Semiconductors

AN10313

Reduce CPU overhead with Intelligence Interrupt Arbitration (I2A) feature

Rev. 01 — 25 June 2004 Application note

Document information

Info	Content
Keywords	I2A, Interrupt Arbitration, Arbitration
Abstract	The Intelligent Interrupt Arbitration technology is an enhanced feature to the Impact Line of Philips Semiconductors Industrial UARTs. The advantage of implementing the I2A is explained and the operation of the arbitrating systems is described in this application note. This application note is applicable to the following Philips Semiconductors Industrial UARTs: SC28L202, SC28L194, and SC28L198.

PHILIPS

Revision history

Rev	Date	Description
1	20040625	Application note (9397 750 13347)

Contact information

For additional information, please visit: **http://www.semiconductors.philips.com**

For sales office addresses, please send an email to: **sales.addresses@www.semiconductors.philips.com**

1. Introduction

Philips Semiconductors, the world's leading supplier of Universal Asynchronous Receiver Transmitter (UART) devices for industrial applications, developed the Intelligence Interrupt Arbitration (I2A) feature to significantly reduce the amount of time required for interrupt handling. The I2A technology improves system performance by minimizing the processor time for servicing interrupt requests because the I2A technology can identify the highest priority interrupt request and channel it first to the processor. In addition, the I2A arbitrates multiple interrupt requests and reports the interrupt context through the modification of the interrupt vector. The information contained in the interrupt vector includes the device location, type of interrupt, priority, and the fill level of the FIFO. The microprocessor can read the information to determine the interrupt sources.

Previously, the processing of the interrupt arbitration was done by the microprocessor. The microprocessor relied on the traditional interrupt service by searching and testing various status registers on the assertion of the interrupt request signal. The traditional interrupt processing took a great deal of the microprocessor time to service and diminished the time available for the microprocessor for handling other functions.

The I2A system with the enhancement of the bus cycle time helps free up the microprocessor and also reduces the interrupt overhead when used in higher speed environment. The use of the I2A feature improves the overall system performance. The I2A feature is available in the SC28L202 Dual (2-channel) UART, the SC28L194 Quad (4-channel) UART, and the SC28L198 Octal (8-channel) UART, which are the Impact Line of Philips Semiconductors Industrial UARTs.

This application note describes the operation of the I2A feature and the process of the interrupt arbitration system. The enabling/programming of the feature and the mechanism of the arbitration are discussed in the next paragraph.

2. The I2A operation

The I2A operation shown in Figure 1 and described below is mainly about the process of the arbitration system. The purpose of the arbitration process is to give the user the flexibility of having a single activity, namely the asserting of IACKN signal, which steers the interrupt service directly to the routing for the device generating the interrupt and providing the interrupt context to the host.

The interrupt sources and four registers are involved in the operation of the I2A. The interrupt sources, which are capable of generating an interrupt output, are receivers, transmitters, two detectors (change of state and break change), and counters/timers. The four registers, which have effects on the arbitration process, are IMR (Interrupt Mask Register), ISR (Interrupt Status Register), ICR (Interrupt Control Register), and CIR (Current Interrupt Register).

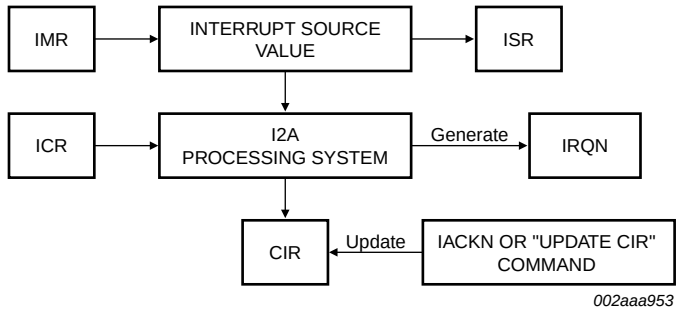

Fig 1. The operation of I2A

The details function of the four registers in the arbitration processing is described in the following Sections 2.1 through 2.4.

2.1 IMR — Interrupt Mask Register

Programming the IMR bits can enable the interrupt sources. The IMR selects which bits in the ISR enter the interrupt arbitration process. If both the ISR bit and the corresponding IMR bit are set, the interrupt source of the ISR bit is entered the arbitration process. If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the interrupt (IRQN) output.

2.2 ISR — Interrupt Status Register

The ISR provides the status of all potential interrupt sources. When generating an interrupt arbitration value, the IMR masks the contents of the ISR.

2.3 CIR — Current Interrupt Register

The CIR captures the value that is winning the interrupt arbitration. The CIR is updated at the beginning of an interrupt acknowledge (IACKN) bus cycle or in response to an update CIR command. The contents remain in the CIR until another IACKN cycle or update CIR command occurs.

The CIR contains a complete description of the interrupting source, which contains the channel number, type of interrupts source, and the fill level of FIFO. A single read operation to the CIR provides all the information needed to get the most common interrupt sources. The value of the interrupting source in the CIR is used to drive the interrupt vector modification and the global interrupt registers, which are indirect address to the CIR.

2.4 ICR — Interrupt Control Register

The ICR provides the interrupt threshold value for use by the interrupt arbiter. The interrupt arbiter uses the value of the threshold to be compared with the value of the highest priority interrupt source. Writing a value to the ICR can program the threshold value.

3. Interrupt arbitration process

The interrupt arbitration process is used to determine which an interrupt request should be presented to the host. The arbitration is carried out between the interrupt source's value and the threshold value. The value is derived from three fields: the channel number, type of interrupt source, and the fill level of FIFO.

The interrupt arbitration process works as follows:

1. The IMR controls the enabling of the interrupt sources and allows the interrupt sources to enter the arbitration process.
2. The interrupt source value entered the arbitration process is compared with the threshold value, which can be programmed by writing a value to the ICR.
3. If the interrupt source value exceeds the threshold value, the interrupt (IRQN) output will be asserted (LOW). If the interrupt sources values are equal, then the highest channel number prevails in the arbitration process.
4. The host either asserts the IACKN signal or writes the update CIR command so the CIR captures the value of the interrupt source. The value contains the channel number, type of interrupt's source, and the fill level of FIFO.

Here is the example of the four interrupts' source values entering the arbitration process with the following sequence:

1. 0xAD means 5 bytes, Receiver without error, Channel B
2. 0xAF means 5 bytes, Receiver without error, Channel D
3. 0x6C means 3 bytes, Receiver without error, Channel A
4. 0x4E means 2 bytes, Receiver without error, Channel C

Remark: The first sequence entered the arbitration process earlier than the second sequence, and so on.

In the arbitration process, the four interrupts' source values arbitrate themselves to determine their priority as follows:

1. 0xAF means 5 bytes, Receiver without error, Channel D
2. 0xAD means 5 bytes, Receiver without error, Channel B
3. 0x6C means 3 bytes, Receiver without error, Channel A
4. 0x4E means 2 bytes, Receiver without error, Channel C

Remark: The first priority is higher than the second priority, and so on.

After prioritizing, the four interrupts' source values are being compared with the programmable threshold value. For example:

- If the threshold value is programmed to 0x40, then the Receiver D interrupts first, followed by the Receiver B, then the Receiver A, and last the Receiver C.
- If the threshold value is programmed to 0x80, then the Receiver D interrupts first, followed by the Receiver B. The Receiver A and C do not interrupt because their value does not exceed the threshold value.

- If the threshold value is programmed to 0xC0, then none of the Receivers interrupt because their value does not exceed the threshold value.

4. Conclusion

As applications increase in complexity, there will be a need to minimize any CPU overhead. The implementation of the I2A feature embedded in the Impact Line of Philips Semiconductors Industrial UARTs greatly reduces the microprocessor time required to service the UART interrupts. The I2A system allows the microprocessor to retrieve the information of the highest priority of the interrupt occur without the need of the microprocessor to arbitrate the interrupts.

5. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

6. Contents

Date of release: 25 June 2004
Document order number: 9397 750 13347

Published in the U.S.A.