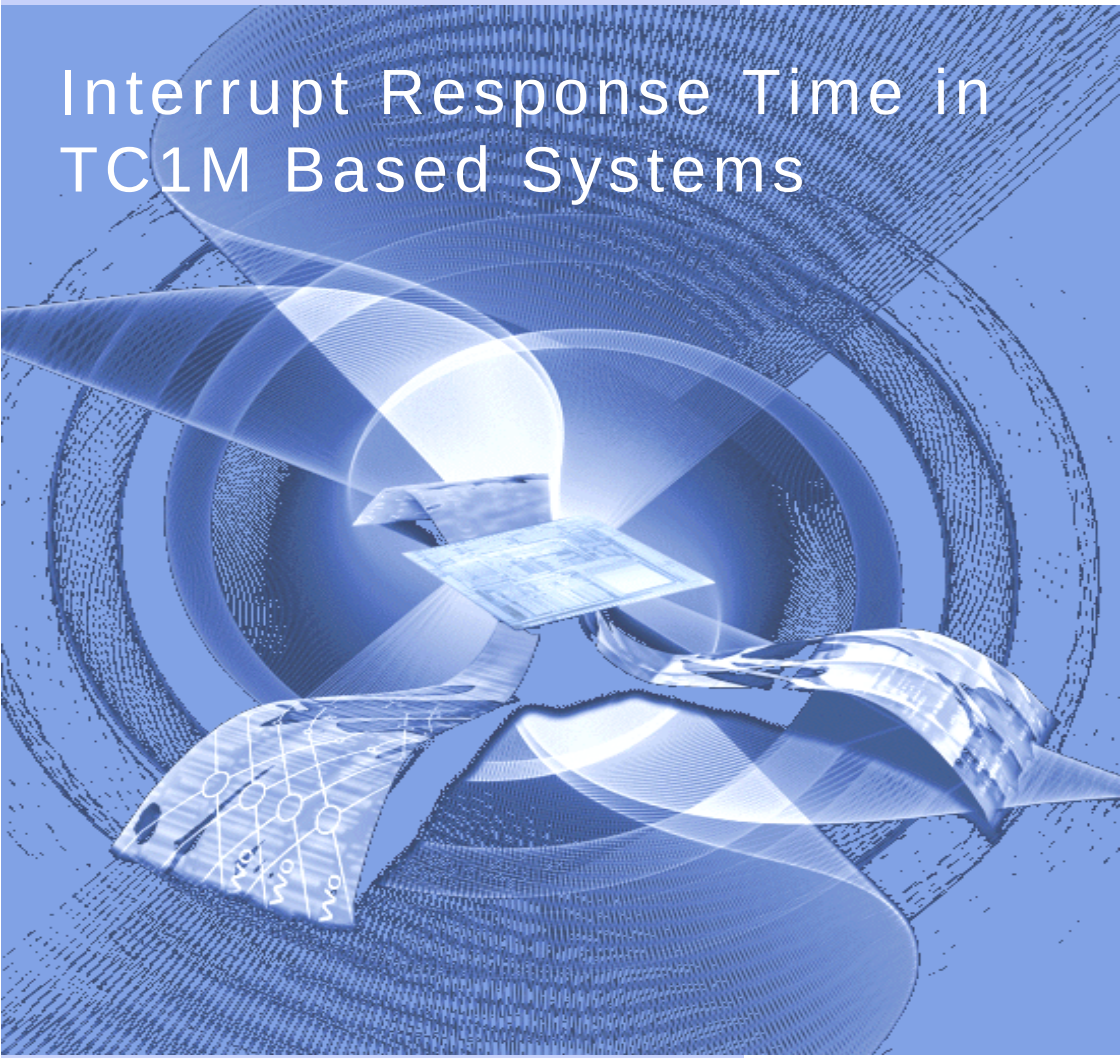
TriCore™
32-bit Unified Processor

Interrupt Response Time in TC1M Based Systems



IP Cores

Author: Sorin Zarnescu

Application Note: AP32076

Never stop thinking.

Edition 3rd July 2002

Published by Infineon Technologies AG,
St.-Martin-Strasse 53,
D-81541 München, Germany

Attention please!

The information herein is given to describe certain components and shall not be considered as warranted characteristics.
Terms of delivery and rights to technical change reserved.
We hereby disclaim any and all warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.
Infineon Technologies is an approved CECC manufacturer.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office in Germany or our Infineon Technologies Representatives worldwide (see www.infineon.com for contact details).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.
Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Table of Contents **Page**

1 Scope

TC1M is a licensable core implementing version 1.3 of the TriCore architecture. This application note gives a brief discussion of the interrupt response time in a TC1M-based system. Because of the many variables that may influence access time in a real system, some assumptions have been necessary for simplification. Where assumptions have been made, they are clearly identified in the text.

2 Interrupt Response Time (IRT)

The Interrupt Response Time (IRT) is defined as the time it takes from the assertion of the interrupt request signal by the interrupting device, until the start of the execution of the first useful instruction in the Interrupt Service Routine (ISR). The first useful instruction is the first instruction that is directly related to the actions requested by the interrupting device and ***not*** related to interrupt processing.

Generally speaking, when an interrupt request is asserted in any processor system, several actions need to take place (the order may vary depending on the processor):

2.1 Arbitration (ARB) Phase

Since there are many potential sources that can assert the Interrupt Request Signal, the arbitration phase is required in order to determine the highest priority interrupt when multiple interrupts are active at the same time.

2.2 Priority Check (PRI) Phase

Once the highest priority interrupt has been selected, its priority level is compared with the priority level of the task currently being performed by the CPU. This task could be the ISR of another interrupt, or a regular CPU background task. If the requesting interrupt has a priority level higher than that of the current CPU task, it will be serviced; i.e. the present context is saved and the first ISR instruction is fetched.

2.3 Context Save (CON) Phase

This phase ensures that the context of the task currently running in the CPU is saved, so that at the end of the ISR the CPU can resume processing from the point of interruption.

2.4 Vector to ISR (VEC) Phase

In this phase the CPU is usually vectored to an interrupt table. From that table it will fetch at least the first instruction(s) of the ISR.

3 IRT Calculation

The order in which the phases described in **Section 2** are performed, may differ from processor to processor. However, all of these tasks need to be carried out to ensure the correct functioning of the processor system.

After completing all of the phases described, the CPU starts executing the first IRT instruction.

Interrupt systems are implemented to ensure the fastest means of the CPU executing instructions required by the interrupting device. Any instruction that is not directly related to servicing the interrupt is in effect an overhead and delays the servicing of the interrupt.

The Interrupt Response Time is therefore defined as:

IRT = ARB + PRI + CON + VEC

4 TC1M Based IRT

This section details the time taken by each of the interrupt phases (see **Section 2**), in a TriCore TC1M based system. The time is indicated by the number of CPU cycles.

Note: *An important assumption is that the CPU is not stalled by the execution of a complex instruction that requires a long time to complete (such as a load/store to slow memories, for example). If this is the case, the time to perform the complex instruction should be added to the interrupt response time.*

4.1 ARB Phase

The time taken by the arbitration (ARB) phase depends on the total number of interrupts implemented in the system:

Total Number of Interrupts	Number of Cycles
0 - 3	1
4 - 15	2
15 - 63	3
64 - 255	4

It is important to note that in all TriCore systems the arbitration is carried out by specialized hardware. This hardware runs in parallel with the CPU and does not 'disturb' the CPU; i.e. The CPU continues with its normal operation while arbitration takes place. This specialized arbitration hardware is located in the Interrupt Control Unit (ICU) that is itself part of the CPU Slave (CPS).

At the end of arbitration the CPS sends an interrupt request signal to the CPU. This adds an extra cycle to each of the **Number of Cycles** figures shown in the table above. In a system where the total number of interrupts is 15 for example, the ARB phase takes 3 cycles (Number of Cycles is 2, plus 1 for the CPS interrupt request signal = 3).

Note: *Although this number must be included in the overall interrupt response time, it does not affect the system throughput.*

4.2 PRI Phase

The Priority Check (PRI) phase takes 1 cycle.

As with the ARB phase, PRI is performed in parallel with the CPU's normal operation.

4.3 CON Phase

The Context Save (CON) phase takes advantage of the wide 128-bit bus to local data memory, that allows 4 registers to be saved in one clock cycle. TriCore automatically saves 16 registers on entry and restores them on return.

TriCore's use of shadow registers means that the number of cycles varies depending on how many interrupts are nested. For the first two occurrences (i.e. the very first two interrupts being processed - a nesting level of two), it will take 3 cycles. This include 2 cycles of context switching, plus 1 cycle to send an acknowledge signal back to the Interrupt Control Unit (ICU).

If the nesting level goes beyond 2, the CON phase will take 5 cycles for interrupt 3, 4 and so on.

4.4 VEC Phase

Once the interrupt has been accepted (i.e. its priority number is higher than the priority number of the task currently being executed), the CPU fetches the first instruction of the Interrupt Service Routine (ISR). This instruction is located in the Interrupt Vector Table (IVT). In order to get the fastest interrupt response, it is recommended to locate the IVT in the fast local program memory.

The Vector to ISR (VEC) phase takes 3 clock cycles.

5 IRT Examples

Below are two examples of calculating the Interrupt Response Time (IRT):

5.1 Example 1

Assume a system with 15 interrupts and a CPU clock of 200 MHz.

For the first occurring interrupt:

IRT = 3 + 1 + 3 + 2 = 9 clock cycle

IRT = 45 ns

The sequence of operations is as follows:

I1 (interrupting device sends request to ICU for arbitration)

Note: *The following assumptions are made:*
I1 has a higher priority than the task currently executed by the CPU, the interrupts are enabled and the CPU executes non-complex instructions.

ARB_I1	**ARB phase**
ARB_I1	
ARB_I1 (ICU sends request to CPU)	
PRI_I1 (CPU check priority of CPU vs. priority of pending interrupt)	**PRI phase**
CON_I1	
CON_I1	**CON phase**
ACK (CPU sends acknowledge to ICU)	
VEC_I1 (fetch request to memory)	
VEC_I1 (fetch data from memory)	**VEC phase**
(fetch stage of the 1st instruction in ISR_I1)	**ISR execution**
(decode stage of the 1st instruction in ISR_I1)	
(execute stage of the 1st instruction in ISR_I1)	
(write back stage of the 1st instruction in ISR_I1)	
‘	
‘	
‘ body of the ISR)	
‘	
‘	

5.2 Example 2

For the same example as above (**Example 1**), if the interrupt is not the first or second in multiple interrupt occurrences, the CON phase is 5 cycles long (4 CON cycles are required to save 16 registers).

Note: *The examples given consider the case in which only 16 registers need to be saved. If all 32-bit registers need to be saved, than the first instruction of the ISR will be one that saves the remaining 16 registers.*

Infineon goes for Business Excellence

“Business excellence means intelligent approaches and clearly defined processes, which are both constantly under review and ultimately lead to good operating results.

Better operating results and business excellence mean less idleness and wastefulness for all of us, more professional success, more accurate information, a better overview and, thereby, less frustration and more satisfaction.”

Dr. Ulrich Schumacher

http://www.infineon.com

Published by Infineon Technologies AG