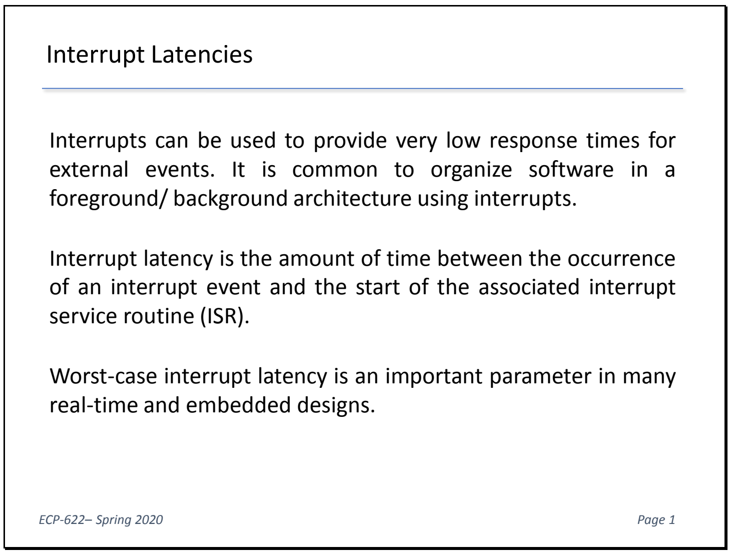


Interrupt Latencies

Interrupts can be used to provide very low response times for external events. It is common to organize software in a foreground/ background architecture using interrupts.

Interrupt latency is the amount of time between the occurrence of an interrupt event and the start of the associated interrupt service routine (ISR).

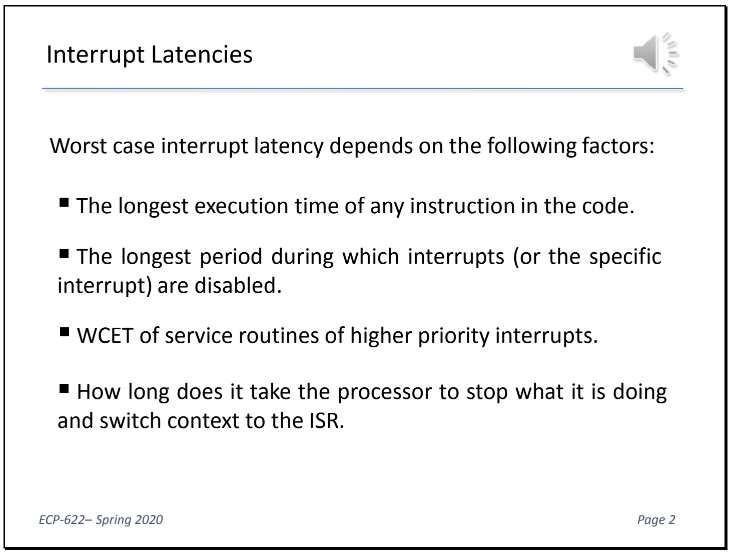
Worst-case interrupt latency is an important parameter in many real-time and embedded designs.

Slide 1

As we mentioned before, aperiodic events can be treated either by periodic polling or using interrupts. Associating an event with a high priority interrupt allows fastest response to this event. Interrupt handlers are sometimes called foreground tasks that operate with high priority, while less important tasks run in the background.

Slide 2

Note in this definition that latency is measured as the time passed until the ISR starts, and not ends. Hence, the execution time of the ISR itself is not included in the interrupt latency.



Interrupt Latencies

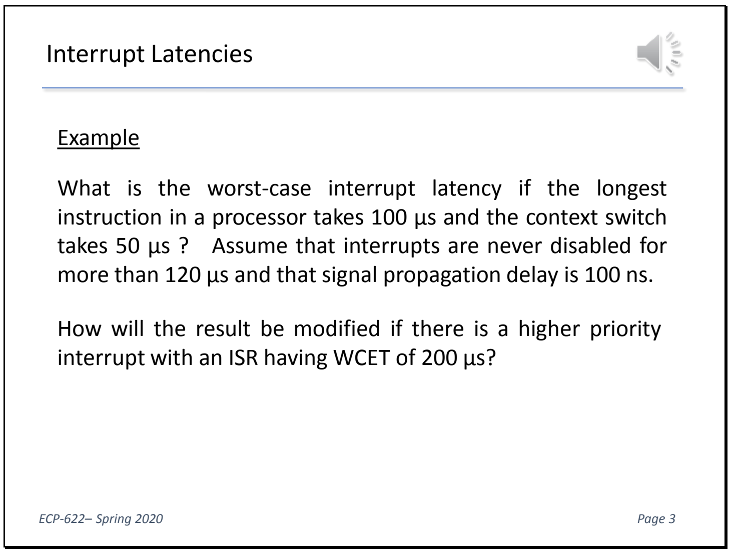
Worst case interrupt latency depends on the following factors:

- The longest execution time of any instruction in the code.
- The longest period during which interrupts (or the specific interrupt) are disabled.
- WCET of service routines of higher priority interrupts.
- How long does it take the processor to stop what it is doing and switch context to the ISR.

ECP-622– Spring 2020 Page 2

Slide 4

Since interrupt can only be checked at the end of an instruction, response may be delayed if interrupt request arrives during the execution of a long instruction. Delay may also occur since interrupts are disabled within some tasks. Another potential delay source occurs when request is accompanied by requests of higher priority interrupts, thus their ISRs run first. Before start of ISR, necessary context switching will cause further delay.



Interrupt Latencies

Example

What is the worst-case interrupt latency if the longest instruction in a processor takes 100 µs and the context switch takes 50 µs ? Assume that interrupts are never disabled for more than 120 µs and that signal propagation delay is 100 ns.

How will the result be modified if there is a higher priority interrupt with an ISR having WCET of 200 µs?

ECP-622– Spring 2020 Page 3

Slide 5

Note that delay may be caused either by waiting for long instruction OR waiting for interrupt re-enabling, but not both. Signal propagation delay is the time that may be needed for the processor to detect the interrupt request signal after the event.

Interrupt Latencies

In the first case:

worst-case interrupt latency= Max (100 µs, 120 µs)+ 50 µs+ 100 ns
= 170.1 µs

In the second case, higher priority interrupt runs first, and context switching occurs two times:

worst-case interrupt latency=
Max (100 µs, 120 µs)+ 200 µs + 2x50 µs+ 100 ns = 420.1 µs

Interrupt Latencies

Some of the previous factors depend on processor design and are not controlled by software. On the other hand, good programming practices can minimize the effect of other factors.

- ISRs should be as short as possible: long routine will affect the response time for every other interrupt at lower priority. Any nonessential code should be removed from the ISR, moving as much processing as possible into background tasks.
- Time intervals of interrupt disabling must be minimized.

Interrupt latencies in RTOS environments

RTOS must have guaranteed worst-case interrupt latency and context switch times. These are usually considered among the selection criteria for an RTOS.

RTOS will typically reduce the times of interrupt disabling in its functions.

An interrupt routine must not call any RTOS function that might block the caller.