Compound Adder Design Using Carry-Look-ahead / Carry Select Adders

Jayaprakash M*, Dr.A. Shanmugam**
*AP, Department of ECE, SCAD Institute of Technology, Coimbatore, India
**Principal, Bannari Amman Institute of Technology, Sathyamangalam, India

ABSTRACT

To reduce fan-out load at the final multiplexor stage and to relieve the speed requirement of compound adders, a parallel global carry generation is used for a hybrid carry-look ahead/ carry-select adder design. To develop the speed and to minimize silicon area, a new logical decomposition is derived. The new architecture has been explained with a 64-bit adder design using dynamic CMOS circuit implementations. The 64-bit adder has the delay of 525ps in 0.225µm bulk CMOS technology.

Keywords: Carry look ahead adder, Carry select adder, CMOS, Fan-out, multiplexor.

1. INTRODUCTION

A fast tree structure is employed in carry look ahead adder (CLA) in global and local carry generation. The CLA architecture can be combined with carry select schemes for speed improvement [1][2]. Two conditional addition results are pre-computed for each specific block in the hybrid scheme. Each global carry is used to select one of the two results. As a result, in the hybrid adder architecture the delay of the local carry generation is eliminated. In this paper we proposed fast CMOS circuits for the hybrid carry look ahead / carry select adder design.

A simple parallel scheme is proposed to relieve the fan-out load at the final multiplexor stage for the global carry generation in the hybrid adder. In the look ahead logic and exhaustive duplication of look ahead cells, the parallelism does not require intermediate outputs. The majority of look ahead cells are shared among multiple look ahead trees.

A new compound adder design is projected for the carry-select scheme in the hybrid architecture. To minimize the area of the compound adder internal carry generation logic is shared in the new compound adder. To improve the speed of the carry look ahead logic the new compound adder is designed to be used with Ling’s pseudo carry scheme.

2. ARCHITECTURE OF A CARRY-LOOK AHEAD / CARRY SELECT ADDER

2.1 Block Diagram

Fig. 1 indicates the block diagram of the hybrid carry –look ahead / carry select adder architecture. For fast computation of the global carries, usually tree architecture is used [3][4]. Each global carry selects one of the two results of the corresponding compound adder.
2.2 Parallel Carry Generation

Figure 2 (a) shows 8-bit spaced global carry generation. It has two standard 16-bit spaced global carry generation trees where one of them is a simply 8-bit shifted version of the other. It has more uniformly distributed fan-out load in the global carry generation at the expense of more redundant look ahead cells than the spanning tree network of Lynch, Figure 2 (b) shows a 4-bit spaced global carry generation using a 4-way parallelism.

As the level of parallelism is increased, the compound adder design is simplified. The extreme case of such parallelism is Kogge-Stone tree, where there are 16 global carry generation trees for a 64-bit adder and the resulting global carry spacing is one bit [9]. In this case, only a simple XOR is required to calculate the final sum bits and power consumption overhead of multiple tree networks is the limiting factor in having such exhaustive parallelism.

2.3 Logical Decomposition For A Compound Adder

Let \( a_i \) and \( b_i \) denote the two operand bits at \( i \)-th bit position. \( g_i \) and \( p_i \) are defined as generate and propagate signals at the position \( i \), respectively.

\[
p_i = a_i + b_i \\
g_i = a_i \cdot b_i
\]

Let \( g_{i:j} \) denotes that a carry is generated in a group of bits from \( i \) to \( j \) inclusive and propagated to bit position \( (i+1) \) [1]. \( p_{i:j} \) is similarly defined as a group propagate signal denoting that the input carry of bit position \( j \) is propagated to bit position \( (i+1) \).
The compound adder is designed using carry look ahead architecture. For convenience, only two level look ahead hierarchy is used. In the hierarchy of the carry look ahead tree, for the \( i \)-th bit position, there is a block in the first level look ahead stage where the \( i \)-th bit belongs. Denote \( f_1(i) \) as the LSB (Least Significant Bit) position of the block. Then, the carry-in bit of the \( i \)-th bit position can be represented as below.

\[
c_i = g_{i-1:1} + p_{i-1:1}c_0
\]

Now, the two conditional carries are represented as below.

\[
c_0^i = g_{i-1:1} + p_{i-1:1}g_{f_1(i)-1:0}
\]
\[
c_1^i = g_{i-1:1} + p_{i-1:1}g_{f_1(i)-1:0} + g_{f_1(i)-1:0} + p_{i-1:1}g_{f_1(i)-1:0}
\]

The conditional sum bits, \( s_0^i \) and \( s_1^i \), are obtained through the final exclusive OR stage as follows.

\[
p_1^i = (a_i \oplus b_i)
\]
\[
s_0^i = (c_0^i \oplus p_1^i)
\]
\[
s_1^i = (c_1^i \oplus p_1^i)
\]

### 3. CIRCUIT IMPLEMENTATION OF 64 BIT HIGH-SPEED ADDER DESIGN

The 2 way parallel global carry generation and the 8-bit compound adder are explained with a 64-bit adder design. For high speed implementation delayed-reset dynamic CMOS circuits are used\[6\]. For further speed improvement, links pseudo carries are employed \[4\][7][8]. The first level look ahead logic and the 8-bit compound adder are modified to accommodate the use of Ling’s pseudo carries.

#### 3.1 Carry Generation

For a hierarchical carry generation scheme for the 64-bit adder, \( G_4 \) and \( P_4 \) are defined as a group generate and a group propagate of a group of 4-bits. With the use of an identity, \( g_{i}=p_{i}g_{i} \), the following equations are obtained for \( i=0,4,8,12...56 \) and 60.

\[
G_4i/4 \equiv g_{i+3:i}
\]
\[
= g_{i+3:i} + p_{i+3}g_{i+2} + p_{i+3}g_{i+2} + p_{i+3}p_{i+2}g_{i+1} + p_{i+3}p_{i+2}g_{i+1}g_{i}
\]
\[
= p_{i+3}G_4^*i/4
\]

\[
P_4i/4 \equiv p_{i+3}p_{i+2}p_{i+1}p_i
\]

The 16-bit spaced global carries can be obtained by the following equations.

\[
G_4^*i/4 \equiv g_{i+3:i} + p_{i+3}g_{i+2} + p_{i+3}g_{i+2} + p_{i+3}p_{i+2}p_{i+1}g_{i}
\]

Here, \( G^*_4 \) and \( P^*_4 \) are pseudo group generates and pseudo group propagates in the Ling’s pseudo carry scheme \[4\][8]. \( G_{16}, P_{16}, G_{16}^* \) and \( P_{4}^* \) are similarly defined for 16-bit groups as in Quanch[4] except for differences in group partitioning. Using the group generates and group propagates, the 16-bit spaced global carries can be obtained by the following equations.
\[
C_{16} = G16_0 + P16_0 C_0 = p_{15} (G16^*_0 + P16^*_0 C_0)
= p_{15} C^*_0
\]
\[
C_{32} = G16_1 + P16_1 G16_0 + P16_1 P16_0 C_0
= p_{31} (G16^*_1 + P16^*_1 G16^*_0 + P16^*_1 P16^*_0 C_0)
= p_{31} C^*_0
\]
Similarly,
\[
C_{48} = p_{47} C^*_0
\]
From the above equations, global carries can be simply obtained from Ling’s pseudo global carries. Hence, pseudo global carries are generated in the look ahead network since the use of pseudo carries simplifies the first stage of look ahead network. The use of pseudo carries can be easily handled in the new compound adder design to get the correct sum bits without hurting critical path delay[5].

Now using equation 3, equation 4 and the identity of \( g_i = g_i \cdot p_i \), the pseudo generate and propagate for the first 4-bit block are reformulated as follows.

\[
G4^*_0 = g_3 + g_2 + p_2 g_1 + p_2 p_1 g_0
= (g_3 + g_2) + p_2 p_1 (g_1 + g_0)
\]
\[
P4^*_0 = p_2 p_1 p_0 p_{-1}
= (p_2 p_1) (p_0 p_{-1})
\]
Hence, \( p_{-1} \) is used to maintain the consistency with other groups though it can be omitted for the first 4-bit block.

Figure 3 shows a dynamic CMOS circuit implementation of the first look ahead stage. The generation of \( p_i \) and \( g_i \) is merged in the first look ahead logic and the newly derived logic equations with the compound dynamic CMOS circuit styles provides a simpler and faster circuit. The new circuit is faster by 67% in G4 than the direct implementation. Similar circuits are used for the following 16-bit group generates and group propagates.

Figure 3: Dynamic CMOS Circuit for the First Look ahead Stage

So far, only 16-bit spaced global carry generation has been explained for the convenience. With the same mechanism other 8-bit offseted global carries can be obtained. They are also 16-bit spaced global carries for another 64-bit group which consists 56 LSB bits of input operands and 8 dummy bits of ‘0’ as seen in Figure 2(a).
3.2 Compound Adder

For a 8-bit block with a LSB at bit position I, the carry-in at the (i+7)-th bit position can be represented as follows.

\[ C_{i+7} = g_{i+6:i} + p_{i+6:i} C_i \]

\[ = (g_{i+6:i+3} + p_{i+6:i+3} g_{i+2:i}) + (p_{i+6:i+3} p_{i+2:i-1}) C^*_{i} \]

Let \( S_{i+7}^{(0)} \) and \( S_{i+7}^{(1)} \) denote the corresponding conditional sum bits at bit position (i+7) respectively. Then,

\[ S_{i+7}^{(0)} = (g_{i+6:i+3} + p_{i+6:i+3} g_{i+2:i}) \oplus p_{i+7} \]

\[ S_{i+7}^{(1)} = (g_{i+6:i+3} + p_{i+6:i+3} (g_{i+2:i} + p_{i+2:i-1})) \oplus p_{i+7} \]

![Figure 4: A New 8-bit Compound Adder](image)

Figure 4 shows a block diagram of the new 8-bit compound adder. As shown in the figure, carry generation circuits are shared for both cases, which leads to area-efficiency. Moreover, the last two logical stages can be merged in one domino stage to minimize the delay and area as shown in Figure 5.

In the figure, \( k \) and \( q \) are needed for dynamic CMOS XOR implementation. They are logical complements of \( p \) and \( g \), respectively, and are defined as follows

\[ k_i = \overline{a_i} \cdot \overline{b_i} \]

\[ q_i = \overline{a_i} + b_i \]

3.3 Performance

The new 64-bit hybrid adder was simulated using 0.225 µm bulk CMOS technology. It has a delay of 524ps on the critical path at 1.62V and 85ºC operating condition. The new design is 15% faster than the conventional design in global carry generation. The main speed improvement comes first look ahead stage where the new design employs Ling’s pseudo carry scheme with a new simplified dynamic CMOS circuit. In compound adder designs, the 8-bit compound adder is fast enough to match the delay of the global carry generation.
4. CONCLUSION

A proposed architecture in this paper is a new fast carry look ahead/carry select adder. The fan-out load of the global carries can be significantly reduced and the design of compound adder is eased by duplicating some parts of global carry generation network with an offset. A new fast area-efficient compound adder has been proposed in order to relieve the area burden and speed requirement in the carry-select scheme. In order to accommodate the use of Ling’s pseudo carries the compound adder has been designed. The proposed 64-bit dynamic CMOS adder has a 13% speed improvement compared with conventional design in 0.225µm COMS technology.

5. REFERENCES