10/2/17

UNIT - II

THE MICROPROCESSOR & ITS ARCHITECTURE

- Introduction

→ The microprocessor as a programmable device

→ The architecture of Intel microprocessors.

→ ways that the family members address the memory system.

→ Addressing modes are described for the real, protected & flat modes of operation.

1.9) Internal Microprocessor Architecture

→ Before a program is written or instruction investigated, internal configuration of the microprocessor must be known.

→ In a multiple core µp each core containing the same programming model.

→ Each core runs a separate task or thread simultaneously.

The programming model.

→ 8086 through Core2 consider two types of registers

1) program visible → registers are used during programming & are specified by the instructions.

2) program invisible → not addressable directly during applications programming.

→ 80286 & above contain program - invisible registers to control & operate protected memory.

→ 80386 through core2 microprocessors contain full 32-bit internal architectures.

→ 8086 through the 80286 are fully upward - Compatible to the 8086 through core2.

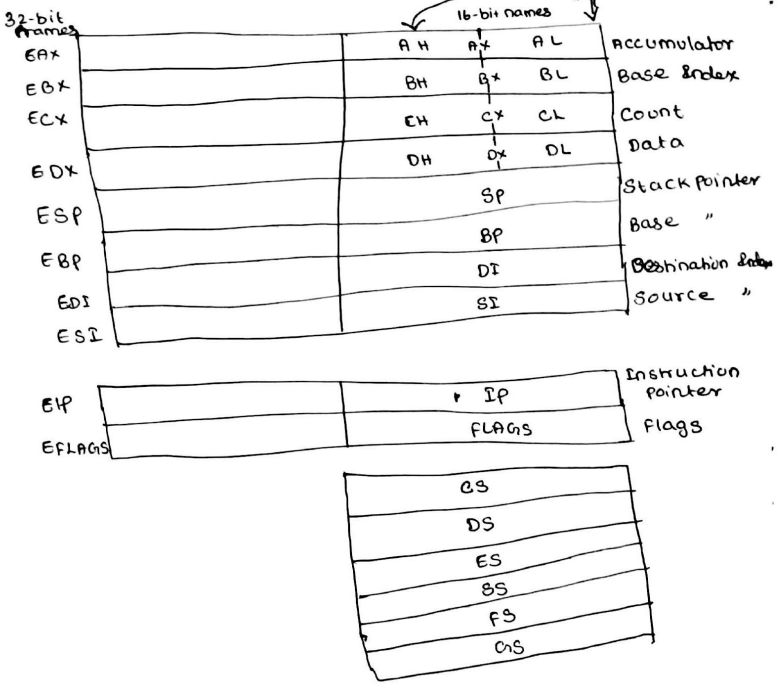


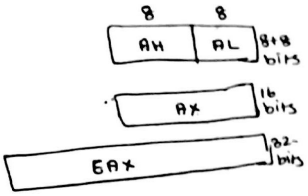
Fig: Programming model of 8086 through core2 MP including 64-bit extensions.

Accessing parts of register.

→ use 8-bit, 16-bit or 32-bit name.

→ Applies to EAX, EBX, ECX & EDX

32-bit	16-bit	8-bit High	8-bit Low
EAX	AX	AH	AL
EBX	BX	BH	BL
ECX	CX	CH	CL
EDX	DX	DH	DL



Multipurpose (General) Registers.

→ The top portion of the programming model contains the general purpose registers are EAX, EBX, ECX, EDX, EBP, ESI & EDI

→ These registers although general in nature each have special purposes & names.

RAX → a 64-bit register (RAX), a 32-bit register (EAX), a 16-bit register (AX) or as either of two 8-bit registers (AH & AL)

→ The accumulator is used for instructions such as multiplication, division and some of the adjustment instructions.

→ Intel plans to expand the address bus to 52 bits to address 4P (peta) bytes of memory.

RBx → addressable as RBx, EBx, Bx, BH & BL.

→ Bx register (Base index) sometimes holds offset address of a location in the memory system in all versions of the µp.

RCx → addressable as RCx, EBx, Cx, CH & CL.

→ Count is a general-purpose register that also holds the count for various instructions.

RDx → addressable as RDx, EDx, Dx, DH & DL

→ data is a general-purpose register that holds a part of the result from a multiplication or part of dividend before a division.

RBp → as RBP, EBP or BP

→ Points to a memory (base pointer) location for memory data transfers.

RDI → addressable as RDI, EDI, or DI

→ often addresses (destination index) string destination data for the string instructions.

RSI → used as RSI, ESI or SI

→ The (source index) register addresses source string data for the string instructions.

→ RDI & RSI also function as a general-purpose register.

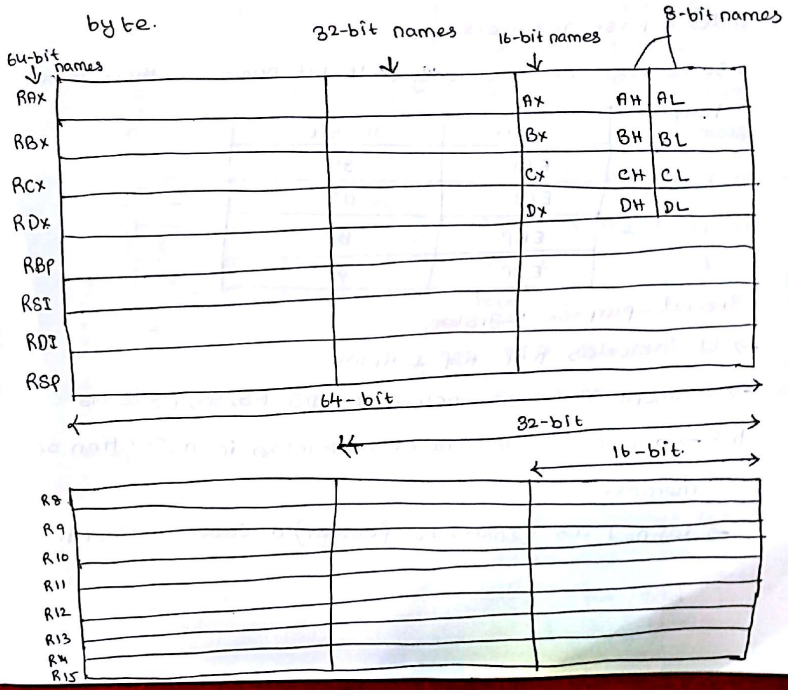
R8-R15 → registers found in the pentium 4 & core2 if 64-bit extensions are enabled.

→ Data are addressed as 64-bit, 32-bit, 16-bit or 8-bit sizes & are of general purpose.

→ most applications will not use these registers until 64-bit processors are common.

→ The 8-bit portion is the rightmost 8-bit only.

→ bit 8 to 15 are not directly addressable as a byte.



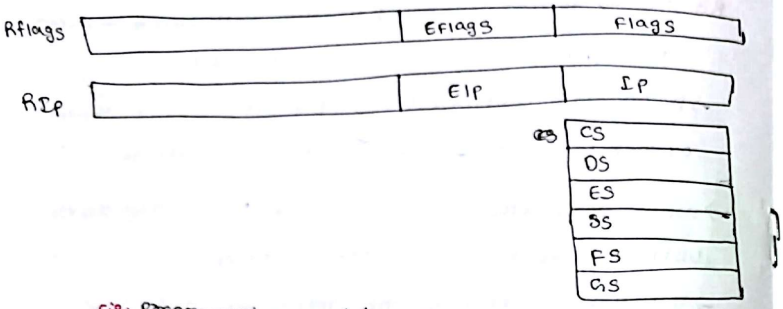


Fig: Programming model of 8086 through core2 IP. including 64-bit extensions.

Index & Base registers.

→ Some registers have only a 16-bit name for their lower half.

32-bit	16-bit
ESI	SI
EDI	DI
EBP	BP
ESP	SP

Special-purpose Registers.

→ It includes RIP, RSP & RFlags.

→ Segment registers include CS, DS, ES, SS, FS & GS.

RIP → addresses the next instruction in a section of memory

→ defined as (Instruction pointer) a code segment.

RSP → Addresses an area of memory called the stack.

→ the (stack pointer) stores data through this pointer.

RFLAGS → indicate the condition of the IP & ctrl its operation.

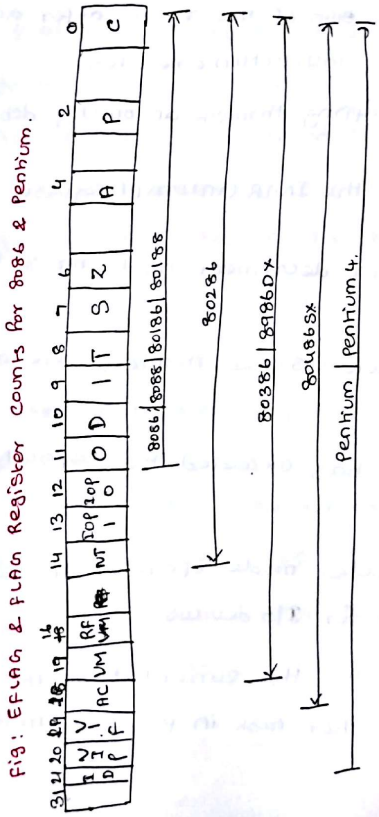


Fig: EFLAG & FLAG Register counts for 8086 & Pentium.

→ Flags are upward-compatible from the 8086/8088 through core2.

→ The rightmost five & overflow flag are changed by most arithmetic & logic operations.

→ Flags never change for any data transfer or program ctrl operation

→ Some of the flags are also used to ctrl features found in the IP.

C → holds the carry after addition or borrow after subtraction.

P → count of 1's in the number expressed as odd or even.
logic 0 for odd
" 1 " even.

A → holds the carry after addition or borrow after subtraction b/w bit 3 & 4 of the result.

Z → The result of an arithmetic or logic operation is zero.

S → holds the arithmetic sign of the result after an arithmetic or logic instruction executes.

T → trap flag enables stepping through an on-chip debugging features.

I → Controls operation of the INTR (interrupt Request) i/p pin.

D → Selects increment or decrement mode for SI and/or DI registers.

O → overflow occurs when signed numbers are added or subtracted.

→ Indicates the result has exceeded the capacity of the machine

IopL → used in protected mode operation to select the privilege level for I/o devices.

NT → nested task indicates the current task is nested within another task in protected mode operation.

RF → resume used with debugging to ctrl resumption of execution after the next instruction

VM → Virtual Mode bit selects virtual mode operation in a protected mode system.

AC → alignment check bit activates if a word or double word is addressed on a non-word or non-double word boundary.

VIF → Virtual Interrupt is a copy of the interrupt flag bit available to the pentium 4.

VIP → Virtual Interrupt pending provides information about a virtual mode interrupt for pentium.

→ used in multitasking environments to provide virtual interrupt flags.

ID → Identification flag indicates that the pentium µPs support the CPUID instruction

→ CPUID instruction provides the system with information about the pentium microprocessor.

Segment Registers.

→ Generate memory addresses when combined with other registers in the µp

→ 4 or 6 segment registers in various versions of the µp.

→ A segment register functions differently in real mode than in protected mode.

→ following is a list of each segment register, along with its function in the system.

CS → Segment holds code (program & procedures) used by the µp.

DS → Segment contains most data used by a program.

→ Data are accessed by an offset address or contents of other registers that hold the offset address.

ES → an additional data segment used by some instructions to hold destination data.

SS → defines the area of memory used for the Stack

→ Stack entry point is determined by the Stack Segment & Stack pointer registers.

→ the BP register also addresses data within the Stack segment.

S.JAGADEESAN

FS & GS → Segments are supplemental segment registers available in 80386 - core2 µps.

→ allow two additional memory segments for access by programs.

Note: Windows uses these segments for internal operations, but no definition of their usage is available.