Avoiding Phantom Interrupts on the TMS470R1X Central Interrupt Module

Frank Noha *Advanced Embedded Control*

ABSTRACT

This application report describes the ways that phantom interrupts can occur on a TMS470R1x device and how to avoid them.

Contents

List of Figures

List of Tables

1 Description of Central Interrupt Module

All the devices of the TMS470R1x family include the central interrupt module (CIM), which controls the generation of interrupts to the CPU if an interrupt is requested by one of the peripheral modules. See Figure 1.

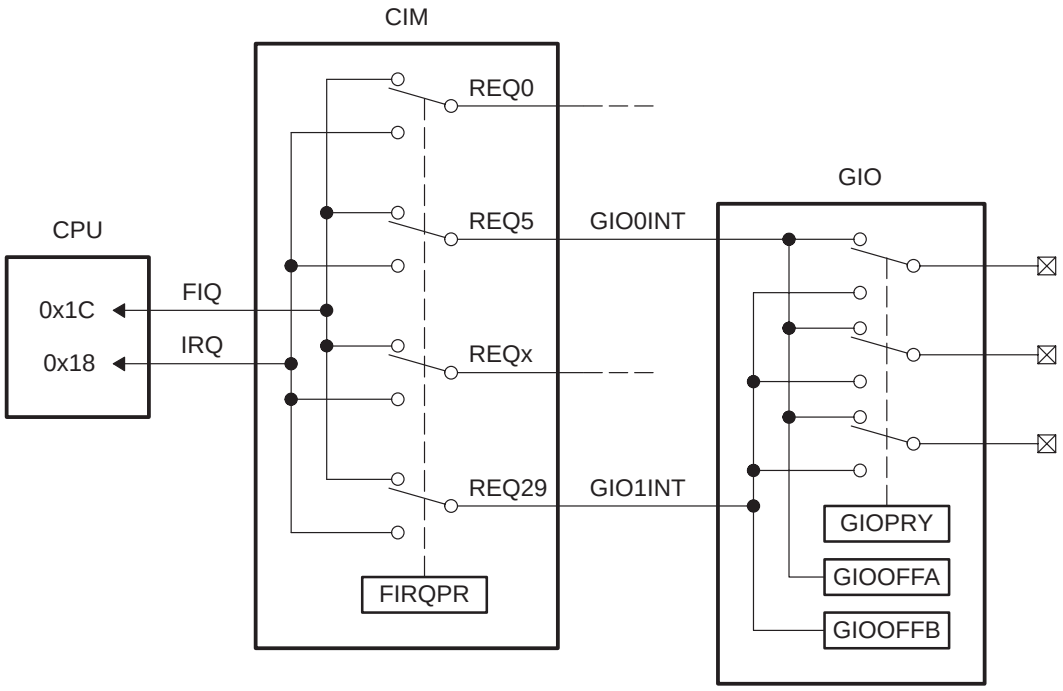

Figure 1. CIM's Role in Generating Interrupts

All interrupts have a corresponding interrupt flag in the INTREQ register. A flag is set if an interrupt was requested by the peripheral module. To allow a fast determination of which module raised an interrupt request, three interrupt offset registers are implemented (IRQIVEC, FIQIVEC, CIMIVEC). IRQIVEC holds the offset of the highest priority IRQ, FIQIVEC holds the offset of the highest priority FIQ, and CIMIVEC holds the highest priority interrupt whether it is an FIQ or an IRQ.

Table 1. Encoding of Interrupt Offsets

Encoding	Description
0x0000	Phantom Interrupt
0x0001	Channel0
0x0020	Channel31

The software can read the offset and determine which peripheral module caused the interrupt without having to poll all the interrupt flags. The interrupt request from the peripheral module goes inactive if the interrupt flag in the peripheral module is cleared.

2 Ways Phantom Interrupts Occur

An offset of 0x0000 means that the interrupt source cannot be determined. There are two possible ways such phantom interrupts can be generated:

Scenario 1:

1. An interrupt was generated by the peripheral module.
2. Before the offset is read out of one of the offset registers, the interrupt flag in the peripheral module is cleared.
3. Because of the cleared flag, the channel goes inactive and the corresponding interrupt flag is reset in the CIM module.
4. Clearing of the flag causes the CIM to put the offset of the next pending interrupt that is in the priority queue in the offset register, or if no other interrupt was requested, to put the phantom interrupt offset in the register. Example 1 shows a possible scenario for such behavior.

Example 1:

The GIO module is configured to generate interrupts with both interrupt request lines. In the CIM, one of the requests is configured as FIQ and one is configured as IRQ. An IRQ is requested by the GIO module. This causes the interrupt handler for the IRQs to execute. Before the handler can read the offset, an FIQ is raised by the GIO module, which causes the FIQ interrupt handler to execute (the IRQ handler will be interrupted). In the FIQ interrupt service routine (ISR), the GIOFLG register, which holds the interrupt flags of the GIO module is completely reset to zero, because of, for example, a software bug. This will also deactivate the request of the IRQ. When the FIQ ISR is finished, the IRQ handler will continue execution and read the offset of the IRQ source. However, the request of the GIO module is no longer active, which results in the IRQ handler reading an offset of zero if no other channel was active.

Scenario 2:

1. An interrupt was generated by the peripheral module.
2. Before the offset is read out of one of the offset registers, the corresponding channel is disabled in the REQMASK register. This causes the CIM not to generate the appropriate offset, since the channel is no longer valid.
3. If no other channel is active an offset of 0x0000 is read. However, the corresponding interrupt flag will not be reset because the request is still active from the peripheral module.
4. By enabling the channel again, the offset is written into the offset register.

This behavior most likely is caused by operating systems, which has to disable all interrupts to execute certain operating system tasks.

Example 2:

The application software disables all interrupts in the REQMASK register. In the same cycle, an interrupt is generated by one of the peripheral modules when the write takes place. This interrupt is signaled to the ARM7. After the write operation (the last instruction has to be finished before responding to an interrupt), the interrupt is serviced. The interrupt handler tries to read the offset to determine the module that generated the interrupt, but it reads the phantom interrupt offset, since all interrupts are disabled. After enabling the interrupts again, the offset is written correctly to the offset register, since the interrupt flag is still set in the INTREQ register.

3 Solutions

Care has to be taken not to generate phantom interrupts. Some suggestions corresponding to the scenarios described in the previous section follow.

Scenario 1:

There is no general recommendation how to respond to interrupts. However, if possible it is best to always read any offsets present in the peripheral module rather than to clear the flag manually. Reading the offset clears the corresponding flag automatically. If there is no offset register present, then care must be taken to clear the flag of only the interrupt to which the application software responded to.

Scenario 2:

Table 1 shows the encoding of the offsets and the associated channels.

If the application software or the operating system has to disable all interrupts during program execution, it is better to disable the interrupts by setting the I and F flag in the CPSR register (ARM7) to one instead of disabling all interrupts in the REQMASK register of the CIM.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265