

# 8-Bit Borrow-Lookahead Subtractor For a Floating Point Adder

Guilherme Godoi, Fernando B. Rossarolla, André L. Aita  
{godoi,fbr}@nupedee.ufsm.br  
aaita@inf.ufsm.br

Universidade Federal de Santa Maria – UFSM  
Departamento de Eletrônica e Computação – DELC  
Faixa de Camobi Km 9, Campus Universitário – Santa Maria – RS – Brasil – 97015-900

## Abstract

*This paper describes the development of a 8-bit CMOS integrated subtractor, using the borrow-lookahead technique. The logical and electrical design, simulation and layout are shown. The Magic tool is used to edit the full-custom layout, in 2 $\mu$ m technology. This circuit is one functional block of the first stage of the pipeline of a floating point adder.*

## 1 Introduction

The addition of floating point numbers requires the alignment of the binary point. This alignment operation requires the significand shifting and the exponent adjustment. Using a magnitude comparator and multiplexers, always is executed the higher exponent minus the lower one. This positive difference is the number of bits that the significand of the lower number must be shifted to the right. Once the exponents are equal, the significands may be added or subtracted. The delay time of this process must be small, ensuring the speed of the first stage of the pipeline. Since the delay of the subtractor is higher than the magnitude comparator and the multiplexers, a fast subtractor must be designed to assure the speed of whole circuit. The development of this fast subtractor is described as following.

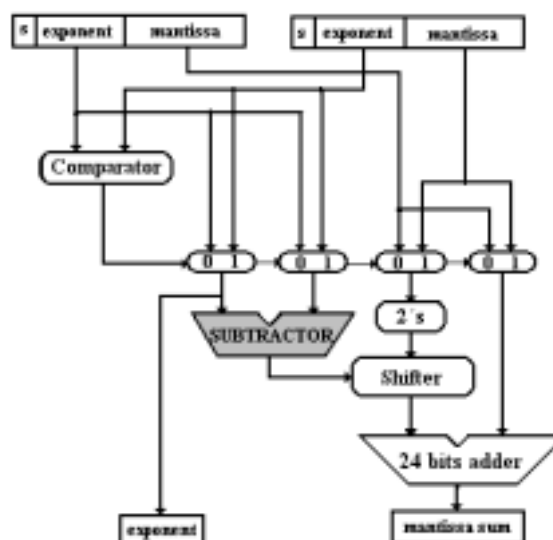


Figure 1 - Block diagram of the first stage of the floating point adder.

## 2 Logical Design

Two 8-bit binary numbers are subtracted column by column, using full subtractors. Each full subtractor performs the difference between the minuend  $X_i$  and the subtrahend  $Y_i$ , also accounting the *borrow-in*, as we move from right to left. The expressions that define the binary subtraction in each *full-subtractor* are:

$$D_i = X_i \oplus Y_i \oplus B_{i-1} \quad (1)$$

$$B_i = G_i + P_i \cdot B_{i-1} \quad (2)$$

where  $G_i = \bar{X}_i \cdot Y_i$ ,  $P_i = \bar{X}_i + Y_i$  and  $0 \leq i \leq 3$ .

The terms  $G_i$  and  $P_i$  are called generate and propagate functions, respectively. When  $G_i = 1$ , the  $i^{\text{th}}$  stage generates the borrow. If there is a borrow from the lower significant stage and  $P_i = 1$ , the borrow is propagated to the more significant one.

Equations (1) and (2) show that the difference  $D_i$  and the borrow  $B_i$  signals depend on the borrow of previous stage  $B_{i-1}$ . If the entire circuit is made by cascading the full subtractors (*ripple-borrow*), the delay would be too high. The adopted solution uses the **borrow-lookahead** technique, similar to the *carry-lookahead* one where the borrow bit necessary in each stage is determined direct from the inputs. Thus, the circuit speed increases. However, the circuit becomes more complex.

Consider the Eq. (2) to the least significant bits ( $X_0$  and  $Y_0$ ). There isn't borrow-in, which defines this single-bit subtractor as a *half-subtractor*. Then:

$$B_0 = G_0 \quad (3)$$

Making  $B_0$  equal to  $B_{i-1}$  of the next stage, and repeating the procedure for the rest of the bits, we have:

$$B_1 = G_1 + P_1 \cdot G_0 \quad (4)$$

$$B_2 = G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0) \quad (5)$$

$$B_3 = G_3 + P_3 \cdot (G_2 + P_2 \cdot (G_1 + P_1 \cdot G_0)) \quad (6)$$

Now  $B_i$  depends only on the generate and propagate bits, which are determined direct from the circuit inputs. Then, the cascading is eliminated. However, same procedure for  $B_4$  implies in logical gates with large fan-ins, which have high delays and questionable performance. To solve this problem, the circuit is separated in two 4-bits blocks, where the *borrow-out* of the first block is the *borrow-in* of the second. Thus, there is a *ripple-borrow* between the two blocks. Now the *borrow*s for the next stages are similar to the other ones:

$$B_4 = G_4 + P_4 \cdot B_3 \quad (7)$$

$$B_5 = G_5 + P_5.(G_4 + P_4 . B_3) \quad (8)$$

$$B_6 = G_6 + P_6.(G_5 + P_5 .(G_4 + P_4 .B_3)) \quad (9)$$

So, the complete circuit of the 8-bit subtractor can be constructed joining the *borrow* signal to each full subtractors.

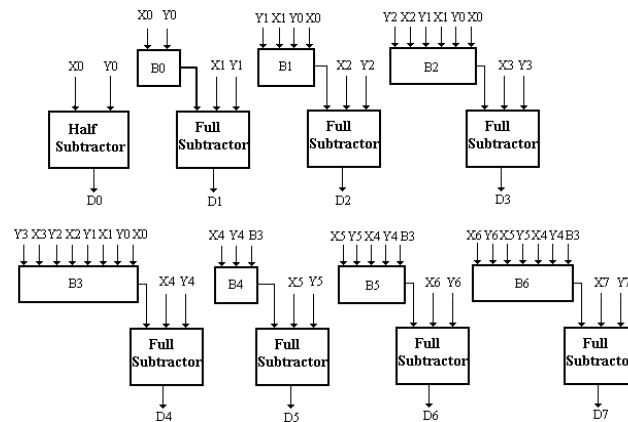


Figure 2 - Block diagram of the 8-bit subtractor.

### 3 Electrical Design

The *borrow* generator circuits were implemented with complex gates. This resulted in an optimized circuit with a small number of transistors and lower delay time. However, due to electrical performance, the limit of four transistors in series between Vdd/Gnd and output node must not be exceeded.

The *half-subtractor* (used in the first stage) and the full ones were implemented with XOR gates. The non-static logic was preferred in the XOR gate implementation due to the reduced number of transistors required. In the Fig. 3, the electrical schematic of both circuits can be shown.

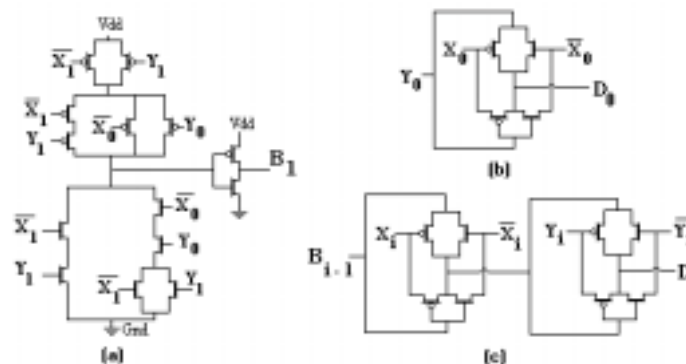


Figure 3 - a) Second stage borrow circuit, b) Half subtractor circuit, c) Full subtractor circuit.

### 4 Layout

The Magic tool was used in the subtractor layout edition. This tool has on-line design rule checking. The *full-custom* layout generates more compact circuits (higher integration density), but requires more layout edition time than the automatic layout methods.

The *borrow* generator and difference circuits, for each bit, are implemented separately, and then interconnected to make the final layout of the 8-bit subtractor. The circuit was composed by 220 transistors that occupy approximately  $133.400\mu\text{m}^2$  of silicon area.

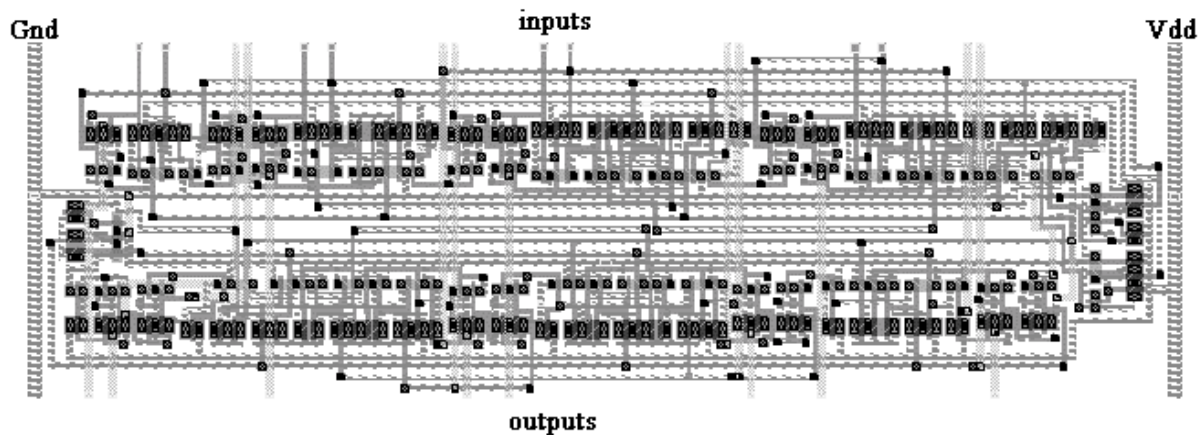


Figure 4 - 8-bit subtractor layout.

## 5 Simulations

The Magic tool also has a circuit extractor. The *netlist* extracted, including parasitic elements, was simulated with PSPICE to verify the functionality of the circuit and the delay of the operation. The circuit presented a correct behavior. With a capacitance load of 0.2 pF, the maximum delay observed was 3,54 ns.

## 6 Conclusions

The *borrow-lookahead* technique, used to implement the 8-bit exponent subtractor to be used in the first stage of a floating point adder, allowed a subtractor faster than other ones, like *ripple-borrow*. The employment of super-gates also contributed reducing delays and the number of transistors needed in the circuits.

Although the *full-custom* methodology was used, no great layout optimization was possible, because of the circuit irregularity. And, because of the large numbers of input and also due to the fact that the inputs are used in several circuits, the routing occupied a large area of the circuit. Simulations results showed the well functioning of the circuit.

## 7 References

- [MAR 00] MARTIN, K., *Digital Integrated Circuits Design*, Oxford University Press, Inc. New York, New York, 2000.
- [PAT 00] PATTERSON, D.A. and HENNESSY, J.L., *Organização e Projeto de Computadores – A Interface Hardware/Software*, Editora LTC, Rio de Janeiro, RJ, 2000.
- [REI 00] REIS, R.A.L., *Concepção de Circuitos Integrados*, Série Livros Didáticos, Número 7, 1º Edição, Instituto de Informática da UFRGS – Porto Alegre: Editora Sagra Luzzato, 2000.