32-bit FLOATING POINT MULTIPLIER IMPLEMENTATION

Version 1.0 (Wednesday, April 17, 2019)

PREFACE:

We[1] have created this multiplier to handle the multiplication process of two 32-bit *IEEE 754* based numbers as a part of *FPU project*[2].

---

[1] Team members:
- Hussien Mostafa
- Abdulrahman Ragab
- Abdullah Khaled
- Abdallah Mohamed
- Mahmoud Hassan

Supervisor:
- Dr. Muhammad Mahmoud Muhammad Ibrahim

[2] **FPU project:** the floating point unit to be used in our implementation of MIPS based processor

CONTENT:

CHAPTER 1:
"IEEE 754" REPRESENTATION

"This kind of refreshing part for the sake of comprehension."

FLOATING-POINT REPRESENTATION:

Floating-point numbers are usually a *multiple of the size of a word*. The representation of a MIPS floating-point number is shown below, where s is the sign of the floating-point number (1 meaning negative), exponent is the value of the 8-bit exponent field (including the sign of the exponent), and fraction is the 23-bit number.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s				exponent											fraction																

1 bit   8 bits   23 bits

In general, floating-point numbers are of the form

$$(-1)^S \times F \times 2^E$$

These formats go beyond MIPS. They are part of the IEEE 754 floating-point standard, found in virtually every computer invented since 1980. This standard has greatly improved both the ease of porting floating-point programs and the quality of computer arithmetic.

Before we go on board we have to take these notes with us:

1- Number must be normalized in an understandable language (1.01000 * 2^E not 101.000* 2^E-3)
2- Exponents are biased by 127 which means 0 = 127 to get the the real exponent form IEEE754 one subtract 127 from it.

CHAPTER 2:
OUR IMPLEMENTATION JOURNEY