Transition Reduction in Carry-Save Adder Trees

Patrik Larsson and Chris J. Nicol
Bell Laboratories, 101 Crawfords Corner Rd., Holmdel, NJ 07733
pal@bell-labs.com, chrisn@bell-labs.com

***Abstract* — By taking advantage of the redundancy in a 4-2 compressor, we reduce the number of transitions in carry-save adder trees that are common in large multipliers. Three new 4-2 compressors are proposed. These are used in different configurations to reduce the probability of a transition in the global carry wires by up to 40% over current techniques. Power reductions are demonstrated with the use of a 4-tap FIR filter module and a 54×54-bit multiplier. Transistor level circuit simulations indicate 5-6% power reduction with no increase in delay.**

I. Introduction

In a logic circuit there are two different types of transitions on capacitive nodes causing power consumption. The necessary transitions, named 'valid' transitions in [1], describe the differences in states of internal nodes of a circuit between two consecutive clock cycles. Spurious transitions or 'glitches' are generated at the output of a logic circuit when the inputs are not applied simultaneously. The spurious transitions do not contribute to the signal processing but increase power consumption.

In [1], the spurious transitions were reduced by delay matching the input signals to a multiplier. In this work, we focus on reducing the number of valid transitions by selecting proper Boolean functions. This method reduces both valid and spurious transitions, since valid transitions generate spurious transitions. By choosing a proper Boolean function, the transition reduction is guaranteed over all process, temperature and power supply variations, which might not be the case for the delay matching method. Furthermore, the proposed method is a viable supplement to delay matching, since it will also work in perfectly balanced logic.

Logic restructuring has been proposed to reduce power consumption of state-machines [2] and logic containing "don't-care" states [3]. However, for highly optimized datapaths, "don't-care" states seldom exists and there is no freedom in selecting states to reduce power consumption. In this paper, we propose to explore redundancy in a common subcell in datapaths; the 4-2 Carry-Save Adder.

II. Redundancy in 4-2 CSA

High-speed multipliers [4,5] often contain a Wallace tree of carry-save adders to sum the partial products. The most commonly used circuit is the 4-2 compressor (5-3 counter) shown in Fig. 1. C_S and C_L represent a small and a large parasitic capacitor respectively, reflecting the fact that C_{out} is a local connection while Cry often is a wire routed over several cells.

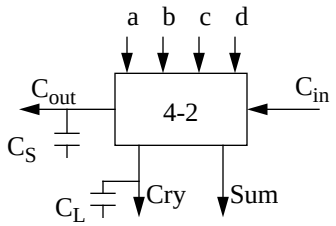

Fig. 1. Signal labeling in a 4-2 compressor.

The weight of C_{out}, Cry and Sum is 2, 2 and 1 respectively. Following the notation in [6], the truth table is as shown in Table I, where Ones represents the number of 1's in $\{a,b,c,d,C_{in}\}$.

Table I Truth table for 4-2 compressor

Ones	Sum	Cry	C_{out}	P
0	0	0	0	1/32
1	1	0	0	5/32
2	0	0/1	1/0	10/32
3	1	0/1	1/0	10/32
4	0	1	1	5/32
5	1	1	1	1/32

For the cases Ones=2 or 3, we have the freedom of setting either Cry=1 or C_{out}=1, but not both, with the restriction that C_{out} must be independent of C_{in}. Assuming random inputs, the state probabilities are shown in the right-most column. The likelihood of Ones = 2 or 3 is 20/32 so we can expect reasonable power savings by reducing the power consumption of these states.

Our objective in this paper is to select proper Boolean functions for Cry and C_{out} so as to minimize the power consumption. From the basic assumption in Fig. 1 that Cry has larger load than C_{out}, we see that power reduction is obtained by minimizing the number of

ISLPED 1996 Monterey CA USA
0-7803-3571-8/96/$5.00 © 1996

transitions on Cry, which can be done in two different ways:

a) If we have access to the previous value of Cry, we can feed this back into the 4-2 and try to keep the next value of Cry the same as the previous.

b) Defining P(0) and P(1) as the probability of a node being low and high, respectively, and assuming that consecutive states are independent, the transition probability, P_t, is

$$P_t = P(0) \cdot (1 - P(0)) + P(1) \cdot (1 - P(1)) = \ldots$$
$$\ldots = 2 \cdot P(0) \cdot (1 - P(0)) \quad (1)$$

where we have used P(0) = 1 - P(1). This is a continuous function with a single maximum located at P(0) = 0.5. From this equation, we see that it is advantageous to force P(0) to be close to 0 or 1 as mentioned in [3].

III. Reducing Transition Probabilities in CSA Trees

In the following, we explain the method of using feedback and skewing the probability density function to reduce the number of valid transitions for a single 4-2 adder. We also discuss ways to configure multiple 4-2 compressors to sum vectors in a CSA tree.

A. Single 4-2 compressor

The minimum requirement on a 4-2 compressor is reflected in the Karnaugh maps in Fig. 2(a) and (b). The Sum function is not shown, since we only discuss the freedom in selecting Cry and C_{out}. The empty slots should be filled with 0's or 1's such that Cry + C_{out} = 1. Furthermore, since C_{out} should be independent of C_{in}, the bottom two maps must be identical.

The method of skewing the probability density function is easily implemented by setting all empty slots to 0 in the top two maps and 1 in the maps for C_{out}. These are the maps for a cell we name an 'even 4-2'. A corresponding 'odd 4-2' is obtained by setting all 1 in the top maps and 0's in the bottom maps.

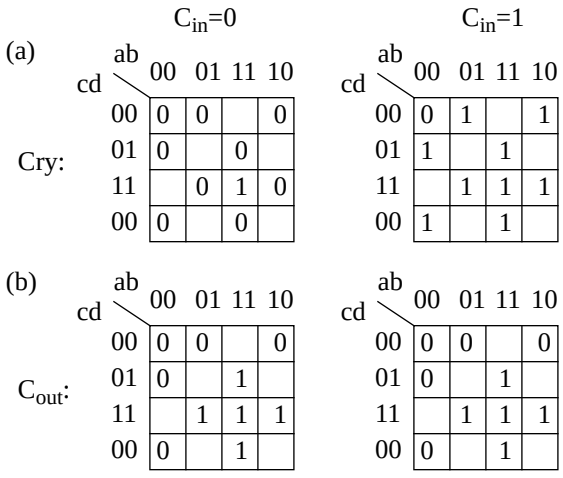

Fig. 2. Karnaugh maps for (a) Cry and (b) C_{out} in a 4-2.

When exploring feedback to reduce the number of transitions, a synchronous approach is taken as shown in Fig. 3, where Cry is the previous value of pC stored in a flip-flop. As indicated by the dashed line, Cry is fed back into the 4-2 so that the 4-2 will try to make pC the same as Cry. The effect of this feedback is that the 4-2 compressor behaves as an even 4-2 when the output of the carry flip-flop is 1 and as an odd 4-2 when the output is 0.

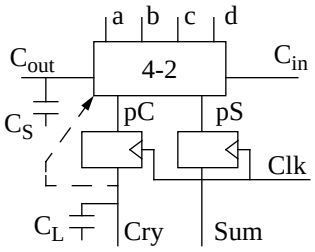

Fig. 3. 4-2 adder with feedback to reduce transitions.

Assuming random inputs with 50% transition probability at a, b, c, d and C_{in}, Table II shows the probability of having a valid transition (ignoring glitches) at the C_{out}, Cry and Sum outputs of several 4-2 compressors: using 2 full adders, the circuit given in [4], and the proposed even, odd and feedback circuit. The probability of having Cry=0 is shown on the right, which indicates that a skewed probability gives reduced number of transitions. For the feedback circuit, equation (1) is not valid since consecutive states are not independent. So, 0.31 is the minimum rather than the maximum transition probability even though P(Cry=0)=0.5.

Table II Transition probability of various 4-2 adders

Type	$P(t_{Cout})$	$P(t_{Cry})$	$P(t_{Sum})$	P(Cry=0)
2xFA	0.5	0.5	0.5	0.5
[4]	0.49	0.49	0.5	0.44
Odd	0.43	0.43	0.5	0.31
Even	0.43	0.43	0.5	0.69
fbk	0.45	0.31	0.5	0.5

B. Row of 4-2 Compressors

When cascading several 4-2's into a vector, we need some caution. An even 4-2 has high probability of having a 1 at the C_{out} output as indicated in the top right hand corner of Fig. 4.If C_{in}=1 of an even 4-2, it is not successful at keeping its Cry output at 0 as intended when selecting the Karnaugh maps. Therefore, it is better to feed the C_{out} of an even 4-2 into an odd 4-2 as done at the top of Fig. 4.

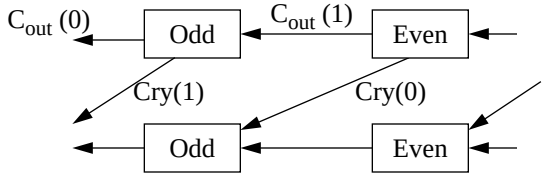

Fig. 4. Alternating odd/even 4-2 adders with most likely logic states given in parenthesis.

Table III shows the probability of having a transition on each output when adding four 32-bit random vectors. The odd/even row corresponds to the use of odd and even compressors in the odd and even bit positions respectively. It is clear that the most effective ways to reduce transitions on the Cry outputs is to use either the odd/even configuration or the feedback circuit.

Table III Transition probabilities in parallel CSA's

Type	$P(t_{Cout})$	$P(t_{Cry})$	$P(t_{Sum})$
2xFA	0.48	0.48	0.5
[4]	0.48	0.48	0.5
Odd	0.42	0.46	0.5
Even	0.42	0.46	0.5
odd/even	0.42	0.33	0.5
fbk	0.43	0.29	0.5

C. Tree of 4-2 Compressors

When implementing an adder tree, there will be a 2D structure of 4-2's as indicated in Fig. 4. It might seem counterintuitive to feed Cry from an even cell into an odd, since Cry is most likely 0, making it less likely for the odd to generate a 1 at its Cry output. However, a detailed study of the Karnaugh maps for the odd cell proves that a 0 from Cry of an even cell as input to one of a,b,c or d increases the probability of giving a 0 at C_{out} of the odd cell. Keeping C_{out}=0 of the odd cell generates less transitions than keeping Cry=1

IV. Circuit Implementations

Circuit examples of the even and odd 4-2 compressors are shown in Fig. 5 and 6, respectively. Noninverting gates can be merged into the next gate, creating complex CMOS gates. The even cell contains 68 transistors when XOR gates are assumed implemented with 10 transistors.

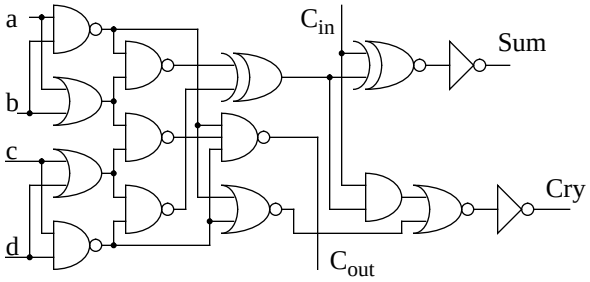

Fig. 5. Example of even 4-2 compressor.

The complexity and speed of this 4-2 is similar to the common 4-2 cell used in [4], which has 58 transistors when buffering both Cry and Sum. Both circuits have a critical path equal to 3 XOR gates from the a,b,c,d inputs to the sum output. Another important factor is the balancing of the Cry and Sum outputs so that spurious transitions in the tree are kept to a minimum. The even 4-2 compressor circuit maintains this balanced timing behavior. The odd 4-2 in Fig. 6 is slightly more complex but can be realized with 70 transistors in standard CMOS logic.

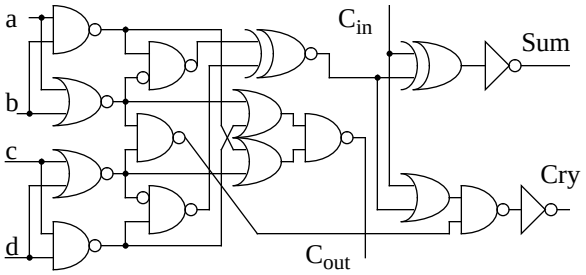

Fig. 6. Example of odd 4-2 compressor.

Fig. 7 shows an implementation of a 4-2 cell with feedback of the previous value of Cry. The feedback increases the number of transistors to 104. Note here that the path from Cry_{in} to pC is noninverting. This implies that the flip-flop assumed at the output of this cell in Fig. 3 is not strictly necessary. However, removing the flip-flop makes C_{out} a function of C_{in}, which is not allowed in a 4-2 adder. To circumvent this, we can feed back C_{out} instead of Cry. The circuit will be identical to Fig. 7, but the inverse of C_{out} connected to Cry_{in}.

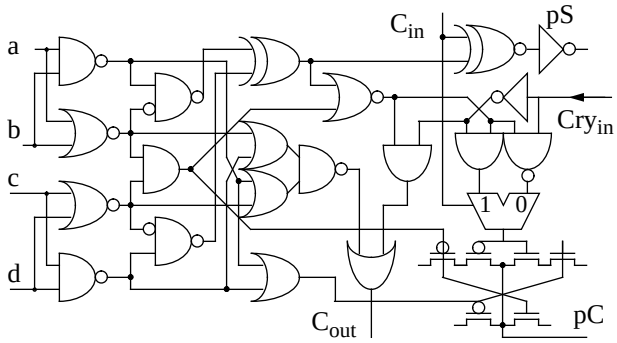

Fig. 7. Example of 4-2 adder with feedback.

V. Simulation Examples

This section describes an FIR architecture and a 54×54-bit multiplier that make extensive use of carry-save adder trees. The FIR module is to be used in a low power adaptive equalizer suitable for use in portable communications devices. The multiplier is presented as a general purpose example.

A. FIR module with 4 multipliers

Fig. 8 shows an FIR module with four multipliers. The 12-bit weights, W, are Booth encoded giving six partial products in each multiplier. These are fed into two full adders and a 4-2 is used to create a carry and a sum vector. The output of each multiplier is fed into a carry-save adder tree to produce a sum of the four products.

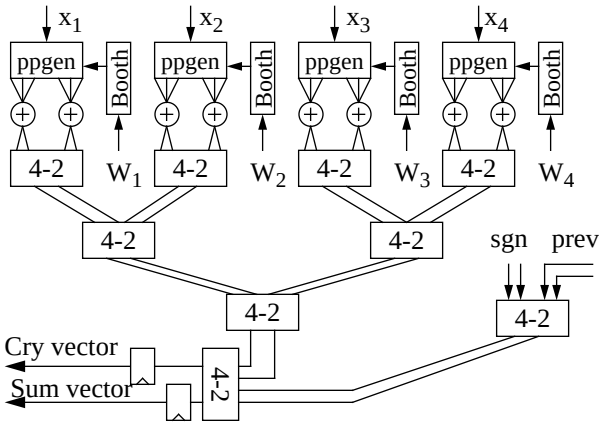

Fig. 8. Simplified schematic of a 4-tap FIR module.

B. 54×54-bit multiplier

The next example is an implementation of a 54×54-bit multiplier suitable for use in a double-precision floating-point processor. The 54-bit multiplier is Booth encoded to produce 28 partial products (including sign and guard bits) which are summed using a CSA tree with a depth of 4. The architecture is similar to that described in [5]. The results from simulations are shown in Table IV. In this example the carry propagate adder has been removed because we concentrate on power reduction in the 4-2 tree.

C. Simulation results

Process parameters from a 0.5µm standard 3.3V CMOS process were used when simulating the FIR4 module and the 54-bit multiplier for various combinations of 4-2 compressors. A timing simulator [7] was used and we assumed fixed-size source and drain diffusion diodes. Capacitance of global wires were estimated from a floorplan based on the layout of subcells. The power consumption for different cases are listed in Table IV.

Table IV Power of multipliers with different cells

Cells	FIR4(mW)	Mul54(mW)
2xFA	79.2	-
[4]	70.8	626
even/even	69.9	618
odd/even	70.3	-
even/[4]	67.9	586
even/[4]+fbk	74.2	-

* Power simulated at Vdd=3.3V and f_{clk}=75MHz

The best results were obtained by using the even/[4] combination where the odd 4-2 compressor is replaced with that described in [4]. The compressor in [4] has a similar truth table to the odd compressor but can be implemented with a simpler (and therefore more power efficient) circuit. The other advantage of using the even/[4] adder combination is that the circuits have balanced timing characteristics resulting in fewer spurious transitions. This is not true in the odd/even combination because the C_{out} path has additional delay in the odd circuit. This will cause an imbalance in the sum and carry outputs of every other bit resulting in spurious transitions throughout the adder tree. However if the truth table of the odd compressor could be implemented more efficiently, better results could be expected for the odd/even combination as seen in Table II and III.

The increased complexity of the 4-2 compressor with feedback causes more power consumption than the method saves. This is of course related to the total output capacitance of the register in Fig. 8, which in this case was estimated at 200fF.

Table IV indicates that the actual circuit implementations presented here do not give as much power saving as predicted in Table II and III. This is mainly due to the increased transistor count of the proposed circuits compared to the cell in [4]. However, as technology scales, the capacitance of wires will dominate over the parasitic capacitance of the transistors, thereby reducing the cost of additional transistors and increasing the gain of the proposed transition reduction method.

VI. Conclusions

We have demonstrated that power reductions are possible in carry-save adder trees by selecting appropriate Boolean functions for the carry outputs of the 4-2 compressors with the intention of minimizing the transitions in high-capacitance nodes. The proposed transistor implementation does not add delay to the critical path but increases the number of transistors by about 10%. The probability of a logic transition on a global carry signal was reduced by up to 40%. However, this translated into only a 6% power saving in a 54×54-bit multiplier.

References

[1] T. Sakuta, W. Lee, P.T. Balsara, "Delay Balanced Multipliers for Low Power/Low Voltage DSP Core", *IEEE Symposium on Low Power Electronics*, Oct. 1995.

[2] K. Roy, S. Prasad, "SYCLOP: Synthesis of CMOS Logic for Low Power Applications", *IEEE Int. Conf. Computer Design (ICCD)*, pp. 464-467, 1992.

[3] A. Chen, A. Ghosh, S. Devadas, K. Keutzer, "On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks", *IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, pp. 402-407, 1992.

[4] G. Goto, T. Sato, M. Nakajima, T. Sukemura, "A 54×54-b Regularly Structured Tree Multiplier", *IEEE J. of Solid-State Circuits*, vol. 27, no. 9, pp. 1229-1236, Sept. 1992.

[5] N. Ohkubo, M. Suzuki, T. Shinbo, T. Yamanaka, A. Shimizu, K. Sasaki, Y. Nakagome, "A 4.4ns CMOS 54 x 54-b Multiplier Using Pass-Transistor Multiplexer", *IEEE J. of Solid-State Circuits*, vol. 30, No. 3, March 1995.

[6] Z. Guan, P. Thomson, A.E.A. Almaini, "A Parallel CMOS 2's Complement Multiplier Based on 5:3 Counter", *IEEE Int. Conf. Computer Design (ICCD)*, pp. 298-301, 1994.

[7] B. Ackland, R. Clark, "Event-EMU: An Event Driven Timing Simulator for MOS VLSI Circuits", *IEEE Int. Conf. on Computer-Aided Design (ICCAD)*, pp. 80-83, 1989.