DEVICE DRIVERS AND INTERRUPTS SERVICE MECHANISM

Lesson-11: Interrupt latency and Service deadline

1. Interrupt Latency

Interrupt Latency

- A period between occurrence of an interrupt and start of execution of the ISR
- The time taken in context switching is also included in a period, called interrupt latency period, T_{lat}.

Minimum Interrupt-latency period

- T_{lat}, is the sum of the periods as follows.
- Time to be taken is for the response and initiating the ISR instructions.
- This includes the time to save or switch the context (including the program counter and registers) plus the time to restore its context.

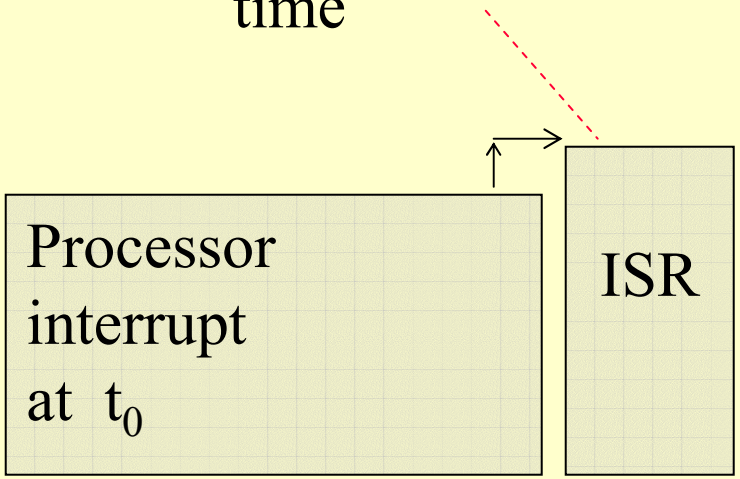
For example, in the ARM7 processor, this period equals two clock cycles plus zero to twenty clock cycles for finishing an ongoing instruction plus zero to three cycles for aborting the data.

Minimum Latency = context switching period

- When the interrupt service starts immediately on context switching the interrupt latency = T_{switch} = context switching period. When instructions in a processor takes variable clock cycles, maximum clock cycles for an instructions are taken into account for calculating latency

Latency in case of interrupt service starting immediately

Starts after time t_0+ t' only where t' is context switch time

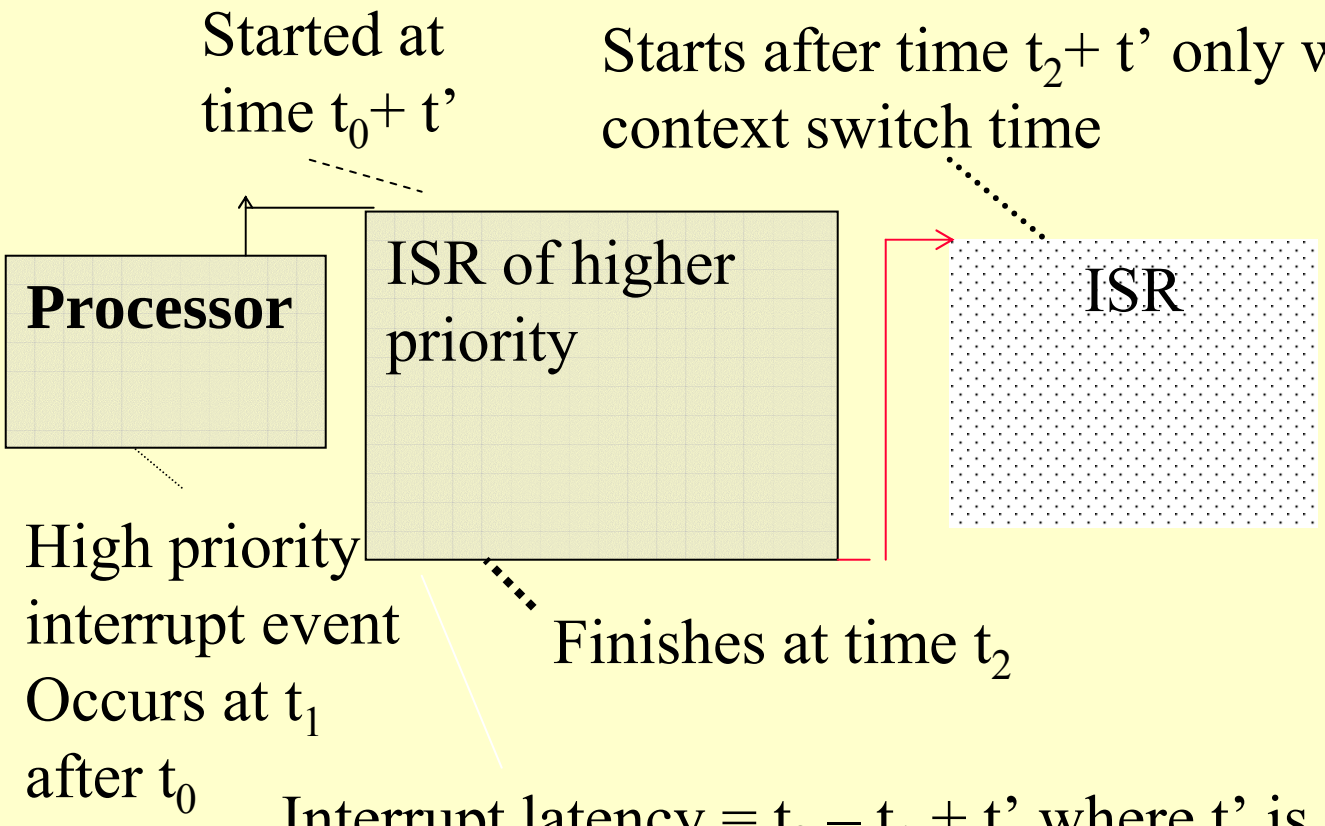


Interrupt latency = t' only where t' is context switch time for saving the running program context and loading the new routine context

Latency on context switch to an higher priority interrupt

- When the interrupt service does not starts immediately by context switching but context switching starts after all the ISRs corresponding to the higher priority interrupts complete the execution. If sum of the time intervals for completing the higher priority ISRs = ΣT_{exec} , then interrupt latency = $T_{switch} + \Sigma T_{exec}$.

Latency in case of interrupt service starts after ISRs of higher priority than the present interrupt finish the execution

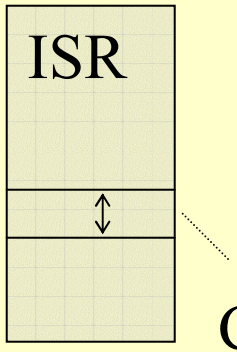


Interrupt latency = $t_2 - t_1$ + t' where t' is context switch time for saving the running program context and loading the new routine context

Latency due to execution of Disable Interrupt Instruction in a critical section

- $T_{disable}$ is the period for which a routine is disabled in its critical section. The interrupt service latency from the routine with interrupt disabling instruction (due to presence of the routine with critical section) for an interrupt source will be $T_{switch} + \Sigma T_{exec} + T_{disable}$

Interrupt latency as sum of the periods for T_{switch}, ΣT_{exec} and $T_{disable}$ when presently running routine to be interrupted is executing critical section codes



Critical section codes (Time spent = $T_{disable}$)

Interrupt latency = ΣT_{exec} + $T_{disable}$ + T_{switch} where T_{switch} is context switch time for saving the running program context and loading the new routine context , $T_{disable}$ is time for which interrupts remained disabled in critical period and ΣT_{exec} is time for which other high priority routines executed

Worst case latency

- Sum of the periods Tswitch , ΣTexec and Tdisable where sum is for the interrupts of higher priorities only.

Minimum latency

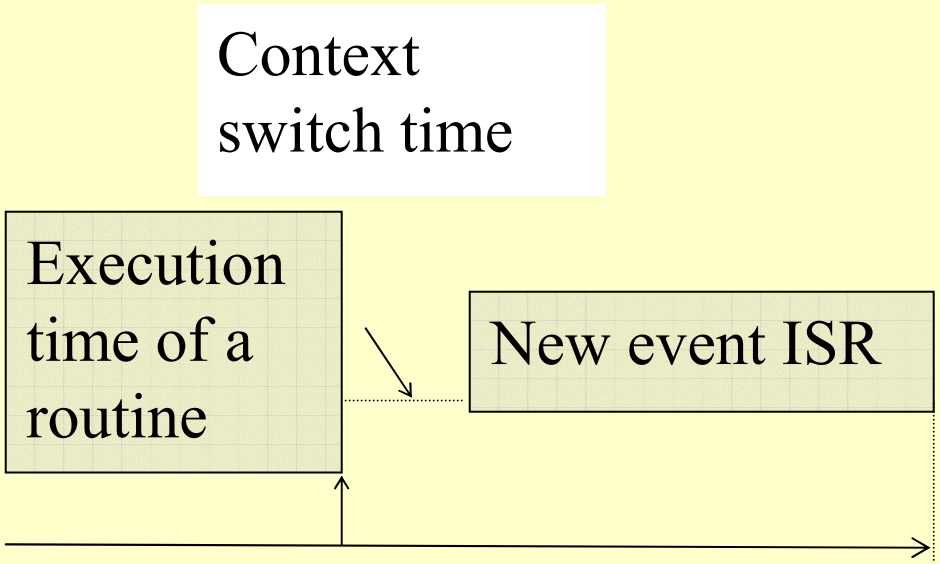
- Sum of the periods T_{switch} and $T_{disable}$ when the interrupt is of highest priority.

2. ISR or Task Deadline

ISR or Task Deadline

- For every source there may be a maximum period only up to which the service of its all ISR instructions can be kept pending. This period defines the Deadline period, T_d during which the execution must be completed.

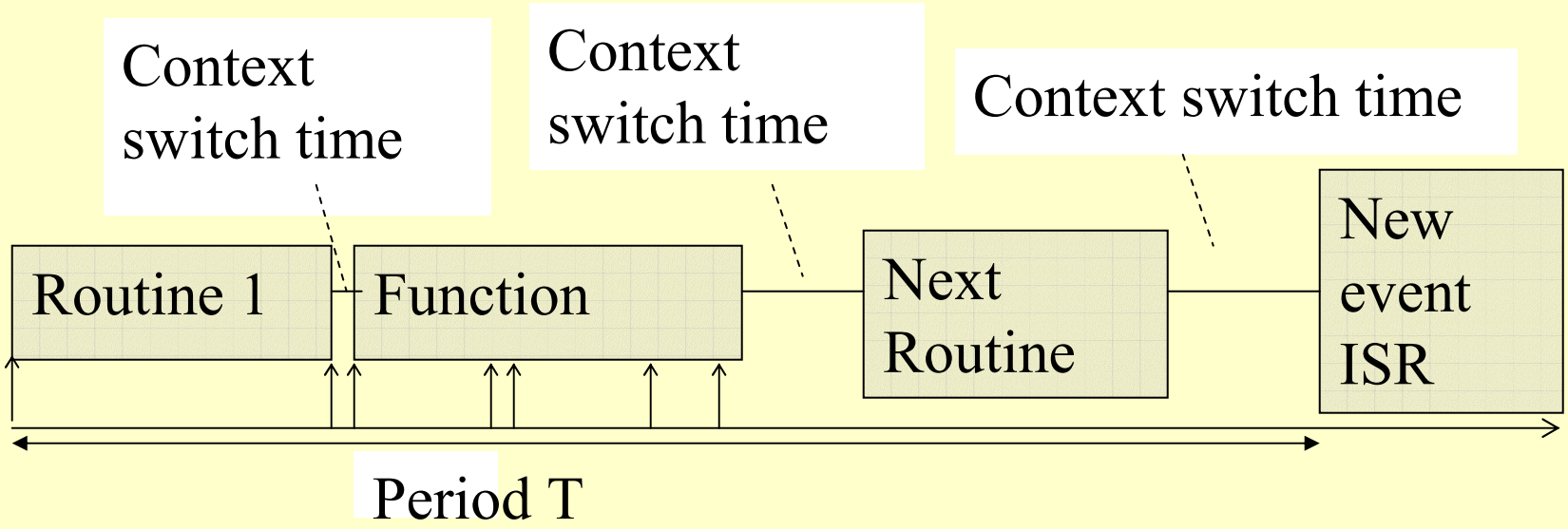
Interrupt latency period and deadline for an interrupt



Interrupt latency should be such that a deadline of a new event ISR is not missed

Must finish at time t_d

Short ISR and functions, which run at later instances so that the other ISR deadlines are not missed



Interrupt latency = T should be such that a deadline of a new event ISR is not missed

Example

- Video frames in video conferencing reach after every 1 ÷15 s. The device on getting the frame interrupts the system and the interrupt service deadline is 1 ÷15 s, else the next frame will be missed

Example

- A 16-bit timer device on overflow raises TF interrupt on transition of counts from 0xFFFF to 0x0000.
- To be responded by executing an ISR for TF before the next overflow of the timer occurs, else the counting period between 0x0000 after overflow and 0x000 after next to next overflow will not be accounted.
- If timer counts increment every 1μs, the interrupt service deadline is 2^{16} μs = 65536 μs.

To keep the ISR as short as possible

- In case of multiple interrupt sources
- To service the in-between pending interrupts and leave the functions that can be executed afterwards later for a later time.
- Use of interrupt service threads, which are the second level interrupt handlers.
- When this principle is not adhered to, a specific interrupting source may not be serviced within in the deadline (maximum permissible pending time) for that

3. Assignment of priorities to Meet Service Deadlines

Assignment of priorities to Meet Service Deadlines

- By following an EDF (Earlier Deadline First) strategy for assigning the priorities to the ISRs and Tasks, the service deadlines are met.

Software overriding of Hardware Priorities to Meet Service Deadlines

- It is first decided among the ISRs that have been assigned higher priority by in the user software. If user assigned priorities are , and then if these equal then among the that highest priority, which is pre-assigned by at the processor internal- hardware.

Summary

We learnt :

- Interrupt Latency— Each running-program when interrupts, the interrupting source service routine takes some time before starting the servicing codes.
- It is sum of execution time of higher priority interrupts and context switching period.

We learnt

- If the interrupted routine is in critical section (interrupts disabled), the interrupt latency increases by the period equal to interrupts disabled period
- Use of processor, which saves the context fast, use of priority assignments, enabling and disabling of interrupts are resorted to meet the deadlines to finish all the ISRs and tasks within the deadlines.

End of Lesson 11 of Chapter 4