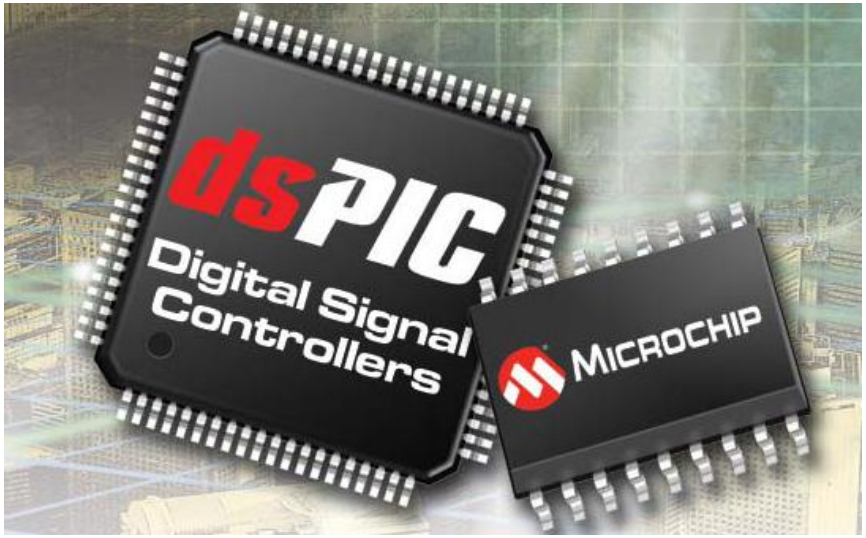
EE427 Advanced Microcontrollers



Interrupts (Contd)

Disable Interrupt Instruction

- **`DISI`** (disable interrupts) instruction has the ability to disable interrupts for up to 16384 instruction cycles.
- This instruction is useful when time critical code segments must be executed.
- The **`DISI`** instruction only disables interrupts with priority levels 1-6.
- Priority level 7 interrupts and all trap events still have the ability to interrupt the CPU when the DISI instruction is active.
- The DISI instruction works in conjunction with the DISICNT register.
- When the DISICNT register is non-zero, priority level 1-6 interrupts are disabled.

Disable Interrupt Instruction (contd)

- The DISICNT register is decremented on each subsequent instruction cycle.
- When the DISICNT register counts down to ‘0’, priority level 1-6 interrupts will be re-enabled.
- The DISICNT register is readable and writable.
- The user can terminate the effect of a previous **`DISI`** instruction early by clearing the DISICNT register.
- The amount of time that interrupts are disabled can also be increased by writing to or adding to DISICNT.

Return from interrupt and Nesting

- **`RETFIE`** (Return from Interrupt) instruction will unstack the PC return address, IPL3 status bit, and SRL (low byte of the Processor Status register) register to return the processor to the state and priority level prior to the interrupt sequence.

- Interrupts, by default, are nestable.
- Any ISR that is in progress may be interrupted by another source of interrupt with a higher user assigned priority level

Wake-up mode

- Any source of interrupt that is individually enabled, can wake-up the processor from Sleep or Idle mode.
- When the interrupt status flag for a source is set and the interrupt source is enabled via the corresponding bit in the IEC Control registers, a wake-up signal is sent to the dsPIC30F CPU.
- When the device wakes from Sleep or Idle mode, one of two actions may occur:
  - If the interrupt priority level for source is greater than the current CPU priority level, then the processor will process the interrupt and branch to the ISR for the interrupt source.
  - If the user assigned interrupt priority level for the source is less than or equal the current CPU priority level, then the processor will simply continue execution, starting with the instruction immediately following the PWRSAV instruction that previously put the CPU in Sleep or Idle mode.

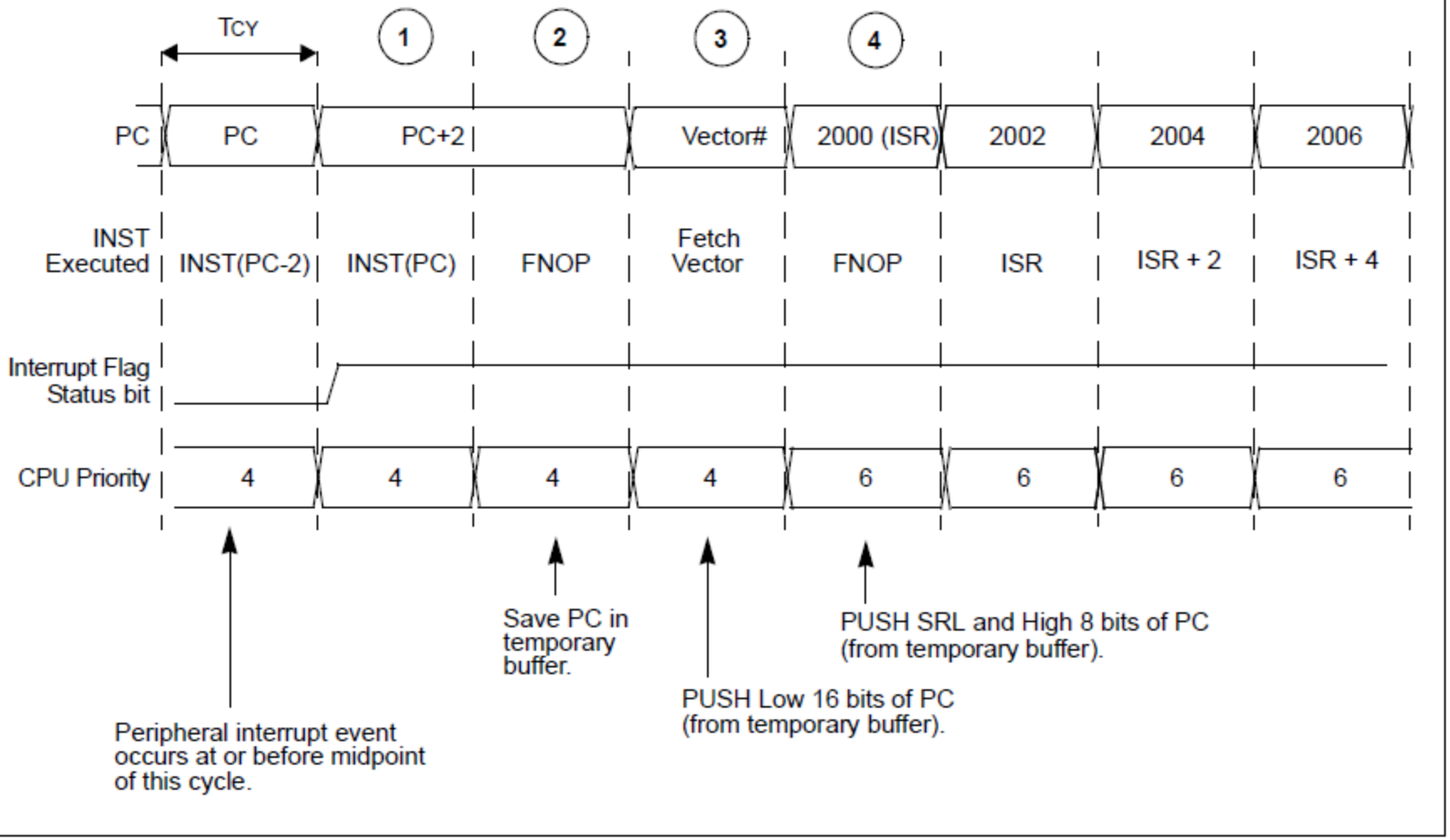
Interrupt Processing Timing

- **Interrupt Latency for One-Cycle Instructions**
  - The interrupt process takes four instruction cycles.
  - The interrupt flag status bit is set during the instruction cycle after the peripheral interrupt occurs.
  - The current instruction completes during this instruction cycle.
  - In the second instruction cycle after the interrupt event, the contents of the PC and SRL registers are saved into a temporary buffer register.
  - The second cycle is executed as a NOP to maintain consistency with the sequence taken during a two-cycle instruction**.** fetched.

Interrupt Processing Timing

- In the third cycle, the PC is loaded with the vector table address for the interrupt source and the starting address of the ISR is fetched.
- In the fourth cycle, the PC is loaded with the ISR address.
- The fourth cycle is executed as a NOP while the first instruction in the ISR is fetched

Interrupt Timing During a One-Cycle Instruction

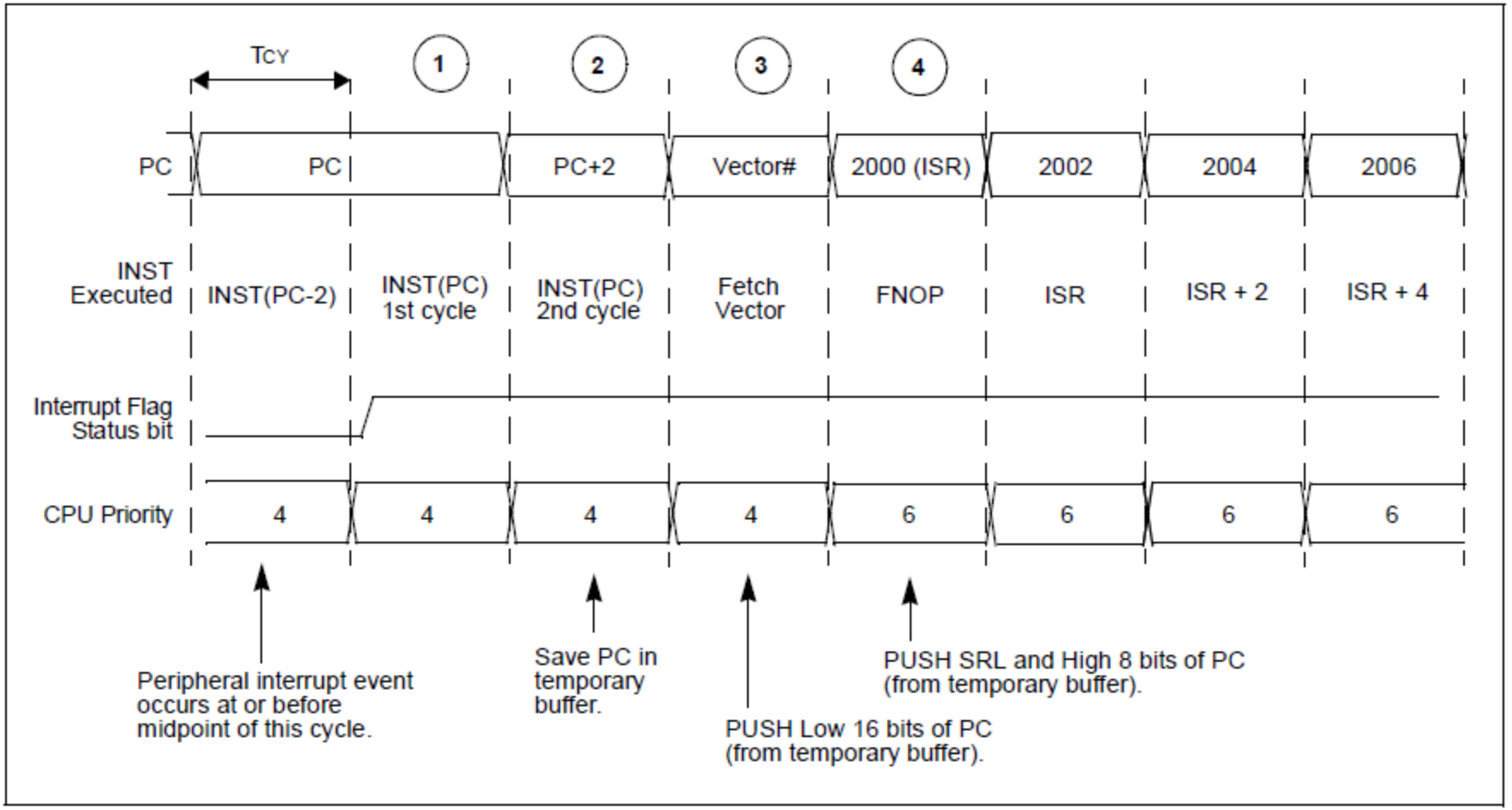


Interrupt Latency for Two-Cycle Instructions

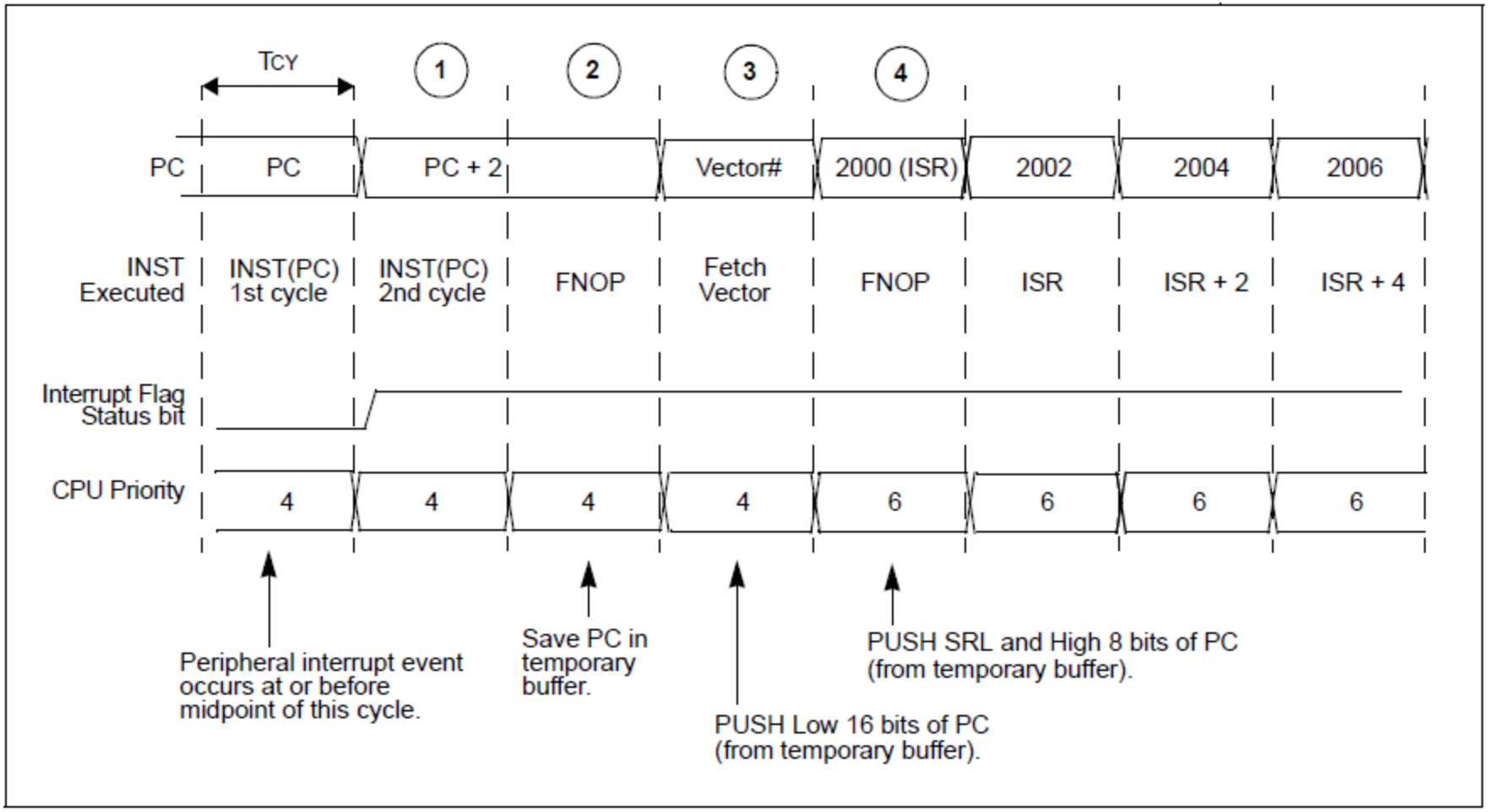
- The interrupt latency during a two-cycle instruction is the same as during a one-cycle instruction.

- The first and second cycle of the interrupt process allow the two-cycle instruction to complete execution.

Interrupt Timing During a Two-Cycle Instruction



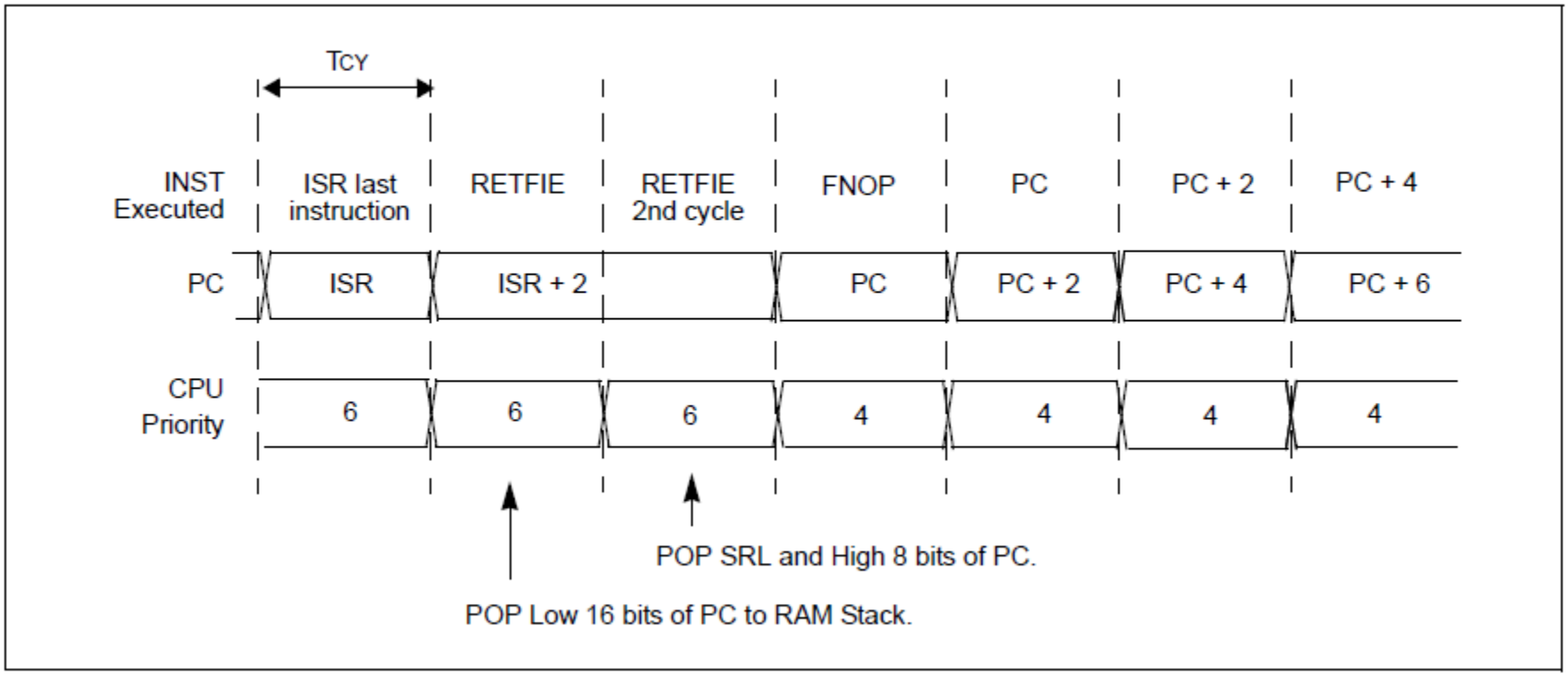
Interrupt Timing, Interrupt Occurs During 1st Cycle of a 2-Cycle Instruction



Returning from Interrupt

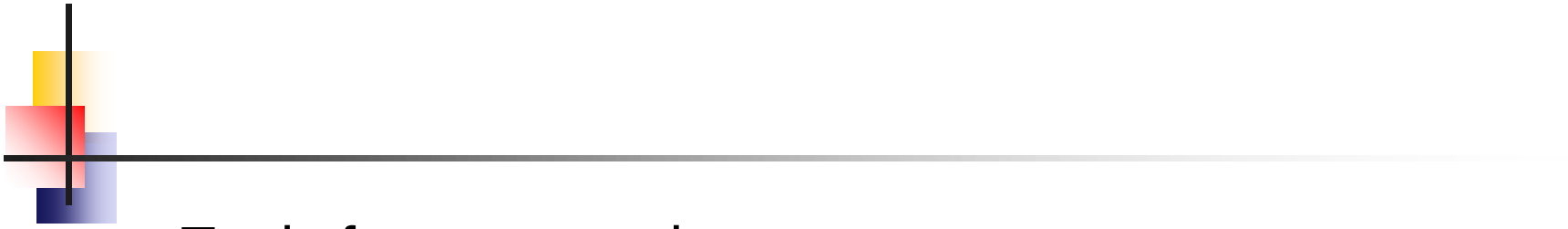
- "Return from Interrupt" instruction, **`RETFIE`**, exits an interrupt or trap routine.
- During the first cycle of a `RETFIE` instruction, the upper bits of the PC and the SRL register are popped from the stack.
- The lower 16 bits of the stacked PC value are popped from the stack during the second cycle.
- The third instruction cycle is used to fetch the instruction addressed by the updated program counter.

Return from Interrupt Timing



Interrupt Control and Status Registers

- INTCON1, INTCON2 Registers
- IFSx: Interrupt Flag Status Registers
- IECx: Interrupt Enable Control Registers
- IPCx: Interrupt Priority Control Registers
- SR: CPU Status Register
- CORCON: Core Control Register



- Topic for next session
  - Oscillators
  - Watch Dog Timer (WDT)