

## 8087 Instruction Set

The instruction set of 8087 starts with F, stands for floating point. The instruction of 8087 numeric data processor can be classified into following six groups:

1. Data transfer instructions
2. Arithmetic instructions
3. Compare Instructions
4. Transcendental instructions
5. Load constant instructions
6. Processor control instructions

### 1 Data Transfer Instructions

#### (a) Real Transfers

| S. No. | Instruction      | Description with example                                                                                                                                                                                                                                                                                                                                                    |
|--------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1      | FLD source       | <p>Decrements stack pointer by one and copies a real number from a stack element or memory to the new ST. A short – real or long-real number from memory is automatically converted to temporary real format by the 8087 before it is put in ST.</p> <p>Examples:</p> <p>FLD ST(2) ; Copies ST(2) to ST</p> <p>FLD [BX] ; Number from memory pointed by BX copied to ST</p> |
| 2      | FST Destination  | <p>Copies ST to a specified stack position or to a specified memory location.</p> <p>Examples:</p> <p>FST ST(3) ; Copy ST to ST(3)</p> <p>FST [BX] ; Copy ST to memory pointed by [BX]</p>                                                                                                                                                                                  |
| 3      | FSTP destination | <p>Copies ST to a specified stack element or memory location and increments stack pointer by one to point to the next element on the stack. This is a stack POP operation.</p>                                                                                                                                                                                              |

|   |                  |                                                                                                                                                                                    |
|---|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4 | FXCH destination | Exchanges contents of ST with the contents of a specified stack element. If no destination is specified, then ST(1) is used.<br><br>Example:<br><br>FXCH ST(4) ; Swap ST and ST(4) |
|---|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

#### (b) Integer transfers

| S. No. | Instruction       | Description with example                                                                                                                                                                                              |
|--------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5      | FILD source       | Integer load. Converts integer number from memory to temporary real format and pushes converted number on 8087 stack.<br><br>Example:<br><br>FILD DWORD PTR [BX] ; Short integer from memory location pointed by [BX] |
| 6      | FIST destination  | Integer store. Converts number from ST to integer form, and copies to memory.<br><br>Example:<br><br>FIST INT_NUM ;ST to memory locations named INT_NUM                                                               |
| 7      | FISTP destination | Integer store and pop. Similar to FIST except that stack pointer is incremented after copy.                                                                                                                           |

#### (c) Packed Decimal Transfers

| S. No. | Instruction | Description with example                                                                                                                                                   |
|--------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8      | FBLD source | Packed decimal (BCD) load. Convert number from memory to temporary-real format and push on top of 8087 stack.<br><br>Example:<br><br>FBLD AMOUNT ;Ten byte BCD number from |

|   |       |                                                                                                                                                                                                                                             |
|---|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   |       | memory location AMOUNT to ST                                                                                                                                                                                                                |
| 9 | FBSTP | BCD store in memory and pop 8087 stack. Pops temporary – real from stack, converts to 10-byte BCD, and stores result to memory.<br><br>Example:<br><br>FBSTP MONEY ; Contents from top of stack are converted to BCD, and stored in memory. |

## 2. Arithmetic Instructions

| S. No. | Instruction                     | Description with example                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1      | FADD<br>destination,<br>source  | Will add real number from specified source to real number at specified destination. Source can be stack element or memory location. Destination must be a stack element. If no source or destination is specified, then ST is added to ST(1) and the stack pointer is incremented so that the result of the addition is at ST.<br><br>Examples:<br><br>FADD ST(2), ST ; Add ST to ST(2), result in ST(2)<br><br>FADD ST, ST(5) ; Add ST(5) to ST, result in ST<br><br>FADD SUM ; Real number from memory + ST<br><br>FADD ; ST + ST(1), pop stack-result at ST |
| 2      | FADDP<br>destination,<br>source | Adds ST to specified stack element and increments stack pointer by one.<br><br>Example:<br><br>FADDP ST(2) ; Add ST(2) to ST<br><br>; Increment stack pointer so ST(2)<br>; becomes ST                                                                                                                                                                                                                                                                                                                                                                         |

|   |                                 |                                                                                                                                                                                                                                                                                                                                                                            |
|---|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3 | FIADD source                    | <p>Adds integer from memory to ST, Stores the result in ST.</p> <p>Example:</p> <p>FIADD CARS_SOLD ;Integer number from memory<br/>+ ST</p>                                                                                                                                                                                                                                |
| 4 | FSUB<br>destination,<br>source  | <p>Subtracts the real number at the specified source from the real number at the specified destination and puts the result in the specified destination.</p> <p>Examples:</p> <p>FSUB ST(3), ST ; ST(3) <math>\leftarrow</math> ST(2) – ST</p> <p>FSUB DIFFERENCE ; ST <math>\leftarrow</math> ST-real from memory</p> <p>FSUB ; ST <math>\leftarrow</math> (ST(1)-ST)</p> |
| 5 | FSUBP<br>destination,<br>source | <p>Subtracts ST from specified stack element and puts result in specified stack element. Then increments stack pointer by one.</p> <p>Examples:</p> <p>FSUBP ST(2) ; ST(2) – ST . ST(1) becomes new ST.</p>                                                                                                                                                                |
| 6 | FISUB source                    | <p>Subtracts integer number stored in memory from ST and stores result in ST.</p> <p>Example:</p> <p>FISUB DIFFERENCE ; ST <math>\leftarrow</math> ST-integer from memory</p>                                                                                                                                                                                              |
| 7 | FSUBR<br>destination,           | <p>These instructions operate same as FSUB instructions discussed earlier except that these instructions subtract the</p>                                                                                                                                                                                                                                                  |

|    |                                  |                                                                                                                                                                                                                                                                                                                                                                                          |
|----|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | source                           | contents of the specified destination from the contents of the specified source and put the difference in the specified destination.                                                                                                                                                                                                                                                     |
| 8  | FSUBRP<br>destination,<br>source | [Normal FSUB instruction subtracts source from destination.]                                                                                                                                                                                                                                                                                                                             |
| 9  | FISUBR source                    |                                                                                                                                                                                                                                                                                                                                                                                          |
| 10 | FMUL<br>destination,<br>source   | <p>Multiply real number from source by real number from specified destination, and put result in specified stack element.</p> <p>Examples:</p> <p>FMUL ST(2), ST ; Multiply ST(2) and ST, result in ST(2)</p> <p>FMUL ST, ST(5) ; Multiply ST(5) to ST, result in ST</p>                                                                                                                 |
|    | FMULP<br>destination,<br>source  | <p>Multiplies the real number from specified source by real number from specified destination, puts result in specified stack element, and increment stack pointer by one. With no specified operands FMULP multiplies ST(1) by ST and Pops stack to leave result at ST.</p> <p>Example:</p> <p>FMULP ST(2) ; Multiply ST(2) to ST.<br/>increment stack pointer so STI(1) becomes ST</p> |
| 11 | FIMUL source                     | <p>Multiply integer from memory at ST and put result in ST.</p> <p>Example:</p> <p>FIMUL DWORD PTR [BX]</p> <p>;Integer number from memory pointed by BX x ST and</p>                                                                                                                                                                                                                    |

|    |                                 |                                                                                                                                                                                                                      |
|----|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                                 | result in ST                                                                                                                                                                                                         |
| 12 | FDIV destination,<br>source     | <p>Divides destination real by source real, stores result in destination.</p> <p>Example:</p> <p>FDIV ST(2), ST ; Divides ST by ST(2)</p> <p>; stores result in ST</p>                                               |
| 13 | FDIVP<br>destination,<br>source | <p>Same as FDIV, but also increments stack pointer by one after DIV</p> <p>Example:</p> <p>FDIV ST(2), ST ; Divides ST by ST(2), stores result in ST and increments stack pointer</p>                                |
| 14 | FIDIV source                    | <p>Divides ST by integer from memory, stores result in ST.</p> <p>Example:</p> <p>FIDIV PERCENTAGE; ST ← ST/integer number</p>                                                                                       |
| 15 | FDIVR<br>destination,<br>source |                                                                                                                                                                                                                      |
| 16 | FDIVP<br>destination,<br>source |                                                                                                                                                                                                                      |
| 17 | FIDIVR source                   | <p>These three instructions are identical in format to the FDIV, FDIVP and FIDIV instructions above except that they divide the source operand by the destination operand and put the result in the destination.</p> |

|    |         |                                                                                                                                                                                                                                                                                     |
|----|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 18 | FSQRT   | Contents of ST are replaced with its square root.<br><br>Example:<br><br>FSQRT                                                                                                                                                                                                      |
| 19 | FSCALE  | Scales the number in ST by adding an integer value in ST(1) to the exponent of the number in ST. Fast way of multiplying by integral powers of two.                                                                                                                                 |
| 20 | FPREM   | Partial remainder. The contents of ST(1) are subtracted from the contents of ST over and over again until the contents of ST are smaller than the contents of ST(1)<br><br>Example:<br><br>FPREM                                                                                    |
| 21 | FRNDINT | Round number in ST to an integer. The round – control (RC) bits in the control word determine how the number will be rounded.                                                                                                                                                       |
| 22 | FXTRACT | Separates the exponent and the significant parts of a temporary real number in ST. After the instruction executes, ST contains a temporary – real representation of the significant of the number and ST(1) contains a temporary real representation of the exponent of the number. |
| 23 | FABS    | Replaces ST by its absolute value. Instruction simply makes sign positive.                                                                                                                                                                                                          |
| 24 | FCHS    | Complements the sign of the number in ST.                                                                                                                                                                                                                                           |