A Logarithmic Shifter for a Floating-Point Adder

Fernando D. Franke, Diego S. Picada, André L. Aita
franke@nupedee.ufsm.br
spencer@nupedee.ufsm.br
aaita@inf.ufsm.br

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

Abstract

This paper describes the design stages, layout and simulation of a logarithmic shifter for a floating-point adder, using CMOS 2μm technology.

1 Introduction

Nowadays, the growing circuit integration has been allowed the development of smaller and faster systems. The high processing power usually means architectures with floating-point arithmetic units.

This work exposes the design of a logarithmic shifter trying to address a fast and efficient solution for the problem of the binary point alignment present in the digital floating-point addition/subtraction operations.

2 The Floating Point Addition

The representation of a floating-point number, in single precision (32 bits), according to the IEEE 754 standard, requires three parts: exponent (e), mantissa (m) and sign (s). The exponent, of 8 bits, is represented in excess 127, so it is always a binary positive number, which facilitate its manipulation. The mantissa (23 bits) is the fractional part of the number, and its sign is represented by the bit s. The decimal value of a normal number is $(-1)^s \cdot 2^{(e-127)} \cdot 1.m$, where 1.m is called the significand. In the floating-point addition, we should initially align the binary point. The alignment operation requires the mantissa shifting and exponents adjustment, in order to equal the exponent of the smallest number to the exponent of the largest one. The logarithmic shifter performs this operation.

With a magnitude comparator and multiplexers, the largest exponent is subtracted from the smallest one, always. The difference is equal to the number of bits that the mantissa of the smallest number should be shifted to the right. With equals exponents, the mantissas can be added.

A floating-point adder presents a higher design complexity when compared with a fixed-point adder. However, in applications where the magnitudes varies a lot and a reasonable precision is necessary, the addition using fixed point become inappropriate, because very long registers must be used to avoid large errors.

The circuit of a floating-point adder is composed of several functional units. Fig. 1 shows the hardware of the floating-point adder required to implement the first stage of this unit.

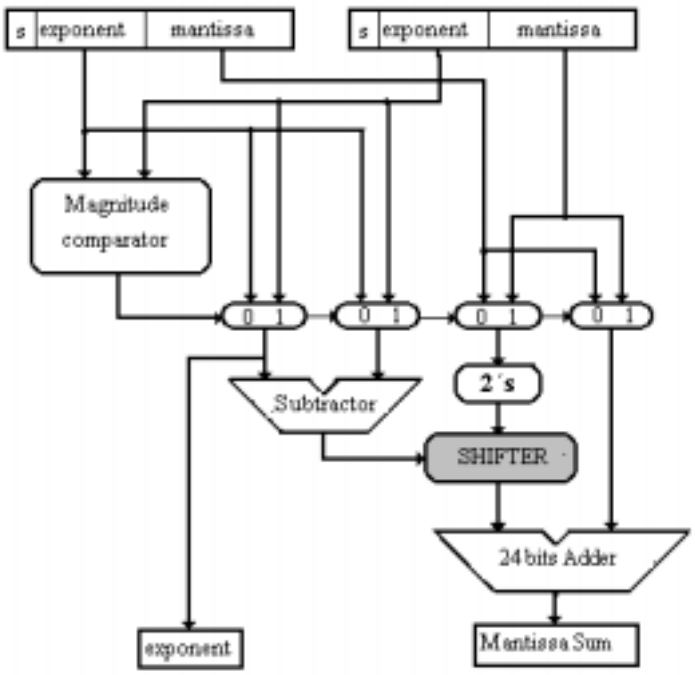

Figure 1 - Schematic of the first stage of the pipeline of a floating-point adder.

3 Description of the Shifter

A logarithmic shifter was chosen to compose the floating-point adder. In general, for small shifts, shifters as the barrel shifter are more efficient than the logarithmic shifter. However, when a long shift is necessary, as in the present case, the logarithmic shifter is more speed and area efficient, according to [RAB 96]. Besides, it does not need a decoder since the inputs that control the number of shifts are coded. This is allowed by the circuit structure, where the total shift value is decomposed into shifts over powers-of-two. Thus, the output of subtractor, the difference between the exponents, can be directly feed to the shifter control.

3.1 Electrical Circuit and Operation

The logarithmic shifter has five bits Shi, to control the number of shifts, where i represents the number of shifts, besides the main input and the shifted output, both 24 bits-wide.

In fig. 2, the electrical circuit of the shifter is shown (partial view). Four generic inputs Ix, Iy, Iz and Iw and corresponding outputs Ox, Oy, Oz, and Ow are detailed.

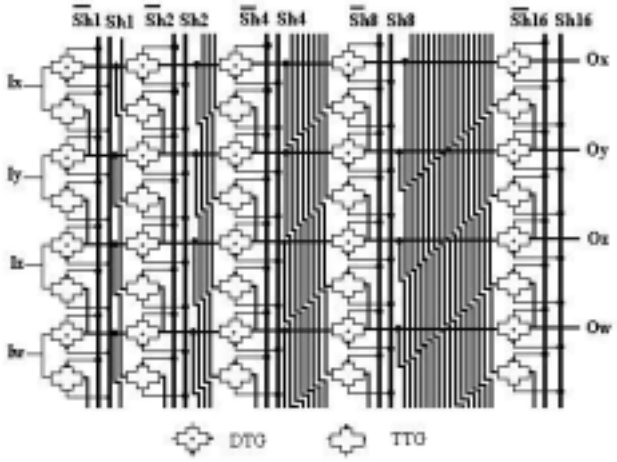

Figure 2 - Partial electrical circuit of logarithmic shifter.

The circuit allows a maximum shifting of 32 bits. However, for a single precision, 24 bits are enough. Beyond this value there is no more significant bits to shift.

The transmission gates define the signal path from input to output. The direct transmission gate (DTG) defines the direct path while the transversal transmission gate (TTG) enables the shifting operation in all shifter stages.

3.2 Layout

The shifter layout was edited with the Magic tool. The circuit regularity allowed a high area optimization. Unfortunately, the routing area was considerable. The shifter circuit has 418 transistors, which are arranged in an area of 395.109 μm2.

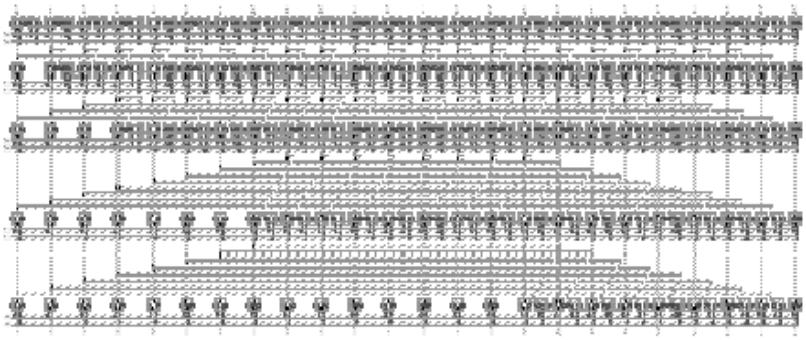

Figure 3 – Shifter layout.

4 Simulation

The electrical simulation was performed in order to verify the shifter operation and to analyze its performance. In one of the simulated cases, a pulsed signal was introduced at 23rd input keeping the other ones at 0V, while the control input assumed the following four cases:

C1 = 10111, 23 shifts;

C2 = 10110, 22 shifts;

C3 = 10101, 21 shifts and

C4 = 10100, 20 shifts.

The simulation results, detailing the least significant output bits, are shown in fig. 4. The correct operation of the circuit can be observed.

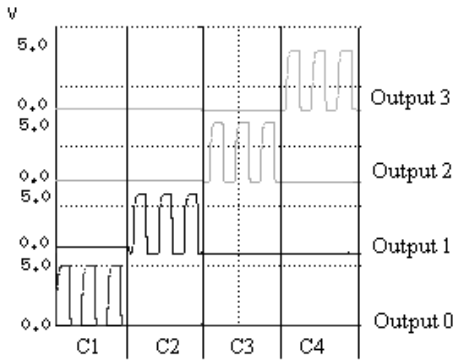

Figure 4 - Simulation results detailing the four least significant output bits.

The average propagation time evaluated by simulation was 1,70 ns, showing a nominal operation frequency about 250 MHz.

5 Conclusions

This paper presented the design of a logarithmic shifter to be used as a shifting unit in a floating-point adder. The shifting operation must be performed in order to align the binary point. Simulation results shown the correct operation and its good performance.

6 References

[PAT 00] PATTERSON, D. A.; J. L. HENNESSY, ***Organização e Projeto de Computadores – A Interface Hardware/Software***, Segunda Edição, Editora LTC, Rio de Janeiro, RJ, 2000.

[RAB 96] RABAEY J.M. ***Digital Integrated Circuits, A Design Perspective***, Prentice Hall: 1996, p. 416-417.

[REI 00] REIS, R. A. L., ***Concepção de Circuitos Integrados***, Série de Livros Didáticos, Número 7, Primeira Edição, Instituto de Informática da UFRGS – Porto Alegre, Editora Sagra Luzzato, 2000.