Introduction to 8086

Introduction to 8086

- 8086 is a 16-bit processor, which implies that
  - 16-bit data bus
  - 16-bit ALU
  - 16-bit registers
- 8086 has a 20-bit address bus can access up to 2^{20} memory locations.

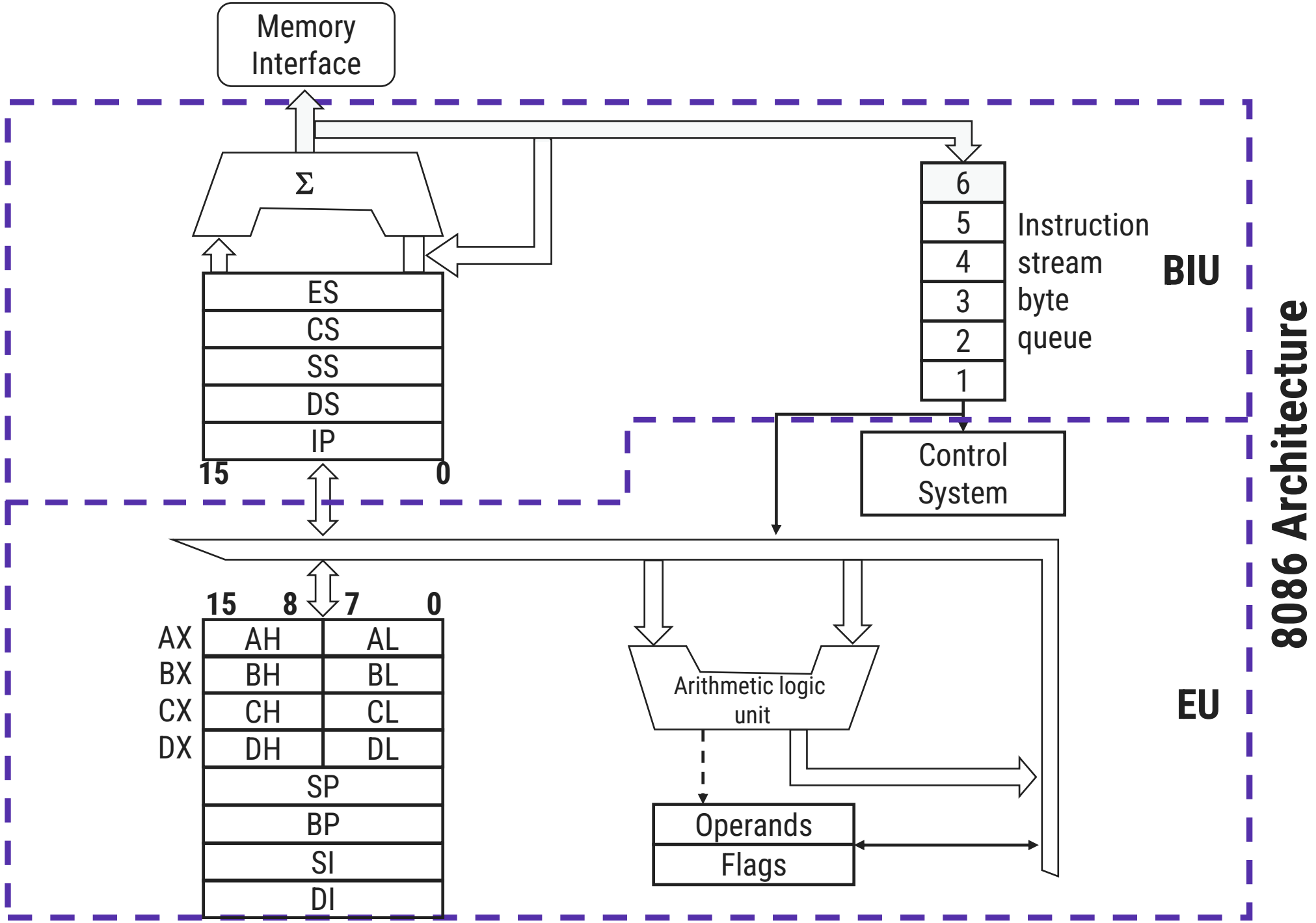
  (2^{20}=1048576 bytes =1 MB)
- It can support up to 64K I/O ports. (2^{16} I/O ports:2^{16}=65536)
- 8086 has 256 vectored interrupt.
- 8086 contains powerful instruction set, that also supports multiply and divide operation.

Introduction to 8086

- 8086 can operate in two modes:
  i. **Minimum mode**: A system with only one processor i.e.8086
  ii. **Maximum mode**: A system with multiple processors.
  e.g. 8086 + math co-processor(8087),
  8086+ I/O processor (8089)
- 8086 uses memory segmentation. Segmentation means dividing memory into logical components.
- In 8086 memory is divided into 16 segments of capacity 2^{16} bytes each and used as code, stack, data and extra segment respectively.

8086 Architecture

Block Diagram


Memory Interface
Σ
ES
CS
SS
DS
IP
15
0
6
5
4
3
2
1
Instruction stream byte queue
BIU
Control System
15
8
7
0
AX
BX
CX
DX
AH
AL
BH
BL
CH
CL
DH
DL
SP
BP
SI
DI
Arithmetic logic unit
Operands
Flags
EU
8086 Architecture

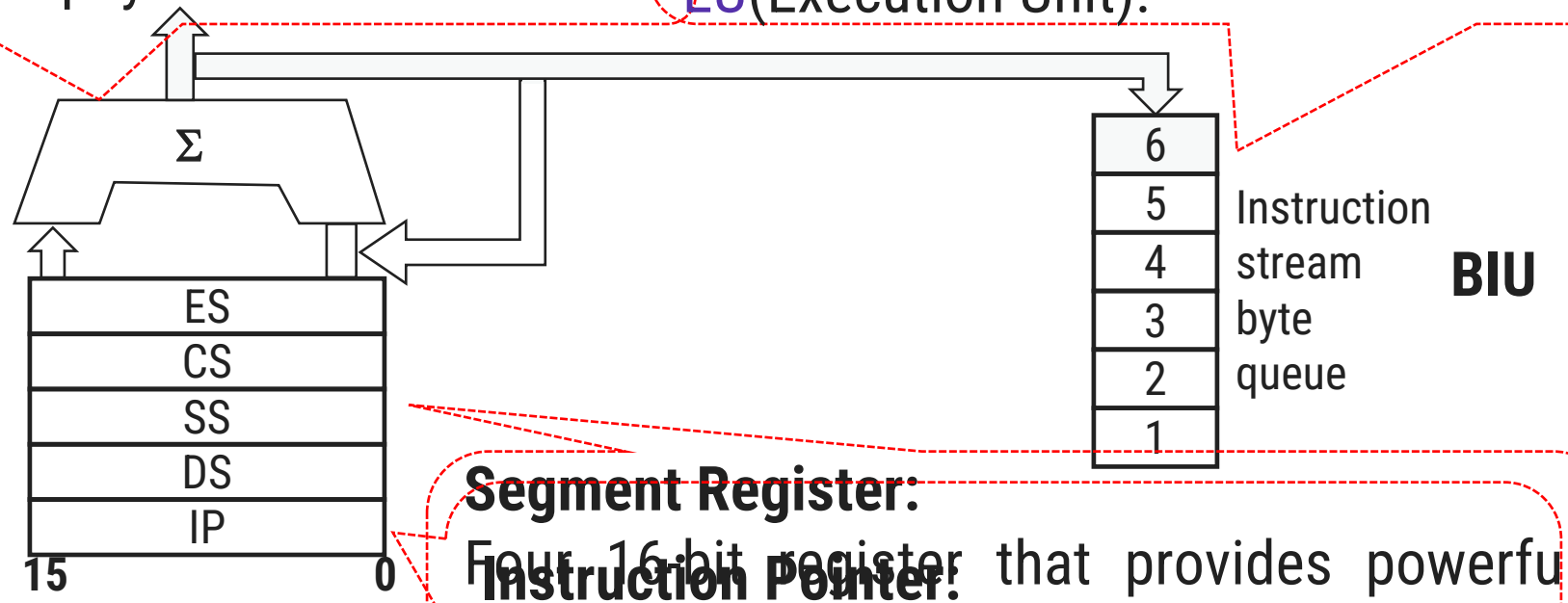
8086 Architecture

- In 8086 CPU is divided into two independent functional units:
  1. **BIU (Bus Interface Unit)**
  2. **EU (Execution Unit)**
- Dividing the work between these two units speeds up the processing.

Components: BIU(Bus Interface Unit)

Address Generator: Generates 20-bit of physical address.

Instruction Queue:
It holds the instruction bytes of the next instruction to be executed by EU(Execution Unit).



Segment Register:
Four 16-bit register that provides powerful memory management mechanism.
ES (extra segment) , **CS** (code segment), **SS** (stack segment) , **DS** (data segment).

Instruction Pointer:
Register that holds 16-bit address or offset of next code byte within code segment.

Task of BIU

1. Fetch instructions from memory.
2. Read/write instructions to/from the memory.
3. Input/output of data to/from peripheral ports.
4. Address generation for memory reference.
5. Queuing instructions.

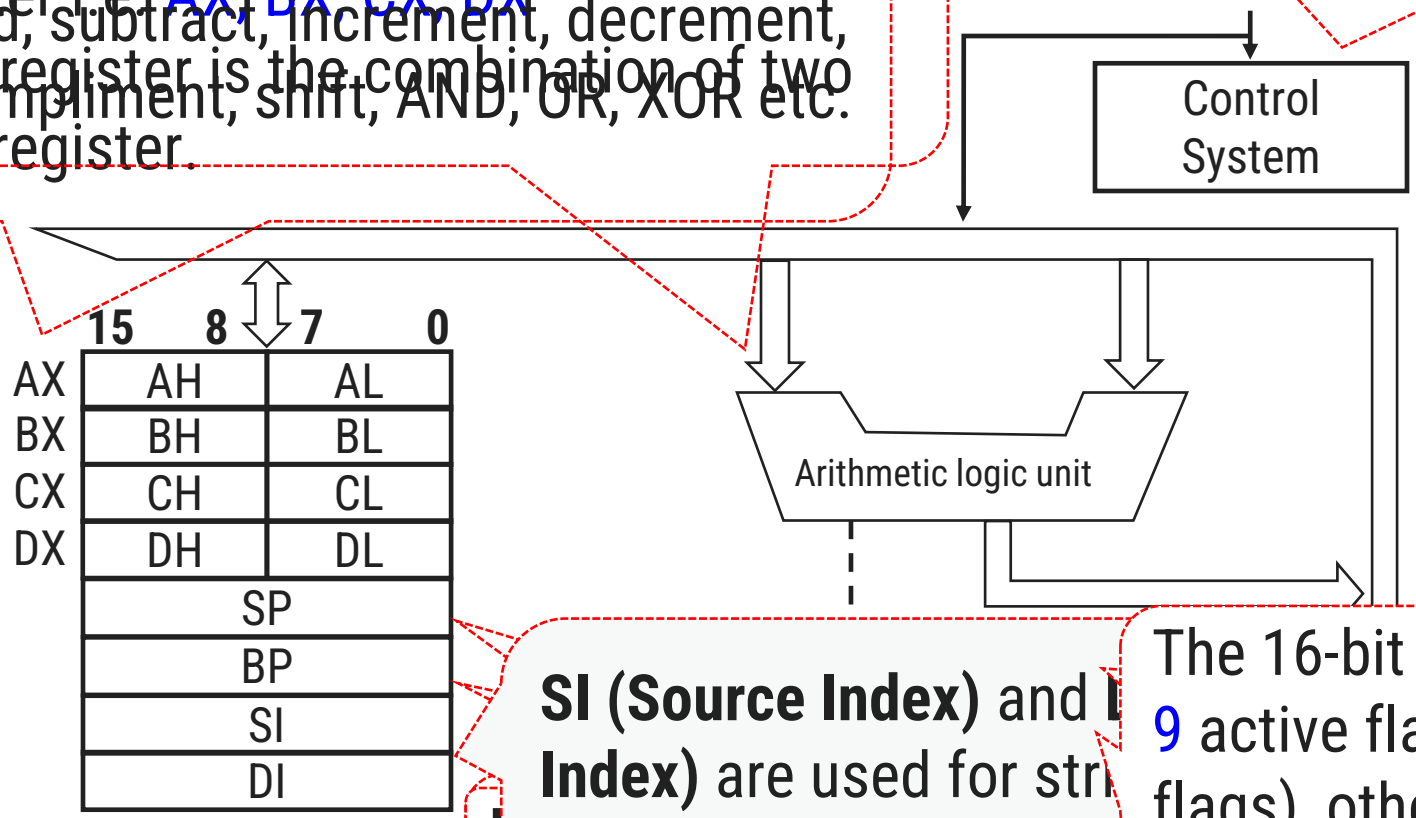
Thus, BIU handles all transfer of data and address.

Components: EU(Execution Unit)

EU has 4 general purpose, 16-bit register i.e. AX, BX, CX, DX each register is the combination of two 8-bit register.

Contains 16-bit ALU, that performs add, subtract, increment, decrement, compliment, shift, AND, OR, XOR etc.

Performs various internal operations.

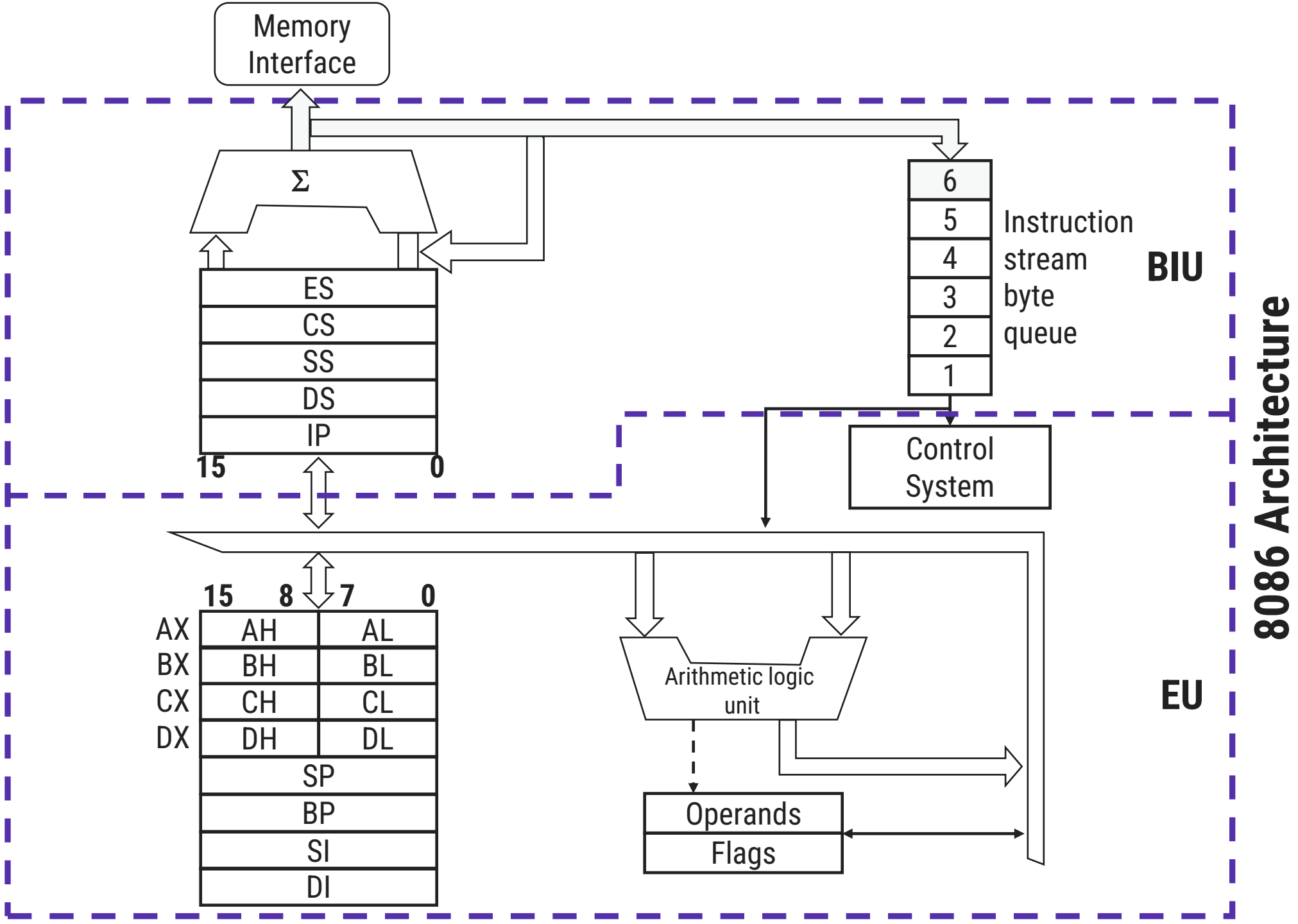


SI (Source Index) and **I Index)** are used for stri operation and for movin memory from one location to another.

The 16-bit flag register of 8086 contains 9 active flags (6 conditional & 3 control flags), other 7 flags are undefined.

Task of EU (Execution Unit)

1. Decodes the instruction.
2. Executes decoded instructions.
3. Tells BIU from where to fetch the instruction.
4. EU takes care of performing operation on the data.
5. EU is also known as **execution heart** of the processor.


Memory Interface
Σ
ES
CS
SS
DS
IP
15
0
6
5
4
3
2
1
Instruction stream byte queue
BIU
Control System
15
8
7
0
AX
BX
CX
DX
AH
AL
BH
BL
CH
CL
DH
DL
SP
BP
SI
DI
Arithmetic logic unit
Operands
Flags
EU
8086 Architecture

Segment Register in 8086

Segment Register in 8086

1. **Code Segment (CS)**: Stores **executable** program.
2. **Data Segment (DS)**: Contains **data** used by a program. Data can be accessed from this by an offset address.
3. **Stack Segment (SS):** Defines an area of memory used for the **stack**.
4. **Extra Segment (ES):** ES an additional **data** segment.

Segmentation in 8086

Segmentation in 8086

What is Segment?

An area in memory.

What is Segmentation?

The process of dividing memory into segments of various sizes is called **Segmentation**.

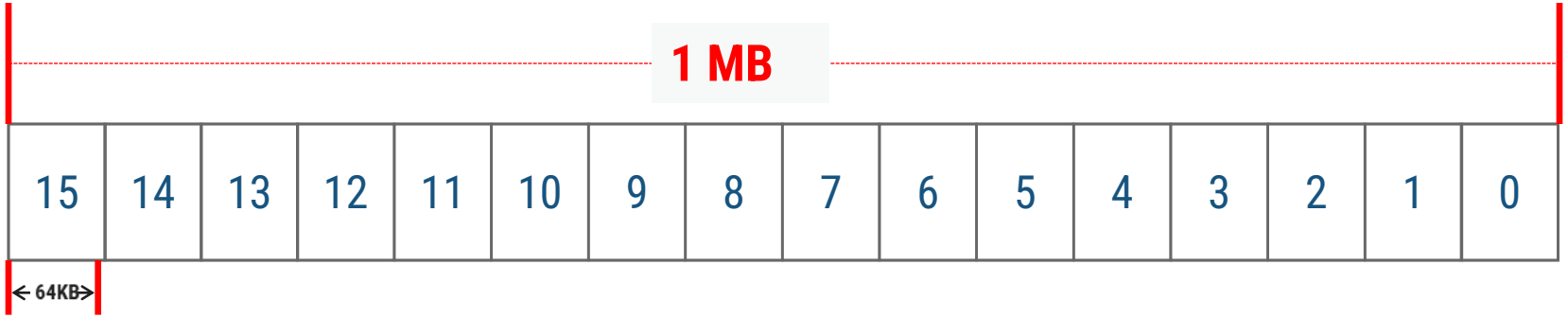
What is need of segmentation in 8086?

What is the need of segmentation in 8086?

- Memory is huge collection of bytes.
- In order to organize these bytes in an efficient manner segmentation is used.

E.g. $$\text{No. of segments} = \frac{\text{Total memory available}}{\text{size of each segment}}$$

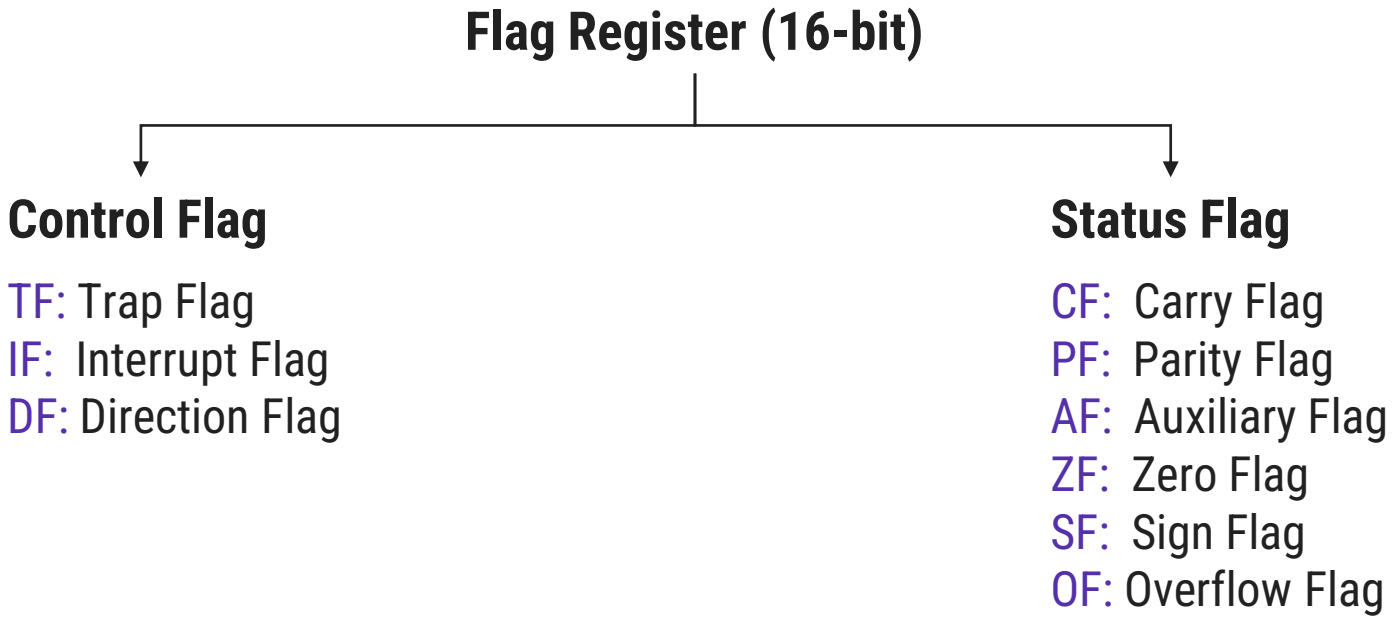
$$\text{No. of segments} = \frac{1\ \text{MB}}{64\ \text{KB}} = \frac{1024\ \text{KB}}{64\ \text{KB}} = 16\ \text{segments}$$



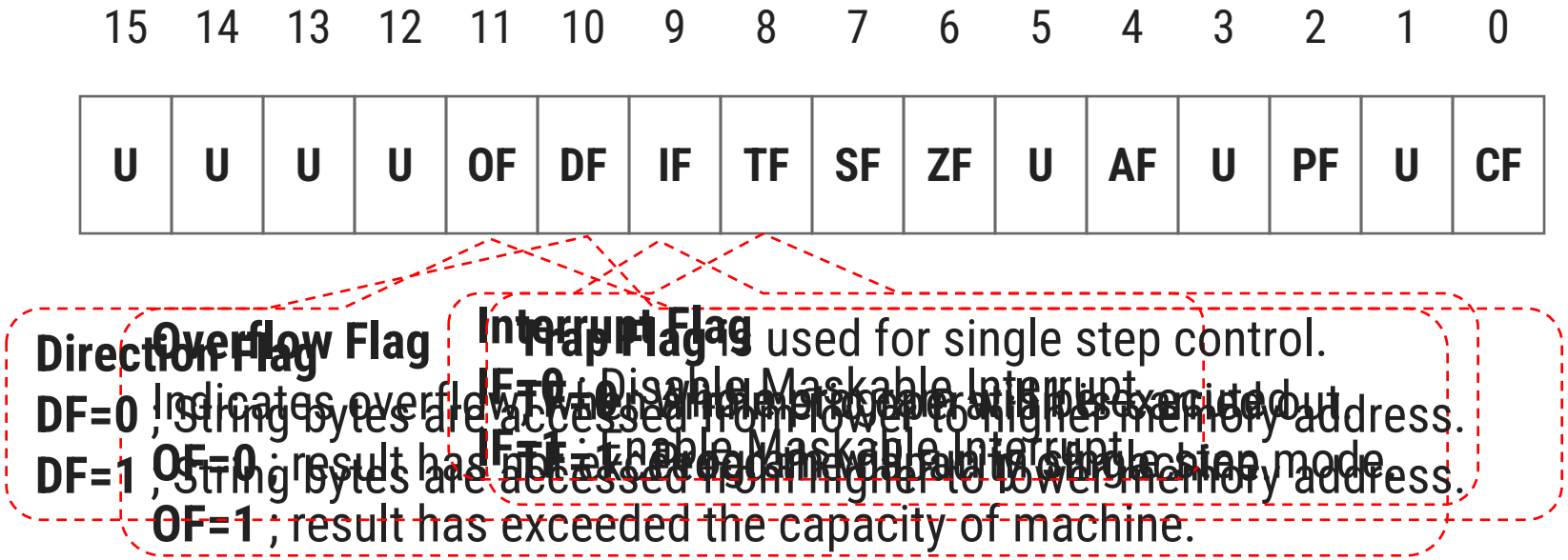
8086 Flag Register

8086 Flag Register

- The 16-bit flag register of 8086 contains 9 active flags (6 conditional & 3 control flags), other 7 flags are undefined.



8086 Flag Register



8086 Flag Register

- **Carry Flag (CF):** Set(1) if arithmetic operation results in carry; otherwise reset(0).
- **Auxiliary Flag (AF):** If an operation performed in ALU generates a carry/barrow from lower nibble (i.e. $D_0 - D_3$) to upper nibble (i.e. $D_4 - D_7$), the AF flag is set i.e. carry given by D_3 bit to D_4 is AF flag. This is not a general-purpose flag, it is used internally by the processor to perform Binary to BCD conversion.
- **Parity Flag (PF):** This flag is used to indicate the parity of result. If lower order 8-bits of the result contains even number of 1's, the Parity Flag is set and for odd number of 1's, the Parity Flag is reset.
- **Zero Flag (ZF):** It is set(1), if the result of arithmetic or logical operation is zero else it is reset(0).
- **Sign Flag (SF):** In sign magnitude format the sign of number is indicated by MSB bit. If the result of operation is negative, sign flag is set(1).