MATH CO-PROCESSOR 8087

Gursharan Singh Tatla
professorgstatla@gmail.com
**INTRODUCTION**

- 8087 was the first math coprocessor for 16-bit processors designed by Intel.
- It was built to pair with 8086 and 8088.
- The purpose of 8087 was to speed up the computations involving floating point calculations.
- Addition, subtraction, multiplication and division of simple numbers is not the coprocessor’s job.
- It does all the calculations involving floating point numbers like scientific calculations and algebraic functions.
INTRODUCTION

- By having a coprocessor, which performs all the calculations, it can free up a lot of CPU’s time.

- This would allow the CPU to focus all of its resources on the other functions it has to perform.

- This increases the overall speed and performance of the entire system.

- This coprocessor introduced about 60 new instructions available to the programmer.

- All the mnemonics begin with “F” to differentiate them from the standard 8086 instructions.

- For e.g.: in contrast to ADD/MUL, 8087 provide FADD/FMUL.
**INTRODUCTION**

Math coprocessor is also called as:

- Numeric Processor Extension (NPX)
- Numeric Data Processor (NDP)
- Floating Point Unit (FPU)
ARCHITECTURE OF 8087

- 8087 coprocessor is designed to operate with 8086 microprocessor.
- The microprocessor and coprocessor can execute their respective instructions simultaneously.
- Microprocessor interprets and executes the normal instruction set and the coprocessor interprets and executes only the coprocessor instructions.
- All the coprocessor instructions are ESC instructions, i.e. they start with “F”.
ARCHITECTURE OF 8087

Control Unit
- Control Word
- Status Word
- Data Buffer
- New Instruction
- Data
- Status
- Address
- Addressing & Bus Tracking
- Exception Pointers

Numeric Execution Unit
- Exponent Module
- Exponent Bus
- Interface
- Fraction Bus
- Programmable Shifter
- Arithmetic Module
- Temporary Registers
- Tag Word
- Register Stack (0-7)

80 Bit
ARCHITECTURE OF 8087

The internal structure of 8087 coprocessor is divided into two major sections:

- Control Unit (CU)
- Numerical Execution Unit (NEU)
**CONTROL UNIT (CU)**

- It interfaces coprocessor to the microprocessor system bus.
- It also synchronize the operation of the coprocessor and the microprocessor.
- This unit has a Control Word, Status Word and Data Buffer.
- If an instruction is ESC instruction, then coprocessor executes it.
- If not, then microprocessor executes it.
**NUMERIC EXECUTION UNIT (NEU)**

- This unit is responsible for executing all coprocessor instructions.
- It has an 8 register stack that holds the operands for instructions and result of instructions.
- The stack contains 8 registers that are 80-bits wide.
- Numeric data is transferred inside the coprocessor in two parts:
  - 64-bit mantissa bus
  - 16-bit exponent bus
STATUS REGISTER
**Status Register**

- Status Register tells the overall status of 8087 coprocessor.
- It is a 16-bit register.
- It is accessed by executing the FSTSW instruction.
- This instruction stores the contents of status register into memory.
- Once the status is stored in memory, the bit positions of the status register can be examined.
STATUS REGISTER

- **Busy**: It indicates that the coprocessor is busy executing the task.

- **Condition Codes (C₀-C₃)**: They indicate various conditions about the coprocessor.

- **Top of Stack**: It indicates a register as top of stack register, out of the eight stack registers.

- **Exception Flag**: It is set if any of the exception flag bits (SF, PR, UF, OF, ZD, DN, IO) are set.
**Status Register**

- **Stack Fault**: It is not available in 8087. It is active only in 80387 and above.

- **Precision**: It indicates that the result has exceeded the selected precision.

- **Underflow**: It tells if the result is too small to fit in a register.

- **Overflow**: It tells if the result is too large to fit in a register.
**STATUS REGISTER**

- **Zero Divide**: It indicates that you try to divide a non-zero value by zero.

- **Denormalized**: It indicates that at least one of the operand is de-normalized.

- **Invalid Operation**: It indicates an invalid operation. For e.g.: pushing more than eight items onto the stack, attempting to pop an item off an empty stack or taking the square root of a negative number.
CONTROL REGISTER

Round:
00 - To nearest or even
01 - Down
10 - Up
11 - Truncate result

Precision Control:
00 - 24 bits
01 - reserved
10 - 53 bits
11 - 64 bits

Exception Masks:
Precision
Underflow
Overflow
Zero Divide
Denormalized
Invalid Operation

Reserved
Control Register controls the operating modes of 8087.

It is also a 16-bit register.

It performs rounding control and precision control.

It is also used to do masking and unmasking of the exception bits that correspond to the rightmost six bits of the status register.

FLDCW instruction is used to load the value into control register.
CONTROL REGISTER

- **Rounding Control:** It determines the type of rounding or truncating to be done.

- **Precision Control:** It sets the precision of the result.

- **Exception Masks:** It determines whether an error affects the exception bits in the status register.
  
  - If it is one, then the corresponding error is ignored.
  
  - If it is zero and the corresponding error occurs, then it generates an interrupt, and the corresponding bit in status register is set.
## Tag Register

<table>
<thead>
<tr>
<th>TAG 7</th>
<th>TAG 6</th>
<th>TAG 5</th>
<th>TAG 4</th>
<th>TAG 3</th>
<th>TAG 2</th>
<th>TAG 1</th>
<th>TAG 0</th>
</tr>
</thead>
</table>

**Tag Values:**
- 00 = Valid
- 01 = Zero
- 10 = Invalid
- 11 = Empty
Tag Register

- Tag Register is used to indicate the contents of each register in the stack.
- There are total 8 tags (Tag 0 to Tag 7) in this register and each tag uses 2 bits to represent a value.
- Therefore, it is a 16-bit register.

Tag Values:
- 00 = Valid
- 01 = Zero
- 10 = Invalid
- 11 = Empty
PIN DIAGRAM OF 8087

GND 1 40 V_{CC}
AD_{14} 2 39 AD_{15}
AD_{13} 3 38 A_{16}/S_3
AD_{12} 4 37 A_{17}/S_4
AD_{11} 5 36 A_{18}/S_5
AD_{10} 6 35 A_{19}/S_6
AD_9 7 34 BHE/S_7
AD_8 8 33 RQ/GT_1
AD_7 9 32 INT
AD_6 10 31 RQ/GT_0
AD_5 11 30 NC
AD_4 12 29 NC
AD_3 13 28 \overline{S_2}
AD_2 14 27 \overline{S_1}
AD_1 15 26 \overline{S_0}
AD_0 16 25 QS_0
NC 17 24 QS_1
NC 18 23 BUSY
CLK 19 22 READY
GND 20 21 RESET

www.eazynotes.com
INTERFACING OF 8086 AND 8087

- Multiplexed address-data bus lines are connected directly from 8086 to 8087.

- The status lines and the queue status lines are connected directly from 8086 to 8087.

- The Request/Grant (RQ/GT\(_0\) and RQ/GT\(_1\)) signals of 8087 are connected to RQ/GT\(_0\) and RQ/GT\(_1\) of 8086.

- BUSY signal of 8087 is connected to TEST pin of 8086.
Thank You 😊

Have a Nice Day