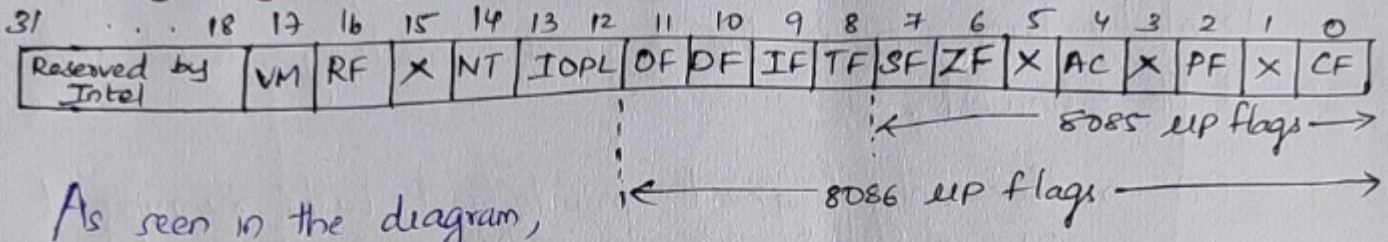
Academic Year: 2022-23 **Semester: IV**
Class/Branch: SE **Subject: MP**

Flag Register of 80386 (EFLAGS - Extended Flag)



As seen in the diagram, the lower 12 bits (11...0) of EFLAGS are same as those in 8086. These are the only flags available when µP is in Real Mode.

The additional 5 flags are only available once µP enters Protected mode by making PE bit = 1, in CR0 register.

IOPL : I/O Priviledge level.

Priviledge levels are assigned to entities (either data or program) stored in memory.

In mly, everything is stored in segments. Segments are like files in mly and priviledge levels are assigned to segments. Each segment has a descriptor and the priviledge level of the segment is stored in the descriptor.

Priviledge levels are also assigned to I/O devices. IOPL bits defines the numerically maximum Priviledge level (logically lowest) at which a task must be running to access I/O devices.

80386 has 4 priviledge levels used for protection mechanism. Priviledge level = 0 is the highest Priviledge level and 3 is the lowest.

00 → then only highest priviledged tasks running at PL = 0 can perform I/O instructions.

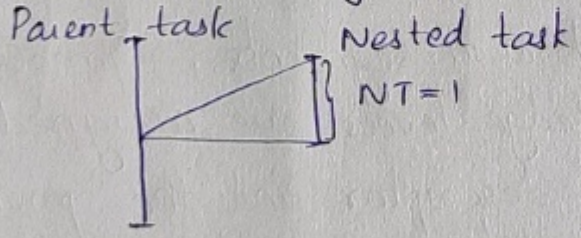
01 → PL = 1

10 → PL 2

11 → PL3 - All tasks at any priviledge level can perform I/O instructions.

NT: Nested Task

NT flag is used to indicate that the current task is nested. ie, it is invoked by another task.



Each task has its own TSS (Task state Segment). The TSS has a "back link" pointing to the parent task. In the TSS of the parent task also there will be a back link but it will be a NULL pointer.

If NT=1 then the current task is nested and has a valid back link in its TSS to the TSS of the previous task

Academic Year: 2022-23 **Semester: IV**
Class/Branch: SE **Subject: MP**

RF: Resume Flag

On 80386 up, some fault handless (ISRs) return back to same instruction that caused the fault instead of returning back to the next instruction. By keeping RF = 1, we ensure that the program resumes after such a fault instead of repeatedly generating breakpoint faults on the same instruction.

If RF = 1, then any debug fault in the next instruction will be ignored and RF is automatically reset after the next instruction.

Amount
Amount
Amount
Amount

Suppose you are writing a financial application and calculating total amount at the end. You are getting a wrong answer (logical error). On 8086 we can do single stepping and insert breakpoints.

On 80386, we can put breakpoints on amount variable. So wherever in the program amount is affected it will stop there and show the result.

Suppose we have found that 2nd amount is wrong. We will fix the logic and set RF = 1. So next time it will skip the previous amount breakpoint we have already checked it previously and fixed it correct. So after skipping this breakpoint, RF automatically becomes 0.

VM: Virtual 8086 Mode.

If VM = 1, enter virtual 8086 mode. V86 mode is basically used to run 8086 programs in a faster environment of 80386 using multi tasking and protection.
If VM = 0, come back to protected mode.