Serialized IRQ Support for PCI Systems

Compaq Computer Corporation
Cirrus Logic Incorporated
National Semiconductor Corporation
OPTi Incorporated
Standard Microsystems Corporation
Texas Instruments Incorporated
VLSI Technology Incorporated

Revision 6.0
September 1, 1995

Scope of this Revision

This is the first public release of the Serialized IRQ Support for PCI Systems specification.

Revision History

Revision	Date	Comments
6.0	9/1/95	First public release

Table of Contents

1.0 Scope

This document describes an approach for supporting standard ISA IRQs in a PCI-based system. This approach provides a mechanism for communicating IRQ status between ISA legacy components, PCI components, and PCI system controllers.

A serial interface is specified that provides a means for transferring IRQ and/or other information from one system component to a system host controller. This document is limited to the description of the serial protocol used to transfer the required information, leaving the detailed internal implementation and any specific interrupt mapping requirements to the designer.

This specification was developed through the collaboration of eight of the PC industry's leading suppliers of systems, system controllers, and peripheral components. Companies involved in the creation of this spec are:

Cirrus Logic/PicoPower	Standard Microsystems Corporation (SMC)
Compaq Computer Corporation	Texas Instruments
National Semiconductor	VLSI Technology
OPTi	

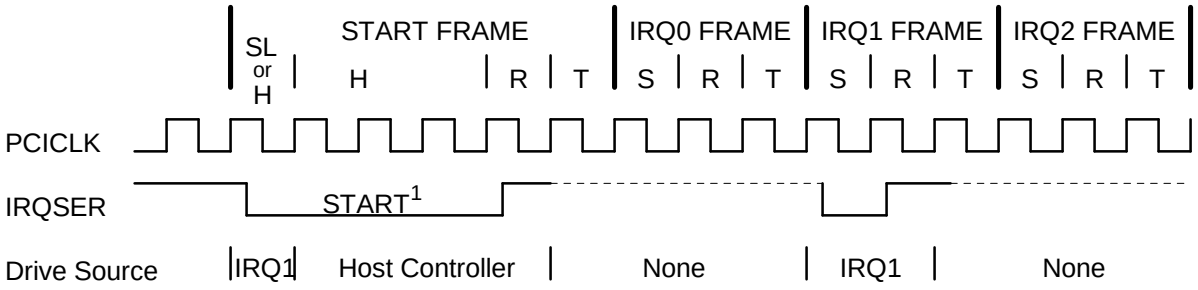
2.0 Architectural Overview

The IRQ/Data serializer is a Wired-OR structure that simply passes the state of one or more device's IRQ(s) and/or Data to the host controller. The transfer can be initiated by either a device or the host controller. Both high and low transitions are reported in this protocol. A transfer, called an IRQSER Cycle, consists of three frame types: one Start Frame, several IRQ/Data Frames, and one Stop Frame. This protocol uses the PCI Clock as its clock source and conforms to the PCI bus electrical specification.

3.0 Functional Specification

3.1 Timing Diagrams For IRQSER Cycle

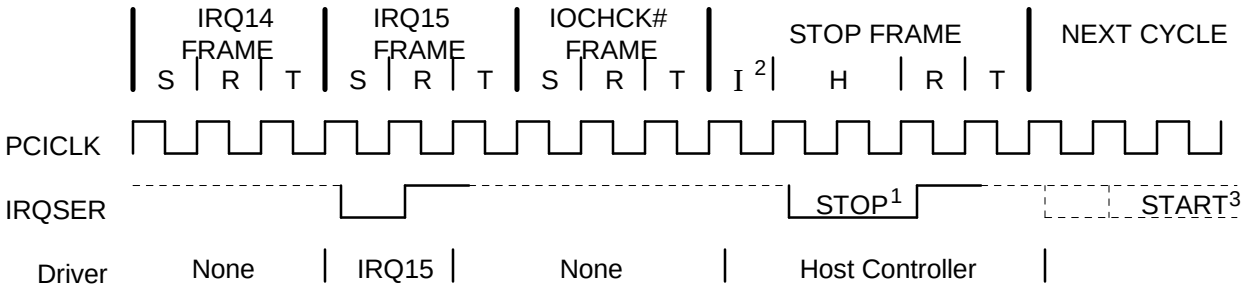
3.1.1 Start Frame timing with source sampled a low pulse on IRQ1



H=Host Control SL=Slave Control R=Recovery T=Turn-around S=Sample

1. Start Frame pulse can be 4-8 clocks wide. See Sec. 4.1 for description.

3.1.2 Stop Frame Timing with Host using 17 IRQSER sampling period



H=Host Control R=Recovery T=Turn-around S=Sample I= Idle.

1. Stop pulse is 2 clocks wide for Quiet mode, 3 clocks wide for Continuous mode. See Section 3.2 for mode definition.
2. There may be none, one or more Idle states during the Stop Frame. See Section 4.3 for explanation
3. The next IRQSER cycle's Start Frame pulse may or may not start immediately after the turn-around clock of the Stop Frame.

3.2 IRQSER Cycle Control

There are two modes of operation for the IRQSER Start Frame.

1) **Quiet (Active) Mode** : Any device may initiate a Start Frame by driving the IRQSER low for one clock, while the IRQSER is Idle. After driving low for one clock the IRQSER must immediately be tri-stated without at any time driving high. A Start Frame may not be initiated while the IRQSER is Active. The IRQSER is *Idle* between Stop and Start Frames. The IRQSER is *Active* between Start and Stop Frames. This mode of operation allows the IRQSER to be Idle when there are no IRQ/Data transitions which should be most of the time.

Once a Start Frame has been initiated the Host Controller will take over driving the IRQSER low in the next clock and will continue driving the IRQSER low for a programmable period of three to seven clocks more (See section 4.1 below.). This makes a total low pulse width of four to eight clocks. Finally, the Host Controller will drive the IRQSER back high for one clock, then tri-state.

Any IRQSER Device which detects any transition on an IRQ/Data line for which it is responsible must initiate a Start Frame in order to update the Host Controller unless the IRQSER is already in an IRQSER Cycle and the IRQ/Data transition can be delivered in that IRQSER Cycle.

2) **Continuous (Idle) Mode** : Only the Host controller can initiate a Start Frame to update IRQ/Data line information. All other IRQSER agents become passive and may not initiate a Start Frame. IRQSER will be driven low for four to eight clocks by Host Controller. This mode has two functions. It can be used to stop or idle the IRQSER or the Host Controller can operate IRQSER in a continuous mode by initiating a Start Frame at the end of every Stop Frame.

An IRQSER mode transition can only occurs during the Stop Frame. Upon reset, IRQSER bus is defaulted to Continuous mode, therefore only the Host controller can initiate the first Start Frame. Slaves must continuously sample the Stop Frames pulse width to determine the next IRQSER Cycle's mode.

3.2.1 IRQ/Data Input Detection

To assure that an IRQSER agent does not miss a narrow low going IRQ/Data input pulse, it is recommended that an IRQSER device hold a low going IRQ/Data input through a low level extender. Also, according to the interrupt controller specification, a low going pulse must be at least 100nS wide to guarantee its recognition. Therefore if Legacy compatibility is required of the IRQSER agent, it is further recommended that a low going pulse of less then 40nS to 80nS be digitally filtered (i.e. ignored). IRQSER agents should hold any high to low transition on an IRQ/Data input that passes through the filter and keep it held low until the low state has successfully been transmitted to Host through the IRQSER bus. On the other hand, a short high going pulse is considered a glitch also in AT system and should be filtered by the IRQSER agent. A high going pulse of less then 40nS to 80nS should be digitally filtered. In order to prevent adding latency with the filters, it is also recommended that the START generation logic bypass the high and low going filters and generate a START immediately upon detection of a transition on the IRQ/Data input. This will sometimes cause an IRQSER Cycle without a change of state on any of the IRQ/Data signals.

3.3 IRQ/Data Frame

Once a Start Frame has been initiated, all IRQSER Devices must watch for the rising edge of the Start Pulse and start counting IRQ/Data Frames from there. Each IRQ/Data Frame is three clocks: Sample phase, Recovery phase, and Turn-around phase. During the Sample phase the IRQSER Device must drive the IRQSER low, if and only if, its last detected IRQ/Data value was low. If its detected IRQ/Data value is high, IRQSER must be left tri-stated. During the Recovery phase the IRQSER Device must drive the IRQSER high, if and only if, it had driven the IRQSER low during the previous Sample Phase. During the Tum-around Phase all IRQSER Devices must be tri-stated. All IRQSER Devices must drive the IRQSER line low at the appropriate sample point if its associated IRQ/Data line is low, regardless of which device initiated the Start Frame.

The Sample Phase for each IRQ/Data follows the low to high transition of the Start Frame pulse by a number of clocks equal to the IRQ/Data Frame times three, minus one. (e.g. The IRQ5 Sample clock is the sixth IRQ/Data Frame, (6 x 3) - 1 = 17th clock after the rising edge of the Start Pulse.)

IRQSER Sampling Periods		
IRQ/Data Frame	**Signal Sampled**	**# of clocks past Start**
1	IRQ0	2
2	IRQ1	5
3	SMI#	8
4	IRQ3	11
5	IRQ4	14
6	IRQ5	17
7	IRQ6	20
8	IRQ7	23
9	IRQ8	26
10	IRQ9	29
11	IRQ10	32
12	IRQ11	35
13	IRQ12	38
14	IRQ13	41
15	IRQ14	44
16	IRQ15	47
17	IOCHCK#	50
18	INTA#	53
19	INTB#	56
20	INTC#	59
21	INTD#	62
32:22	Unassigned	95

At the end of each Sample Phase the Host Controller will sample the state of the IRQSER line and replicate the status of the original IRQ/Data line at the input to the 8259 Interrupt Controller. Slot # 21-18 (INTA# / INTB# / INTC# / INTD#) is optional. The required minimum IRQ/Data Frames is 17 (See Sec. 3.4.3 for length definition).

3.4 Stop Cycle Control

Once all IRQ/Data Frames have completed the Host Controller will terminate IRQSER activity by initiating a Stop Frame. Only the Host Controller can initiate the Stop Frame. A Stop Frame is indicated when the IRQSER is low for two or three clocks. If the Stop Frame's low time is two clocks then the next IRQSER Cycle's sampled mode is the Quiet mode; and any IRQSER device may initiate a Start Frame in the second clock or more after the rising edge of the Stop Frame's pulse. If the Stop Frame's low time is three clocks then the next IRQSER Cycle's sampled mode is the Continuos mode; and only the Host Controller may initiate a Start Frame in the second clock or more after the rising edge of the Stop Frame's pulse.

3.4.1 IRQSER Cycle Maximum Sampling Period

Based on the maximum of 32 sampling slots assignment, plus Start / Stop pulse width variations, the longest IRQSER Cycle is approximately 110 PCI clocks.

3.4.2 Undefined Sample Periods

There are 17 IRQ/Data Frames defined for IRQs, SMI#, and IOCHCK#, and there are four optional IRQ/Data Frames, 18 through 21. The remaining eleven Frames are unassigned and may be utilized for System specific information transfer.

3.4.3 Minimum IRQ/Data Time Slot

The Host Controller must support IRQ/Data Frames 1 to 17. This produces the minimum IRQSER Cycle. The Host controller will have the Minimum IRQSER sampling period defined (programmable or hard-wired) in IRQSER Control Register bit [5..2] (See Section 6.0). All IRQSER Devices must decode the STOP Frame to terminate IRQSER Cycle sampling. For PCI to PCI bridge, IRQSER sampling period must be programmable and be set to same value as the Host's Controller.

3.4.4 IRQs Local to the Host Controller

The Host Controller is not required to broadcast its local IRQs on the IRQSER bus, therefore not all IRQs are visible on the IRQSER bus.

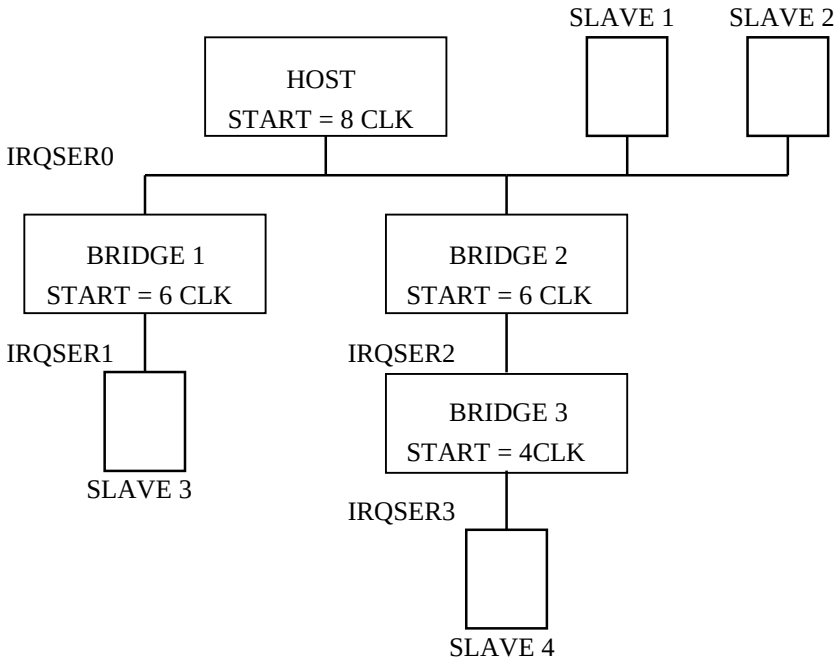
4.0 PCI to PCI Bridges

PCI to PCI Bridges must buffer the IRQSER from the secondary to the primary PCI Bus in order to support the IRQSER protocol. A bridge may transfer IRQSER status from the secondary to the primary PCI Bus either synchronously or asynchronously.

In a synchronous system, the Host Controller and the PCI to PCI Bridge must provide for a programmable Start pulse width of four, six, or eight clocks.

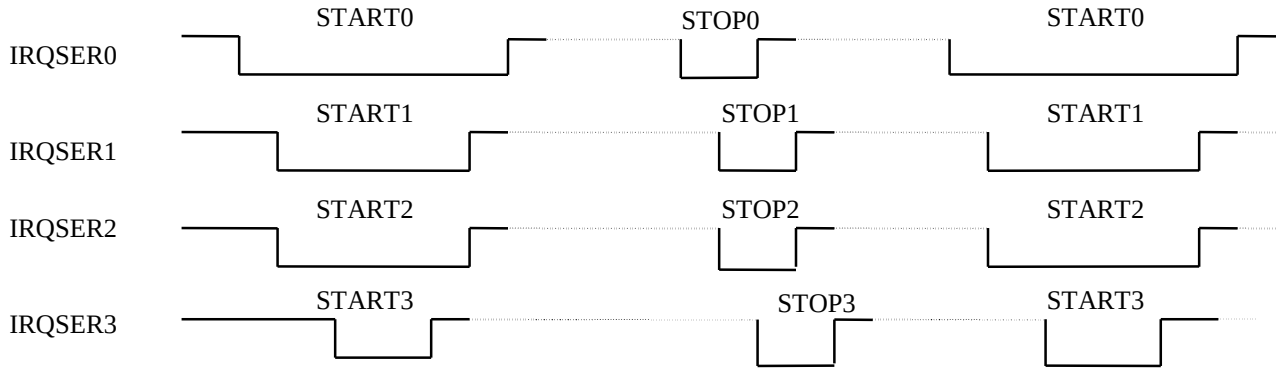
The following diagram shows a hierarchical IRQSER system with a synchronous bridge design. IRQSER1/2/3 is synchronous to IRQSER with different Start Frame pulse width settings.

4.1 Block Diagram

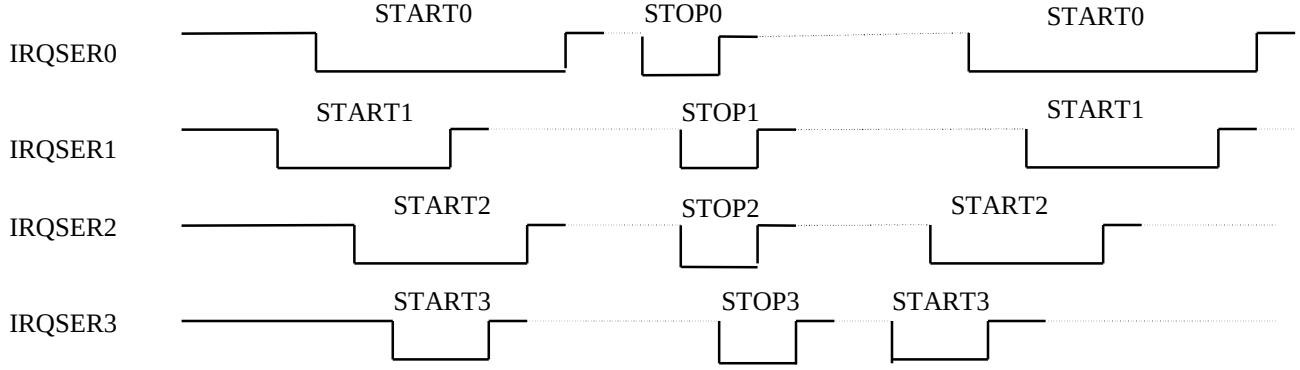


4.2 Timing Diagrams

Host initiated (down-stream) IRQSER Cycle



Slave initiated (up-stream) IRQSER Cycle (Slave 3 followed by Slave 4)



4.3 Cycle synchronization and Stop pulse propagation

The bridge is responsible for synchronizing its secondary IRQSER Cycle with the primary Host's IRQSER Cycle. An IRQSER Cycle may be initiated by either a primary or a secondary IRQSER agent(s). In case of an up-stream (secondary bus initiated) cycle, the Bridge will buffer secondary IRQSER input and report to primary bus as an IRQSER slave. For a down-stream (primary bus initiated) IRQSER Cycle, the Bridge will drive Start Frame on its secondary bus as the IRQSER Host would. In both cases the Bridge will operate the Start Frame on the secondary bus as if it were a Host.

The Stop Frame will propagate in an up-stream (from primary to secondary) direction only. By counting the primary side IRQ/Data Frames, the Bridge can detect the primary bus Stop Frame pulse and generate the secondary side Stop Frame pulse. The Stop pulse width must be same for the primary and the secondary bus. If the secondary bus's IRQ/Data Frames finish before the primary bus (up-stream case), the Bridge must insert idle state(s) in its secondary IRQSER bus while waiting for the primary bus's Stop Frame pulse. This will guarantee that the primary bus Host finishes the IRQSER Cycle first and most up-stream secondary bus finishes the IRQSER Cycle last.

5.0 Latency

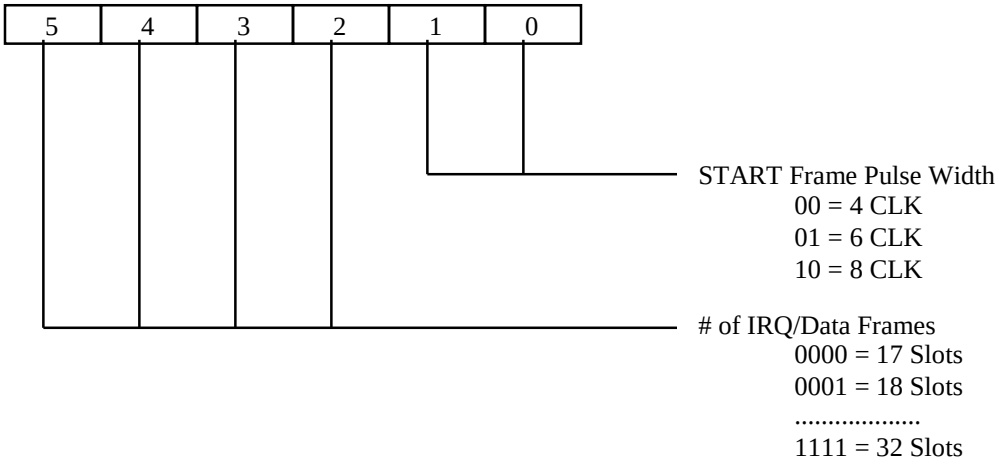
Latency for IRQ/Data updates over the IRQSER bus in bridge-less systems with the minimum IRQ/Data Frames of seventeen, will range up to 96 clocks (3.84uS with a 25MHz PCI Bus or 2.88uS with a 33MHz PCI Bus). If one or more PCI to PCI Bridge's are added to a system, the latency for IRQ/Data updates from the secondary or tertiary buses will be a few clocks longer for synchronous buses, and approximately double for asynchronous buses.

5.1 EOI/ISR Read Latency

Any serialized IRQ scheme has a potential implementation issue related to IRQ latency. IRQ latency could cause an EOI or ISR Read to precede an IRQ transition that it should have followed. This could cause a system fault. The host interrupt controller is responsible for ensuring that these latency issues are mitigated. The recommended solution is to delay EOIs and ISR Reads to the interrupt controller by the same amount as the IRQSER Cycle latency in order to ensure that these events do not occur out of order.

6.0 Host and Bridge IRQSER Control Register

Size 6 bits or larger
Type Write/Read
Power up xxxx00h



Bits 0-1 are the Start Frame's pulse width selection.

Bits 2-5 are number of IRQ/Data Frames. Host is required to support minimum of 17 IRQ/Data Frames and maximum of 32 IRQ/other Frames. For a Host that doesn't have programmability in the number of IRQ/Data Frames, these four bits can be hardwired. In order to match the Host Controller's IRQ/Data Frames setting, a PCI to PCI bridge must implement the number of IRQ/Data Frames register .

*** NOTE : There is no programming bit required for IRQSER Slaves.
*** NOTE : The bit location shown in above chart is for reference only. Actual implementation may have different bit combination.

7.0 AC/DC Specifications

All IRQSER agents must drive/sample IRQSER synchronously related to the rising edge of PCI bus clock. IRQSER signal conforms to the electrical specification of PCI bus. Electrical parameters will follow PCI specification section 4, sustained tri-state.

8.0 Reset and Initialization

The IRQSER bus uses PCIRST# as its reset signal and follows the PCI bus reset mechanism. The IRQSER pin is tri-stated by all agents while PCIRST# is active. With reset, IRQSER Slaves and Bridges are put into the (continuous) IDLE mode. The Host Controller is responsible for starting the initial IRQSER Cycle to collect system's IRQ/Data default values. The system then follows with the Continuous/Quiet mode protocol (Stop Frame pulse width) for subsequent IRQSER Cycles. It is Host Controller's responsibility to provide the default values to 8259's and other system logic before the first IRQSER Cycle is performed. For IRQSER system suspend, insertion, or removal application, the Host controller should be programmed into Continuous (IDLE) mode first. This is to guarantee IRQSER bus is in IDLE state before the system configuration changes.