24-bit Bi-directional Programmable Logarithmic Shifter for Normalization of Floating Point Numbers

Fernando D. Franke, Diego S. Picada, André L. Aita
{franke, spencer}@mail.ufsm.br, aaita@inf.ufsm.br
Universidade Federal de Santa Maria – UFSM
Centro de Tecnologia, Departamento de Eletrônica e Computação – DELC

Abstract

This paper describes the design of a 24-bit bi-directional programmable shifter. Control signals determine the direction of shifting as well the number of left shifts. This shifter is required in the normalization step of floating point operations [PAT 96]. Electrical and logical simulations are presented and also the full-custom layout. The shifter allows a fast and efficient shifting operation.

1. Introduction

According to the IEEE 754/854 standard, a number represented in floating point format, single precision (32 bits) has three parts: exponent (*e*), mantissa (*m*) and signal (*sig*). The 8-bit exponent is represented in excess 127, so it is always a binary positive number, which facilitates its manipulation. The mantissa, with 23 bits, is the fractional part of the number. The number signal is represented by the signal bit *sig*. The equivalent number of a floating point has the following format $(-1)^{sig} \cdot 2^{(e-127)} \cdot 1.m$, where 1.m is the significand *S* of the number (24 bits). A floating point number that has its significand in format *S=1,m* is called normalized.

The sum or subtraction of two normalized floating-point numbers starts with the binary point alignment step which in fact de-normalizes one operand. After the addition or subtraction, the result will be in the general case non-normalized, so a normalization step is required to put the number according to the IEEE standard. The analysis of the present case showed that a single-right programmable left shifter is necessary to full-fill the requirements in this step.

2. Shifters

Shifters are functional units that can shift a binary word input. The direction of shifting and the number of shifts can be programmable.

There are several types of shifters. The more appropriate one depends one the case. In general, to perform small shifts, barrel shifters are more efficient than logarithmic ones. However when a high number of shifts is necessary, as in the case of a floating-point adder, the logarithmic shifter is preferred, in terms of area and speed [RAB 96]. This occurs because the circuit structure allows shifting in stages, which are increased in terms of powers of two. The name logarithmic shifter arose from this functional characteristic.

This way it was verified that the logarithmic shifter would be the ideal type for the adder case, so much for the exponents adjustment process, as for the results normalization process.

3. The Bi-directional Shifter

Depending on the result, the shifting operation of the normalization step can vary. In other words, the shifting direction and the number of shifts will differ according to the resulted number. Three distinct cases are identified and they are illustrated in the fig. 1.

The first is the case when a single right shift is necessary, since the sum resulted in a carry-out and the most significant bit C should be introduced in result significand.

Another situation occurs when there is no carry out and the most significant bit of the sum significand is 1. In this case shifting is not necessary because the result is already normalized.

The third case is identified when there is no carry-out and the most significant bit of sum significand is zero. So shifting to the left will be necessary to normalize the number. Since the position of the most significant bit of the sum significand can be anywhere along of it, a programmable left shift will be necessary.

The programming is obtained from the 24-bit leading bit detector [BAR 02], which informs the number of left shifts to implement. When left shifting, guard bit (*g*) and zeroes are introduced.

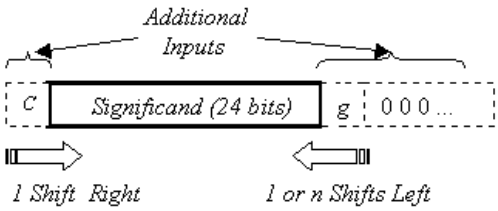

Fig. 1 – Shift cases and the additional input bits

3.1 Shifter Operation

The bi-directional programmable logarithmic shifter has two control bits, *D1* and *D0*. These signals determine the shifter operation mode: one right, one left, a variable left and no shifting mode. If a variable left shifting is defined, five auxiliary control inputs, *C0*, *C1*, *C2*, *C3* e *C4* specify the number of left shifts. The logarithmic shifter has also two additional

inputs, the left in (*Li*) and the right in (*Ri*), which are used to fill the empty positions. All these signals are shown in fig. 2.

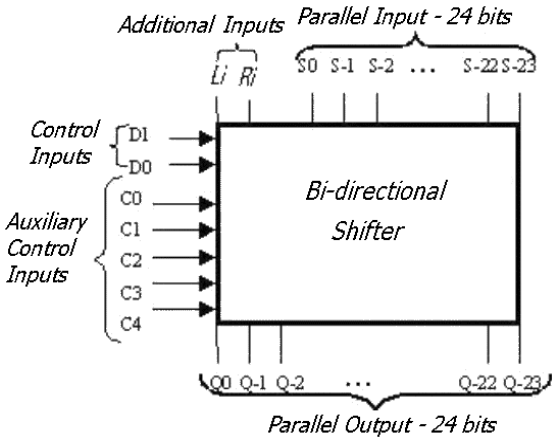

Fig. 2 - Logarithmic shifter block diagram

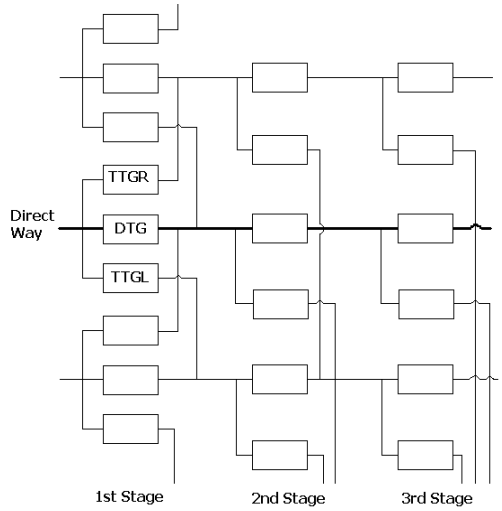

Fig. 3 - Partial shifter electrical schematic

The functional behavior of the circuit can be understood with fig. 3 that shows partially the shifter electrical schematic. Depending on the mode of operation, the transmission gates are opened and/or closed, defining the required input signal path along the shifter. The transmission gates can be classified according the path they enable. The transmission gates responsible for the direct path are called Direct Transmission Gates (*DTG*). Transversal Transmission Gates, Right or Left (*TTGR or TTGL* respectively), allow the signal shifting in any of the five stages of the shifter. All stages have the same structure, except the first one since in this stage both right and left shift are possible. Figure 4 details the partial layout of shifter, where the TGs can be clearly observed.

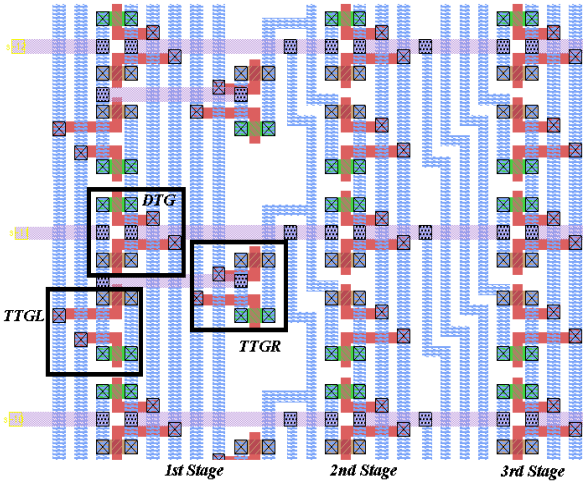

Fig. 4 - Partial layout of the logarithmic shifter, detailing the transmission gates and the metal connections.

There are three main cases to consider:

Right Shift: In this case the input signal is shifted one position to the right through the *TTGR*. In the other stages it flows directly through the DTGs. The carry-out of the sum is used as right input *Ri*.

Left Shift: In this case, the input signal is shifted one or "n" positions to left by the *TTGLs*. According to the number of shifts, bit *g* (through the left input *Li*) and zeroes are shifted in to fill the empty positions.

In each stage, the signal can run straightforward to the output or it can be shifted 1, 2, 4, 8 and/or 16 positions to the left depending on the programming. Although 32-position shifting is possible, only 24 are allowed by the actual shifter structure.

No Shift: In this case only the *DTGs* are enabled. So the inputs are not shifted and the output is equal to the input.

Tab. 1 – Shift cases and respective control inputs of each case.

Parallel Input - 24 bits					
S0	S-1	S-2...	S-21	S-22	S-23
a	b	c	x	y	z

Inputs									Outputs				Action
Control							Serial		Parallel 24-bits				
C4	C3	C2	C1	C0	D1	D0	Ri	Li	Q0	Q-1...	Q-22	Q-23	
1	0	1	1	0	0	0	X	0	y	z	0	0	n = 22 shift left
1	0	1	1	1	0	0	X	1	z	1	0	0	n = 23 shift left
X	X	X	X	X	0	1	X	1	b	c	z	1	1 shift left
X	X	X	X	X	0	1	X	0	b	c	z	0	1 shift left
X	X	X	X	X	1	0	X	X	a	b	y	z	No shift
X	X	X	X	X	1	1	1	X	1	a	x	y	1 shift right
X	X	X	X	X	1	1	0	X	0	a	x	y	1 shift right

3.2 Simulation Results

Simulations had shown severe signal degradation if single pass transistors are used instead of transmission gates. Considering the five stage shifter and a signal input of 5V, a voltage level of 3.74 V was observed with single pass transistors against 4.8 V observed when using transmission gates. So, with transmission gates, no buffering is necessary.

The logarithmic shifter circuit, designed with transmission gates, has 646 transistors while the same circuit designed with single pass transistors has only 382. An estimation of the number of transistors required in he case of a shifter using pass transistors instead of TGs and only one buffer stage, indicates an approximate number of 478 transistors.

Figure 5 shows some results from logical simulation. The input signal (where only the least significant (*S-23*) is one and *g*=0) suffers a variable (0 to 23) number of shifts. Other results also confirmed the correct operation of the shifter.

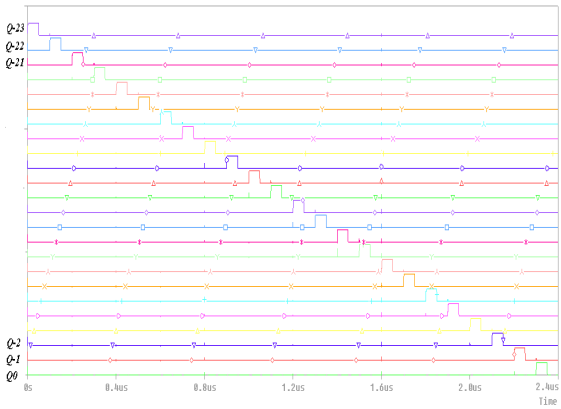

Fig. 5 - Outputs for successive shifts from 0 to 23.

Analyzing the input sign propagation time in relation to output, using a load of 1pF, the frequency operation for the shifter obtained was around of 70 MHz.

Figure 6 shows the shifter layout, edited with the *Magic* tool, using full-custom methodology and the 2µm technology. The layout occupies a core area of 535946 μm^2.

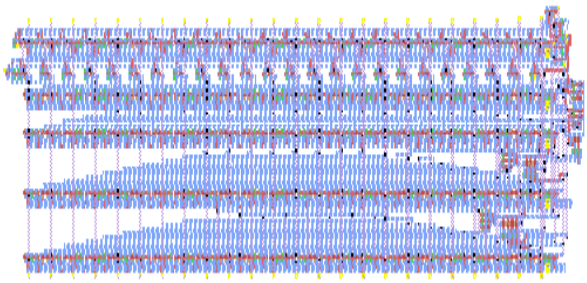

Fig. 6 - Bi-directional Logarithmic Shifter complete layout.

4. Conclusion

This paper presented the design, simulation and layout edition of a bi-directional programmable logarithmic shifter for normalization of floating point numbers.

The full-custom methodology was used to make the layout of the shifter, because it is ideal for circuits with great regularity, which is the present case. This allows a better use of the silicon area.

The simulations showed that, with transmission gates implementing the logarithmic shifter, the signal degradation could be ignored. So, buffers are avoided, which is not true if pass transistor were used.

5. Acknowledgments

This work is partially supported by CNPq and FAPERGS.

6. References

[BAR 02] BARROS, Taiser T. T.; AITA, André L. in **Leading Bit Position Detector for Normalization of Floating Point Numbers - A 24-bit Priority Encoder,** submitted to XVII SIM – South Symposium on Microelectronics.

[FRA 01] FRANKE, Fernando D.; PICADA, Diego S.; AITA, André L. in **A Logarithmic Shifter for a Floating-Point Adder**, pp. 37-40, Proceedings of the 1st Student Forum Microelectronics, Pirenópolis, Brazil, 2001.

[PAT 00] PATTERSON, D. A.; HENNESSY J. L. **Organização e Projeto de Computadores – A Interface Hardware/Software**, 2nd. edition, Ed. Livros Técnicos e Científicos, Rio de Janeiro, RJ, 2000.

[PAT 96] PATTERSON, D. A. **Computer Architecture: A Quantitative Approach**, J. L. Hennessy, 2nd. Edition, pp. A-13-A-28, Academic Press, 1996.

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