COURSE: UCS1502 - MICROPROCESSORS AND INTERFACING

Interrupts

Dr. K. R. Sarath Chandran
Assistant Professor, Dept. of CSE

This presentation covers

• Details of 8086 interrupts

Learning Outcome of this module

• To understand about 8086 interrupts and its implementation

Interrupts

- Microprocessors allow normal program execution to be interrupted by some external signal or by a special instruction in the program.
- The microprocessor responds to that interrupt by stop executing its current program and calls a procedure called ISR (Interrupt Service Routine), which is a short program to instruct the microprocessor on how to handle the interrupt.
- An IRET instruction at the end of the ISR returns execution to the interrupted program.

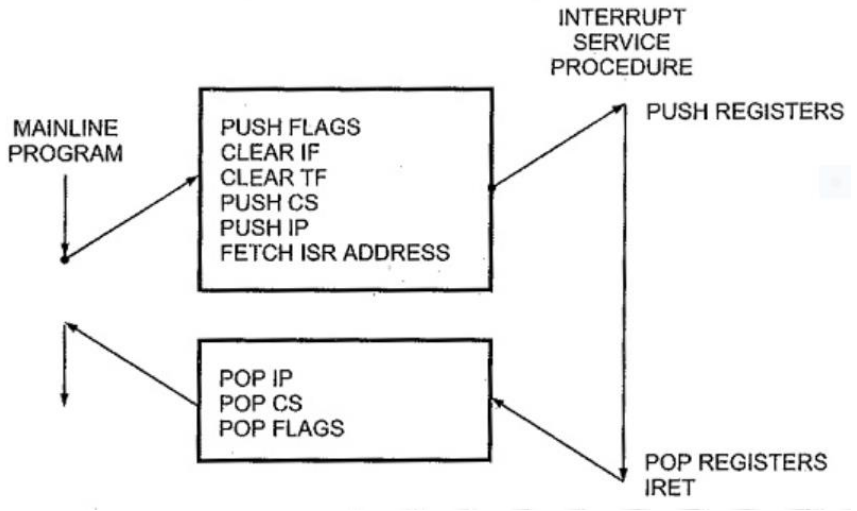
Three sources

1. External signals (through INTR and NMI) -Hardware interrupts
2. Interrupt instructions (INT N - Software interrupts)
3. Error conditions (E.g.: divided by zero)

Interrupts

At the end of each instruction cycle, 8086 checks to see if any interrupts have been requested. If it has been requested, 8086 responds to the interrupt through the following series of major actions

- Decrements stack pointer by 2 and pushes the flag register to the stack
- Disables INTR input by clearing interrupt flag (IF)
- Resets trap flag (TF)
- Decrements stack pointer by 2 and pushes CS
- Decrements stack pointer by 2 and pushes IP
- Performs an indirect far jump to the start of the ISR



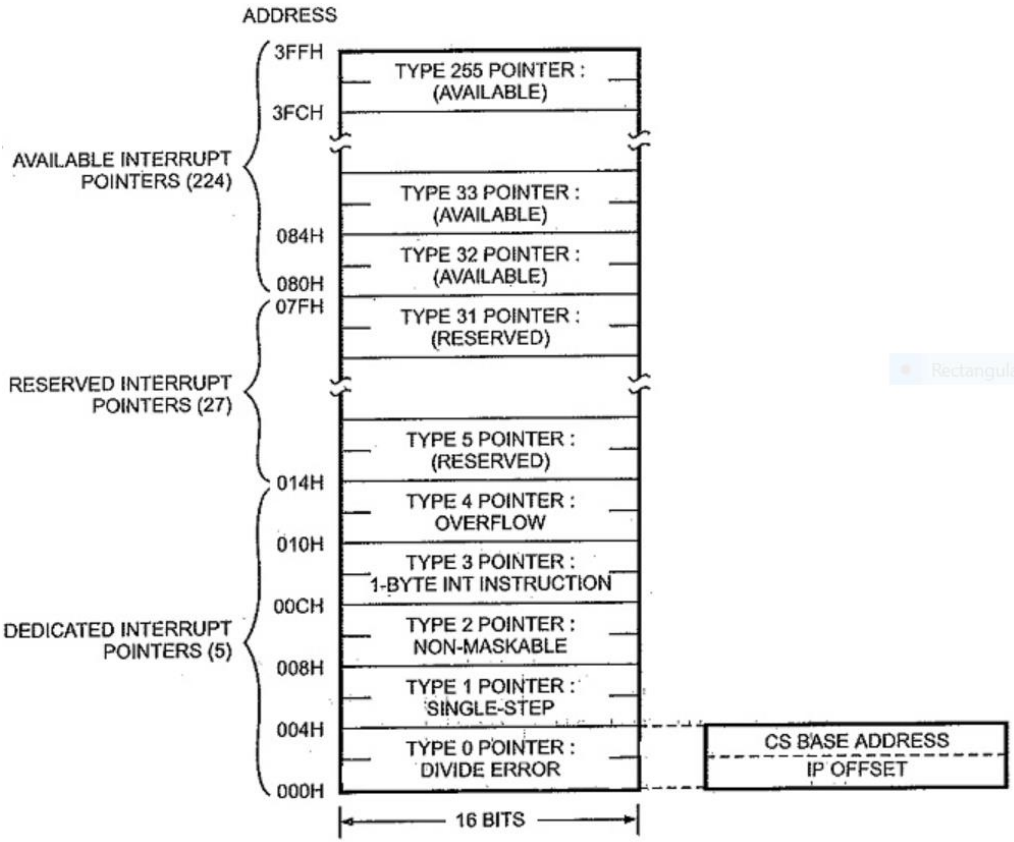
An IRET instruction at the end of the interrupt service procedure returns execution to the main program.

Interrupts

Interrupt Vector Table

How to get the new CS and IP value of ISR?

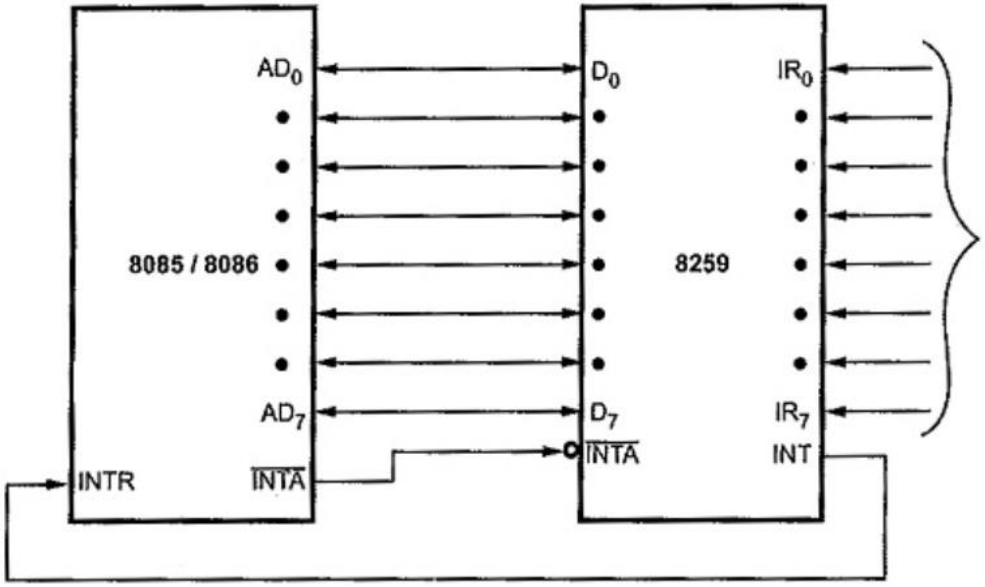
- The 8086 gets the new values of CS and IP register from four memory addresses.
- When it responds to an interrupt, the 8686 goes to memory locations to get the CS and IP values for the start of the interrupt service routine.
- In an Interrupt Structure of 8086 system the first 1 Kbyte of memory from 00000H to 003FFH is reserved for storing the starting addresses of interrupt service routines.
- This block of memory is often called the **Interrupt Vector Table in 8086** or the **interrupt pointer table**.

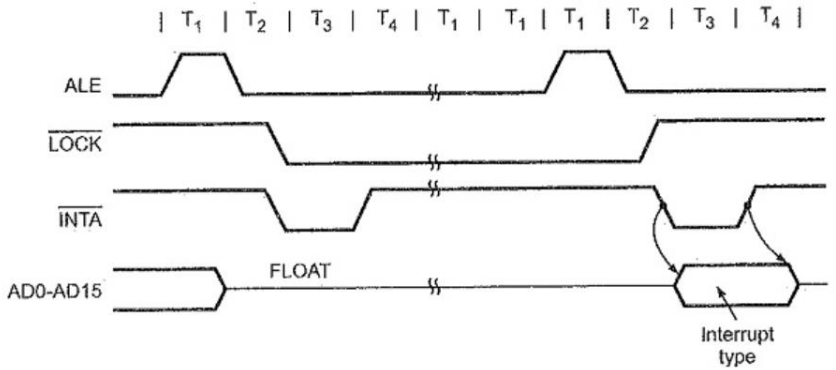


Each interrupt type is given a number between 0 to 255 and the address of each interrupt is found by multiplying the type by 4 e.g. for type 11, interrupt address is 11 x 4 = 44_{10}= 0002CH

Interrupts

Features of 8259 Programmable Interrupt Controller

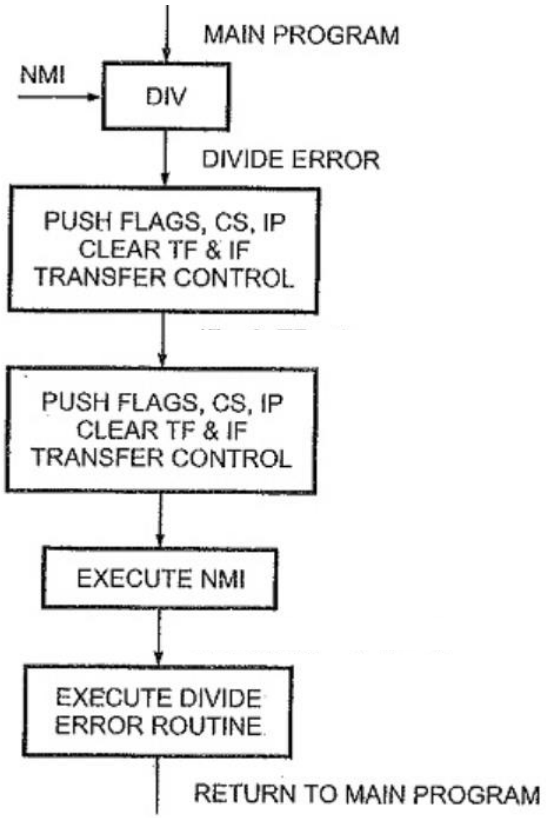




- Interrupts to 8259
- 8259 gives INTR to 8086
- 8086 floats AD0-AD15 and sends first $\overline{INTA}$
- (it is to tell 8259 to get ready with next input)
- 8086 sends 2nd $\overline{INTA}$ to 8259
- In response to this 8259 sends INT type number(N) on AD0-AD7
- 8086 will execute INT N instruction

Interrupt priority

- Software interrupts (All interrupts except single step, NMI and INTR interrupts) have the highest priority, followed by NMI followed by INTR. Single step has the least priority.
- The interrupt flag is automatically cleared as part of the response of an 8086 to an interrupt.
- This prevents a signal on the INTR input from interrupting a higher priority interrupt service routine.
- The 8086 allows NMI input to interrupt higher priority interrupt, for example suppose that a rising edge signal arrives at the NMI input while the 8086 is executing a DIV instruction, and that the division operation produces a divide error.
- Since the 8086 checks for internal interrupts before it checks for an NMI interrupt, the 8086 will push the flags on the stack, clear TF and IF, push the return address on the stack, and go to the start of the divide error service routine.
- The 8086 will then do an NMI interrupt response and execute non-maskable interrupt service routine. After completion of NMI service routine an 8086 will return to the divide error routine. It will execute divide error routine and then it will return to the main program



Interrupt priority

- If any interrupt comes, 8086 disables IF and TF.
- Single step operation in ISR is disabled by default, so it has lowest priority
- If single stepping is needed in ISR, TF need to be set in initial portions of ISR.

Interrupt	Priority
Divide Error, Int n, INTO	HIGHEST
NMI	↓
INTR	↓
SINGLE - STEP	LOWEST

References

- Doughlas V. Hall, “Microprocessors and Interfacing, Programming and Hardware”, Second Edition, TMH.

Thank you