ELectronics EXpress

IEICE Electronics Express, Vol.4, No.22, 696–700

15 GHz low-voltage-swing carry-lookahead adder

Fatemeh Kashfi[a)], **Amir Agah, S. Mehdi Fakhraie, and Saeed Safari**

Silicon Intelligence and VLSI Signal Processing Laboratory, School of ECE, University of Tehran, Tehran 14395–515, Iran

a) *f.kashfi@ece.ut.ac.ir*

Abstract: We describe a high speed adder that employs a carry-lookahead structure and uses low-voltage-swing pass-transistor-based Manchester carry chain. This structure is implemented in 65 nm technology and accommodates 15 GHz clock frequency at the slowest corner which is 20% higher than the highest speed in the previously studied high-speed structures.

Keywords: carry-lookahead adder, low-voltage-swing

Classification: Integrated circuits

References

[1] D. J. Deleganes, M. Barany, G. Geannopoulos, K. Morrise, D. Milliron, A. P. Singh, and S. Wijeratne, "Low Voltage Swing logic circuits for a Pentium® 4 processor integer core," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 36–43, Jan. 2005.

[2] F. Kashfi and S. M. Fakhraie, "Implementation of a high speed low-power 32-bit adder in 70 nm technology," in *Proc. International Symposium on Circuits and Systems*, ISCAS 2006, pp. 9–12, May 2006.

[3] S. B. Wijerante, N. Siddaish, S. K. Mathew, M. A. Anders, R. K. Krishnamurthy, J. Anderson, M. Ernest, and M. Nardin "A 9-GHz 65-nm Intel Pentium 4 processor integer execution unit," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 26–37, Jan. 2007.

[4] A. Agah, S. M. Fakhraie, and A. Emami, "Tertiary-tree 12-GHz 32-bit adder in 65 nm technology," in *Proc. International Symposium on Circuits and Systems*, ISCAS 2007, pp. 3006–3009, May 2006.

[5] K. Martin, *Digital Integrated Circuit Design*, Oxford University Press, 1999.

1 Introduction

Addition is one of the main operations in processor execution unit and also in digital signal processors. Low-voltage-swing topologies [1, 2] and carry-lookahead structures with sparse tree carry-generate circuits [3, 4] are two important techniques used in the latest high speed adders.

The proposed adder structure in this article has the advantage of using both of these techniques simultaneously. Carry-generate and sum-generate

DOI: 10.1587/elex.4.696
Received September 16, 2007
Accepted October 04, 2007
Published November 25, 2007

circuits in this carry-lookahead structure are made of pass-transistor-based Manchester carry chains in which the internal signals have low voltage swing. This structure is shown to provide a very high speed design at acceptable power. All simulations are done in HSPICE environment with 65 nm CMOS technology.

2 Adder structure

This adder uses a carry-lookahead (CLA) structure with low-voltage-swing (LVS) technique, so we call it LVS-CLA adder. A CLA adder is based on calculating the carry-out for a block of bits in parallel to, and independent of, calculating the sum outputs of the block. In this structure the carry-out is given by [5]:

$$C_{\text{out-i}} = G_{\text{i}} + P_{\text{i}}G_{\text{i-1}} + P_{\text{i}}P_{\text{i-1}}G_{\text{i-2}} + ... + P_{\text{i}}...P_1G_0 + P_{\text{i}}...P_1P_0C_{\text{i}} \quad (1)$$

where P_i's are the propagate, and G_i's are the generate signals, and some adders also use the kill, K_i, signal given by:

$$G_{\text{i}} = A_{\text{i}}.B_{\text{i}}, P_{\text{i}} = A_{\text{i}} \oplus B_{\text{i}}, K_{\text{i}} = \overline{A_{\text{i}} + B_{\text{i}}} \quad (2)$$

To produce the LVS signals in the internal nodes, pass-transistor-based Manchester carry chain logic is used. As the LVS signals have to flow in the circuit differentially [1], we use two complement branches of this chain in the whole circuit. The building block of our carry-generate circuit is shown in the bubble ($i = 0$ in this building block) in Fig. 1. The stage 1 only generates or kills the carry signal, whereas the stage 2 can propagate the carry from the previous stage or generate or kill the carry signal itself. The equation that describes CO_{i+1} can be written as:

$$\begin{aligned} CO_{\text{i+1}} &= \overline{K}_{\text{i+1}}\left(G_{\text{i+1}} + CO_{\text{i}}P_{\text{i+1}}\right) \\ &= \left(A_{\text{i+1}} + B_{\text{i+1}}\right)\left[A_{\text{i+1}}.B_{\text{i+1}} + CO_{\text{i}}\left(\overline{A}_{\text{i+1}}.B_{\text{i+1}} + A_{\text{i+1}}.\overline{B}_{\text{i+1}}\right)\right] \\ &= A_{\text{i+1}}.B_{\text{i+1}} + CO_{\text{i}}\left(A_{\text{i+1}} \oplus B_{\text{i+1}}\right) \end{aligned} \quad (3)$$

2.1 Carry-generate circuit

Being a CLA adder, it is using a sparse tree structure in which every fourth carry-out (C_3, C_7,..., C_{27}) is generated. The complete structure of the sparse tree carry-generate circuit is shown in Fig. 1.

To explain this structure, assume that a carry signal is generated in bit 5 ($G_4 = 1$). Therefore, the $\overline{G}_4$-PMOS transistor will set its output node to one in the main branch and the G_4-NMOS transistor will kill the carry-complement signal in the complement branch. If carry isn't killed or generated in bit 6 ($G_5 = 0$ and $K_5 = 0$) the P_5-pass-transistors will propagate the carry and its complement signals. In the case that the carry also doesn't generate or kill by bits 7 and 8 (means $P_6 = 1$ and $P_7 = 1$) the output of the pass-transistors controlled by $P_{6,7}$ ($= P_6.P_7$) will determine the carry

DOI: 10.1587/elex.4.696
Received September 16, 2007
Accepted October 04, 2007
Published November 25, 2007

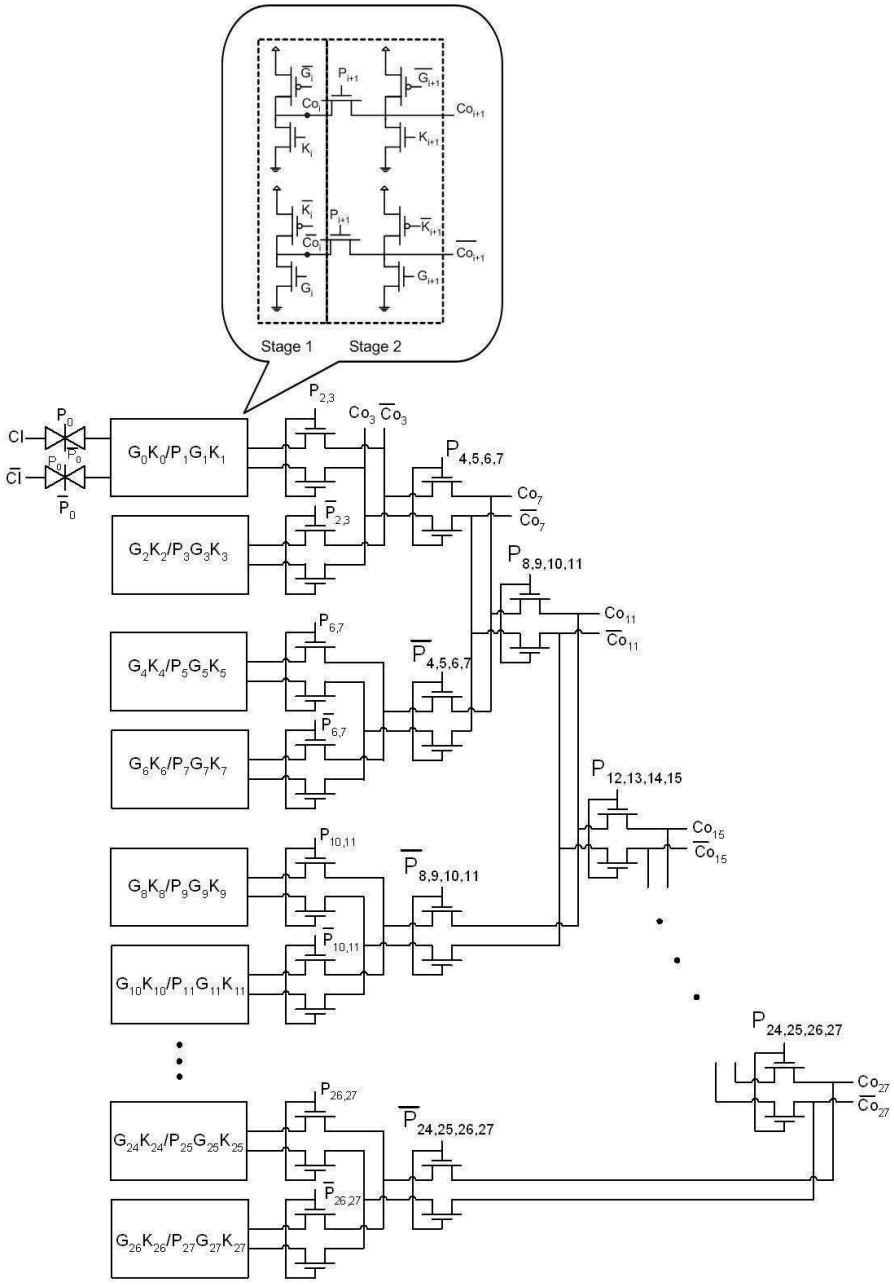

Fig. 1. Carry-generate circuit of the proposed adder.

signal up to that node. The output of a $P_{6,7}$-pass-transistor has a wire-OR connection with the output of a $\overline{P}_{6,7}$-pass-transistor in both the main and its complement branch. The $\overline{P}_{6,7}$-pass-transistor is placed because the carry and its complement don't propagate in the parallel branches containing higher bits. If carry is generated or killed in bits 7 or 8, then $\overline{P}_{6,7} = 1$ and the wire-OR output node again determines the correct carry signals. The four-input ANDs have the same operation similar to the explained two-input ANDs.

As soon as the signal passes through a pass-transistor, it is converted to LVS signal. Carry-generate sparse tree in recent high speed adder structures [3, 4] was implemented with static-dynamic logics where the internal

DOI: 10.1587/elex.4.696
Received September 16, 2007
Accepted October 04, 2007
Published November 25, 2007

nodes have the full signal swing. Our proposed tree has the advantage of using LVS signals in the internal nodes, and this will reduce propagation delay significantly. In comparison with 32-bit LVS carry-select adder with blocks of 16-bit carry skip adders, in which there is a successive carry generation [2], as the proposed adder has a CLA structure, parallel carry generation is fulfilled. Every forth bit can independently generate its carry-out. In the worst case situation where carry has to propagate from carry-in input to the bit 27, it has to pass through 9 pass-transistors in the carry-generate circuit to reach the 8th sum-generate circuit. For the LVS carry skip structure [1] it is mentioned that the LVS signal should not propagate through more than 6 pass-transistors to keep its strength, however, in comparison as there are no pass-transistor-based XOR gates after each pass-transistor in our carry-generate tree (as the sum signals generate separately), the carry signal keeps its strength at the worst case too. To keep the signal strength, the input carry of the whole circuit and its complement are applied to the first carry-generate stage with transmission gates controlled by P_0 signal (Fig. 1).

2.2 Sum-generate circuit

Each fourth differential carry-out feeds the input of a 4-bit carry-ripple sum-generate adder again made up of Manchester carry chain shown in Fig. 2. In these 8 adders, the carry signal will be XORed with propagate signal of the next stage to generate sum output of that stage.

In the output nodes of the circuit which are sum and 31st carry outputs, sense amplifiers are used to amplify differential small signals to the standard levels of zero and one [1]. Propagate, generate, and kill signals are produced with static logics, while two-input and four-input AND gates are implemented with dynamic logic. The reset network is only applied to the sum-generate circuit in which the dynamic outputs are generated. Carry-generate circuit doesn't need any resetting phases and its situation is correctly modified with

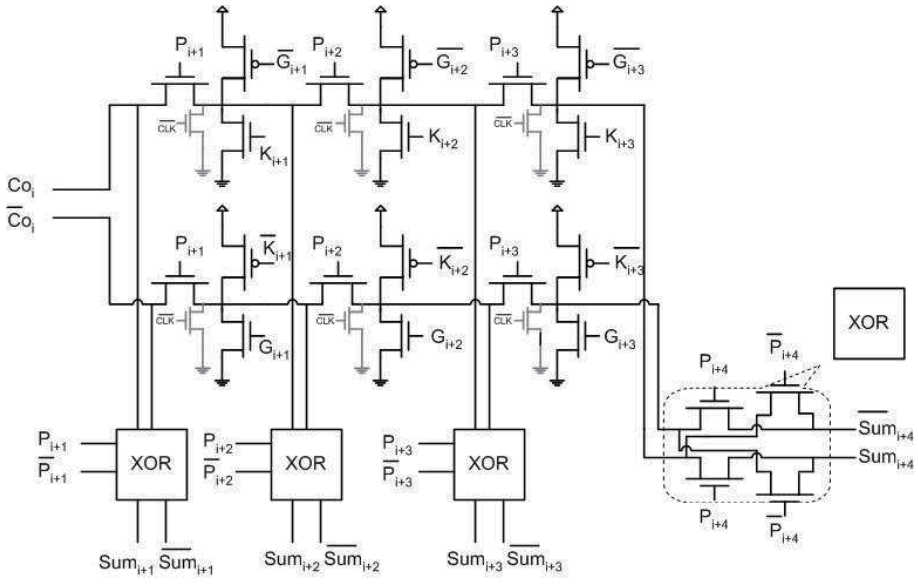

Fig. 2. four-bit carry-ripple sum-generate structure.

DOI: 10.1587/elex.4.696
Received September 16, 2007
Accepted October 04, 2007
Published November 25, 2007

the changes of the inputs.

3 Results and Conclusion

Using the explained techniques, we achieved 15 GHz clock frequency at all process corners with minimum 1.2 V voltage supply. The circuit with its peripheral circuits takes an area of 1740 μm^2.

We compared our design with three previous high speed 32-bit adders [2, 3, 4]. We implemented these adders in 65 nm technology with 1.3 V voltage supply and compared them with our proposed adder in Table I. These adders are guaranteed to accommodate the shown frequency at all process corners. The LVS-CLA adder is 20% faster than LVS carry-select adder [2] with 52% less power dissipation. This achieved for usage of fewer transistors used in the main circuits of the LVS-CLA adder. More area occupied by the LVS-CLA adder is due to static and dynamic peripheral circuits with little power dissipation. In comparison to static-dynamic sparse tree structures [3, 4], LVS pass-transistor-based Manchester chain logic is more appropriate for very high speed structures while static-dynamic sparse trees can dissipate less power in some implementations. Therefore, for achieving very high speed and low power dissipation LVS-CLA adder is much more appropriate among the latest structures while occupying more area.

Table I. Comparison of Different Adder Structures.

Different Adders	VDD (V)	Frequency (GHz)	Power (mW/GHz)	Area (μm²)
Sparse Tree [3]	1.3	10.4	1.13	1126
Tertiary Tree [4]	1.3	11.1	2.1	1170
LVS carry-select [2]	1.3	12.5	3.28	1235
LVS-CLA (this work)	1.3	15	1.55	1740

DOI: 10.1587/elex.4.696
Received September 16, 2007
Accepted October 04, 2007
Published November 25, 2007