DATA FORMATS OF 8087

The data types supported by 8087 are as follows:

Integers	Decimal	Floating Point Numbers
1. Word Integer	1. Packed BCD	1. Short Real
2. Short Integer		2. Long Real
3. Long Integer		3. Temporary Real

Integers

- Here **MSB** represents the **sign** of the number.
- The **remaining** bits are used to represent the **magnitude** of the number.
- For **-ve numbers** the magnitude is stored in **2's complement form**.
- There are **three** data **types Word Int**., **Short Int**. and **Long Int**. of sizes 2 Bytes, 4 Bytes and 8 bytes respectively.

Packed BCD

- **Total size** of the number is **10 Bytes** (80 bits).
- Here a **number** is represented as a **series of 18 Packed BCD digits i.e. 9 bytes**.
- Hence each byte has two BCD digits.
- The **MSB of** the **10th byte** carries the **sign bit**, the **remaining 7 bits** are "**don't care**".
- Here a **–ve no** is **NOT stored** in its **2's complement form**.

Floating Point Numbers

- In some numbers, which have a fractional part, the position of the decimal point is not fixed as the number of bits before (or after) the decimal point may vary.
- **Eg: 0010.01001, 0.0001101, -1001001.01** etc.
- As shown above, the position of the decimal point is not fixed, instead it **"floats"** in the number. **For doubts contact Bharat Sir on 98204 08217**
- Such numbers are called Floating Point Numbers.
- Floating Point Numbers are stored in a "Normalized" form.

Normalization of a Floating Point Number

- Normalization is the process of shifting the point, left or right, so that there is only one non-zero digit to the left of the point. #Please refer Bharat Sir's Lecture Notes for this ...

Eg:

Floating Point Number		Normalized Number
0010.01001	⋯⋯➤	$1.001001*2^{1}$
0.0001101	⋯⋯➤	$1.101*2^{-4}$
-1001001.01	⋯⋯➤	$-1.00100101*2^{-6}$

- As seen above a Normalized number is represented as:

$(-1^{S}) \times (1.\mathbf{M}) \times (2^{E})$ Where: S = Sign, M = Mantissa and E = Exponent.

- As Normalized numbers are of the 1.M format, the "**1**" is not actually stored, it is instead **assumed**. This saves the storage space by 1 bit for each number.
- Also the Exponent is stored in the biased form by adding an appropriate bias value to it so that -ve exponents can be easily represented.

Advantages of Normalization.

1. Storing all numbers in a standard for makes **calculations easier** and **faster**.
2. By **not storing** the **1** (of 1.M format) for a number, considerable **storage space** is **saved**.
3. The **exponent** is **biased** so there is **no need** for **storing** its **sign bit** (as the biased exponent cannot be -ve).

There are three data types for Floating Point Numbers supported by 8087:

1. Short Real Format

- **32 bits** are used to store the **number**.
- **23 bits** are used for the **Mantissa**.
- **8 bits** are used for the Biased **Exponent**.
- **1 bit** used for the **Sign** of the number.
- The **Bias** value is $(\mathbf{127})_{10.}$
- The range is $\pm\mathbf{1*10^{-38}}$ **to** $\pm\mathbf{3*10^{38}}$ approximately.
- It is called as the **Single Precision Format** for Floating-Point Numbers.

2. Long Real Format

- **64 bits** are used to store the **number**.
- **52 bits** are used for the **Mantissa**.
- **11 bits** are used for the Biased **Exponent**.
- **1 bit** used for the **Sign** of the number.
- The **Bias** value is $(\mathbf{1023})_{10.}$
- The range is $\pm\mathbf{10^{-308}}$ **to** $\pm\mathbf{10^{308}}$ approximately.
- It is called as the **Double Precision Format** for Floating-Point Numbers.

3. Temporary Real Format

- **80 bits** are used to store the **number**.
- **64 bits** are used for the **Mantissa**.
- **15 bits** are used for the Biased **Exponent**.
- **1 bit** used for the **Sign** of the number.
- The **Bias** value is $(\mathbf{16383})_{10.}$
- The range is $\pm\mathbf{10^{-4932}}$ **to** $\pm\mathbf{10^{4932}}$ approximately.
- **1** (of 1.M format) is **present at 63rd bit**, hence the decimal point is assumed between the 62nd and the 63rd bit. **For doubts contact Bharat Sir on 98204 08217**
- **8087 stores** numbers **internally** in **this format** as it has the **biggest range**.
- It is also called as **Extended Precision Format** or the **internal format** of 8087.

Short Real – 32 bit format – Bias value 127

S	E	M
Sign (1)	Biased Exponent (8)	Mantissa (23)

Long Real – 64 bit format – Bias value 1023

S	E	M
Sign (1)	Biased Exponent (11)	Mantissa (52)

Temp Real – 80 bit format – Bias value 16383

S	E	M
Sign (1)	Biased Exponent (15)	Mantissa (64)