Early Branch Prediction Circuit for High Performance Digital Signal Processors

Aamir A. Farooqui[1], Vojin G. Oklobdzija[2]
[1]Department of Electrical and Computer Engineering,
University of California, Davis, CA 95616.
e-mail: aamirf@ece.ucdavis.edu.
[2]Integration Berkeley, California.
email: vojin@nuc.berkeley.edu.
http://www.integr.com/

Abstract

In this paper, design and VLSI Implementation of an Early Branch Prediction (EBP) circuit, based on a variation of Carry Look-ahead scheme is presented. The key features of this design are low area, high speed (2 Ø log n/2ø + 1), and high modularity. This design out performs all the EBP designs presented so far. For 64-bit word length the early branch prediction is obtained in 679 ps as simulated for 0.2-mm technology under typical conditions. Simulation and layout results for 0.2-mm CMOS technology show a 30% increase in speed with 25% decrease in area as compared, to recently published results.

1. Introduction

Handling of the conditional branches is an important issue in high-performance computer design. Conditional branch instructions create a "critical path" in many processors. The reason is that the evaluation of the condition (true or false) takes additional time in addition to the execution of branch instruction. Many attempts were made for the early branch prediction (EBP) [1,2,5,6,8]. Recently, David *et. al*. [1] have proposed a circuit based on Prefix-And method. They claim it as the fastest possible circuit, with a delay of *log* n+3 (where n is the number of bits).

In this paper, we propose a new scheme for EBP which requires a delay of 2 ⌈*log* n/2⌉ + 1, with minimum hardware. In this design only one circuit is used to evaluate the Greater than (GT), Less than (LT) and Equal to Zero (ETZ) conditions. In all the EBP presented so far, one circuit is used for the evaluation of GT and LT condition each, and a separate XOR-AND tree for the evaluation of ETZ condition. In order to make a fair comparison between the proposed circuit and a recently published design [1] the two circuits were laid-out in a single chip for comparison purposes.

2. Architecture

Let A = (a_{n-1}, …, a_0) and B = (b_{n-1}, …, b_0) be the two operands to be compared, with a_{n-1} and b_{n-1} be the sign bits. The three conditions to be tested are A>B (GT), A<B (LT), and A=Z (ETZ). The conditions A≥B and A<B can be evaluated by subtracting B from A and looking at the carry out of the result, and the sign bits of A and B (see Table 1). The detection of carry out signal requires the delay of a carry-propagate adder.

In this paper we have used a modified Carry Look-ahead scheme. It is based on the fact that the carry generated (*G*) at bit position *i* ($G_i = a_i \bullet b_i$, where $\bullet$ is the AND operation) can be dropped in the computation of the group carry, if the two input bits at any position *j (j > i)* are zero. We call this function *NZ* (no zero), and it's value at bit position *i* is, $NZ_i = \overline{a_i + b_i}$, where + is the OR operation.

A_n	B_n	Condition
0	1	A>B
1	0	A<B
0	0	A≥B if C_{out}= 1, else A<B
1	1	A≥B if C_{out}= 1, else A<B

Table 1. *Detection of A‡B, and A<B, based on An-1, Bn-1, and Cout.*

Since, the functions G_i and NZ_i (*i= n-1 to 0*) can be generated simultaneously from A and B, the group carry C_{out} can be computed in 3 logic levels (if there is no limitation on fan in/out).

$$C_{out} = NZ_{n-1} \bullet \ldots\ldots \bullet NZ_2 \bullet NZ_1 \bullet G_0 + NZ_{n-1} \bullet \ldots \ldots \bullet NZ_2 \bullet G_1 + \ldots\ldots + G_{n-1} \qquad (1)$$

Using *NZ* and *G* functions we can evaluate ETZ condition as follows:

$$(\overline{NZ_{n-1} + \ldots\ldots + NZ_0}) \bullet (\overline{G_{n-1}} \bullet \cdots\cdots \bullet \overline{G_0}) = (a_{n-1} \oplus b_{n-1}) \bullet \ldots\ldots \bullet (a_0 \oplus b_0) \qquad (2)$$

3. Circuit Design

Based on (1) and (2) a circuit (as shown in Fig. 1) has been designed for the detection of C_{out} and ETZ condition. Due to fan in/out limitations the circuit has been designed for four bits only. The C_{out} (group carry) is produced in 3-gate delays without any Xor gate or tri-state buffer as required in [1,2,3]. The NZ_{out} (group not zero) is produced after 2-gate delays while the ETZ_{out} (group equal) is produced in 3-gate delays with hardware cost of two And gates. The evaluation of A>B, or A<B, based on a_{n-1}, b_{n-1}, C_{out} and A=B, requires a delay of two more additional Xor gates for all methods [1].

Fig-2 shows the transistor level implementation of Fig-1. In this circuit only n-transistors with CMOS inverters are used [6]. The CMOS inverters not only shift the threshold voltage, but also drive the capacitive load The cascading of the 4-bit circuit for 16-bit is shown in Fig-3.

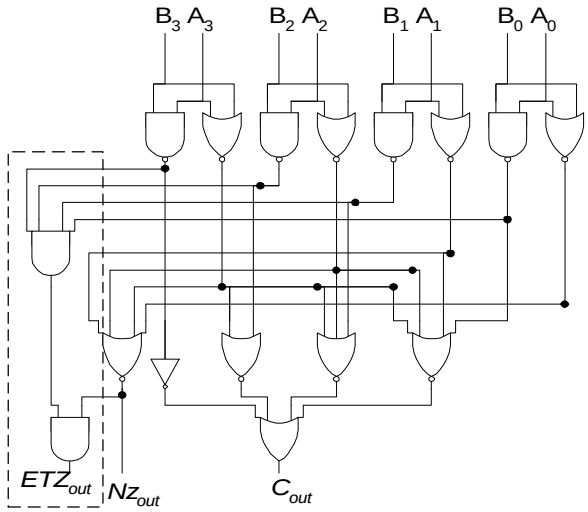

Fig. 1. *Schematic diagram of the 4-bit fast carry and equal to detector.*

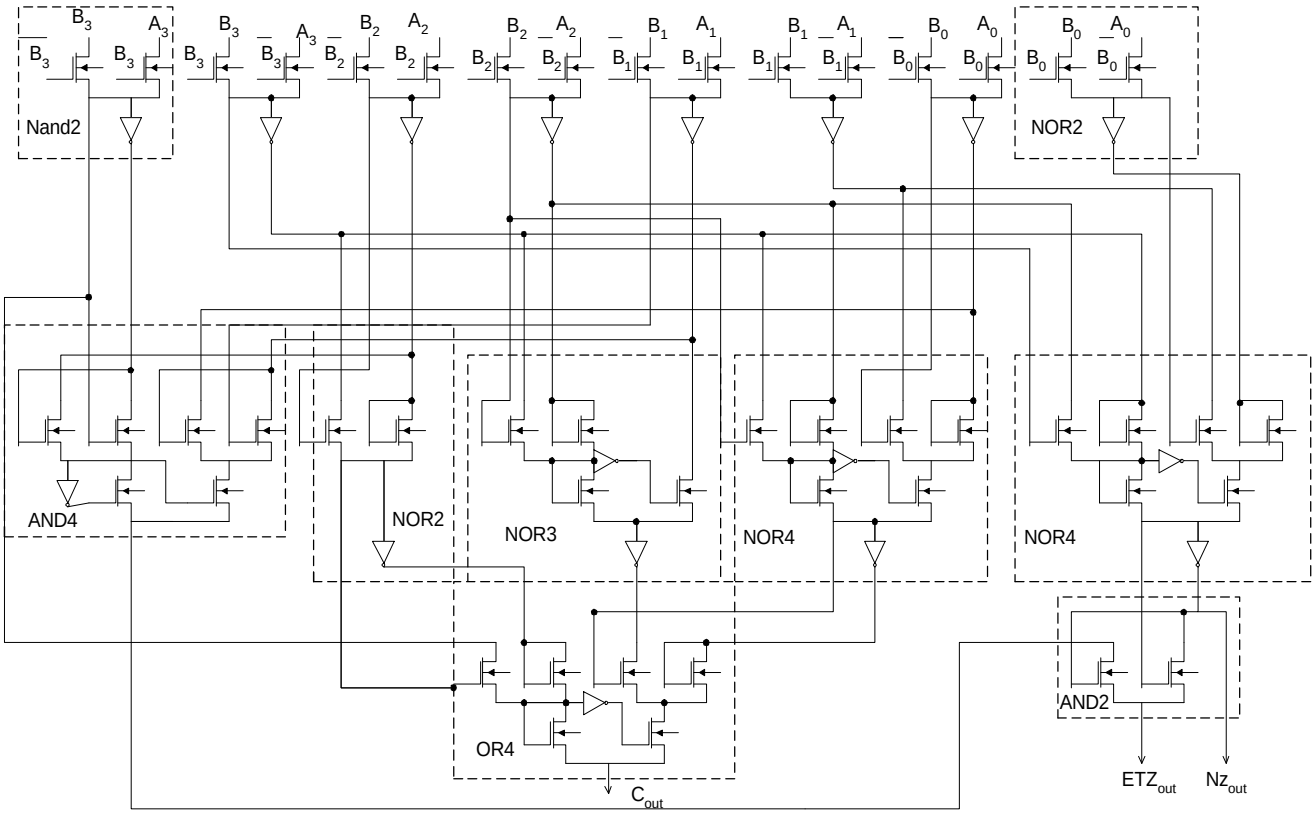

Fig. 2. *Transistor level schematic diagram of the 4-bit fast carry and equal to zero detector.*

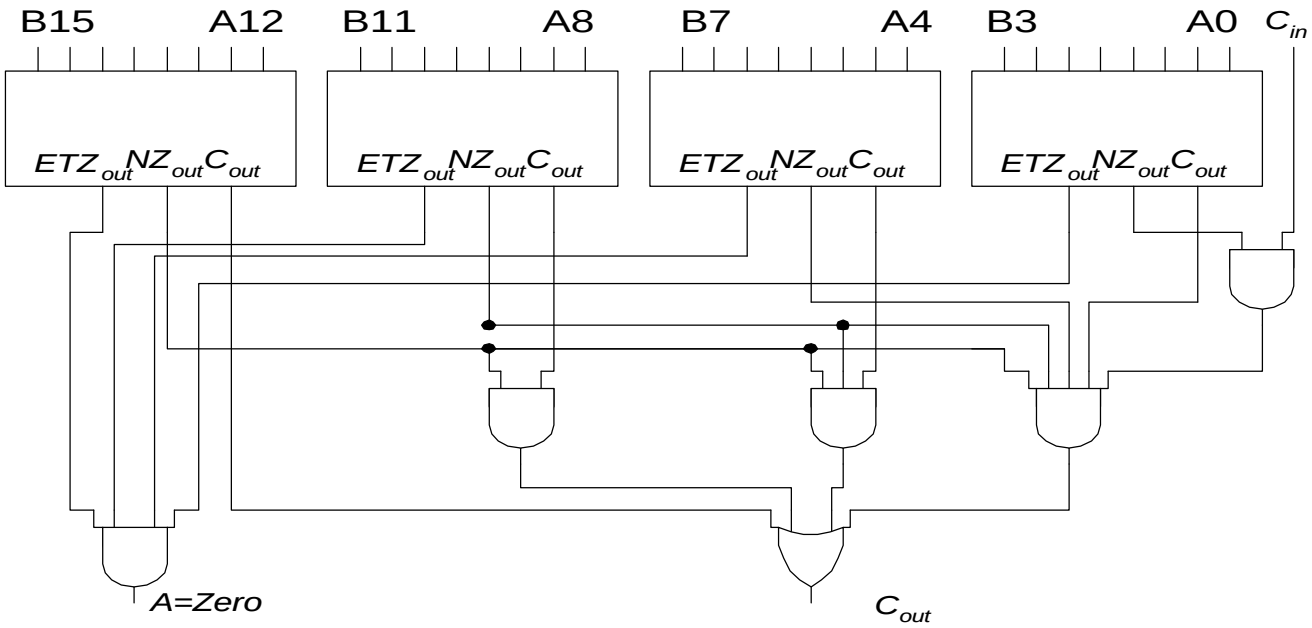

Fig. 3. *Cascading of 4-bit fast carry and equal to modules to obtain 16-bit module.*

4. VLSI Implementation

The VLSI implementation (layout) was done in order to compare the chip area required by [1] and the proposed circuit, and to obtain more realistic parameters (parasitic) for accurate HSPICE simulation. The complete chip layout is shown in Fig. 4. The simulations on the extracted layout using HSPICE for the case A=11111111, B=0000000x (b_0=1/0), and C_{in} = 0 are presented in Fig. 5. The delay and area of the two circuits are shown in Table 2. The simulations are performed for 2-µm n-well CMOS technology with a 2.5V supply, and at 25°C. The simulations show a 30% increase in speed and 25% decrease in area. It should be noted here that in [1] the circuit for the ETZ is not included.

Circuit	Area	Delay
[1]	417ux618	1.04
Propose	402ux509	0.78

Table 2. *Area and delay (with pad frame) of the chip for 8-bit operands.*

5. Comparison

Comparisons of the cost and delay of the four algorithms for the detection of C_{out} has been made by [2]. We have used the same criteria and values, to compare our design with the existing designs. Table 3 shows the asymptotic analysis, and Table 4 shows the analysis for n≤64. We have also simulated our design for n = 4, 16, 32, and 64 bits, the simulation results are presented in Table-5. The simulations are performed for 0.2-µm technology, with 2.5V supply and 25°C. These results show a 64-bit carry generation in 679*ps*.

Method	Delay	Gates
Carry-	2 log n + 1	5n-3
Brent	log n + O($\sqrt{\log n}$)	<7n
Bit prefix-and	log n + 3	<7n
Group prefix-	log n + 3	5n-4
Proposed	$2\lceil \log n/2 \rceil + 1$	4n

Table 3. *Asymptotic cost of different schemes for the detection of carry.*

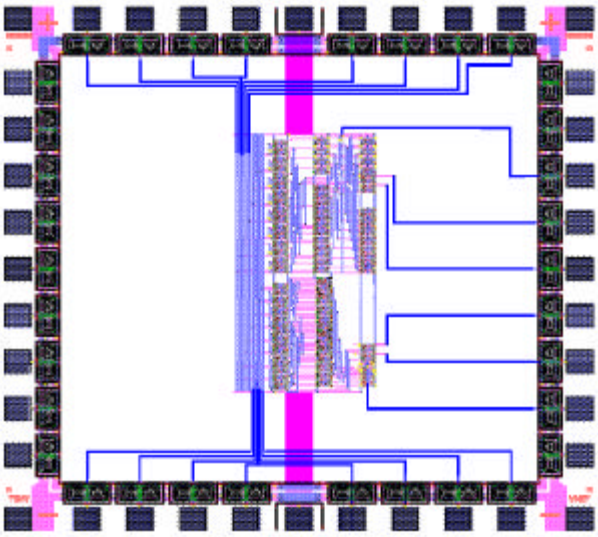

Fig. 4. *VLSI Layout of the chip in 2-mm n-well CMOS technology.*

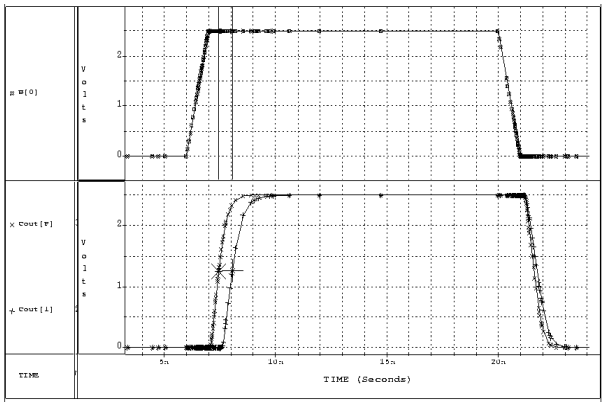

Fig. 5. *HSPICE Simulation results of the chip, top input, and bottom Cout[P] (proposed, legend*) and Cout[1](legend +).*

Method	n = 4		n = 16		N = 64	
	S	G	S	G	S	G
Carry-	5	1	9	7	1	3
Brent	5	1	8	7	1	3
Bit prefix-and	5	1	7	8	9	3
Group prefix-	5	1	7	7	9	3
Proposed	4	1	5	6	7	2

Table 4. *Number of stages(St.) and gates(Gt.) required by different schemes for the detection of carry.*

	n =	n=16	n=32	n=64
C_{out}	28	483	642	679
EQZ	27	344	408	418

Table 5. *HSPICE simulation results for n=4, 16, 32, and 64 bit.*

6. Conclusions

A fast and area efficient technique for the detection of A=B, A>B, and A<B has been developed, and implemented in VLSI. We have shown that this technique, out performs all the existing techniques in terms of area and speed. Moreover, this design calculates the ET condition without extra hardware. The high modularity of this technique makes it ideally suited for VLSI implementation. In this design the use of Xor and tri-state buffers, as done by [1,2,3] has been avoided.

References:

[1] David R. Lutz and D. N. Jayasimha, "The half adder form and early branch condition resolution", 13th Symposium on Computer arithmetic; 266-273, 1997.

[2] David R. Lutz and D. N. Jayasimha, "Comparison of two's complement numbers", International Journal of Electronics; 513-523, April 1996.

[3] Arnold Weinberger, "High Speed zero sum detection", 4th Symposium on Computer arithmetic; 200-207, 1975.

[4] Brent R. P., "On the addition of binary numbers", IEEE Transactions on Computer; 758-759, 1970.

[5] M. Putrino, and S. Vassiliades, "Resolution of branching with prediction", International Journal of Electronics; 163-172, February 1989.

[6] Jordi Cortadella, and M. Llaberia, "Evaluation of A+B=K conditions without carry propagate", IEEE Transactions on Computer; 1484-1488, 1982.

[7] Aamir A. Farooqui, "Early Branch Prediction", ACSEL Report #01-98, University of California, Davis, January 1998.

[8] Burns J. L. and K. J. Nowka, "Parallel condition code generation for high frequency PowerPC Microprocessors", 1998 Symposium on VLSI Technology and VLSI Circuits.