Floating Point Multiplier Project

Leandro César Paiva
DCCE/IBILCE/UNESP
Rua Cristovão Colombo, 2265
São José do Rio Preto - SP
+55 17 221-2200
paiva@dcce.ibilce.unesp.br

ABSTRACT

This document describes the development of a digital circuit to multiply single precision 32 - bit floating point binary numbers, according to IEEE 754 standard. This circuit was developed and simulated using the Altera Max+Plus II design environment, available in the scientific laboratory of DCCE (Departamento de Ciências da Computação e Estatística) of IBILCE (Instituto de Biociências, Letras e Ciências Exatas), campus of Unesp at São José do Rio Preto, Brazil.

INTRODUCTION

This work is being developed as the graduation project in the Computer Science course at DCCE/Ibilce/Unesp. Its goal is to develop a digital circuit to multiply single precision 32 - bit floating point binary numbers, according to IEEE 754 standard. The multiplier circuit when designed and simulated will be used in a reconfigurable digital system. The algorithm of floating point multiplication consists of multiplying the significants, add the exponents and verify the overflow occurrence. If the product of the significants is greater than or equal to 2, shift it one position to the right and increment the exponent by 1. It's important to remember to subtract 127 (the bias value) from the sum of the exponents, to normalize them.

DEVELOPED ACTIVITIES DESCRIPTION

The multiplier circuit can be divided in three main modules: sum of the exponents and eventual adjustments, multiplication of the input significants and the determination of the output signal. The circuit also verifies the occurrence of overflow. The computation of the resulting exponent of the floating point multiplication in the "EXP" module uses one adder and two multiplexers, that choose the correct inputs to be added in sequence. One external synchronous counter controls the multiplexers. When the counter is 0, the sum of the initial exponents occurs. Then, both multiplexers select the result of the first sum for bias subtraction and, thus, the exponent doesn't receive the *bias* value twice. Finally, if the multiplication of the significants results in a value larger than 2, the "EXP" module increments 1 to the value of the resulting exponent. Thus, the result is always in agreement to the required floating point characteristics. The *overflow* needs to be verified in two cases: *overflow* in the subtraction of the *bias* and *overflow* in the last sum of the adder. In the computation of the significant that results from the multiplication of two floating point numbers, both values of the mantissas are multiplied at "MULT_24". The result of this multiplication is stored in the bit vector "result_m[24..0]" and eventually shifted at "DESLOC" to be attributed as the output of the system. If "resul_m[24]" has the value "1", the value of "resul_m[24..0]" is shifted to the right and the exponent is incremented by 1 at "EXP_ACRES. The determination of the resulting signal of the floating point multiplier is done through the use of one "exclusive-or" gate. If both inputs, "sinal_A" e "sinal_B", have different logic values, the resulting signal will be 1, else it will be 0. Figure 1 shows the floating point multiplier with its main modules. Figure 1.a presents the sum of the exponents, the Figure 1.b illustrates the computation of the resulting significant and, finally, the Figure 1.c displays the determination of the resulting signal.

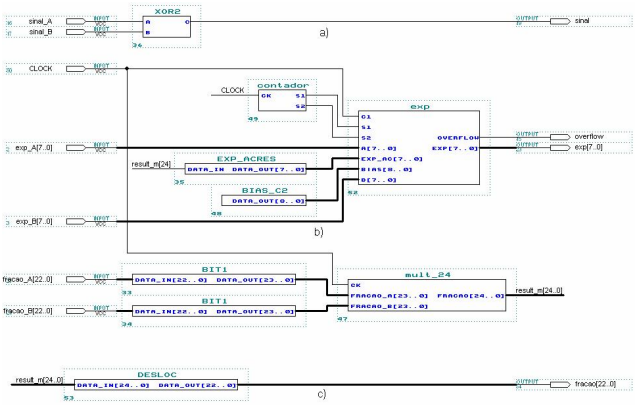

Figure 1 – Floating point multiplier

RESULTS ANALYSIS

The circuit described in the last section was compiled with Altera Max+Plus® II, without errors, using the device 'EPF10K30RC208-3'. The total logic cells and I/O pins used were 1660/1728 (96%) and 91/141 (64%), respectively.

One of the results of the exponent computation is presented in the remaining of this section. Two exponents were input: "exp_A[7..0]" received the value $(01111111)_2$ and the "exp_B[7..0]", the value $(00000000)_2$. The bias is already included in these values, The exponent computation, without

mantissa shift, is $(00000000)_2$. The corresponding computation is presented below.

$[(01111111)_2 - (127)_{10}] + [(00000000)_2 - (127)_{10}] = [(00000000)_2 - (127)_{10}]$.

CONCLUSIONS

This floating point multiplier circuit has been designed, simulated and will be concluded within the end of the year. To accomplish this, multiplier circuits will have to be improved and optimized before they can be included in the target reconfigurable digital system.

ACKNOWLEDGMENTS

L.C.Paiva would like to thank CNPq for the scholarship during part of the development of this work.

REFERENCES

[1] – M. M. Mano *Digital Design*, second edition, Prentice Hall, 516 pages, 1991.

[2] – Eletrônica Digital - Universidade Estadual de Maringá. Centro de Ciências Exatas. Departamento de Informática *http://www.din.uem.br/~edigital/*, visited in 07/20/2003

[3] - M. M. Mano e C. R. Kime *Logic and Computer Design Fundamentals*, Prentice Hall Inc., 1997.