CSE2006
Microprocessor & Interfacing

Module – 6
Co-Processor

Dr. E. Konguvel

Assistant Professor (Sr. Gr. 1),

Dept. of Embedded Technology,

School of Electronics Engineering (SENSE),

konguvel.e@vit.ac.in

9597812810

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Syllabus

CSE2006	MICROPROCESSOR AND INTERFACING	L	T	P	J	C
		2	0	2	4	4
Pre-requisite	**CSE2001-Computer Architecture and Organization**	**Syllabus version**				
		v1.1				

Course Objectives:

1. Students will gain knowledge on architecture, accessing data and instruction from memory for processing.
2. Ability to do programs with instruction set and control the external devices through I/O interface
3. Generate a system model for real world problems with data acquisition, processing and decision making with aid of micro controllers and advanced processors.

Expected Course Outcome:

1. Recall the basics of processor, its ways of addressing data for operation by instruction set.
2. Execute basic and advanced assembly language programs.
3. Learn the ways to interface I/O devices with processor for task sharing.
4. Recall the basics of co-processor and its ways to handle float values by its instruction set.
5. Recognize the functionality of micro controller, latest version processors and its applications.
6. Acquire design thinking capability, ability to design a component with realistic constraints, to solve real world engineering problems and analyze the results.

Syllabus

Student Learning Outcomes (SLO):	**2, 5, 9**	
Module:1	**INTRODUCTION TO 8086 MICROPROCESSOR**	**6 hours**
Introduction to 8086, Pin diagram, Architecture, addressing mode and Instruction set		
Module:2	**INTRODUCTION TO ALP**	**5 hours**
Tools- Assembler Directives, Editor, assembler, debugger, simulator and emulator. E.g., ALP Programs-Arithmetic Operations and Number System Conversions, Programs using Loops, If then else, for loop structures		
Module:3	**Advanced ALP**	**2 hours**
Interrupt programming using DOS BIOS function calls, File Management		
Module:4	**Introduction to Peripheral Interfacing-I**	**5 hours**
PPI 8255, Timer 8253,Interrupt controller-8259		
Module:5	**Introduction to Peripheral Interfacing-II**	**4 hours**
IC 8251 UART, Data converters (A/D and D/A Converter), seven segment display and key- board interfacing		

Syllabus

Module:6	Co-Processor	4 hours
Introduction to 8087, Architecture, Instruction set and ALP Programming		

Module:7	Introduction to Arduino Boards	2 hours
Introduction to Microcontroller- Quark SOC processor, programming, Arduino Boards using GPIO (LED, LCD, Keypad, Motor control and sensor), System design application and case study.		

Module:8	Contemporary issues	2 hours
Architecture of one of the advanced processors such as Multicore, Snapdragon, ARM processor in iPad		

Text Book(s)	
1.	A.K. Ray and K.M. Bhurchandi Advanced Microprocessors and Peripherals, third Edition, Tata McGraw Hill, 2012.
2.	Barry B Bray , The Intel Microprocessor 8086/8088, 80186,80286, 80386 and 80486 Arcitecture, programming and interfacing, PHI, 8th Edition, 2009.
Reference Books	
1.	Douglas V. Hall, SSSP Rao Microprocessors and Interfacing Programming and Hardware. Tata McGraw Hill, Third edition, 2012.
2.	Mohamed Rafiquazzaman, Microprocessor and Microcomputer based system design, Universal Book stall, New Delhi, Second edition, 1995
3.	K Uday Kumar, B S Umashankar, Advanced Micro processors IBM-PC Assembly Language Programming, Tata McGraw Hill, 2002.
4.	Massimo Banzi,Getting Started with Arduino , First Edition, pub. O'Reilly, 2008.
5.	John Uffenbeck and 8088 Family. 1997. The 80x86 Family: Design, Programming, and Interfacing (2nd ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.
Mode of Evaluation: CAT / Assignment / Quiz / FAT / Project / Seminar	

Syllabus

List of Challenging Experiments (Indicative)		
1.	Arithmetic operations 8/16 bit using different addressing modes.	2.5 hours
2.	Finding the factorial of an 8 /16 bit number.	2.5 hours
3.	(a) Solving nCr and nPr (b) Compute nCr and nPr using recursive procedure. Assume that n and r are non-negative integers	2.5 hours
4.	Assembly language program to display Fibonacci series	2.5 hours
5.	Sorting in ascending and descending order	2.5 hours
6.	(a) Search a given number or a word in an array of given numbers. (b) Search a key element in a list of n 16-bit numbers using the Binary search algorithm.	2.5 hours
7.	To find the smallest and biggest numbers in a given array.	2.5 hours
8.	ALP for number system conversions.	2.5 hours
9.	(a) String operations(String length, reverse, comparison, concatenation, palindrome)	2.5 hours
10.	ALP for Password checking	2.5 hours
11.	Convert a 16-bit binary value (assumed to be an unsigned integer) to BCD and display it from left to right and right to left for specified number of times	2.5 hours
12.	ALP to interface Stepper motor using 8086/ Intel Galileo Board	2.5 hours
	Total Laboratory Hours	30 hours

Module 6: Co-Processor

- **Introduction**
- 8087 Numeric Data Processor
- Block Diagram
- Pin Description
- Interfacing 8087 with 8086
- Data Formats
- Instruction Sets
- Assembly Language Programs

Introduction

- The general-purpose microprocessors have an upper limit of data-processing capability and these general processors require complex programming to perform any mathematical calculations.
- For any mathematical operations, processors use high-level language for programming and a library of floating point objects has to be obtained from the manufacturer.
- To increase the operating speed, several microprocessors are connected together using a certain network topology, called as multiprocessor system.
- The simplest multiprocessors system consists of a microprocessor and a Numeric Data Processor (NDP).

Introduction

- The numeric data processor has an independent math processing unit and it can do complex numeric calculations very fast compared to the main processor.
- The most commonly used numeric processors are 8087, 80287 and 80387.
- The 8087 was first released in 1980 by Intel and this can work with 8086, 8088, 80186 and 80188 processors.
- After introducing 80286 CPU, Intel developed a redesigned 80287 NDP to operate with 80286 and 80386 processors.
- To improve the performance of 80386, the 80387 NDP was developed.
- The Intel 80486 and Pentium chips provide high performance as a numeric data processor is directly in built in the CPU.

Module 6: Co-Processor

- Introduction
- **8087 Numeric Data Processor**
- Block Diagram
- Pin Description
- Interfacing 8087 with 8086
- Data Formats
- Instruction Sets
- Assembly Language Programs

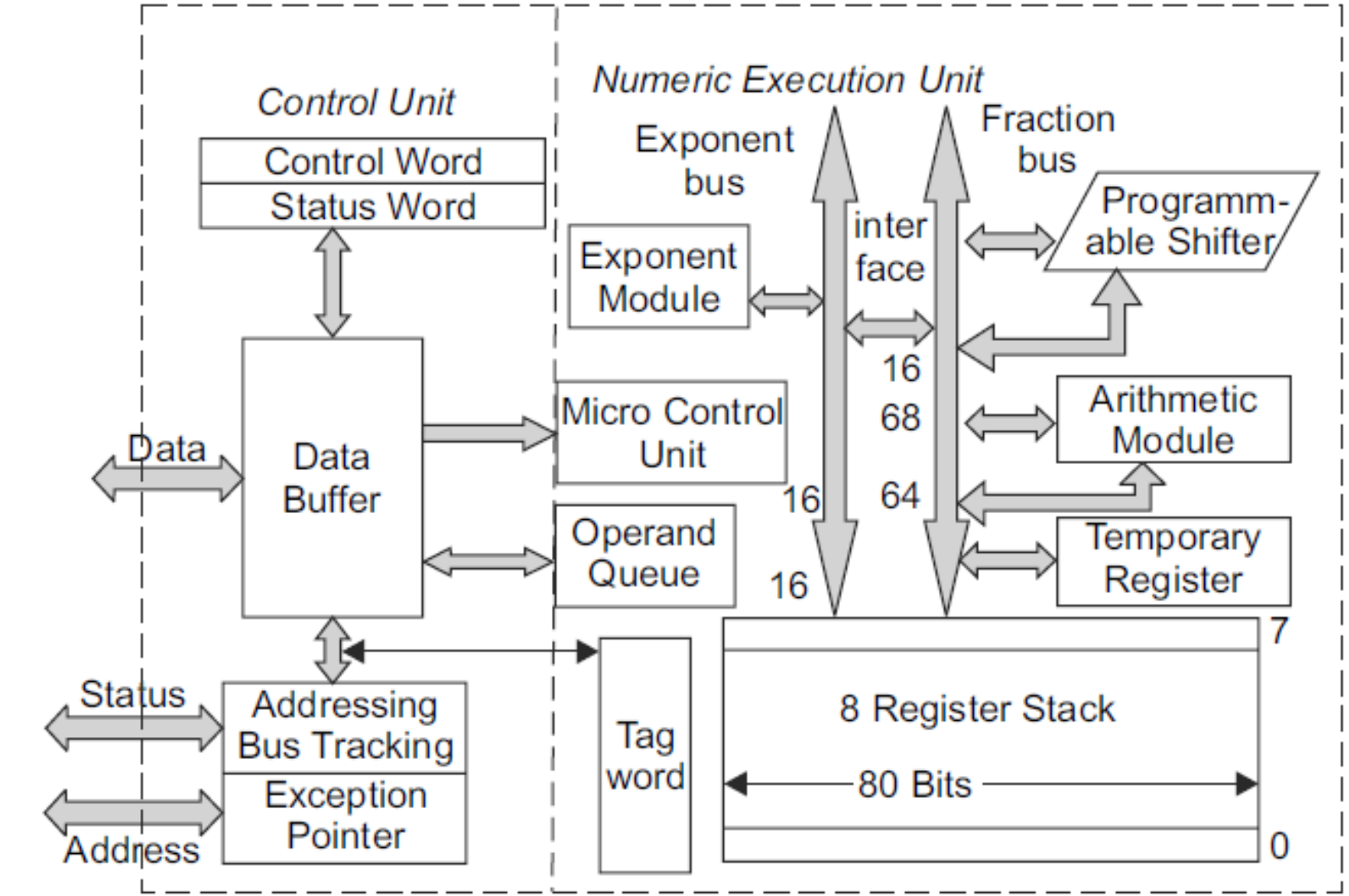
8087 NDP

- The 8087 Numeric Data Processor (NDP) is called a high-speed math co-processor.
- This math co-processor is also known as Numeric Processor Extension (NPX) or Numeric Data Processor (NDP) or Floating Unit Point (FUP).
- The 8087 is available in 40-pin DIP packages in 5 MHz, 8 MHz, and 10 MHz versions and it is compatible with 8086 and higher-version processors.
- 8086 performs the Opcode fetch cycles and identify and allot instructions for 8087.
- 8086-8087 couplet implements instruction level master-slave configuration.
- 8087 adds 68 new instructions to the instruction sets of 8086.

Module 6: Co-Processor

- Introduction
- 8087 Numeric Data Processor
- **Block Diagram**
- Pin Description
- Interfacing 8087 with 8086
- Data Formats
- Instruction Sets
- Assembly Language Programs

Block Diagram



Block Diagram

Control Unit:

- Used to synchronize the operation between the main processor and co-processor.
- Receives the instruction opcode, and then it decodes and reads or writes operands from memory.
- Provides the communication between the processor and memory, and it also coordinates the internal coprocessor execution.
- Continuously monitors the data bus to find instructions for the 8087 co-processor.

Block Diagram

Control Unit:

- Maintains a parallel queue just like the queue of the main processors.
- Monitors the BHE/S7 line to detect the processor type that is either 8086 or 8088.
- For the 8086 processor, the queue length will be 6 bytes, but for 8086 processor, the queue length will be 4 bytes.
- 8087 uses the queue status input pins QS0 and QS1 to identify the instructions fetched by the main processor.
- All instruction codes of 8087 have 11011 as the most significant bits of their first code byte.
- Main processor identifies the co-processor instruction using the ESCAPE code.
- Once the main processor recognizes the ESCAPE code, it sends a trigger signal so that the execution of the numeric processor instruction starts in 8087.

Block Diagram

Control Unit:

- During execution, the ESCAPE code identifies the co-processor instruction which can be operated with or without a memory operand.
- If the co-processor instruction requires a memory operand that will be fetched from memory then the physical address of the memory location will be computed using any one of the addressing modes in 8086 and a dummy read cycle must be initiated by the main processor.
- Thereafter, the 8087 co-processor reads the operand and proceeds for execution.
- If the co-processor instruction does not require any operand then the instruction can be directly executed.
- After execution of the instruction, the 8087 is ready with the execution results and the control unit obtains the control of the bus from 8086 and executes a memory write cycle to write the results at the specified memory location.
- The control unit consists of a control word, a status word and a data buffer which are explained later.

Block Diagram

Numeric Execution Unit (NEU):

- The Control Unit (CU) consists of a data bus buffer, status and control register and the Numeric Execution Unit (EU) has eight data register stacks, microcode control unit and a programmable shifter.
- These units duplicate the functions performed by the microprocessor control and ALU blocks.
- The 8087 NEU and CU can work independently.
- The CU works to maintain synchronization with the main 8086/8088 processor while the NEU is performing numeric operations.
- The numeric extension unit performs all operations that access and manipulate the numeric data in the co-processor's registers.
- In NEU, the numeric registers are 80 bits wide and the numeric data is routed by a 64-bit mantissa bus and a 16-bit sign/exponent bus.
- The Numeric Execution Unit (NEU) executes all numeric processor instructions such as arithmetic, logical, transcendental and data-transfer instructions.

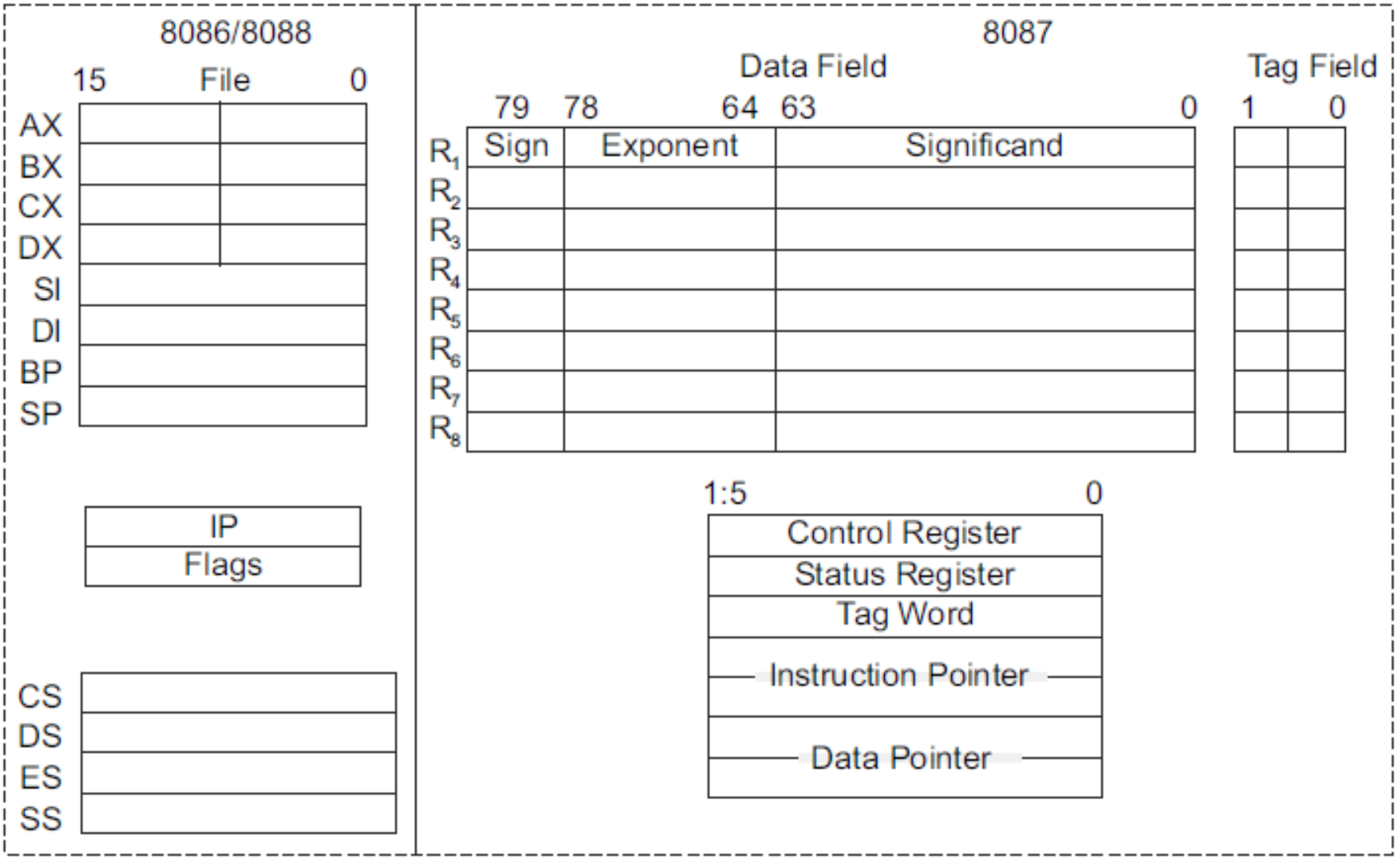
Block Diagram

Numeric Execution Unit (NEU):

- The Operation of CU and NEU is asynchronous with each other.
- The internal data bus is 84 bits wide which consist of 64-bit fraction, 15-bit exponent and a sign bit.
- When the NEU starts execution of an instruction, it always pulls up the BUSY signal, connected with TEST input signal of the 8086.
- Therefore, 8086 must be waiting till the BUSY pin of 8087 or the TEST input pin of 8086 becomes low.
- The microcode control unit generates the control signals which are required for execution of the instructions.
- The programmable shifter is used for shifting the operands during the execution of instructions.
- The data bus interface is able to connect the internal data bus of 8087 with the main processor data bus.

Block Diagram

Registers:



Block Diagram

Floating Point Data Registers:

- The data registers are theoretically divided into three fields such as sign (1-bit), exponent (15-bits) and significant (64-bits).
- The actual use of these fields varies with the type of data being operated with the instruction.
- When the 8087 receives numeric data words, it stores and holds them in a format called temporary real form.
- This number is expressed as the product of a 64-bit significant base and a 15-bit exponent and the Most Significant Bit (MSB) of the register is reversed as a sign bit to represent either a positive or a negative number.
- The 8087 instructions automatically convert data into this format when loading the registers and return back to the other format when returning them to the system memory.

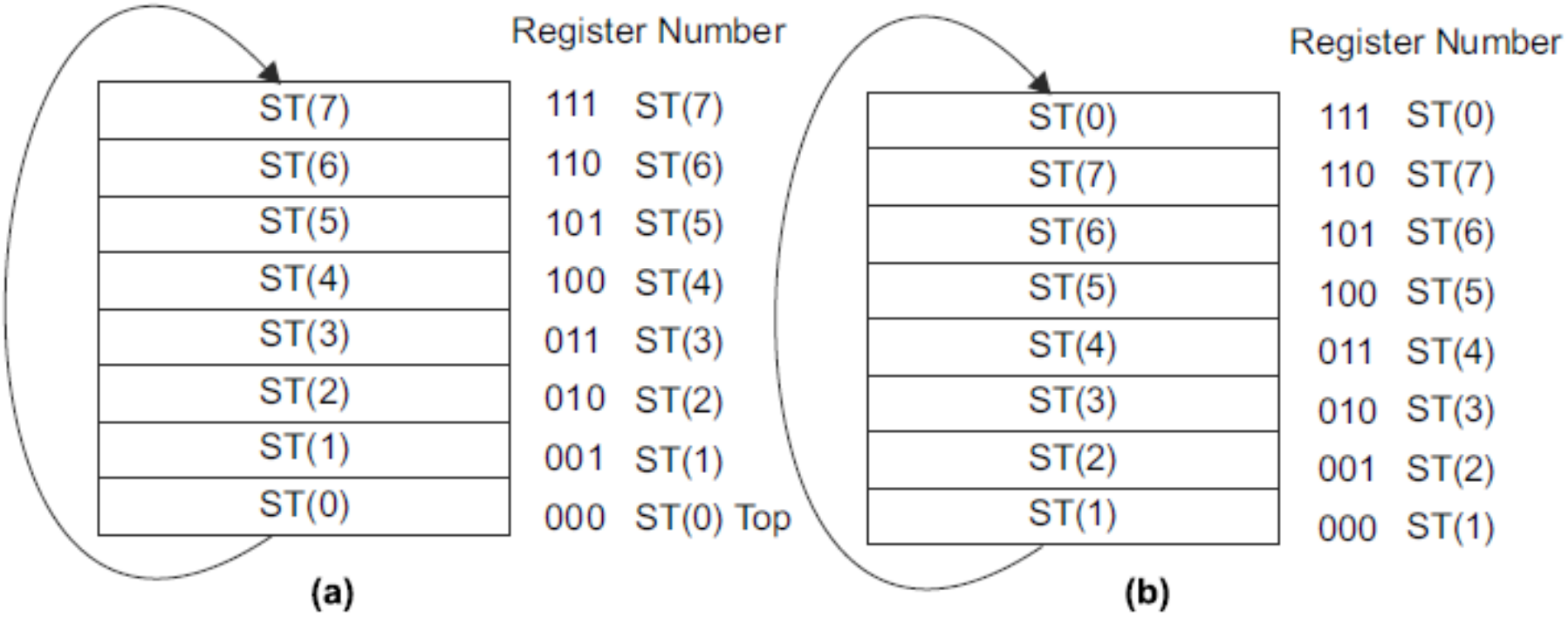
Block Diagram

Floating Point Data Registers:

- These registers can be represented as ST (0), ST (1), ST(2), ST(3) ST(4), ST(5) ST(6), and ST(7).
- When the 8087 co-processor is reset, ST (0) becomes the top of the stack and ST(1) refers to the next register in the stack and other registers will be referred.
- After the first push operation, the register 000 becomes ST(1) and the register 111 refers ST(0).
- During programming, any register can be used as the top of the stack and other registers will be referred according to their position.

Block Diagram

Floating Point Data Registers:



(a) Registers after reset (b) Register after first push operation

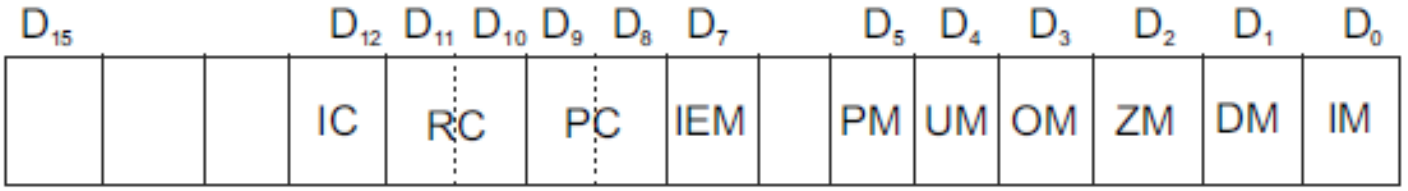
Block Diagram

Control Word Registers:

- The 16-bit control word register is used to control the operation of the 8087.
- The control word register bits can select the required data processing options such as precision control, rounding control and infinity control.
- The bits D5–D0 are used for masking the different exceptions.
- An exception may be masked by setting the respective bit in the control word register.
- The IEM bit is used as a common interrupt mask for all the interrupts.
- When IEM is set, all the exceptions generated will be masked and the execution may continue.

Block Diagram

Control Word Registers:



IC — Infinity control
RC — Rounding control
PC — Precision control
IEM — Interrupt enable mask
PM — Precision mask
UM — Underflow mask
OM — Overflow mask
ZM — Division by zero mask
DM — Denormalized operand mask
IM — Invalid operand mask

(1) Interrupt - Enable Mask:

0 = Interrupts Enabled
1 = Interrupts Disabled (Masked)

(2) Precision Control:

00 = 24 bits
01 = (reserved)
10 = 53 bits
11 = 64 bits

(3) Rounding Control:

00 = Round to Nearest or Even
01 = Round Down (toward − ¥)
10 = Round Up (toward + ¥)
11 = Chop(Truncate Toward Zero)

(4) Infinity Control:

0 = Projective
1 = Affine

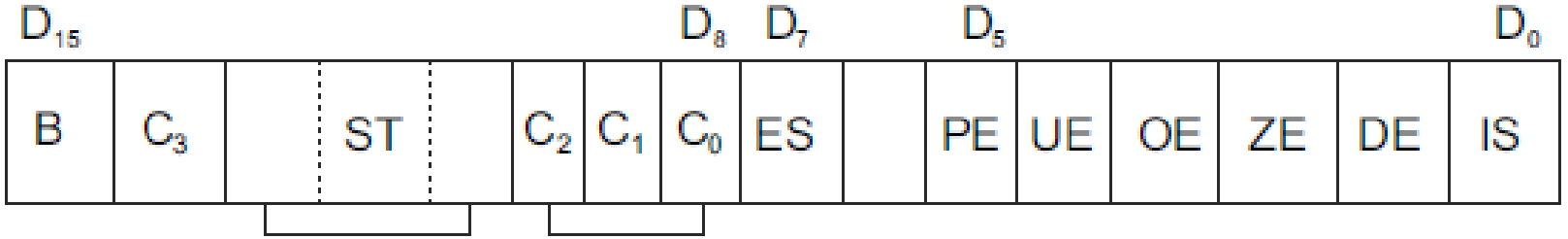
Block Diagram

Status Registers:

- The co-processor status register contains the conditional code bits and the floating point flags.
- During executing an instruction, the 8087 co-processor generates six different exceptions, which are reflected in the status register format.
- Whenever any exception is generated, an interrupt take place to the CPU provided it is not masked.
- Then the 8086 processor will respond if the interrupt flag of the CPU is set.
- When the exceptions are masked, the 8087 continues the execution.
- Therefore, the status register reflects the over all operations of the 8087 co-processor.

Block Diagram

Status Registers:



B — Busy bit

C_3–C_0 — Condition code bits

ST — Top-of-stack (ST)

ES — Error summary

PE — Precision error

UM — Underflow error

OM — Overflow error

ZM — Zero error

DE — Denormalized error

IE — Invalid error

C_3	C_2	C_1	C_0	*Result*
0	0	×	0	ST > Source
0	0	×	1	ST < Source
1	0	×	0	ST = Source
1	1	×	1	ST is not comparable

Block Diagram

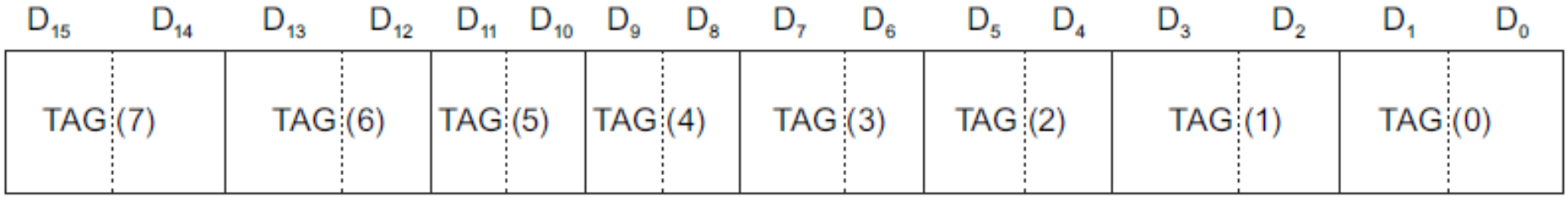
Status Registers:

C_3	C_2	C_1	C_0	*Result*
0	0	0	0	+Un-normal
0	0	0	1	+NAN
0	0	1	0	–Un-normal
0	0	1	1	–NAN
0	1	0	0	Normal
0	1	0	1	$+\infty$
0	1	1	0	– Normal
0	1	1	1	$-\infty$
1	0	0	0	± 0
1	0	0	1	Empty
1	0	1	0	–0
1	0	1	1	Empty
1	1	0	0	+De-normal
1	1	0	1	Empty
1	1	1	0	–De-normal
1	1	1	1	Empty

Block Diagram

Tag Registers:

- The tag register contains several groups of 2 bits that can determine the state of the value of the eight 80-bit stack registers as valid, zero, special or empty.
- The tag word register presents the entire TAG field to the CPU.
- The tag values are 00 = valid, 01 = zero, 10 = special and 11 = empty.



Block Diagram

Instruction and Data Point Registers:

- The instruction and data pointer registers are used to hold the information about the last executed floating point instruction.
- Actually, the information is address of instruction, op-code and operand address.
- Prior to execution of a mathematical instruction, the 8087 forms a table in the memory.
- The table contains the instruction address in the fields of the instruction pointers, the opcode of the instruction, and operand address in the field of data pointers.
- Therefore, the instruction pointer and the data pointer registers contain the address of the currently executed instruction and the corresponding data.

Block Diagram

Instruction and Data Point Registers: