the majority of these applications, multipliers have been a critical and obligatory component in dictating the overall circuit performance when constrained by power consumption and computation speed. Compressors are a critical component of the multiplier circuit, which greatly influence the overall multiplier speed. The authors propose two novel high performance 5 : 2 compressor architectures. The main objective of their designs is to limit the carry propagation to a single stage, thereby reducing the overall propagation delay. The designs are compared with the best one in the literature in terms of delay and are found to have lower values. The analytical techniques use the node capacitances in the signal delay paths to identify the worst delay path. The architectures are implemented with various XOR–XNOR circuits to identify the best one in terms of power and delay. The simulation results of the proposed architectures show lower power and 25% improvement in speed compared to the best architecture reported in the literature for supply voltages ranging from 1.5 V to 3.3 V.

1 Introduction

The concept of digital data manipulation has made a dramatic impact on our society. One has long grown accustomed to the idea of digital computers. Evolving steadily from mainframe and minicomputers, personal and laptop computers have proliferated into our daily lives. In the majority of applications utilising efficient implementation of generic arithmetic logic units and floating point units to execute dedicated algorithms, multipliers have been a critical and obligatory component in dictating the overall circuit performance when constrained by power consumption and computation speed.

Multiplication is basically a two-step process, essentially consisting of the formation of partial products followed by its reduction to give the final binary result. The formation of partial products is a simple process, whereas the summation of partial products contributes to most of the delay and area of the multiplier. One method used to increase the performance is to use encoding techniques, like modified Booth encoding, to reduce the number of partial products generated [1]. However, to achieve even higher performance, advanced hardware multiplier architectures search for faster and more efficient methods for summing the partial products. A powerful technique for improving the performance of partial product reduction is to use carry save adders (CSAs) in Wallace and Dadda trees.

A significant departure from carry save adder arrangement was achieved with the introduction of compressors such as 4 : 2 that involve limited carry propagation [2]. Researchers have tried to explore higher order compressor families like 6 : 2 and 9 : 2 by interconnecting 3 : 2 and 4 : 2 compressors [3]. The focus of the research has been to optimise the circuit structure for high-speed applications. In this paper we focus our attention on the 5 : 2 compressor. We propose two novel 5 : 2 compressor designs for high-speed applications. They are found to perform reliably well under a realistic simulation environment and varying operating conditions.

2 Compressor architectures

Compressors are building blocks used for accumulating partial products during the multiplication process. The basic idea in an $n : 2$ compressor is that n operands can be reduced to two, by doing the addition while keeping the carries and sums separate. This means that all of the columns can be added in parallel without relying on the result of the previous column, creating a two-output adder with a time delay that is independent of the size of its inputs. The full adder is the most primitive compressor and is often referred to as the 3 : 2 compressor since it compresses three operands into two. The sum and carry outputs are given by the following set of equations:

$$\begin{aligned} \text{Sum} &= x_1 \oplus x_2 \oplus c_i \\ \text{Carry} &= x_1 x_2 + x_2 c_i + x_1 c_i \end{aligned} \qquad (1)$$

where x_1 and x_2 are the input operands and c_i is the carry from previous stage. This 3 : 2 compressor has a delay of two XOR gates, and is normally used in carry-save form to sum up the partial products in a multiplier tree. Though the addition used in this manner is much faster than that of a ripple carry adder, the interconnections are very irregular thereby making the structure more complex.

The next higher-level compressor is the 4 : 2 compressor introduced by Weinberger [4], which consists of five inputs and three outputs, and compresses four partial products into two, thus offering a higher compression ratio and a more regular interconnection structure than its 3 : 2 counterpart. The input-output relationship of the compressor can be defined as follows:

$$x_1 + x_2 + x_3 + x_4 + c_i = \text{Sum} + 2 \times C_o + 2 \times \text{Carry} \qquad (2)$$

IEE Proceedings online no. 20050152

doi:10.1049/ip-cds:20050152

Paper first received 28th May and in revised form 1st December 2005

The authors are with the Department of Electrical and Computer Engineering, State University of New York, 75 S. Manheim Blvd., New Paltz, New York 12561, USA

E-mail: damu@engr.newpaltz.edu

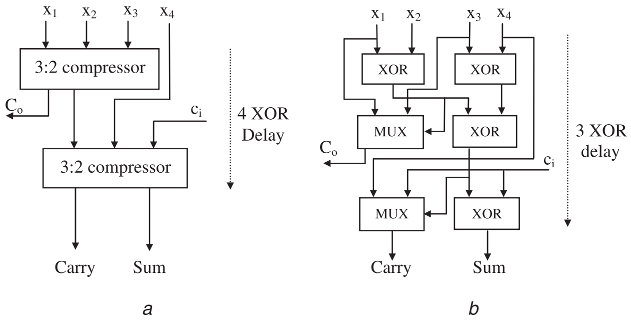


Fig. 1 *4 : 2 compressor*
a Implementation using 3 : 2 compressors
b Logic implementation

The 4 : 2 compressor was initially designed by an intricate connection of two 3 : 2 compressors as shown in Fig. 1*a*. The structure has a delay of four XORs. The advantage of the structure lies in its carry free nature, whereby the carry from the previous stage is not propagated to the next stage. A novel design of a 4 : 2 compressor with XORs and multiplexers (MUX) as building blocks is presented in [5]. This design is based on a modified set of equations for the sum and carry outputs as:

$$\text{Sum} = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus c_i \tag{3}$$

$$\text{Carry} = (x_1 \oplus x_2 \oplus x_3 \oplus x_4)c_i + (\overline{x_1 \oplus x_2 \oplus x_3 \oplus x_4})x_4, \text{ and} \tag{4}$$

$$C_o = (x_1 \oplus x_2)x_3 + (\overline{x_1 \oplus x_2})x_1 \tag{5}$$

An equivalent gate logic realisation is shown in Fig. 1*b*.

A third widely used compressor of significant importance is the 5 : 2 compressor. Its block diagram is shown in Fig. 2*a*. It has seven inputs of which five are direct inputs and two are carry-in bits from a previous stage. Similarly, there are four outputs of which two are carry-out bits to the next stage and the other two are sum and carry bits. All the 5 : 2 compressors of different designs abide by the generic equation:

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 + x_5 + c_{i1} + c_{i2} \\ = \text{Sum} + 2(\text{Carry} + C_{o1} + C_{o2}) \end{aligned} \tag{6}$$

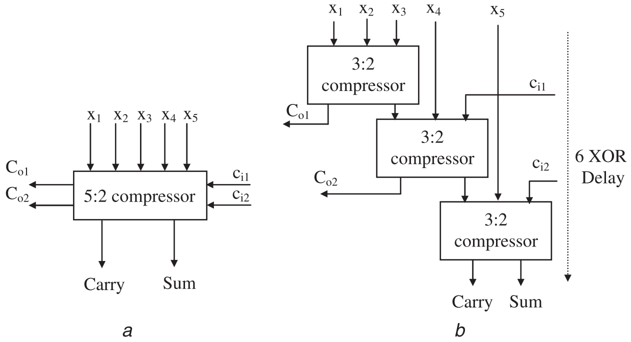


Fig. 2 *5 : 2 compressor*
a Block diagram
b Implementation using 3 : 2 compressors

In its simplest form, a 5 : 2 compressor can be designed by cascading three 3 : 2 compressors as shown in Fig. 2*b*. This structure has a delay of 6 XORs and is slower than the 6 : 2 compressor presented in [3], which has a delay of only five XORs. A faster implementation of the 5 : 2 compressor with 5 XOR delays is presented in [6]. This is shown in Fig. 3*a*, and will be called Arch1 for the rest of this paper. Though the delay of Arch1 is claimed as five XORs in [6], a closer look at the design indicates that the delay is actually four XORs and a MUX.

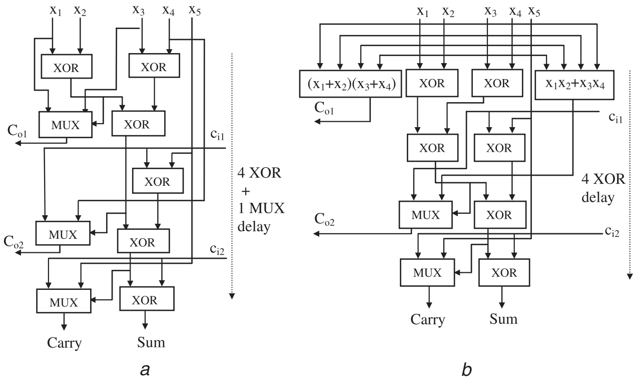


Fig. 3 *Logic implementations of 5 : 2 compressor*
a Arch1
b Arch2

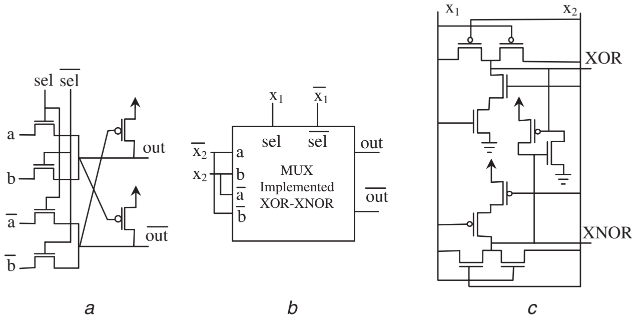


Fig. 4 *Multiplexer/XOR–XNOR modules*
a CPL-based MUX
b CPL-based XOR–XNOR module
c XOR–XNOR module

A second implementation of the 5 : 2 compressor with a delay of four XORs is presented in [7]. This is shown in Fig. 3*b*, and uses a different logic structure for the output bits C_{o1} and C_{o2} than that shown in Fig. 2*b*. This architecture will be called Arch2 in the rest of this paper. But no implementation details are provided in [7] for their design, which makes it difficult to verify the performance of their 5 : 2 compressor. A simple inspection of the above two architectures reveals that Arch2 is faster than Arch1 by one MUX delay.

Comparisons have been made in [6] using simulation between Arch1 and Arch2 by implementing both using the CPL-based MUX cell shown in Fig. 4. This MUX cell can be used to implement either a MUX (Fig. 4*a*) or a XOR–XNOR module (Fig. 4*b*). The simulation results indicated that Arch1 outperforms Arch2 in terms of both speed and power, even though by inspection Arch2 is seen to be faster than Arch1. This suggests that the vertical delay paths as seen in Fig. 3 do not really represent the critical delay paths.

A new set of 5 : 2 compressor implementations with a delay of 4 XORs is presented in [8]. A modified set of output expressions satisfying (6) are used in their design. They are

$$\text{Sum} = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus c_{i1} \oplus c_{i2} \tag{7}$$

$$C_{o1} = x_1x_2 + (x_1 + x_2)x_3 \tag{8}$$

$$C_{o2} = (x_4 \oplus x_5)c_{i1} + (\overline{x_4 \oplus x_5})x_4 \tag{9}$$

$$\text{Carry} = ((x_1 \oplus x_2 \oplus x_3) \oplus (x_4 \oplus x_5 \oplus c_{i1}))c_{i2} + \left(\overline{(x_1 \oplus x_2 \oplus x_3) \oplus (x_4 \oplus x_5 \oplus c_{i1})}\right)(x_1 \oplus x_2 \oplus x_3) \quad (10)$$

A new low power XOR–XNOR cell, which can operate reliably at low supply voltages is also introduced in [8], which is shown in Fig. 4*c*. This cell, together with a few other low power cells for XOR and MUX are used to implement a number of 5 : 2 compressors. These implementations are compared for their delay and power consumption. The simulation results in [8] indicate that two of their proposed 5 : 2 compressor designs consume the lowest power compared to all others. One of their low power designs, denoted as 4del_ebb, is used for comparisons with our designs and is referred hereafter as Arch3 in the rest of this paper.

In an effort to further minimise the critical path delay, we decided to perform a thorough analysis of the signal delay paths in Arch1. We identified three signal propagation paths for Arch1 through which the input signal could propagate to the output. The propagation delay is a function of the node capacitances at the output of the logic blocks along the path of the signal. Therefore, we studied three signal propagation paths of Arch1 by considering the parasitic node capacitances along the path to identify the critical signal delay path. Since some of the signals cross adjacent compressors, more than one compressor is used in the analysis. Hence, unique notations are used to identify the inputs and outputs of various compressors. The inputs are denoted as x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}, c_{i1j}, c_{i2j}, where $j = 1$, 2 or 3 depending on the compressor it represents. Similarly, the outputs are denoted as C_{o1j}, C_{o2j}, Sum*j* and Carry*j*, where again $j = 1$, 2 or 3. A lower value of *j* represents the compressor of lower significance. The setup used for analysis is shown in Fig. 5. The parasitic node capacitances are estimated in the following manner. All capacitances are represented in terms of the gate capacitance C_g of the smallest NMOS transistor used in the design. The following assumptions are made in their calculation [9]:

- Source and drain capacitances are assumed to be half of gate capacitance C_g.
- PMOS transistor capacitances are assumed to be 1.5 times C_g.
- In the case of an inverter, the gate capacitances of NMOS and PMOS transistors are assumed to be two and three times C_g respectively.

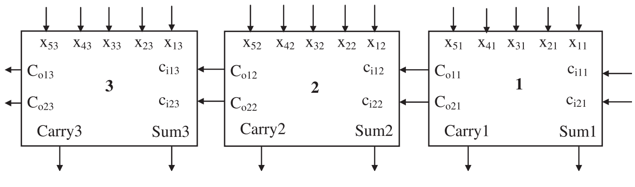


Fig. 5 *Setup for analysis of 5 : 2 compressors*

Three different signal propagation delay paths are used in the following analysis.

2.1 Path 1_{Arch1}: from input x_{11}, through C_{o21} to Sum2

The signal propagation path along with the corresponding node capacitances is shown in Fig. 6*a*. The logic blocks along this path are labelled as XOR1, XOR2, etc. to indicate that XOR1 belongs to compressor 1 and XOR2 belongs to compressor 2 and so on. The analysis uses an implementation using the CPL-based MUX cell shown in Fig. 4. The inverters shown are used to feed complementary signals to the MUX/XOR gates whenever the signal enters a compressor from outside. The capacitance values shown at the outputs of the XOR blocks vary depending on where the output of the XOR block is connected.

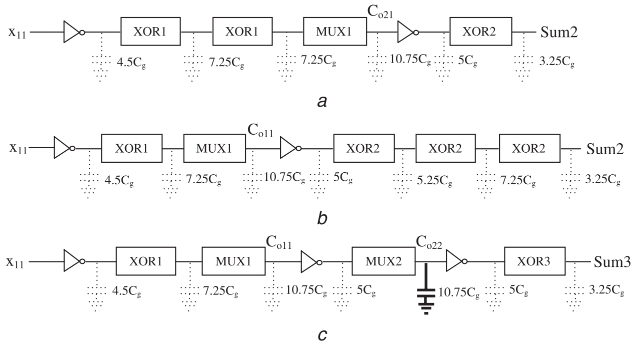


Fig. 6 *Signal propagation paths in 5 : 2 compressors*
a Path 1_{Arch1}
b Path 2_{Arch1}
c Path 3_{Arch1}

2.2 Path 2_{Arch1}: from input x_{11}, through C_{o11} to Sum2

The signal propagates through four XORs and one MUX along its way as shown in Fig. 6*b*. By comparing the delay of Path 2 with that of Path 1 in Fig. 6*a*, we see that path 2 has an additional XOR block, which adds to the delay, thereby resulting in a longer delay than path 1.

2.3 Path 3_{Arch1}: from input x_{11}, through C_{o11} and then through C_{o22} to Sum3

The signal propagates through two XORs and two MUXs as shown in Fig. 6*c*. This indicates that path 2 in Fig. 6*b* with four XORs and a MUX might be a longer delay path. However, before concluding, let us compare Figs. 6*b* and 6*c* based on the node capacitances. The total numbers of capacitances in both figures are equal. But in Path 3, the large capacitance at the output of MUX2 introduces extra delay and overtakes Path 2. So Path 3_{Arch1} has a longer delay than Path 2_{Arch1}. Hence, Path 3 represents the longest delay path for Arch1, and not Path 2 as observed through inspection.

From the above analysis of the delay paths of Arch1 we observe that the delay increases as the signal propagates from one compressor to the other. In the longest delay path (Path 3), the carry propagates through two compressors. So, if we could reduce this carry propagation delay we could improve the speed of the compressor. This is the major emphasis behind our research and is presented in the next Section.

3 High performance 5 : 2 compressors

The method proposed here is to make the carry out signal C_{o2}, independent of the carry in c_i. This will limit the carry propagation delay to one compressor. Two designs are presented. They are denoted as Design 1 and Design 2.

3.1 Design 1

The main objective in our designs is to limit the carry propagation. In the conventional implementation of the 5 : 2 compressor using 3 : 2 compressors shown in Fig. 2*b*, the first compressor generates C_{o1}, the second generates C_{o2} and the third generates the Sum and Carry. The second compressor receives inputs x_4, c_{i1} and the output from the first compressor. Hence, C_{o2} is a function of c_{i1}, thereby propagating c_{i1} across the compressor.

In our design, we feed x_5 to the second 3 : 2 compressor instead of c_{i1}, thereby making C_{o2} independent of c_{i1}. This is illustrated by the modified set of logic equations given in (11) to (14). The main difference lies in the generation of C_{o2} (9). All the other equations are the same as in Arch1 [6].

$$\mathrm{Sum} = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus c_{i1} \oplus c_{i2} \tag{11}$$

$$C_{o1} = (x_1 \oplus x_2)x_3 + (\overline{x_1 \oplus x_2})x_1 \tag{12}$$

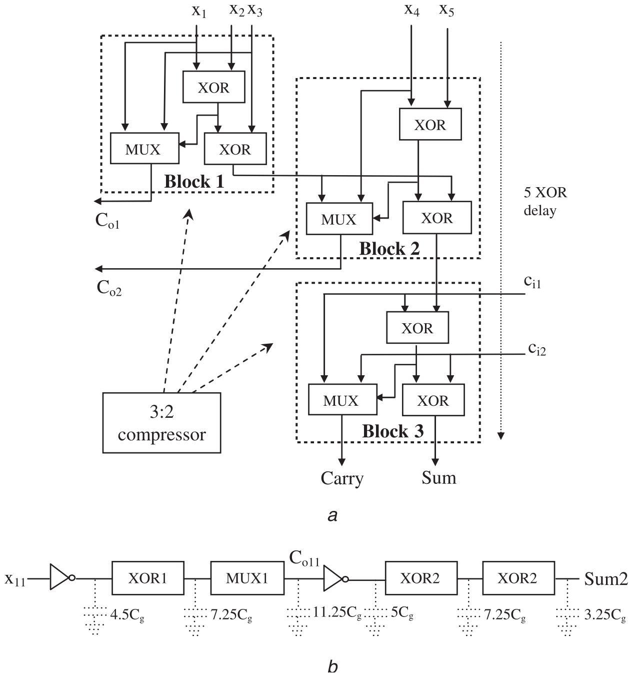
$$C_{o2} = (x_4 \oplus x_5)(x_1 \oplus x_2 \oplus x_3) + (\overline{x_4 \oplus x_5})x_4 \tag{13}$$

$$\begin{aligned}\mathrm{Carry} = {} & (x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus c_{i1})c_{i2} \\ & + (\overline{x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus c_{i1}})c_{i1}\end{aligned} \tag{14}$$

A block schematic of the compressor is shown in Fig. 7*a*. The design is divided into three 3 : 2 compressor modules similar to Fig. 2*b*. The first 3 : 2 compressor (Block 1) is similar to the one in Arch1 and generates C_{o1} using an XOR and a MUX. C_{o2} is generated by the second compressor (Block 2), and is made independent of c_{i1}. This is reflected in (13). The third 3 : 2 compressor (Block 3) generates the Sum and Carry. The vertical delay as indicated is five XOR delays.

Since we succeeded in eliminating the dependence of C_{o2} on c_{i1}, the next step is to identify the critical delay path and compare it with Path 3 of Arch1. For this purpose, we identified the longest delay path from among the three signal paths in Design 1, and used similar techniques as before to identify the critical delay path.

Critical_Delay_Path$_{Design1}$: from input x_{11}, through C_{o11} to Sum2: The signal propagates through three XORs and a MUX as shown in Fig. 7*b*. The largest load capacitance is seen by the MUX, which generates C_{o11}. By comparing Fig. 6*c* (Arch1) and Fig. 7*b* (Design 1) we see that in Arch1 there is an additional load (inverter) owing to the signal propagating into the third compressor. This additional carry propagation increases the propagation delay of Arch1 over Design 1.



a

b

Fig. 7 *5 : 2 compressor, Design 1*
a Logic decomposition of Design 1, 5 XOR delay
b Critical path

3.2 Design 2

We also propose a second design for the 5 : 2 compressor. The logic equations governing the design are as follows:

$$\mathrm{Sum} = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus c_{i1} \oplus c_{i2} \tag{15}$$

$$C_{o1} = x_1x_2 + x_2x_3 + x_1x_3 \tag{16}$$

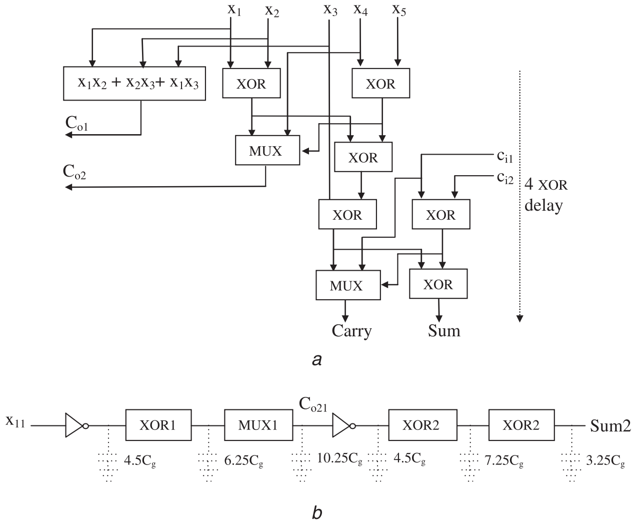
$$C_{o2} = (x_4 \oplus x_5)(x_1 \oplus x_2) + (\overline{x_4 \oplus x_5})x_4 \tag{17}$$

$$\begin{aligned}\mathrm{Carry} = {} & (c_{i1} \oplus c_{i2})(x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5) \\ & + (\overline{c_{i1} \oplus c_{i2}})c_{i1}\end{aligned} \tag{18}$$

A logic decomposition for its implementation is shown in Fig. 8*a*. The vertical delay in this design is reduced to four XORs. C_{o1} is generated using (16), and is implemented similar to the carry generation module of a conventional CMOS full adder. C_{o2} is generated using an XOR and a MUX; however, the inputs are so chosen to make C_{o2} independent of c_i. This can be observed from (17).

The next step is to identify the critical delay path and compare it with Design 1. Similar to our earlier approaches, we identified the longest delay path from among the three paths through which the signal propagates to the output.

Critcal_Delay_Path$_{Design2}$: From input x_{11}, through C_{o21} to Sum2: The signal propagates through three XORs and a MUX to generate the Sum output as shown in Fig. 8*b*. The largest load capacitance is seen by MUX1, which generates C_{o2}. By comparing the load capacitances in Design 1 (Fig. 7*b*) and Design 2 (Fig. 8*b*) it is observed that they are larger in Design 1, and hence Design 1 is a slower circuit compared to Design 2. However, in terms of power dissipation, Design 1 might be a better choice due to its use of MUX for implementing C_{o1}, even though both circuits may have comparable values for power.



a

b

Fig. 8 *5 : 2 compressor, Design 2*
a Logic decomposition of Design 2, 4 XOR delay
b Critical path

4 Simulation results

In Section 3, we proposed two new 5 : 2 compressor circuits. Now we will use simulation tools to validate our claims on speed and power, and to compare our designs against the known architectures available in the literature in terms of

Table 1: Architectures used for simulation

Simulation	Architecture	XOR	XOR (Sum Output)	MUX	# 1 Trans	Trans width (μm)
Sim1	Arch1	Figs. 4*a*, *b*	Figs. 4*a*, *b*	Figs. 4*a*, *b*	68	52
Sim2	Arch2	Figs. 4*a*, *b*	Figs. 4*a*, *b*	Figs. 4*a*, *b*	82	66
Sim3	Arch3	Fig. 4*c*	Fig. 9*b*	Fig. 9*c*	72	67
Sim4	Design 1	Fig. 9*a*	Fig. 9*b*	Fig. 9*c*	60	64
Sim5	Design 1	Figs. 4*a*, *b*	Figs. 4*a*, *b*	Figs. 4*a*, *b*	68	52
Sim6	Design 2	Fig. 9*a*	Fig. 9*b*	Fig. 9*c*	64	67
Sim7	Design 2	Figs. 4*a*, *b*	Figs. 4*a*, *b*	Figs. 4*a*, *b*	74	58

power and delay. The simulation tool used is TSpice Pro V-81 from Tanner Research Corporation. All circuits are simulated using 0.25 μm CMOS process parameters [10]. The circuit capacitance values are estimated using MATLAB models that capture the basic equations for transistors [10]. The measurements are made to obtain the power and delay values of the circuits under different supply voltages ranging from 1.5 V to 3.3 V. These circuits are tested at a frequency of 500 MHz.

The simulation environment is shown in Fig. 5, except that all inputs and outputs are connected through inverters [8]. This provides a realistic simulation environment reflecting the compressor operation in real applications. A sequence of 1024 data words (5-bits each) is randomly generated using MATLAB to feed as inputs to the circuits. The average power consumption of the middle compressor is measured, by excluding the power consumption in the inverters. Since most of the circuits are implemented in pass logic, we measured the power drawn from each of the signal sources feeding the inputs. For the propagation delay measurements we used the traditional definition by measuring the time taken from the input reaching 50% to the output reaching 50%.

The XOR cell is the main component of all the 5 : 2 compressors. Since complementary signals are required at the select inputs of the MUXs, XOR–XNOR cells are used for all the blocks excepting the final XOR, which generates the Sum output. Two implementations are used for each of the proposed compressors. The major difference between them is in the implementation of the XOR and MUX gates. In one of them the CPL-based MUX cell shown in Fig. 4 is used for implementing both XOR and MUX gates. In the other, the 6-transistor XOR–XNOR cell in Fig. 9*a* [11] is used for implementing all XOR gates except the final XOR, which is used to generate the Sum output. This final XOR gate is implemented using the pass logic XOR cell in Fig. 9*b*. This is one of the lowest power-consuming cells with full swing output as found in [12]. An output buffer is added to this cell to improve the driving capability. The MUX gate is implemented using the transmission gate circuit in Fig. 9*c*. Results in [12] indicate that this cell is the best for MUX implementation with respect to power considerations. The output buffers are added at the expense of power dissipation to increase the driving capability of the circuit [8].

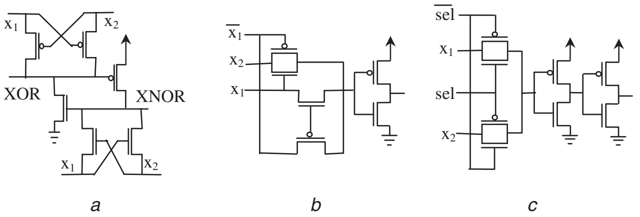


Fig. 9 *Pass logic implementations of XOR–XNOR, XOR and MUX cells*
a Six-transistor XOR–XNOR cell
b XOR
c MUX

Table 1 provides information about the architectures that are simulated. Seven different implementations are used and their simulations are named as Sim1 to Sim7. Sim1 uses Arch1, Sim2 uses Arch2 and Sim3 uses Arch3. The remaining four are based on our proposed architectures, Design 1 and Design 2. In each case, the specific circuit configuration used for each XOR and MUX gate and the total number of transistors used is given in Table 1. Design 1 has the minimum transistor count of 60 when implemented using the six-transistor XOR–XNOR cell, while Arch2 has the maximum of 82 when the CPL-based XOR–XNOR and MUX cells are used. The last column in Table 1 shows the total widths of all transistors added together in each design. Even though this may not be directly related to the area occupied by each design, it may be used as a first estimate for comparing their sizes.

The first set of simulations is aimed at measuring the speed of operation. Table 2 provides the delay characteristics of these different circuits at a frequency of 500 MHz. At supply voltages below 2 V, our designs (Design 1 and Design 2) implemented using the six-transistor XOR–XNOR cell failed to operate correctly owing to their excessive delays. By observing the delay characteristics we notice that Sim7 (Design 2) has the lowest values for delay, which varies from 0.46 ns at 3.3 V to 1.1 ns at 1.5 V. The row corresponding to this design is shown in bold.

Table 3 provides the power characteristics at 500 MHz. From this table, the power consumption is minimum for

Table 2: Delay characteristics of 5 : 2 compressor circuits at 500 MHz

Delay (ns)	1.5 V	1.8 V	2.0 V	2.5 V	3.0 V	3.3 V
Sim1	1.4154	1.0818	0.94624	0.74984	0.64623	0.61043
Sim2	1.4289	1.0962	0.9599	0.7624	0.6576	0.6218
Sim3	1.6205	1.1766	1.0234	0.7665	0.6524	0.61956
Sim4	–	–	1.4608	1.1158	0.94359	0.88102
Sim5	1.1708	0.89382	0.78134	0.61707	0.52935	0.49828
Sim6	–	–	1.2983	0.86347	0.67260	0.64765
Sim7	**1.0959**	**0.84195**	**0.73543**	**0.57478**	**0.48911**	**0.45898**

Table 3: Power characteristics of 5 : 2 compressor circuits at 500 MHz

Power (μW)	1.5 V	1.8 V	2.0 V	2.5 V	3.0 V	3.3 V
Sim1	99.37	152.54	195.39	331.67	515.14	655.51
Sim2	101.43	155.63	197.44	333.91	517.53	658.07
Sim3	**73.08**	**112.19**	**144.93**	**249.18**	**388.91**	**483.45**
Sim4	–	–	164.53	280.58	445.87	575.37
Sim5	97.45	147.65	188.99	323.10	503.21	638.48
Sim6	–	–	164.63	285.17	454.40	583.35
Sim7	100.05	153.10	196.41	332.54	516.15	656.81

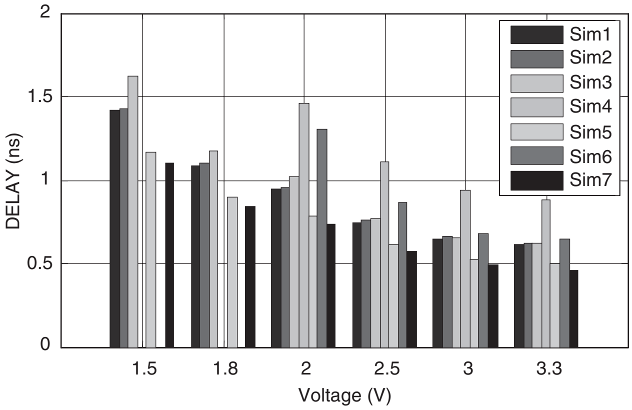


Fig. 10 *Delay at 500 MHz for different supply voltages*

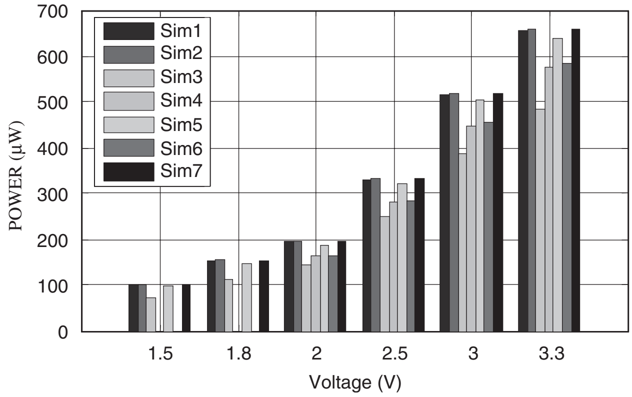


Fig. 11 *Power at 500 MHz for different supply voltages*

Sim3 with values ranging from 484 μW at 3.3 V to 73 μW at 1.5 V. This corresponds to Arch3 implemented with the low power XOR-XNOR cell [8]. This row is marked in bold.

The charts in Figs. 10 and 11 provide a direct comparison of the worst-case delay and power dissipation for different circuits at various supply voltages respectively.

5 Conclusions

In this paper we examined the various 5 : 2 compressors available in the literature and then proposed two new high-speed 5 : 2 compressor architectures, denoted as Design 1 and Design 2. The new designs limit the carry propagation delay to a single compressor stage. They are then analysed for their critical delay paths and are found to be faster than the best one reported in the literature (Arch1). Tspice simulations are performed at 500 MHz; using 0.25 μm CMOS process parameters for supply voltages ranging from 1.5 V to 3.3 V. The architectures are implemented using two different XOR–XNOR cells, and the transistor count varies from a minimum of 60 for Design 1 to a maximum of 74 for Design 2. The least values for delay are for Design 2, which varies from 0.46 ns at 3.3 V to 1.1 ns at 1.5 V. However, Design 1 showed better power characteristics compared to Design 2 with values ranging from 575 μW at 3.3 V to 165 μW at 2.0 V. On the whole our designs showed power savings and about 25% improvement in speed as opposed to Arch1, thereby validating our analytical findings.

6 References

1 McSorley, O.L.: 'High speed arithmetic in binary computers', *Proc. IRE*, 1961, **49**, (1), pp. 67–91

2 Oklobdzija, V.G., and Villeger, D.: 'Improving multiplier design by using improved column compression tree and optimized final adder in CMOS technology', *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 1995, **3**, (2), pp. 292–301

3 Song, P.J., and De Micheli, G.: 'Circuit and architecture trade-offs for high-speed multiplication', *IEEE J. Solid-State Circuits*, 1991, **26**, (9), pp. 1184–1198

4 Weinberger, A.: '4-2 carry-save adder module', *IBM Tech. Discl. Bull.*, 1981, **23**, (8), pp. 3811–3814

5 Ghosh, D., Nandy, S.K., and Parthasarathy, K.: 'TWTXBB: a low latency, high throughput multiplier architecture using a new 4-2 compressor'. 7th Int. Conf. on VLSI Design, Calcutta, India, 1994, pp. 77–82

6 Prasad, K., and Parhi, K.K.: 'Low power 4-2 and 5-2 compressors'. Proc. 35th Asilomar Conf. on Signals, Systems and Computers, 2001, Vol. 1, pp. 129–131

7 Kwon, O., Nowka, K., and Swartzlander, E.E.: 'A 16-bit x 16-bit MAC design using fast 5:2 compressor'. Proc. IEEE Int. Conf. on Application-Specific Systems, Architectures, and Processors, 2000, pp. 235–243

8 Chang, C.H., Gu, J., and Zhang, M.: 'Ultra low voltage, low power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits', *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, 2004, **51**, (10), pp. 1985–1997

9 Uyemura, J.P.: 'Introduction to VLSI circuits and systems' (John Wiley & Sons, Inc, 2002)

10 Rabaey, J.M., Chandrakasan, A., and Nikolic, B.: 'Digital integrated circuits' (Prentice Hall, 2002)

11 Radhakrishnan, D.: 'Low-voltage low-power CMOS full adder', *IEE Proc., Circuits Devices Syst.*, 2001, **148**, (1), pp. 19–24

12 Shams, A.M., and Bayoumi, M.A.: 'A structured approach for designing low power adders'. Proc 31st Asilomar Conf. on Signals, Systems & Computers, 1997, Vol. 1, pp. 757–761