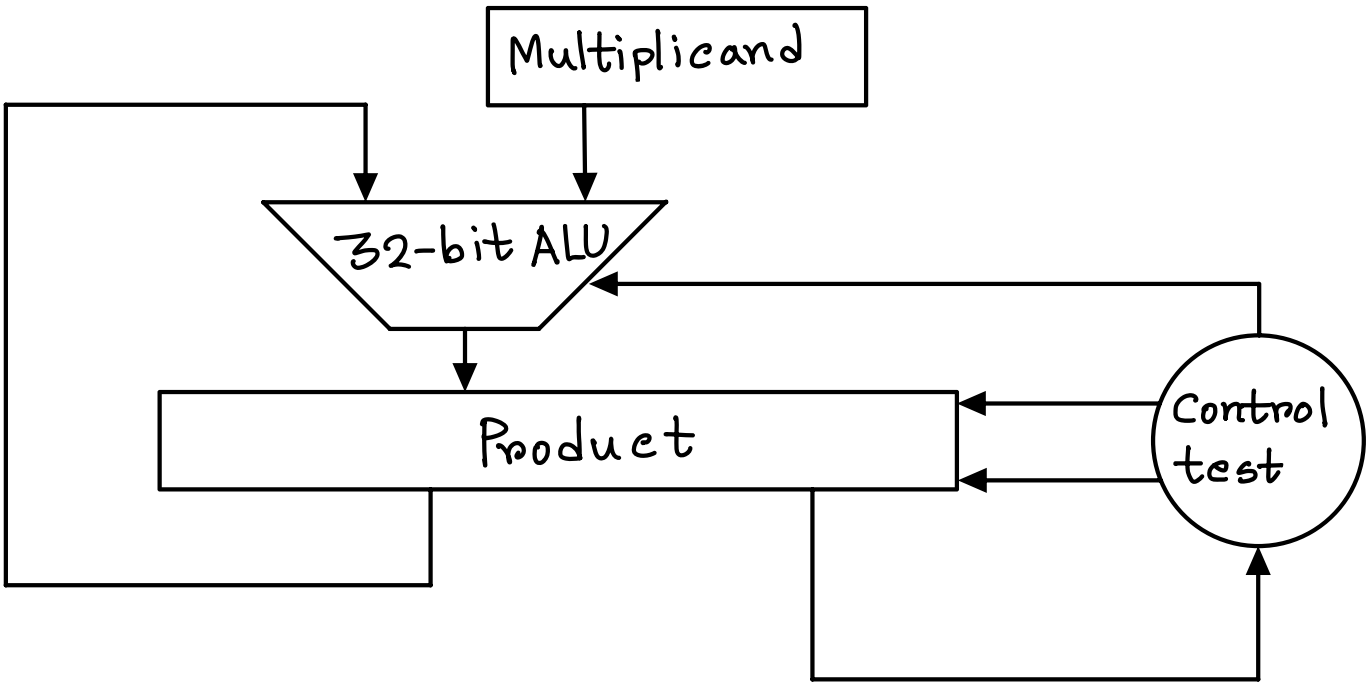
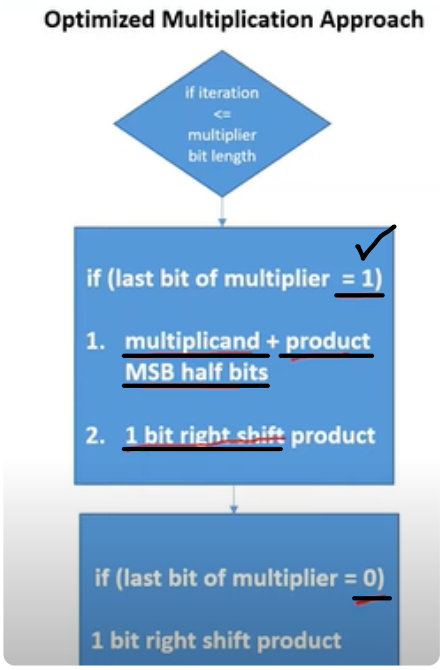
Optimized Multiplier



8 * 9 = 72

Iteration	Multiplicand 1000	Product 00001001
1	1000	10001001 01000100
2	1000	00100010
3	1000	0001 0001
4	1000	1001 0001 0100 1000



32 bit architecture

Product = 64 bit

mult rs, rt

multu rs, rt

mfhi rd

mflo rd

mul rd, rs, rt ---------- mult, mflo

Floating Point Representation

684.32×10^{2} Not normalized

$6.8432 \times 10^{2+2}$ Normalized

0.578×10^{3} Not normalized

$5.78 \times 10^{3-1}$ Normalized

1010.0100×2^{3}

$1.010\ 0010 \times 2^{3+3}$

Floating Point Standard

- Defined by IEEE Std 754-1985
- Developed in response to divergence of representations
  - Portability issues for scientific code
- Now almost universally adopted
- Two representations
  - Single precision (32-bit)
  - Double precision (64-bit)

Normalized Number

- Only One and Non-Zero number before .(decimal point)

5.64×10^{33} → Normalized

$109.64 \times 10^{33} \rightarrow 1.0964 \times 10^{33+2}$
1.0964×10^{35}

The number of times we left shift the (.), will be added with the exponent

$0.675 \rightarrow 6.75 \times 10^{-1}$

The number of times we right shift the (.), will be subtracted from the exponent

- Only One and Non-Zero number before .(binary point)

$1011.1101 \times 2^{33} \rightarrow 1.011101 \times 2^{33+3} = 1.011101 \times 2^{36}$

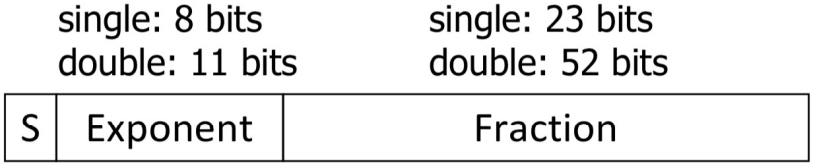
In Binary the Base is 2

$0.0111101 \times 2^{-5} \rightarrow 1.11101 \times 2^{-5-2} = 1.11101 \times 2^{-7}$

Decimal to Floating Point Conversion

- Step 1: Convert the Decimal Number into Binary Number
- Step 2: Normalize the Binary Number
- Step 3: Find out the Biased Exponent
- Step 4: Find out Sign bit and Fraction
- Step 5: Write the Sign bit, Biased Exponent and Fraction in IEEE-754 Floating Point Representation

IEEE Floating-Point Format



$$x = (-1)^S \times (1 + \text{Fraction}) \times 2^{(\text{Exponent} - \text{Bias})}$$

- S: sign bit (0 ⇒ non-negative, 1 ⇒ negative)
- Normalize significand: 1.0 ≤ |significand| < 2.0
  - Always has a leading pre-binary-point 1 bit, so no need to represent it explicitly (hidden bit)
  - Significand is Fraction with the "1." restored
- Exponent: excess representation: actual exponent + Bias
  - Ensures exponent is unsigned
  - Single: Bias = 127; Double: Bias = 1203

Single Precision (32 bit)

Sign bit	Exponent (Biased)	Fraction/ Mantissa
1 bit	8 bit	23 bit

If n = bit length of Exponent Field,
Bias = $2^{(n-1)}-1$

Sign Bit: (0 ⇒ Positive, 1 ⇒ Negative)

Exponent:

8 bit unsigned binary Range= 0 to $2^8 - 1$ = 0 to 255

Exponents 00000000 and 11111111 reserved, So the Range for Biased Exponent Becomes = **1 - 254**

For Exponent being 8 bit, Bias = $2^{(8-1)} - 1 = 127$

For Single Precision

Biased Exponent = Actual Exponent of the Binary number + Bias (127)

Range for Exponent = $2^{-126} - 2^{127}$

1.11101×2^{35}

Biased Exponent = 35 + 127
= 162 = 10100010

1.11101×2^{-8}

Biased Exponent = -8 + 127
= 119
=01110111

$-126+127=1$

$127+127=254$

Example

- **Convert 50.6749 to 32 bit IEEE-754 Floating Point Representation**
- **Step -1 Convert the Decimal Number To Binary Number**

50.67490

Binary of 50 = 110010

Binary of .6749 = 1010110011

Binary of 50.6749 = 110010.1010110011

Binary of .6749

= .6749 x 2 = 1.3498 =1
= .3498 x 2 = 0.6996 =0
= .6996 x 2 = 1.3992 =1
= .3992 x 2 = 0.7984 =0
= .7984 x 2 = 1.5968 =1
= .5968 x 2 = 1.1936 =1
. = 0
. = 0
. = 1
. = 1

Atleast 10 times

- **Step -2 Normalize the Binary Number**

Binary of 50.6749 = 110010.1010110011 x 2^0

Normalized Binary Number =

$1.100101010110011 \times 2^5$

Fraction

- **Convert 50.6749 to 32 bit IEEE-754 Floating Point Representation**

Normalized Binary Number = $1.100101010110011 \times 2^{5}$ (3)

- **Step -3 Find Out The Biased Exponent**

Exponent = 5

Biased Exponent = 5+127

=132

=10000100

- **Step -3 Find Out Sign Bit and Fraction**

Sign Bit = 0

Fraction= 100101010110011 00000000

- **Step -4 IEEE-754 Floating Point Representation**

(Biased)

Sign bit	Exponent	Fraction/ Mantissa
1 bit	8 bit	23 bit
0	10000100	10010101011001100000000

IEEE-754 Floating Point Representation of 50.6749

01000010010010101011001100000000

= 0100 0010 0100 1010 1011 0011 0000 0000

= **0x424AB300**

Double Precision (64 bit)

(Biased)

Sign bit	Exponent	Fraction/ Mantissa
1 bit	11 bit	52 bit

If n = bit length of Exponent Field,
Bias = $2^{(n-1)}-1$

Sign Bit: (0 ⇒ Positive, 1 ⇒ Negative)

Exponent:

11 bit unsigned binary Range= 0 to $2^{11} - 1$

Exponents 00000000 and 11111111 reserved, So the Range for Biased Exponent Becomes = **1 - 2046**

For Exponent being 11 bit, Bias = $2^{(11-1)} - 1 = 1023$

For Double Precision

Biased Exponent = Actual Exponent of the Binary number + Bias (1023)

- **Convert -0.232 to 12 bit IEEE-754 Floating Point Representation, Where Exponent is 4 bit**

(Biased)

Sign bit	Exponent	Fraction/ Mantissa
1 bit	4 bit	7 bit

Binary of 0.232 = 0.00111011

Normalized Binary of 0.232 = 1.11011×2^{-3}

Exponent:

For Exponent being 4 bit, Bias = $2^{(4-1)} - 1 = 7$

If n = bit length of Exponent Field, Bias = $2^{(n-1)}-1$

Exponent = -3

Biased Exponent = -3+7 =4 = 0100

Sign Bit and Fraction:

Sign Bit = 1

Fraction= 1101100

Floating Point Representation

1 0100 1101100
1010 0110 1100 = 0xA6C

Hexadecimal

- Base 16
  - Compact representation of bit strings
  - 4 bits per hex digit

0	0000	4	0100	8	1000	c	1100
1	0001	5	0101	9	1001	d	1101
2	0010	6	0110	a	1010	e	1110
3	0011	7	0111	b	1011	f	1111

- Example: eca8 6420
  - 1110 1100 1010 1000 0110 0100 0010 0000

Floating Point (Single Precision) to Decimal

- 0xF2400240

Sign bit	Exponent (Biased)	Fraction/ Mantissa
1 bit	8 bit	23 bit

Step 1: Hexadecimal to Binary

1111 0010 0100 0000 0000 0010 0100 0000

Step 2: Set the Binary Number as Format

1 11100100 10000000000001001000000

Step 3: Find Out Exponent and Fraction

Biased Exponent =11100100

Biased Exponent (Decimal)=228

Exponent (Decimal)= 228 - 127

=101

For Exponent being 8 bit, Bias = $2^{(8-1)}-1 = 127$

Fraction/ Mantissa = 0.10000000000001001000000

$= 2^{-1}+2^{-14}+2^{-17}$

= 0.500068664

Connected!

$$\text{Decimal Value} = (-1)^{\text{SignBit}} \times (1+\text{Fraction}) \times 2^{(\text{Exponent})}$$

$(-1)^1$ x (1+0.500068664) x 2^{101}

= - 1.500068664 x 2^{101}

= - 3.803125885 x 10^{30}

Upto 5 decimal point with Rounding= - 3.803126 x 10^{30}

Upto 5 decimal point without Rounding= - 3.803125 x 10^{30}