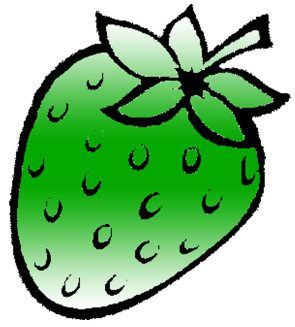
STRAWBERRY



/strawberrydevelopers

/strawberry_app

For more visit:

Strawberrydevelopers.weebly.com

8087 Math Processor

Richa Upadhyay Prabhu

NMIMS's MPSTME

richa.upadhyay@nmims.edu

February 9, 2016

Introduction

Need of Math Processor:

- In application where fast calculation is required
- Also where there is a need to do arithmetic operations on very small and very larger numbers
- 8086 is not designed to do complex mathematical operations
- Highly optimized number crunching programs run slow on general purpose processors

To solve this , special processors with architectures and instruction set optimized for performing complex lengthy calculations.

8087 Introduction

- 8087 is referred as Co-processor, as it is used in parallel with the main processor
- Main processor handles general program execution
- 8087 handles specialized math computations
- 8087 can perform computations 10 times faster than 8086

8087 Introduction

- 8087 is a processor with its own instruction set
- These instructions are written in program as need interspersed with 8086 instructions
- Each processor decodes all the instructions in the fetched instruction byte stream but executes only its own instruction
- Like 8086, 8087 also has an instruction queue
- While decoding if 8087 finds an 8086 instruction then it simply treats it as NOP.

DATA TYPES

There are three general type of data types:

- Binary Integer
- Packed decimal
- Real

Binary Integer Data Type

- Basic format that is used to represent signed binary numbers
- MSB is sign bit; '0' for positive and '1' for negative
- Rest of the bits represent the magnitude of the number
- If number is negative, the magnitude is represented in 2's complement form

Binary Integer Data Type

Below figures show binary integer format in different lengths for various ranges of numbers

(a) Word integer (16 Bit Signed Integer)

S	Magnitude
15	0

Sign bit is 0 for positive and 1 for negative.
Range: $-32768 <= X <= +32767$. Negative number representation in 2's complement form.

Binary Integer Data Type

Below figures show binary integer format in different lengths for various ranges of numbers

(b) Short integer (32 Bit Signed Integer)

S	Magnitude
31	0

Range: $-2 \times 10^9 <= X <= 2 \times 10^9$

Binary Integer Data Type

Below figures show binary integer format in different lengths for various ranges of numbers

(c) Long Integer (64 Bit Signed Integer)

S	Magnitude
63	0

Range: $-9 \times 10^{18} <= X <= 9 \times 10^{18}$

Packed Decimal Numbers

- A number is represented as a string of 18 BCD digits
- Each byte of storage can contain two decimal numbers
- MSB bit is sign bit
- format is handy for working with financial programs, e.g. to represent amount as large as $9,999,999,999,999,999.99
- 8 don't care bits are also there

S	Don't care	Magnitude
79	72	71 ... 0

-99... ... 99<=X<=+99... ...99(18 digits)

Real Numbers

- Real Numbers or Floating point Numbers, which have both a real and a decimal part
- Basic principle is to use one group of bits to represent the digits and another to represent position of binary point w.r.t. these digits
- Number should be written in scientific notation or it should be normalized
- Process of moving decimal point just to the right of most significant non zero digit is called normalization

For example: $0.00857 = 8.57 \times 10^{-3}$

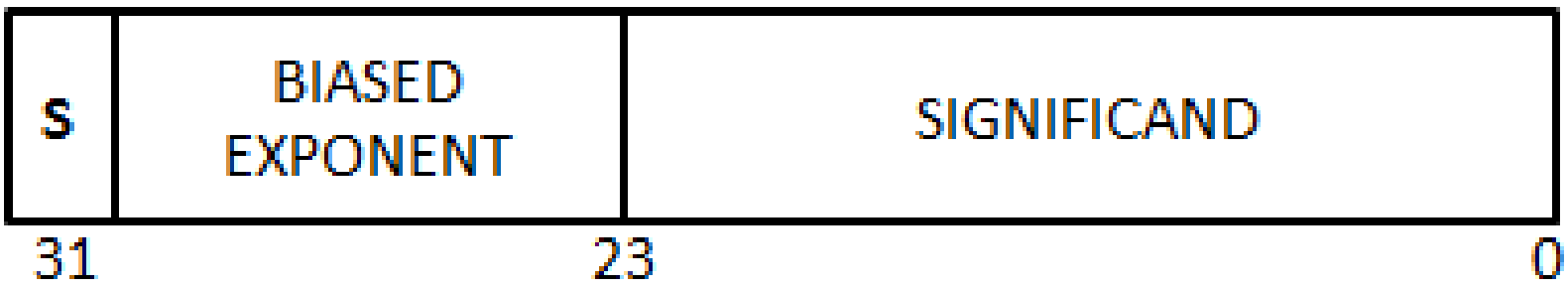
Real Numbers

For example: $0.00857 = 8.57 \times 10^{-3}$

- In the example we have the digit part i.e. *significand* or *mantissa*
- And also the exponent part
- Also sign of the exponent, which indicates whether the magnitude is > 1 or < 1

Real Numbers

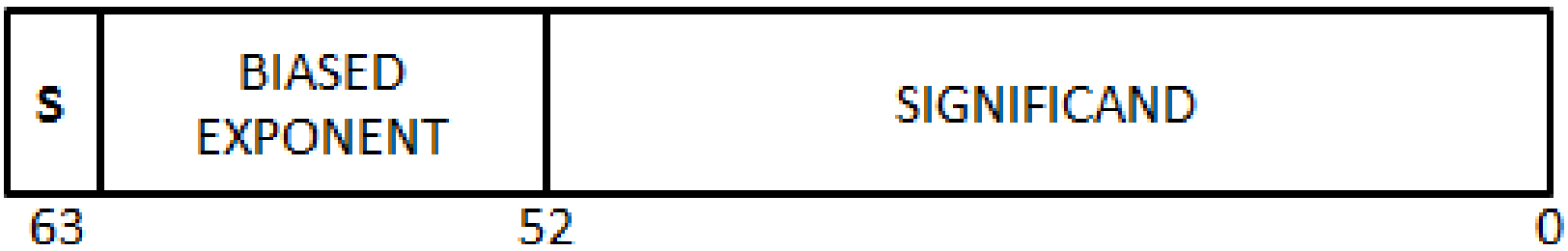
SHORT REAL



$1.2 \times 10^{-38} \leqslant |x| \leqslant 3.4 \times 10^{38}$

Real Numbers

LONG REAL



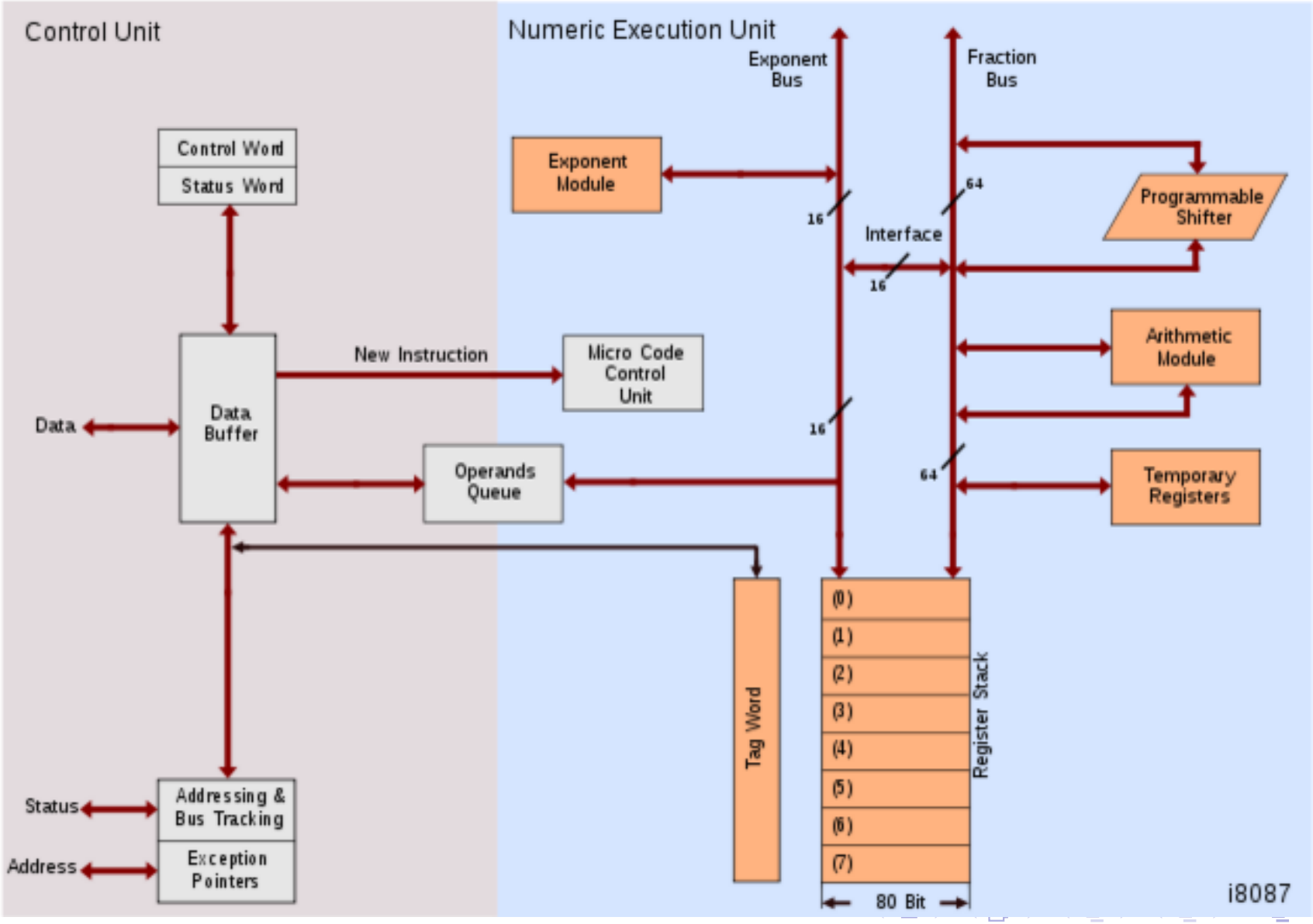
$2.3 \times 10^{-308} \leqslant |x| \leqslant 1.7 \times 10^{308}$

Real Numbers

TEMPORARY REAL

$3.4 \times 10^{-4932} \leqslant |x| \leqslant 1.1 \times 10^{4932}$

8087 INTERNAL ARCHITECTURE



8087 INTERNAL ARCHITECTURE

The 8087 is divided into 2 sections :

- Control Unit
- Numeric Execution Unit

The **numeric execution unit** executes all numeric processor instructions while **control unit** receives, decodes instructions, read and writes memory operands and executes 8087 control instructions.

These 2 units works asynchronously with each other.

The control unit is majorly responsible for establishing communication between CPU and memory and also for coordinating internal co- processor execution.

8087 INTERNAL ARCHITECTURE

CONTROL UNIT

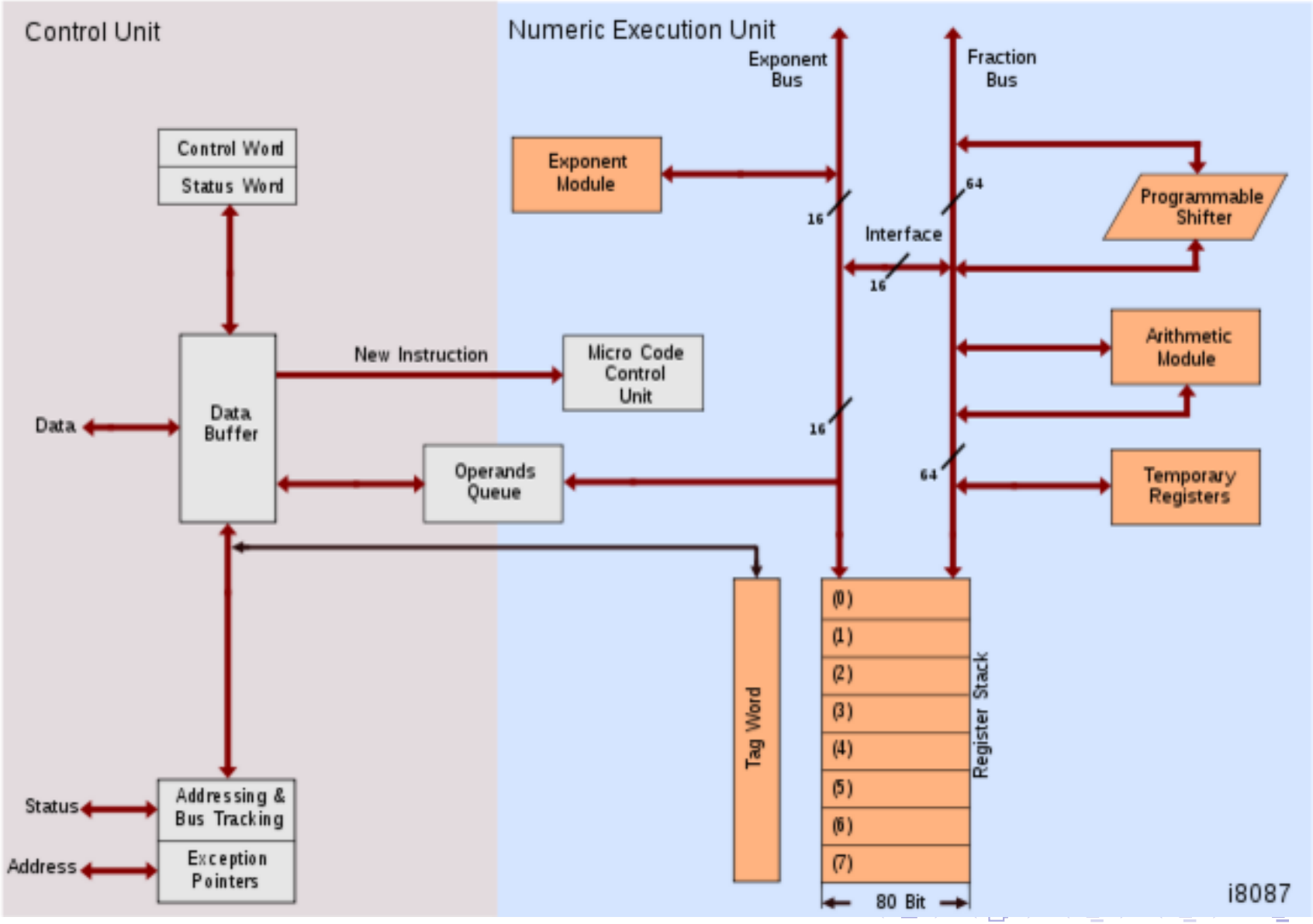
- Used to synchronize the operations between main processor and co-processor
- It receives the instruction opcode, decodes it and reads or write operands from memory
- It continuously monitors data bus to find instruction for 8087
- **Operand Queue**: 8087 maintains a parallel queue similar to the processor, whose length can be adjusted depending on the processor
- For 8086; the queue is of 6 bytes and for 8088 it is of 4 bytes

8087 INTERNAL ARCHITECTURE

CONTROL UNIT

- Queue status input pins QS_0 and QS_1 are used by 8087 to identify instructions fetched by the microprocessor
- 8087 instruction opcodes have 11011 as the most significant bits of first code byte
- Control unit consists of Control word, Status word and Data buffer (will be discussed later)

8087 INTERNAL ARCHITECTURE



8087 INTERNAL ARCHITECTURE

NUMERIC EXECUTION UNIT

- Blocks in this unit duplicate the functions performed by control and ALU blocks in microprocessor
- It performs all operations that access and manipulate numeric data in 8087 registers
- Numeric registers are 80 bit wide and the data is routed by 64 bit mantissa/significand bus and and a 16 bit sign/exponent bus
- While executing an instruction the NEU pulls up the BUSY signal, which is connected to $\overline{TEST}$ input of 8086
- CPU is able to distinguish that the execution is not yet completed

8087 INTERNAL ARCHITECTURE

MICROCODE CONTROL UNIT : generates control signals which are required for execution of instruction

PROGRAMMABLE SHIFTER : used for shifting operands during execution of instruction

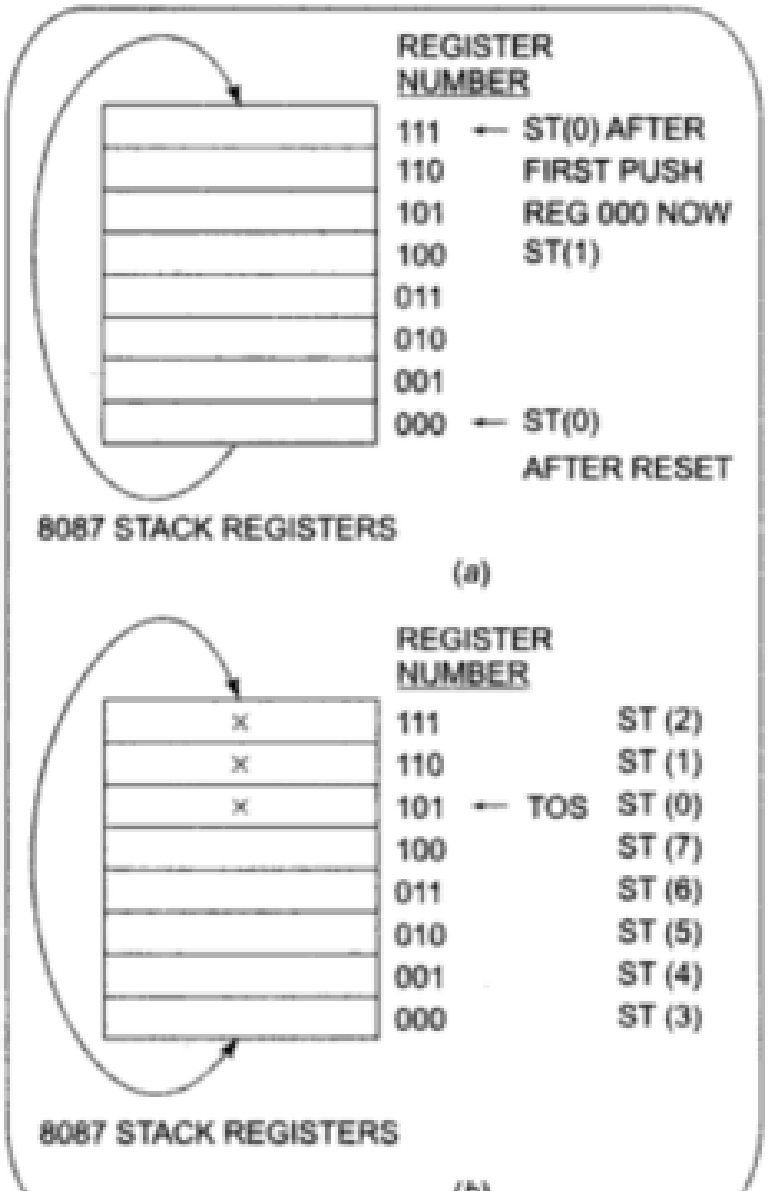
DATA BUS INTERFACE: connects internal data bus of 8087 to main processor data bus

8087 INTERNAL ARCHITECTURE

REGISTER STACK

- 8087 internally works with numbers in 80 bit temporary real format
- To hold these numbers it has a register stack of 80 bits registers, labeled (0)-(7)
- Last in first out stack (same as 8086)
- 8087 has a 3 bit stack pointer, which holds the number of register that is the top of stack

8087 INTERNAL ARCHITECTURE



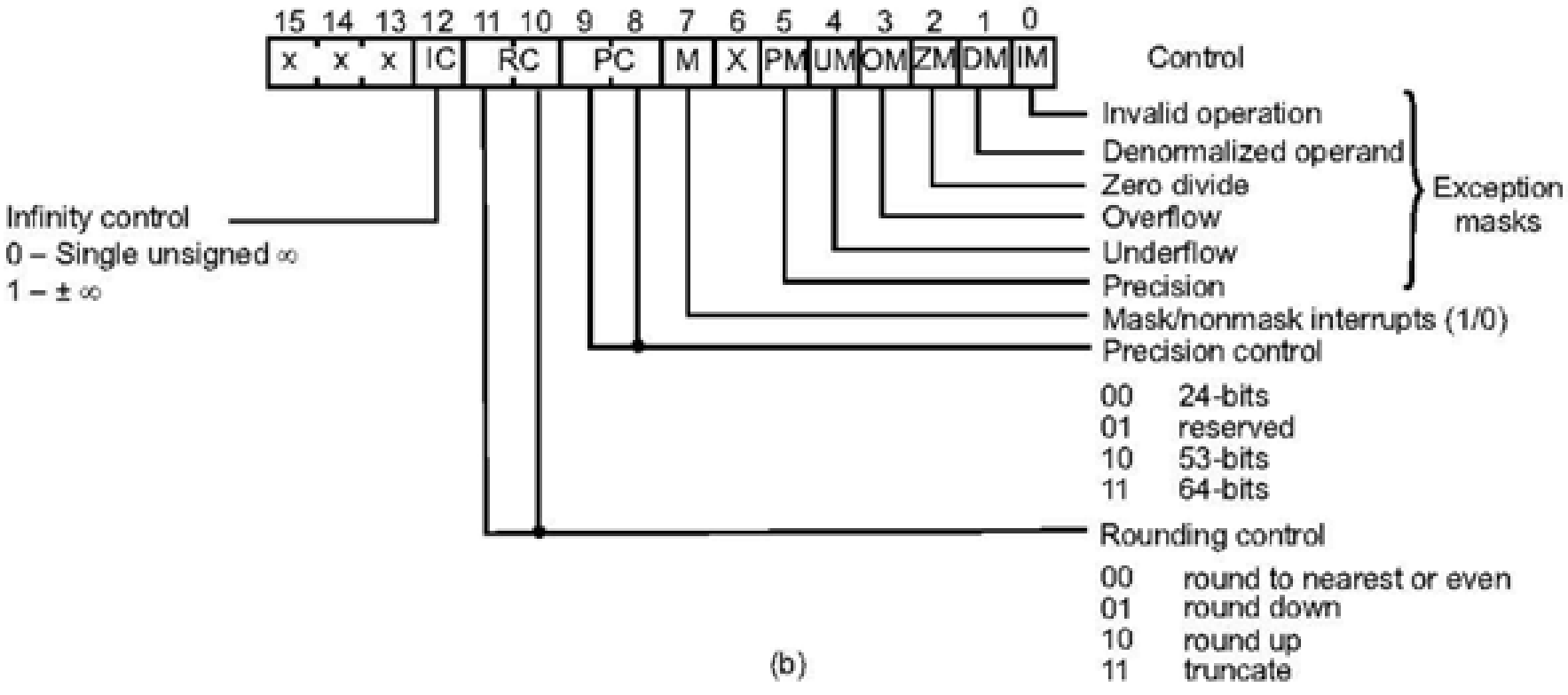
8087 INTERNAL ARCHITECTURE

CONTROL AND STATUS WORD

- Control word is sent to 8087 from 8086, by writing them to a memory location
- 8087 has to execute an instruction which reads the control word from memory
- Status word is sent to 8086 from 8087
- 8087 has to execute an instruction that will write the status word to memory

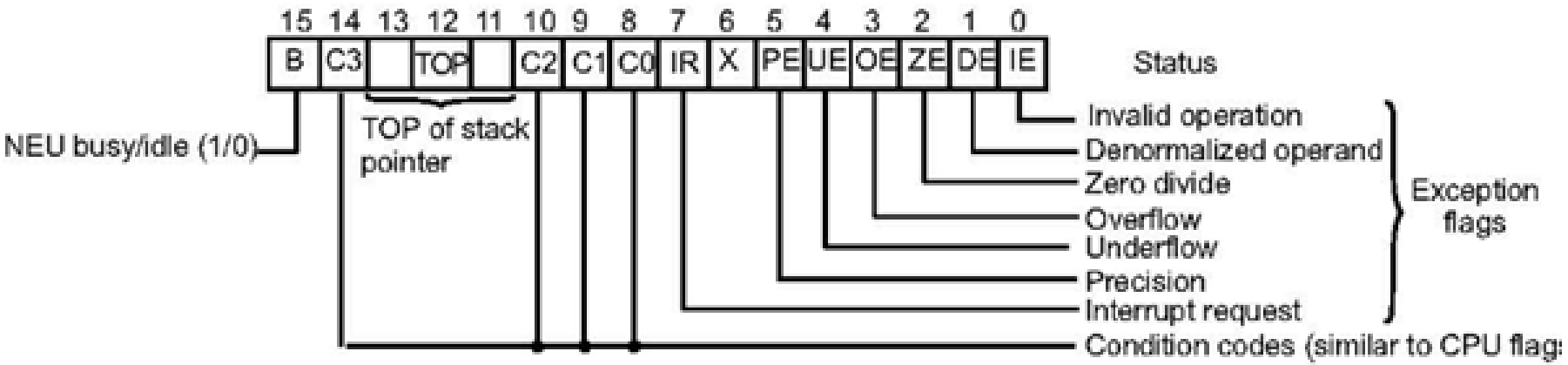
8087 INTERNAL ARCHITECTURE

CONTROL WORD



8087 INTERNAL ARCHITECTURE

STATUS WORD



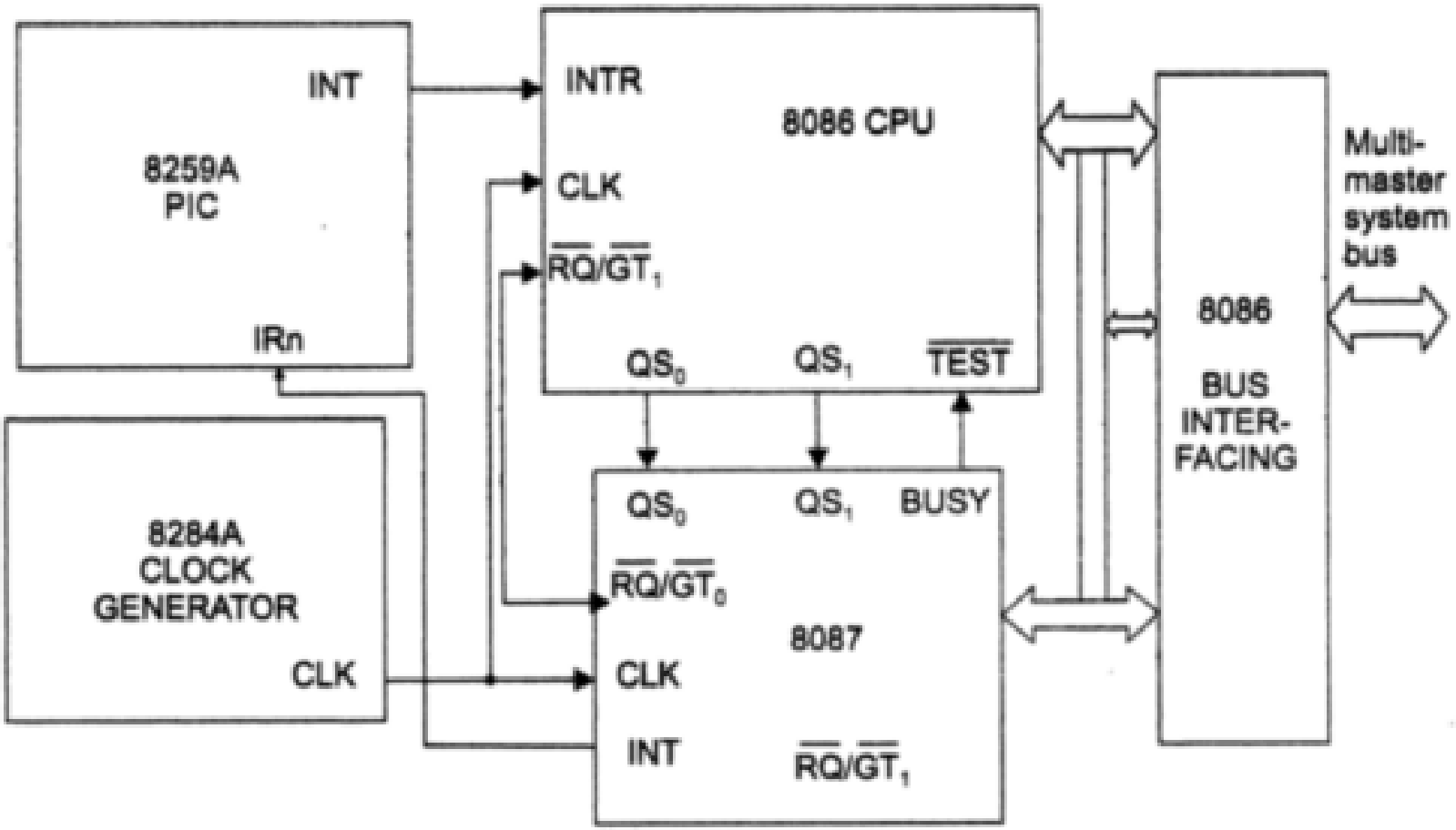
8087 Exceptions

- **Invalid operation**: This includes the attempt to calculate the square root of a negative number or say to take out an operand from an empty register
- **Overflow**: Exponent of the result is too large for the destination real format.
- **Zero divide**: Arises when divisor is zero while the dividend is a non-infinite, non-zero number.
- **Denormalized operation**: It arises when an attempt is made to operate on an operand that is yet to be normalized.
- **Under flow**: Exponent of the result is too small to be represented.
- **Precision**: In case the operand is not made to represent in the destination format, causing 8087 to round the result. Also known as In-exact result.

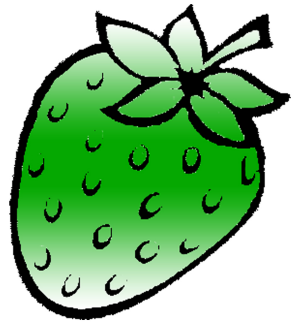
8087 operation inn case of an exceptions

- 8087 sets the appropriate flag bit in the status word in case of occurrence of any one of the exception conditions.
- The exception mask in the control register is then checked and if the mask bit is set i.e., masked, then a built-in fix-up procedure is followed.
- If the exception is unmasked (i.e., mask bit $= 0$), then user-written exception handlers take care of such situations.
- This is done by using the INT pin which is normally connected to one of the interrupt input pins of 8259A PIC.

8087 and 8086 interfacing



STRAWBERRY



/strawberrydevelopers

/strawberry_app

For more visit:

Strawberrydevelopers.weebly.com