8086

1. Draw and explain 8086 Logical Block diagram

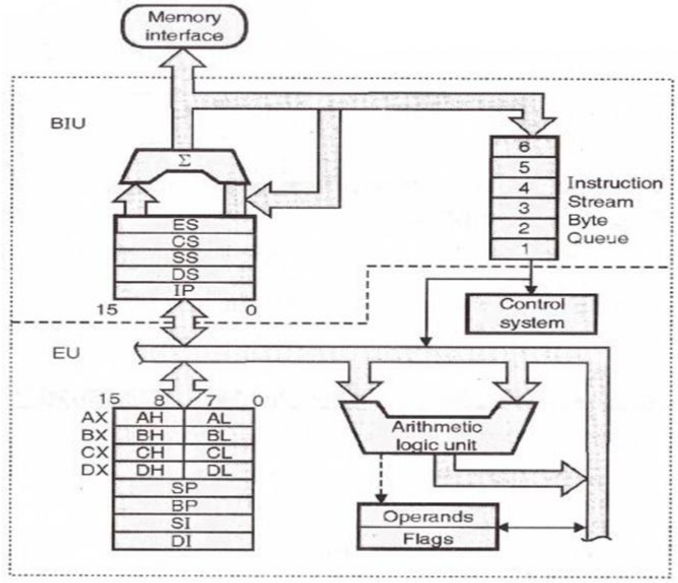


Figure: 8086 Architecture

- In 8086 CPU is divided into two independent functional parts BIU and EU.
- Dividing the work between these two units' speeds up the processing.

BIU (Bus Interface Unit)

Components of BIU

- Instruction queue
  It holds the instruction bytes of the next instruction to be executed by EU
- Segment Registers
  Four 16-bit register that provides powerful memory management mechanism
- ES (extra segment), CS (code segment), SS (stack segment), DS (data segment).
  The size of each register is 64kb.
- Instruction pointer (IP)
  Register that holds 16-bit address or offset of next code byte within code segment
- Address Generation and bus control
  Generation of 20-bit physical address

Task carried out by BIU

- Fetch instruction from memory
- Read/ Write instruction from / to the memory
- Input/ Output (I/O) of data from / to peripheral ports
- Write the data to memory.
- Address generation for memory reference
- Queuing of instruction (The instruction bytes are transferred to the instruction queue)
- Thus, BUI handles all transfer of data and address on the buses for Execution unit.
- BIU works in synchronous with machine cycles

- **EU (Execution Unit)**

Components of EU

- ALU (Arithmetic logic Unit)
  Contains 16-bit ALU, that performs add, subtract, increment, decrement, compliment, shift binary numbers, AND, OR, XOR etc.
- CU (Control Unit)
  Directs internal operation
- Flag Register
  16-bit flag register
  EU contains 9 active flags
- General Purpose Registers (GPR)
  EU has 4 general purpose 16-bit register
  i.e. AX, BX, CX, DX
  each register is the combination of two 8-bit register
  AH, AL, BH, BL, CH, CL, DH, DL where 'L' means Lower byte and 'H' means higher byte.
- Index Register
  16-bit Register is SI (source index) and DI (destination index).
  Both the register are used for string related operation and for moving block of memory from one location to the other.
- Pointers
  16-bit Register.
  i.e. SP (stack pointer), BP (base pointer)
  BP : is used when we need to pass parameter through stack
  SP: It always points to the top of the stack. Used for sequential access of stack segment.
- Decoder (instruction decoder)
  Translates the instruction fetched from into series of action which EU carries out

Task carried out by EU

- Decodes the instruction
- It executes instructions (executes decoded instructions)
- Tells BIU from where to fetch the instruction
- Decodes instruction (decode the fetched instruction)
- EU takes care of performing operation on the data
- EU is also known as execution heart of the processor

2. Explain 8086 Registers

Ans. The 8086 microprocessor has a total of fourteen registers that are accessible to the programmer as follows:-

General Purpose Register

AX: - Accumulator register consists of two 8-bit registers AL and AH, which can be combined together and used as a 16-bit register AX.

- AX works as an intermediate register in memory and I/O operation.
- Accumulator is used for the instruction such as MUL and DIV.

BX: - Base register consists of two 8-bit registers BL and BH, which can be combined together and used as a 16-bit register BX.

- BX register usually contains a data pointer used for based, based indexed or register indirect addressing.

CX: - Count register consists of two 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX. Count register can be used in Loop, shift/rotate instructions and as a counter in string manipulation.

DX: - Data register can be used together with AX register to execute MUL and DIV instruction.

- Data register can be used as a port number in I/O operations.

Segment Register

Types of Segment registers are as follows:-

1. Code Segment (CS): The CS register is used for addressing a memory location in the Code Segment of the memory, where the executable program is stored.
2. Data Segment (DS): The DS contains most data used by program. Data are accessed in the Data Segment by an offset address or the content of other register that holds the offset address.
3. Stack Segment (SS): SS defined the area of memory used for the stack.
4. Extra Segment (ES): ES is additional data segment that is used by some of the string to hold the destination data

Pointer Registers

The pointers IP, BP, SP usually contain offsets within the code, data and stack segments respectively.

1. Stack Pointer (SP): SP is a 16-bit register pointing to program stack in stack segment.
2. Base Pointer (BP): BP is a 16-bit register pointing to data in stack segment. BP register is usually used for based, based indexed or register indirect addressing.
3. Instruction Pointer (IP): IP is a 16-bit register pointing to next instruction to be executed.

Index registers

The Index Registers are as follows:-

1. Source Index (SI): SI is a 16-bit register used for indexed, based indexed and register indirect addressing, as well as a source data addresses in string manipulation instructions.
2. Destination Index (DI): DI is a 16-bit register. DI is used for indexed, based indexed and register indirect addressing, as well as a destination data addresses in string manipulation instructions.

Flag Registers

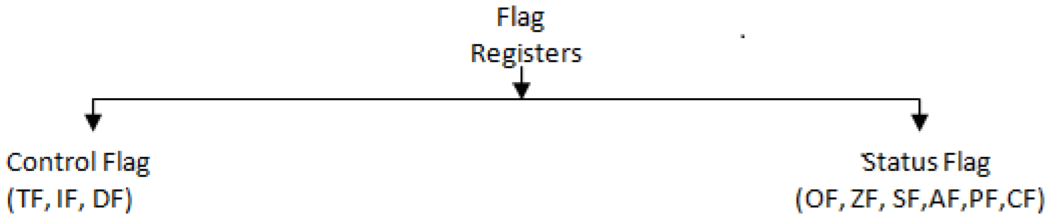
The 16-bit flag register of 8086 contains 9 active flags (six conditional & 3 control flags), other 7 flags are undefined.

3. Draw the format of a Flag register of an 8086 microprocessor.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U	U	U	U	OF	DF	IF	TF	SF	ZF	U	AF	U	PF	U	CF

Figure:8086 Flag Register

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



- Status Flags: It indicates certain condition that arises during the execution. They are controlled by the processor.
- Control Flags: It controls certain operations of the processor. They are deliberately set/ reset by the user.

Control Flags

Control flags are set or reset deliberately to control the operations of the execution unit.

1. **Trap Flag (TF):**
   - It is used for single step control.
   - It allows user to execute one instruction of a program at a time for debugging.
   - When trap flag is set, program can be run in single step mode.
2. **Interrupt Flag (IF):**
   - It is an interrupt enable/disable flag.
   - If it is set, the mask able interrupt of 8086 is enabled and if it is reset, the interrupt is disabled.
   - It can be set by executing instruction sit and can be cleared by executing CLI instruction.
3. **Direction Flag (DF):**
   - It is used in string operation.
   - If it is set, string bytes are accessed from higher memory address to lower memory address.
   - When it is reset, the string bytes are accessed from lower memory address to higher memory address.

Status Flag

1. **Carry Flag (CF):** This flag indicates an overflow condition for unsigned integer arithmetic. It is also used in multiple-precision arithmetic.
2. **Auxiliary Flag (AF):** If an operation performed in ALU generates a carry/barrow from lower nibble (i.e. D0 – D3) to upper nibble (i.e. D4 – D7), the AF flag is set i.e. carry given by D3 bit to D4 is AF flag. This is not a general-purpose flag, it is used internally by the processor to perform Binary to BCD conversion.
3. **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.
4. **Zero Flag (ZF):** It is set; if the result of arithmetic or logical operation is zero else it is reset.
5. **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.
6. **Overflow Flag (OF):** It occurs when signed numbers are added or subtracted. OF=1 indicates that the result has exceeded the capacity of machine.

4. Explain Segmentation in 8086

Segment Register in 8086

Types of Segment registers are as follows:-

1. Code Segment (CS): executable program is stored in CS
2. Data Segment (DS): The DS contains most data used by program. Data are accessed in the Data Segment by an offset address or the content of other register that holds the offset address.
3. Stack Segment (SS): SS defined the area of memory used for the stack.
4. Extra Segment (ES): ES is additional data segment.

Segmentation

In Segmentation, the total memory size is divided into segments of various sizes.

What is Segment?

Segment is just an area in memory.

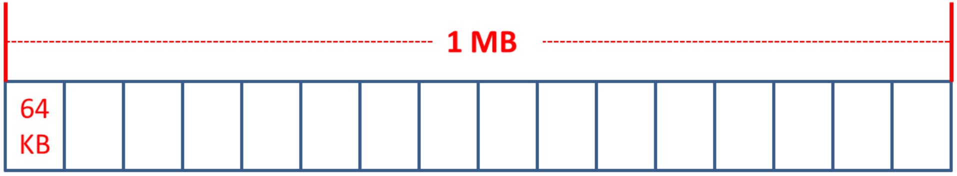
What is Segmentation?

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

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.



Segment = Total memory available/size of each
Segment = 1MB/64KB
= 1024KB/64KB

Total Segments = 16 segments

- Intel 8086 has 20 lines address bus.
- With 20 address lines, the memory that can be addressed is 2^{20} bytes.

2^{20} = 1,048,576 bytes
1 MB = 1111 1111 1111 1111 1111
= **FFFFF H**

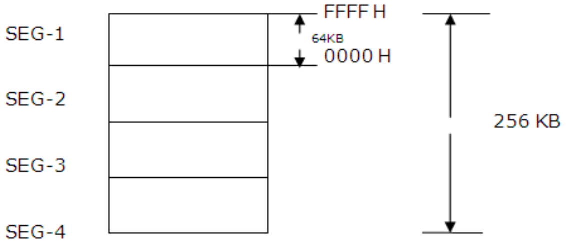


Figure: Segmentation in 8086

- 8086 memory with address ranging from 000000 H to FFFFFF H.
- Size of each Segment Register is 16-bit

2^{16} = 65535 bytes = **64K** [size of each segment]

How to calculate physical address from segment address?

- Segment Registers are used to hold the upper 16-bit of the starting address for each of the segment.
- The 16-bit address is the starting address of the segment from where the BIU is currently fetching instruction code bytes.
- The BIU always inserts zeros for the LSB of the 20-bit address for a segment, as the segment registers cannot store 20 bits, they only store the upper 16 bits.

How is a 20-bit physical address obtained if data bus is of 16-bit?

- The 20-bit address is called its Physical Address (PA).
- PA= Base Address : Offset
- Offset is the displacement of the memory location from the starting location of the segment.

E.g.

Base address DS=2222 H

Step-1: Convert DS 16-bit address to 20-bit address
- BIU appends 0H to the LSBs of the base address.
22220 H

Step-2: Retrieve offset address
- Assuming offset address = 0016 H
PA= Base Address : Offset
PA= 2222 H: 0016 H

Step-3: To calculate the effective address of the memory
Physical Address = Starting Address of Segment(20-bit) + Offset
EA = 2 2 2 2 0 H

```
+ 0 0 1 6 H
----------------
2 2 2 3 6 H
```

5. Explain 8086 pin diagram

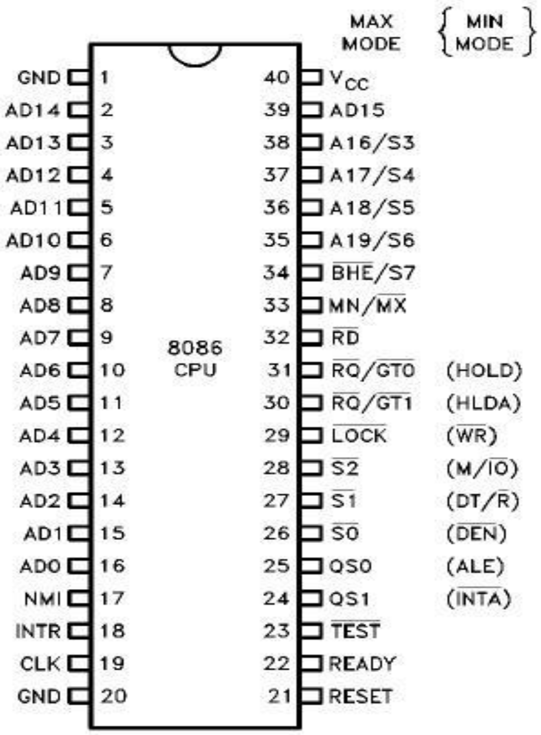


Figure:8086 Pin Diagram

Address and Data pins AD0-AD15 (bidirectional)

- These lines are multiplexed bidirectional address/data bus.
- AD0-AD7 carry lower order byte of data and AD8-AD15 carry higher order byte of data.
- When ALE=1, then Address bus gets enabled, else Data bus will get enabled.

A_{16}/S_3- A_{19}/S_6 (unidirectional)

- These lines are multiplexed and unidirectional address and status bus.
- During T_1, they carry higher order 4-bit address.
- In the remaining clock cycles, they carry status signals.
- S_5 gives the status of Interrupt Flag (IF)
- S_6 goes low, when 8086 controls the shared system bus.
- S_3 and S_4 indicates the segment register

S_4	S_3	Register
0	0	ES
0	1	SS
1	0	CS
1	1	DS

BHE/S_7

- BHE stands for Bus High Enable.
- Active low output signal.
- BHE signal is used to indicate the transfer of data over higher order data bus (D8 – D15).
- 8-bit I/O devices use this signal.
- S7 is reserved for future development.

Interrupt Related pins

NMI

- It is an active high input signal
- It is a non-mask able interrupt signal.

INTR

- It is an active high input signal
- It is an interrupt request signal.

INTA

- It is an active low output signal.
- This is an interrupt acknowledge signal.
- When microprocessor receives INTR signal, it acknowledges the interrupt by generating this signal.

Clock Related pins

CLK

Generates clock signals that synchronize the operation of processor.

RESET: Active high input signal

- When high, microprocessor enters into reset state and Terminates activity of processor
- Processor requires 4 clock cycle to reset. Thus RESET signal must be 1 for at least 4 clock cycles

READY: Active high input signal

- This is an acknowledgement signal from I/O devices or memory.
- When READY= high; it indicates that the device is ready to transfer data.
- When READY=low; microprocessor is in wait state.

Control Pins

TEST

- Active low input signal.
- It is used to test the status of math coprocessor 8087.
- The BUSY pin of 8087 is connected to this pin of 8086.
- If low, execution continues, else microprocessor is in wait state.

MN/MX

8086 works in two modes:

1. Minimum Mode [Active high i/p signal]
2. Maximum Mode [Active low i/p signal]

- If MN/MX is high, it works in minimum mode.
- If MN/MX is low, it works in maximum mode.
- Pins 24 to 31 issue two different sets of signals.
- One set of signals is issued when CPU operates in minimum mode.
- Other set of signals is issued when CPU operates in maximum mode.

Mode Multiplexed pins

S_2	S_1	S_0	**Machine Language**
0	0	0	Interrupt acknowledgement
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	HALT
1	0	0	Code Access
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

DEN

- It is an active low output signal
- This is a Data Enable signal
- This signal is used to enable the transceiver

DT/R

- This is a Data Transmit/Receive signal.
- When high=>data is transmitted out low=>data is received in

M/IO

- This signal is issued by the microprocessor to distinguish memory access from I/O access.
- When high=> memory is accessed.

low=> I/O devices are accessed

QS_1	QS_0	Characteristics
0	0	No operation
0	1	First byte of opcode from queue
1	0	Empty the queue
1	1	Subsequent byte from queue

LOCK

- It is an active low output signal.
- This signal indicates that other processors should not ask CPU (8086) to hand over the system bus.
- This pin is activated by using LOCK prefix on any instruction.

¯WR

- It is an active low output signal.
- It is used to write data in memory or output signal, depending on status of M/IO signal.

HOLD

- It is an active high input signal.
- When DMA controller needs to use address/data bus, it sends a request to the CPU through this pin.
- When microprocessor receives HOLD signal, it issues HLDA signal to the DMA controller.

HLDA

- It is an active high output signal.
- It is a Hold Acknowledge signal.
- It is issued after receiving the HOLD signal.

RQ/GT_0 and RQ/GT_1

- These are Request/Grant bi-directional pins.
- Other processors request the CPU through these lines to release the system bus.
- After receiving the request, CPU sends acknowledge signal on the same lines.
- RQ/GT_0 has higher priority than RQ/GT_1

80286

6. Describe the architecture of the 80286 with a neat block diagram.

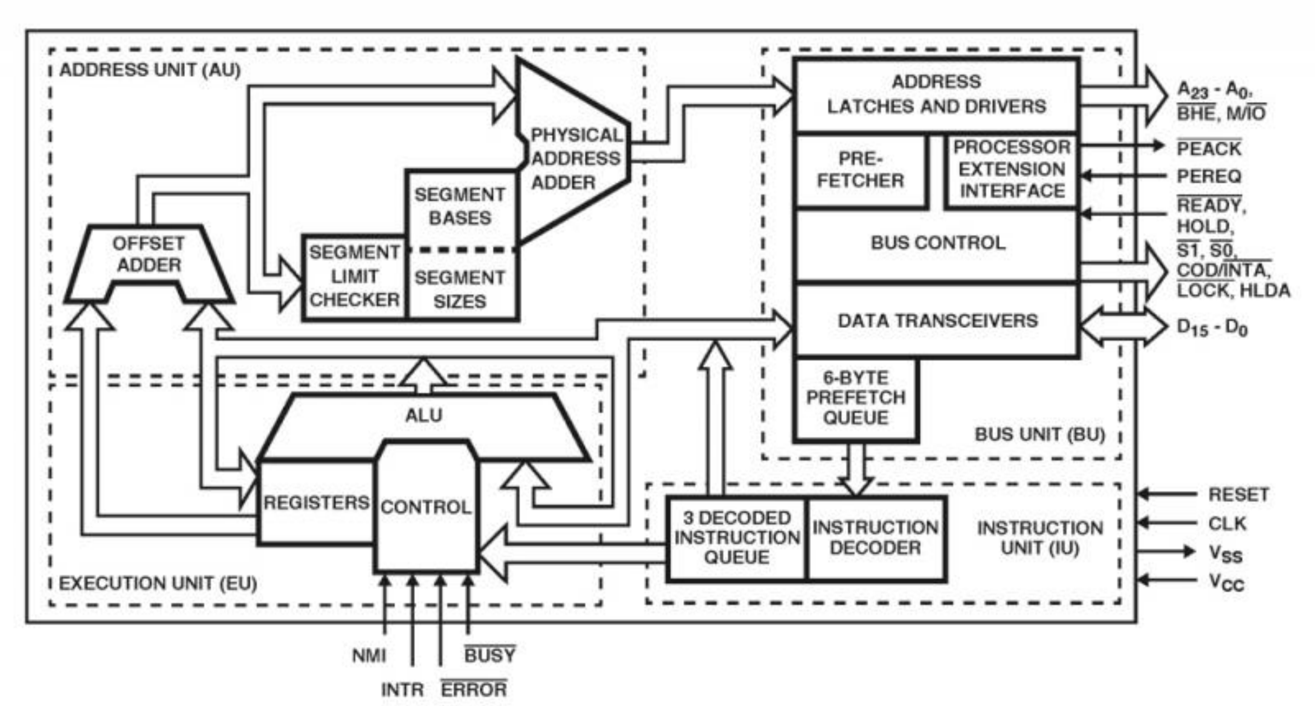


Figure: Architecture of Intel 80286

- The 80286 was designed for multi-user systems with multitasking applications, including communications and real-time process control.
- These were organized into a pipeline, significantly increasing performance.
- It was produced in a 68-pin package including PLCC (Plastic Leaded Chip Carrier), LCC (Leadless chip carrier) and PGA (Pin Grid Array) packages.
- The Intel 80286 had a 24-bit address bus and was able to address up to 16 MB of RAM, compared to 1 MB for its predecessor.

80286 Architecture contains 4 separate processing units.

1. Bus Unit (BU)
2. Instruction Unit (IU)
3. Address Unit (AU)
4. Execution Unit (EU)

Bus Unit (BU):

It has address latches, data transceivers, bus interface and circuitry, instruction pre-fetcher, processor extension interface and 6 byte instruction queue.

Functions :

- To perform all memory and I/O read and write.
- To pre-fetch the instruction bytes.

- To control the transfer of data to and from processor extension devices like 80287 math co-processor.
- Whenever BU is not using the buses for the operation, it pre-fetches the instruction bytes and put them is a 6 byte pre-fetch queue.

Instruction Unit (IU):

It has 3 decoded instruction queue and instruction decoder.

Functions :

- It fully decodes up to three prefetched instructions and holds them in a queue.
- So that EU can access them.
- It helps the processor to speed up, as pipelining of instruction is done.

Execution Unit (EU):

It includes ALU, registers and the Control unit. Registers are general purpose, index, pointer, flag register and 16 –bit Machine Status Word (MSW).

Functions :

- To sequentially execute the instructions received from the instruction unit.
- ALU result is either stored in register bank or sent over the data bus.

Address Unit (AU):

It consists of segment registers, offset address and a physical address adder.

Functions :

- Compute the physical address that will be sent out to the memory or I/O by BU.
- 80286 operate in two different modes
  1. real address mode
  2. Protected virtual address mode.
- When used in Real address mode, AU computes the address with segment base and offset like 8086. Segment register are CS, DS, ES and SS hold base address. IP, BP, SI, DI , SP hold offset.
- Maximum physical space allowed in this mode is 1MB.
- When 80286 operate in protected mode, the address unit acts as MMU.
- All 24 address lines used and can access up to 16MB of physical memory.
- If descriptor table scheme is used it can address up to 1GB of virtual memory.

7. Explain Register Organization of 80286

1. Eight 16-bit general purpose registers (AX, BX, CX, DX, SP, BP, SI, DI)
2. Four 16-bit segment registers (CS, SS, DS, ES)
3. 16-bit Instruction Pointer (IP)
4. 16-bit Flag Register Plus
5. one new 16-bit machine status word (MSW) register

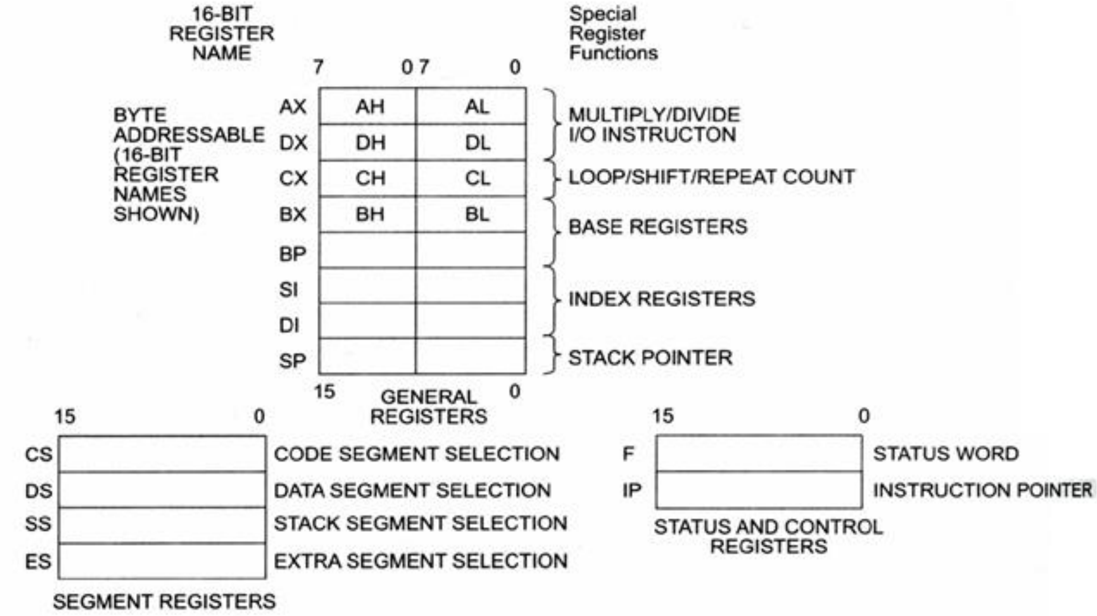


Figure: 80286 Register Set

80286 Flag Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U	**NT**	**IOPL**		**OF**	**DF**	**IF**	**TF**	**SF**	**ZF**	**U**	**AF**	**U**	**PF**	**U**	**CF**

Figure: 80286 Flag Register

- Flag register is of 32 –bit, 15th bit undefined/reserved.
- System flags: reflect the current status of machine.
  i. **IOPL – I/O Privilege Level flag**: 2 –bits are used in protected mode. It holds the privilege level from 0 to 3. ‘0’ assigns to highest privilege whereas ‘3’ assigns to lower privilege level.
  ii. **NT: Nested Task** flag: It is used in protected mode. Bit is set when one task invokes another task.

Machine Status Word (MSW) Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	**TS**	**EM**	**MP**	**PE**

Protected Mode Enable

PE=1; Places 80286 in protected mode

PE=0; It can be only cleared by resetting CPU

Task Switch
TS is automatically set whenever a task switch operation is performed.

Monitor processor extension
If set, this flag allows WAIT instruction to generate a processor extension not present exception

EMulate processor extension flag
If set, The EMulate coprocessor bit is set to cause all coprocessor opcodes to generate a Coprocessor Not Available fault.

Instructions used to load and store MSW:

1. LMSW instruction (Load Machine Status Word)
2. SMSW instruction (Store Machine Status Word)

8. Differentiate between the real mode and protected mode of an 80286 microprocessor.

Real Mode

- Address Unit computes the address with segment base and offset like 8086.
- Maximum physical space allowed in this mode is 1MB.
- When 80286 get reset, it always starts execution in real mode.

Task carried out in Real Mode

- Initializes IP and other registers of 80286
- Initializes the peripheral
- Enables interrupts
- Set up the descriptor table
- Prepares for entering in PVAM(Protected Virtual Address Mode)

Protected Virtual Address Mode (PVAM)

- 80286 is the 1st processor to support the concept of Virtual memory and Memory management.
- Here, the address unit acts as MMU.
- All 24 address lines are used and can access up to 16MB of physical memory.
- If descriptor table scheme is used it can address up to 1GB of virtual memory.

Task carried out in PVAM

- The complete virtual memory is mapped on to the 16Mbyte physical memory.
- If a program larger than 16Mbyte is stored on the hard disk and is to be executed by swapping sequentially as per sequence of execution.
- The huge programs are divided in smaller segments or pages arranged in appropriate sequence.

Real Address Mode	Protected Virtual Address Mode
Can only address 1 MB of system memory and act as fast 8086.	Can address till 16MB of System Memory
Doesn't Supports the concept of Virtual Memory	Supports the concept of Virtual Memory
Real mode provides no support for memory protection, multitasking, or code privilege levels.	Protected mode provides support for memory protection, multitasking, or code privilege levels.
Initially every processor is in Real Mode i.e MSW PE =0	Microprocessor will Switch to this mode by setting MSW PE-bit

Address Calculation: Real Mode

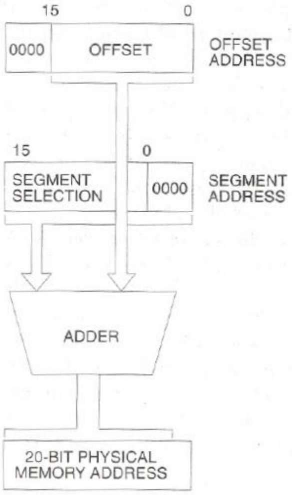


Figure: Real Mode Addressing

- Total 1MB of Memory, divided among 16-segments with each of size 64kb.
- 80286 reserves two fixed areas for
  1. System Initialization
  2. IVT (Interrupt Vector Table)

IVT-1KB of Starting address	00000H – 003FFH
System Initialization	FFFF0H – FFFFFH

Address Calculation: PVAM

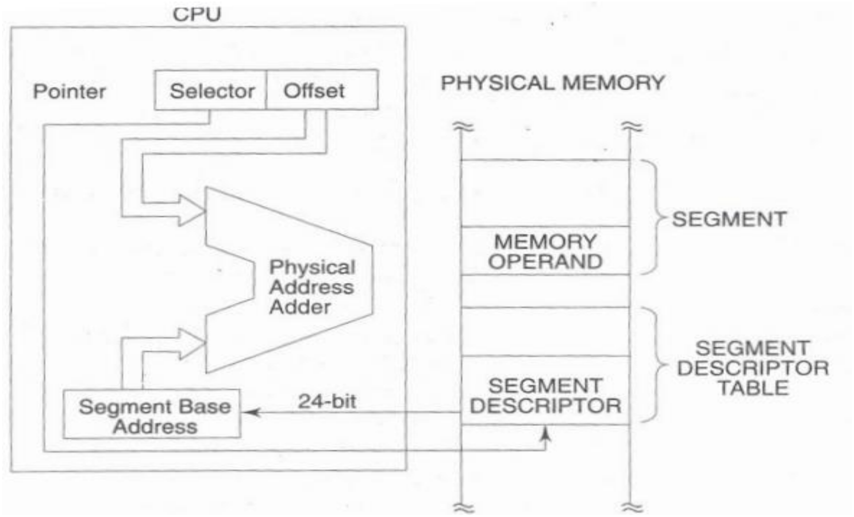


Figure: Physical Address Calculation in PVAM

- 80286 uses the 16-bit content of a segment register as a selector to address a descriptor stored in the physical memory.
- The descriptor is a block of contiguous memory locations containing information of a segment, like segment base address, segment limit, segment type, privilege level, segment availability in physical memory descriptor type and segment.
- Hardware reset is the only way to come out of protected mode

9. What is a descriptor table? What is its use? Differentiate between GDT and LDT.

Descriptor table

The descriptor is a block of contiguous memory location containing information of a segment, like

i. Segment base address
ii. Segment limit
iii. Segment type
iv. Privilege level – prevents unauthorized access
v. Segment availability in physical memory
vi. Descriptor type
vii. Segment use by another task

Use of Descriptor table

- A segment cannot be accessed, if its descriptor does not exist in either LDT or GDT.
- Set of descriptor (descriptor table) arranged in a proper sequence describes the complete program.
- The descriptor describes the location, length, and access rights of the segment of memory.
- The selector, located in the segment register, selects one of descriptors from one of two tables of descriptors.

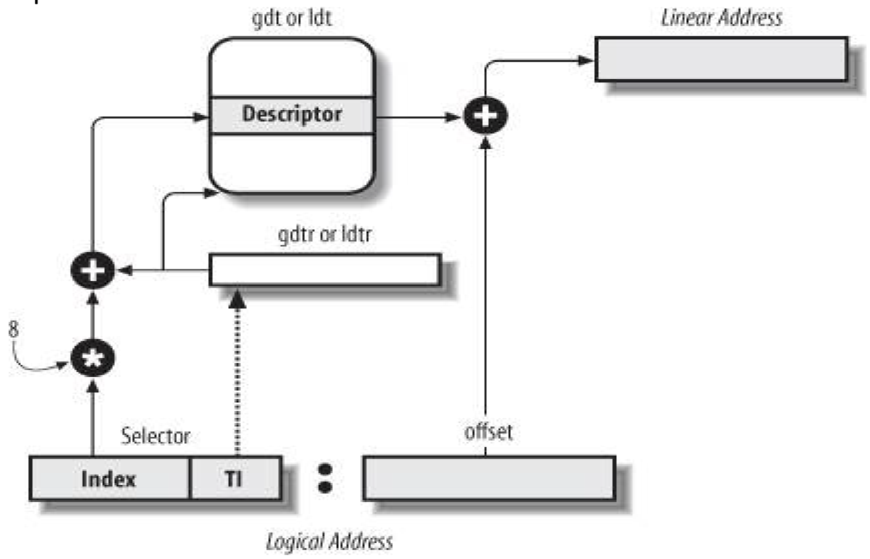


Figure: Protected Mode Addressing with Descriptor Table

Differentiate between GDT and LDT

- Each Descriptor is 8-byte long
- The GDT contains information about segments that are global in nature, that is, available to all programs and normally used most heavily by the operating system.
- The LDT contains descriptors that are application specific.
- A global descriptor might be called a system descriptor, and local descriptor an application descriptor
- The global descriptor table's base address is stored in GDTR
- The local descriptor table's base address is stored in LDTR
- GDT have only one copy in system while LDT can have many
- GDT may not change during execution while, LDT often changes when task switches
- Entry of LDT is saved in GDT.
- Entries in GDT and LDT have the same structure.
- Unlike GDT, LDT also cannot store certain privileged types of memory segments.

80386

10. Explain the Page Table and Page Directory Entry with paging mechanism in an 80386 microprocessor.

Page Directory

- The page directory contains the location of up to 1024 page translation tables, which are each four bytes long.
- Each page translation table translates a logical address into a physical address.
- The page directory is stored in the memory and accessed by the page descriptor address register (CR3).
- Control register CR3 holds the base address of the page directory, which starts at any 4K-byte boundary in the memory system.
- Each entry in the page directory translates the leftmost 10 bits of the memory address. This 10-bit portion of the linear address is used to locate different page tables for different page table entries.

Page Directory Entry

- Total Page Directory Entries are1024
- Each directory entry is of 4 byte

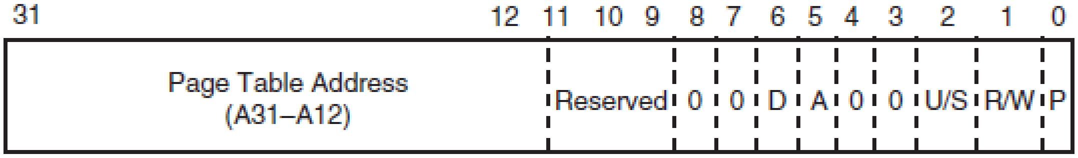


Figure: Page Directory Entry

Page Table

- The page table contains 1024 physical page addresses, accessed to translate a linear address into a physical address.

Page Table Entry

- The page table entries contain the starting address of the page and the statistical information about the page.
- Total Entries are 1024
- Each page table entry is of 4byte

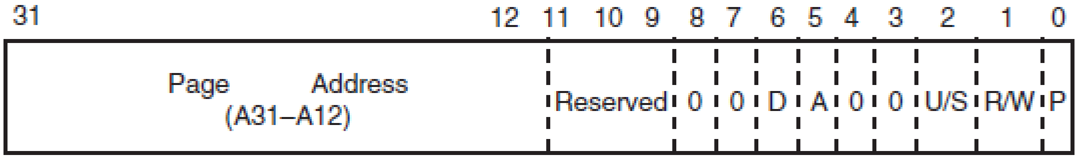


Figure: Page Table Entry

- **D-bit**: Dirty bit is undefined for page table directory entries by the 80386 microprocessor and is provided for use by the operating system.

- **A-bit:** Accessed bit is set to a logic 1 whenever the microprocessor accesses the page directory entry.
- **R/W and Read/write and user/supervisor** are both used in the protection scheme. Both bits combine to develop paging priority level protection for level 3, the lowest user level.

U/S	R/W	Access Level 3
0	0	None
0	1	None
1	0	Read-Only
1	1	Write-Only

- **P-bit**: Present bit, if logic 1 indicates that the entry can be used in address translation. If P = 0, the entry cannot be used for translation. When P = 0, the remaining bits of the entry can be used to indicate the location of the page on the disk memory system.

Page Translation Mechanism in 80386

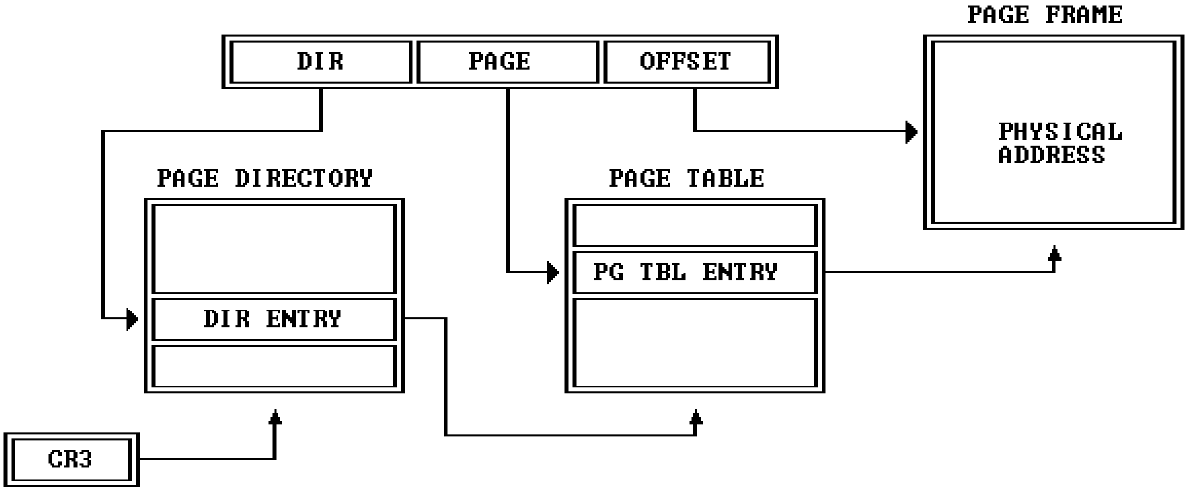


Figure: Page Translation

- A page frame is a 4K-byte unit of contiguous addresses of physical memory. Pages begin on byte boundaries and are fixed in size.
- A linear address refers indirectly to a physical address by specifying a page table, a page within that table, and an offset within that page.
- The below figure of page translation shows, how processor converts the DIR, PAGE, and OFFSET fields of a linear address into the physical address by consulting two levels of page tables.
- The addressing mechanism uses the DIR field as an index into a page directory, uses the PAGE field as an index into the page table determined by the page directory, and uses the OFFSET field to address a byte within the page determined by the page table.

- In the second phase of address transformation, the 80386 transforms a linear address into a physical address.
- This phase of address transformation implements the basic features needed for page-oriented virtual-memory systems and page-level protection.
- Page translation is in effect only when the PG bit of CR_0 is set.

11. Explain Privilege level.

- There are four types of privilege levels
- 00 - kernel level (highest privilege level)
- 01 - OS services
- 10 - OS extensions
- 11 - Applications (lowest privilege level)

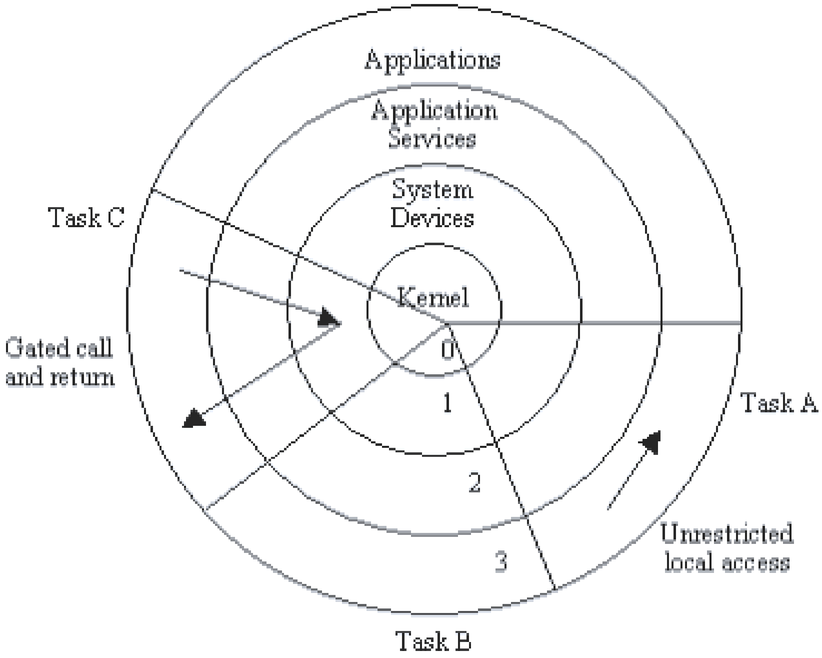


Figure: Privilege Level

- Each task assigned a privilege level, which indicates the priority or privilege of that task.
- It can only change by transferring the control, using gate descriptors, to a new segment.
- A task executing at level 0, the most privileged level, can access all the data segment defined in GDT and LDT of the task.
- A task executing at level 3, the least privileged level, will have the most limited access to data and other descriptors.
- The use of rings allows for system software to restrict tasks from accessing data.
- In most environments, the operating system and some device drivers run in ring 0 and applications run in ring 3.

12. Features of 80386

- The 80386 microprocessor is an enhanced version of the 80286 microprocessor
- Memory-management unit is enhanced to provide memory paging.
- The 80386 also includes 32-bit extended registers and a 32-bit address and data bus.These extended registers include EAX, EBX, ECX, EDX, EBP, ESP, EDI, ESI, EIP and EFLAGS.
- The 80386 has a physical memory size of 4GBytes that can be addressed as a virtual memory with up to 64TBytes.
- The 80386 is operated in the pipelined mode, it sends the address of the next instruction or memory data to the memory system prior to completing the execution of the current instruction
- This allows the memory system to begin fetching the next instruction or data before the current is completed. This increases access time.
- The instruction set of the 80386 is enhanced to include instructions that address the 32-bit extended register set.
- The 80386 memory manager is similar to the 80286, except the physical addresses generated by the MMU are 32 bits wide instead of 24-bits.
- The concept of paging is introduced in 80386
- 80386 support three operating modes:
  1. Real Mode (default)
  2. Protected Virtual Address Mode (PVAM)
  3. Virtual Mode
- The memory management section of 80386 supports virtual memory, paging and four levels of protection.
- The 80386 includes special hardware for task switching.

13. Explain the architecture of the 80386 with a neat block diagram.

- The internal architecture of the 80386 includes six functional units that operate in parallel. The parallel operation is called as pipeline processing.
- Fetching, decoding execution, memory management, and bus access for several instructions are performed simultaneously.
- The six functional units of the 80386 are
  1. Bus Interface Unit
  2. Code Pre-fetch Unit
  3. Instruction Decoder Unit
  4. Execution Unit
  5. Segmentation Unit
  6. Paging Unit

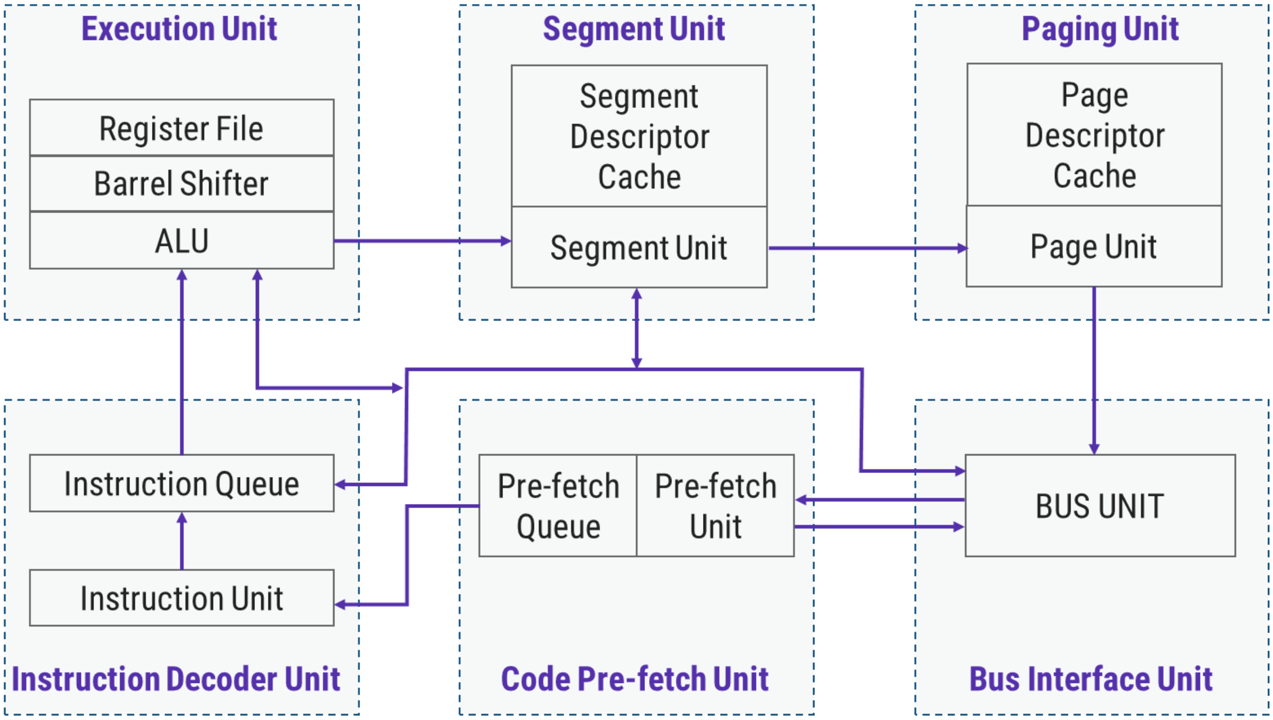


Figure: 80386 Architecture

- The Bus Interface Unit connects the 80386 with memory and I/O. Based on internal requests for fetching instructions and transferring data from the code pre-fetch unit, the 80386 generates the address, data and control signals for the current bus cycles.
- The code pre-fetch unit pre-fetches instructions when the bus interface unit is not executing the bus cycles. It then stores them in a 16-byte instruction queue for decoding by the instruction decode unit.
- The instruction decode unit translates instructions from the pre-fetch queue into micro-codes. The decoded instructions are then stored in an instruction queue (FIFO) for processing by the execution unit.
- The execution unit processes the instructions from the instruction queue. It contains a control unit, a data unit and a protection test unit.
- The control unit contains microcode and parallel hardware for fast multiply, divide and effective address calculation. The unit includes a 32-bit ALU, 8 general purpose registers and a 64-bit barrel shifter for performing multiple bit shifts in one clock. The data unit carries out data operations requested by the control unit.
- The protection test unit checks for segmentation violations under the control of microcode.
- The segmentation unit calculates and translates the logical address into linear addresses at the request of the execution unit.
- The translated linear address is sent to the paging unit. Upon enabling the paging mechanism, the 80386 translates these linear addresses into physical addresses. If paging is not enabled, the physical address is identical to the linear address and no translation is necessary.

14. Register organization of 80386

The Register organization of 80386 is as follows:

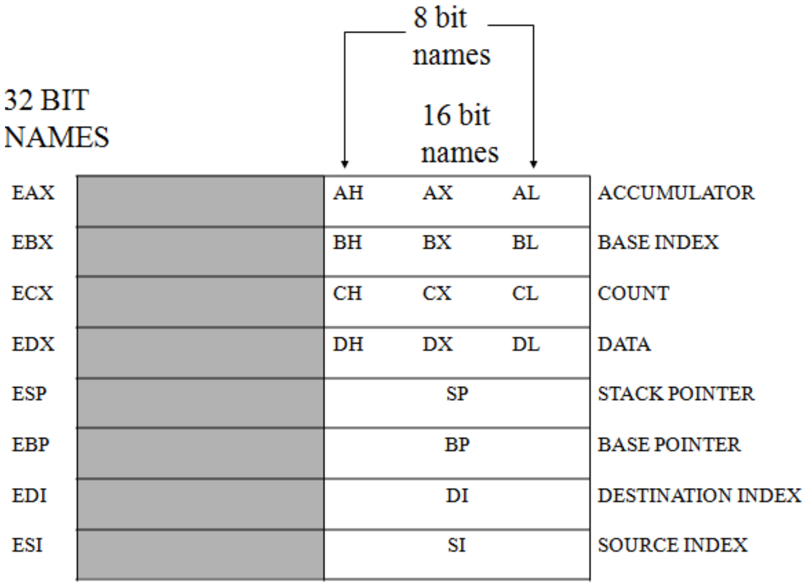


Figure:80386 General Purpose, Index and Pointer Register

General Purpose Register

- Registers EAX, EBX, ECX, EDX, EBP, EDI and ESI are regarded as general purpose or multipurpose registers.
- EAX (ACCUMULATOR): The accumulator is used for instructions such as multiplication, division and some of the adjustment instructions. In 80386 and above, the EAX register may also hold the offset address of a location in memory system.
- EBX (BASE INDEX): This can hold the offset address of a location in the memory system in all version of the microprocessor. It the 80386 and above EBX also can address memory data.
- ECX (count): This acts as a counter for various instructions.
- EDX (data): EDX is a general-purpose registers that holds a part of the result for multiplication or part of the division. In the 80386 and above this register can also address memory data.
- Pointer and Index Register
- EBP (Base Pointer): EBP points to a memory location in all version of the microprocessor for memory data transfers.
- ESP (Stack Pointer): ESP addresses an area of memory called the stack. The stack memory is a data LIFO data structure. The register is referred to as SP if used in 16 bit mode and ESP if referred to as a 32 bit register.
- EDI (Destination index): EDI often addresses string destination data for the string instruction. It also functions as either a 32-bit (EDI) or 16-bit (DI) general-purpose register.
- ESI (Source index): ESI can either be used as ESI or SI. It is often used to the address source string data for the string instructions. Like EDI ESI also functions as a general-purpose registers.

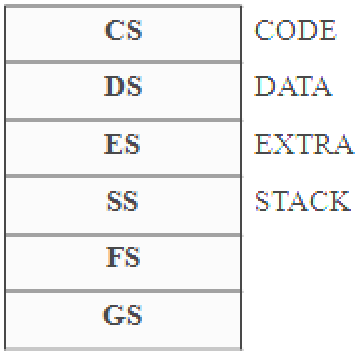


Figure:80386 Segment Register

- CS (Code): The code segment is a section of memory that holds the code used by the microprocessor. The code segment registers defines the starting address of the section of memory holding code.
- SS (Stack): The stack segment defines the area of memory used for the stack. The stack entry point is determined by the stack segment and stack pointer registers. The BP registers also addresses data within the stack segment.
- DS (Data) – The data section contains most data used by a program. Data are accessed in the data segment by an offset address of the contests of other registers that hold the offset address.
- ES (extra) – The extra segment is used to hold information about string transfer and manipulation
- FS and GS – These are supplement segment registers available in the 80386 and above microprocessors to allow two additional memory segments for access by programs.

EIP (Instruction Pointer): EIP addresses the next instruction in a section of memory defined as a code segment. This register is IP (16bit) when microprocessor operates in the real mode and EIP (32 bits) when 80386 and above operate in protected mode

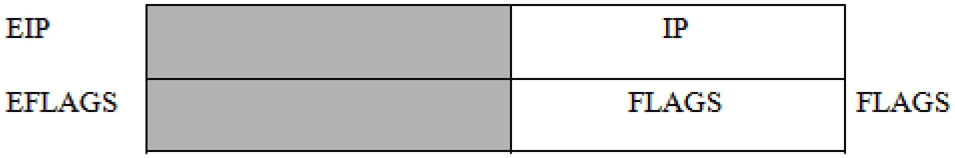


Figure:80386 Instruction Pointer and Flag Register

Flag Register:

Indicates the condition of the microprocessor and controls its operations. Flag registers are also upward compatible since the 8086-80268 have 16bit registers and the 80386 and above have EGLAF register (32 bits)

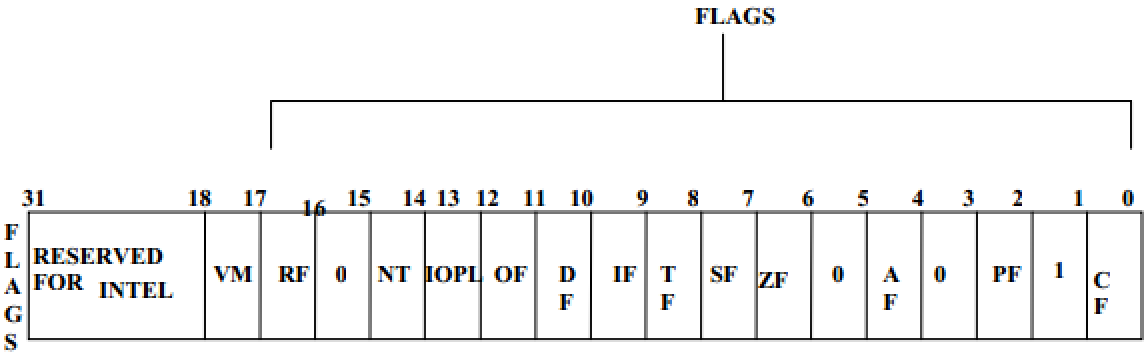
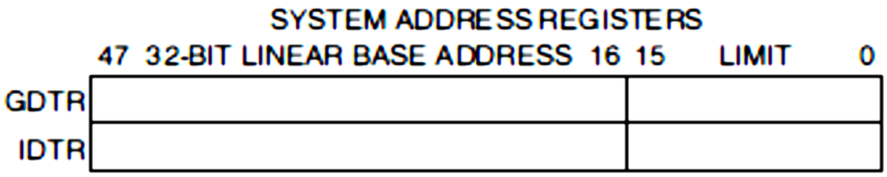
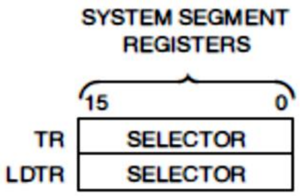


Figure: 80386 Flag Register

- IOPL (I/O Privilege level): IOPL is used in protected mode operation to select the privilege level for I/O devices. If the current privilege level is higher or more trusted than the IOPL, I/O executed without hindrance. If the IOPL is lower than the current privilege level, an interrupt occurs, causing execution to suspend. Note that an IPOL is 00 is the highest or more trusted; if IOPL is 11, it's the lowest or least trusted.
- NT (Nested Task): The nested task flag is used to indicate that the current task is nested within another task in protected mode operation. This flag is when the task I nested by software.
- RF (Resume): The resume flag is used with debugging to control the resumption of execution after the next instruction.
- VM (Virtual Mode): The VM flag bit selects virtual mode operation in a protected mode system.
- Note: All the other flag bit is having similar description as in 8086 flag register.
- System Address Register:
- Four memory management registers are used to specify the locations of data structures which control segmented memory management.
- GDTR (Global Descriptor Table Register) and IDTR (Interrupt Descriptor Table Register) be loaded with instructions which get a 6 byte data item from memory



- LDTR (Local Descriptor Table Register) and TR (Task Register) can be loaded with instructions which take a 16-bit segment selector as an operand.



Special 80386 Register

- Control Register: Four Control Register (CR0-CR3)
- Debug Register: Eight Debug Register (DR0-DR7)
- Test Register: Two Test Register (TR6-TR7)

15. Briefly explain Real, PVAM and Virtual 8086 mode of 80386 microprocessor.

Real Modes of 80386

- Default Mode
- After reset, the 80386 starts from the memory location FFFFFFF0 H under real address mode.
- In real address mode, 80386 works as a fast 8086 with 32 bit registers and data types.
- Real-address mode is in effect after a signal on the RESET pin. Even if the system is going to be used in protected mode, the start-up program will execute in real-address mode temporarily while initializing for protected mode.
- The addressing techniques, memory size, interrupt handling in this mode of 80386 are similar to the real addressing mode of 80286.
- In real address mode, the default operand size is 16 bit but 32 bit operands and addressing modes may be used with the help of override prefixed.
- Maximum physical memory = 1Mega byte (1MB)
- The only way to leave real-address mode is to switch to protected mode.

PVAM of 80386

- 32-bit address bus => access up to 232 bytes = 22. 230 B = 4 GB
- Base address => 32-bit value
- Offset =>16-bit or 32-bit value
- Linear address = base address + offset
- Linear address → physical address with paging
- In protected mode, the segment registers contain an index into a table of segment descriptors.
- Each
  segment descriptor contains the start address of the segment, to which the offset is added to generate the address.
- In addition, the segment descriptor contains memory protection information.
- This includes an offset limit and bits for write and read permission.
- This allows the processor to prevent memory accesses to certain data.
- Protected mode is accessed by placing a logic 1 into the PE bit of CR0
- This system contains one data segment descriptor and one code segment descriptor with each segment set to 4G bytes in length.
- PVAM mode support memory management, virtual memory, multitasking, protection, debugging, segmentation and paging.

Virtual Mode of 80386

- In its protected mode of operation, 80386DX provides a virtual 8086 operating environment to execute the 8086 programs.
- The real mode can also use to execute the 8086 programs along with the capabilities of 80386, like protection and a few additional instructions.
- Once the 80386 enters the protected mode from the real mode, it cannot return back to the real mode without a reset operation.
- Thus, the virtual 8086 mode of operation of 80386, offers an advantage of executing 8086 programs while in protected mode.
- The address forming mechanism in virtual 8086 mode is exactly identical with that of 8086 real mode.
- In virtual mode, 8086 can address 1Mbytes of physical memory that may be anywhere in the 4Gbytes address space of the protected mode of 80386.
- Like 80386 real mode, the addresses in virtual 8086 mode lie within 1Mbytes of memory.
- In virtual mode, the paging mechanism and protection capabilities are available at the service of the programmers.
- The 80386 supports multiprogramming, hence more than one programmer may be use the CPU at a time.
- Paging unit may not be necessarily enable in virtual mode, but may be needed to run the 8086 programs which require more than 1Mbyts of memory for memory management function.
- In virtual mode, the paging unit allows only 256 pages, each of 4Kbytes size.
- Each of the pages may be located anywhere in the maximum 4Gbytes physical memory.
- The virtual mode allows the multiprogramming of 8086 applications.
- The virtual 8086 mode executes all the programs at privilege level 3.
- Any of the other programmers may deny access to the virtual mode programs or data.
- Even in the virtual mode, all the interrupts and exceptions are handled by the protected mode interrupt handler.
- To return to the protected mode from the virtual mode, any interrupt or execution may be used.
- As a part of interrupt service routine, the VM bit may be reset to zero to pull back the 80386 into protected mode.