RTI Errata

- **ATAM893S**
- **ATAM894P**
- **ATAR092P**
- **ATAR892P**
- **ATAR090H**
- **ATAR890H**
- **ATAR080F**
- **ATA6020N**
- **ATAR510F**
- **ATAR862N3/N4/N8**

The latest product enhancement (PCN HC030703 DI enhancement) introduced to all MARC4 products during 2003, leads, under some conditions, to unforeseen and unwanted behavior. The circuit may not process interrupts correctly. All above mentioned MARC4 products are affected. This document describes what happens internally in the controller and how the issue was solved.

Result of Analyzing Misbehavior in Interrupt Processing

Status

Figure 1 on page 2 shows a simplified block diagram of the MARC4 interrupt controller. It consists of a pending register that keeps all interrupt requests until they have been processed, the priority decoder and the priority register that select the interrupt with the highest priority, and the active register that keeps the interrupts in process until the corresponding RTI has been executed.

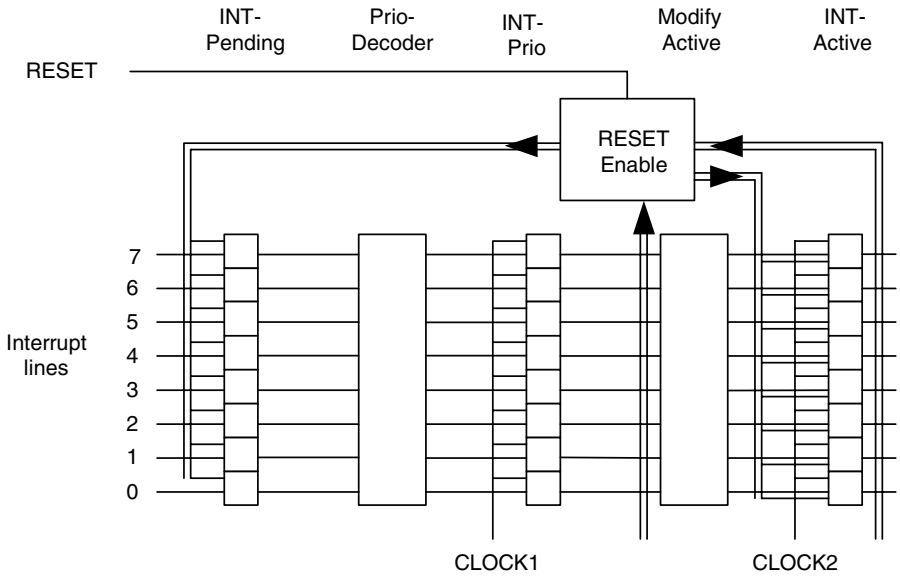
MARC4 4-bit Microcontrollers

ATAM893S
ATAM894P
ATAR092/892P
ATAR090/890H
ATAR080F
ATA6020N
ATAR510F
ATAR862N3/4/8

Errata Sheet

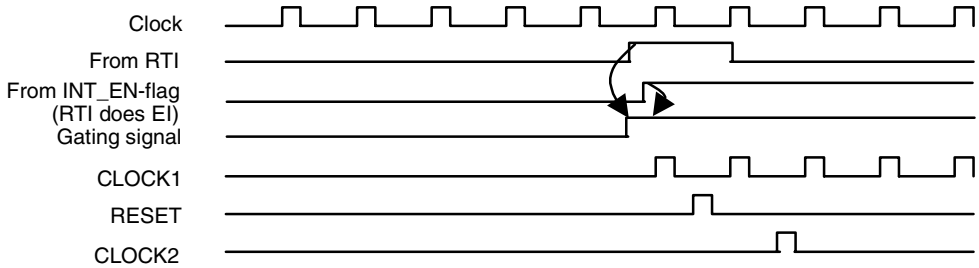
Rev. 4814A–4BMCU–06/04

Figure 1. General Structure of the Interrupt Controller



During the RTI, the RESET Enable block controls that the RESET only removes the just finished interrupt, i.e., the one with the highest priority, from the pending and the active register. The first CLOCK1 is redundant, the next CLOCK1 signal following the RESET transfers the modified information from the pending to the priority register. If more than 1 interrupt is pending, only the flag for the interrupt with the highest priority is set to “1” in the priority register. CLOCK2 updates the active register. If there is already an interrupt active, it has not been overwritten by the new priority. Only interrupts with a priority higher than the one just active are added to the active register. This allows the controller to return to the previous interrupt service routine when it has finished processing the interrupt with the higher priority.

Figure 2. General Timing of the Reset Sequence



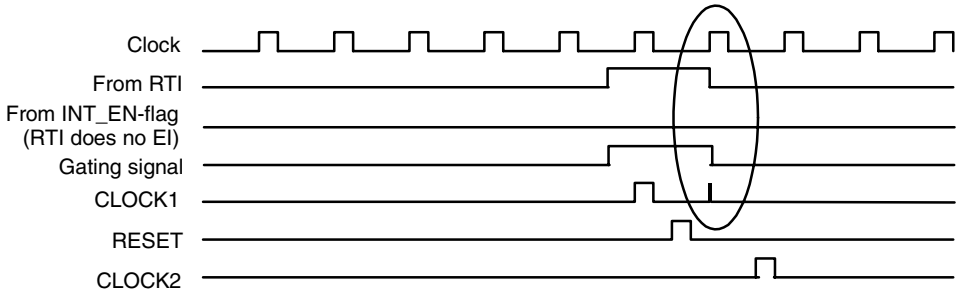
Modification

The modification introduced to all MARC4 circuits has been initiated due to the fact that since “Issue 4” described in the “MARC4 Microcontroller Enhancement Report (revision 4668B-4BMCU-04/04)” might cause faulty behavior in conjunction with the DI/RTI commands. Unfortunately, the measure to remove the EI-function from the RTI resulted in an undetected side effect.

If an interrupt occurs at the same time as a DI-command, the internal timing of the following RTI is slightly changed. The result, visible in the application, additionally depends on the command following the DI (no effect if NOP follows DI, no more interrupts if a CALL follows DI). The behavior and the timing of the reset sequence do not change if the interrupt does not occur simultaneously with a DI command

The root cause is an unforeseen influence of DI, the command following DI and the RTI to the CLOCK1-gating signal. The duration of this decoded signal is shortened if the enable interrupt flag is reset (interrupt disabled). Depending on the command following the DI, the valid and necessary second CLOCK1 is reduced to a spike (NOP) or completely disappears (CALL, LIT_x...).

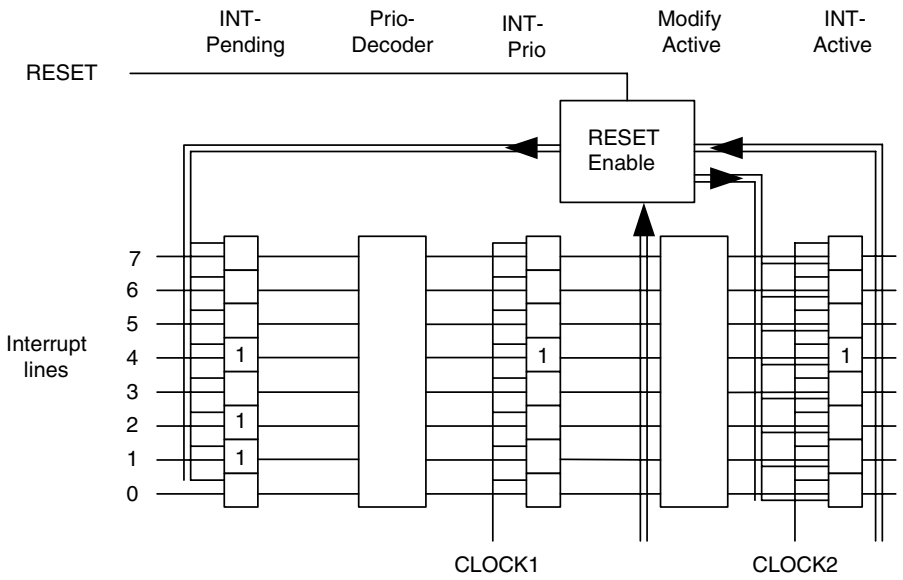
Figure 3. Spike in the Reset Sequence



Example 1

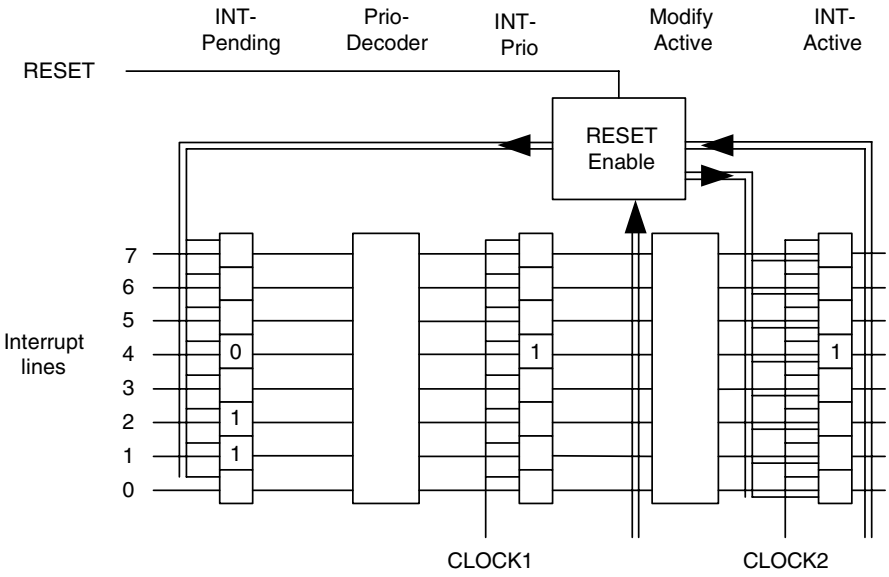
The effect of this behavior to the interrupt registers is the following. If an interrupt exactly meets a DI, it is nevertheless stored in the pending register, transferred to the priority register and then to the active register (Figure 4). It stays in the active register as long as the controller does not return from the corresponding interrupt service routine.

Figure 4. Status Before RTI



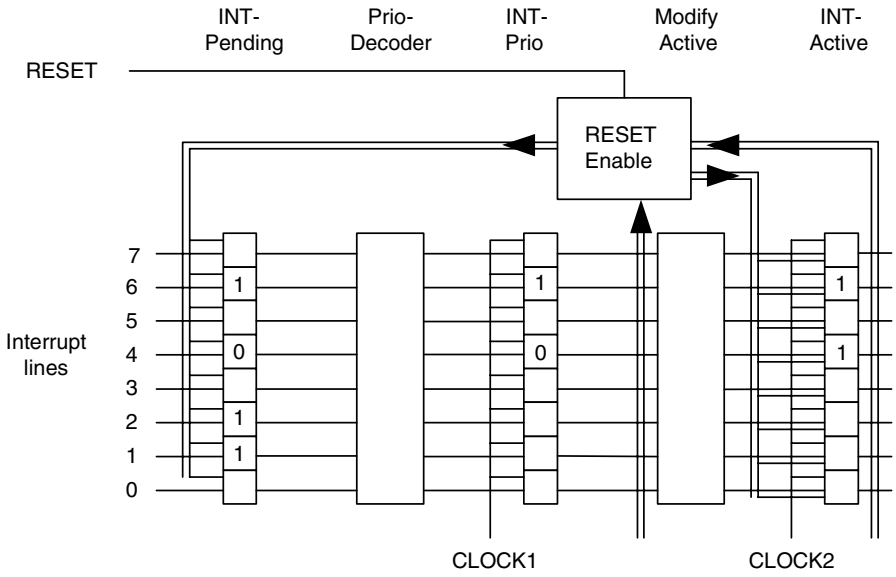
The above described modified reset sequence starts with the RTI. If the command following the DI is a CALL, the first CLOCK1 does not change anything, the RESET forces the relevant bits from pending and active register to zero, the next CLOCK1 signal is missing and leaves the priority register unchanged, the adjacent CLOCK2 takes over the information from the priority to the active register again (Figure 5). As long as the active register is set, the controller cannot execute a new interrupt of the same or a lower priority.

Figure 5. Status After RTI



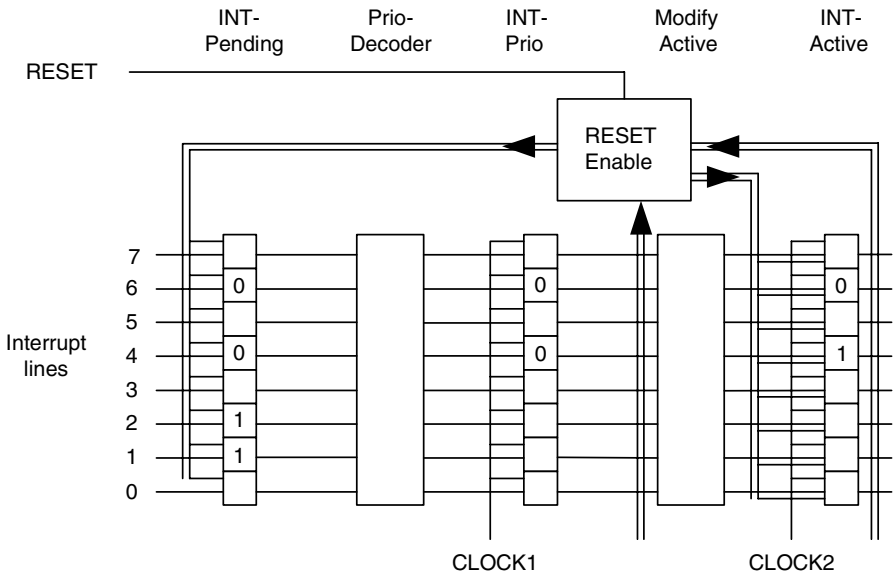
Even an interrupt with a higher priority, entering the pending register, does not finish this lock-up condition. This event leads to an update of the priority register. Only one interrupt, the one with the highest priority, is stored here. It does not delete the wrong contents from the active register but adds the higher priority interrupt to it. Deleting an interrupt from the active register is only possible if the corresponding bit of the priority register is set.

Figure 6. Interrupt with Higher Priority Appears



The RTI resets the bits of the active and the pending register that are marked in the priority register. This new content is evaluated and transferred to the priority register. Then, the higher interrupt will be finished, but the lower one will be still kept in the active register, continuously disabling interrupts of the same or lower priority.

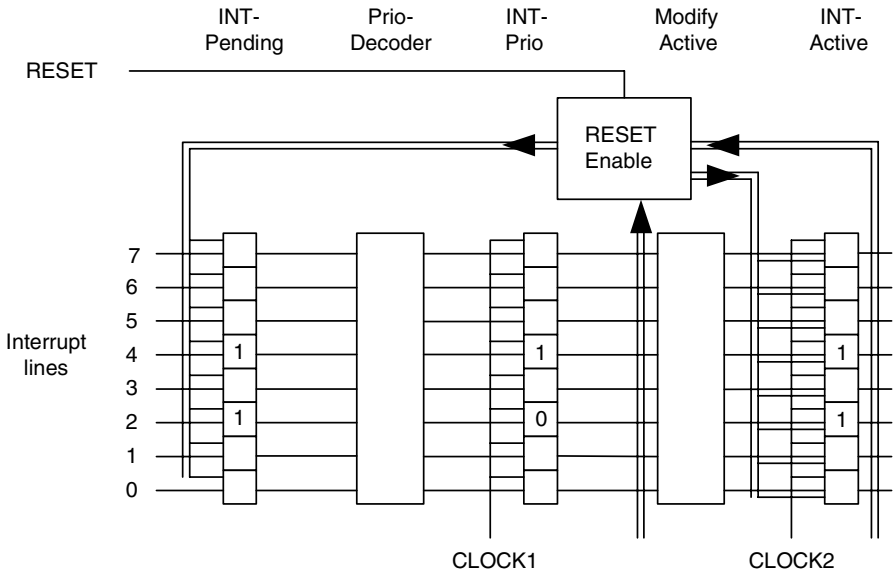
Figure 7. Interrupt with Higher Priority Finished



Example 2

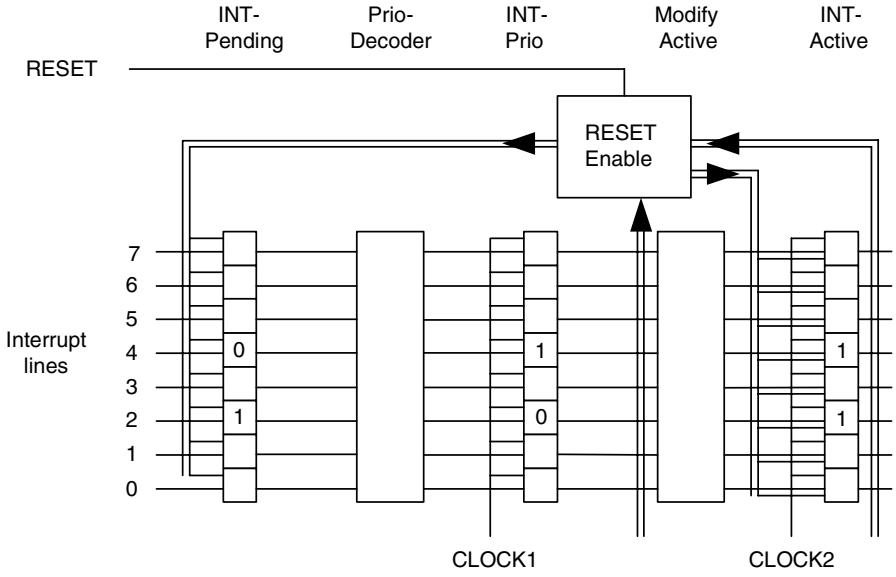
The following example describes a special condition: A first interrupt (for example INT2) has entered the pending register normally, is active and the service routine is just under execution. This routine contains a DI command. If a second interrupt (for example INT4) with a higher priority exactly meets this DI, the DI will be executed but the controller jumps to the service routine related to the higher interrupt. The register status at this moment is given in Figure 8.

Figure 8. Interrupted Interrupt



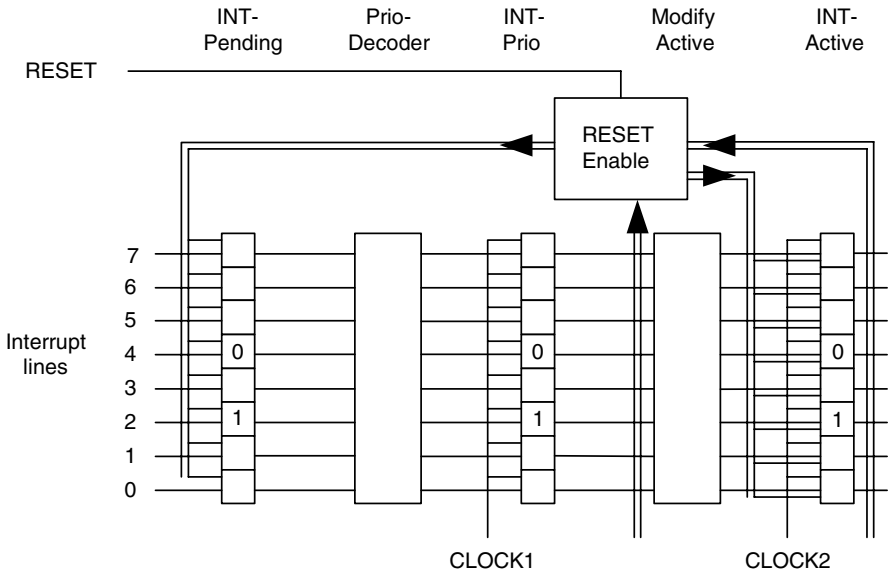
Finishing INT4 causes the program control to switch back to the INT2 service routine that was interrupted before. Due to the missing CLOCK1 after RESET, however, the priority and the active register stay unchanged after the reset sequence, while the INT4 flag in the pending register is deleted.

Figure 9. After Return from INT4



Then, the controller continues processing the INT2 service routine. Leaving this routine with the RTI activates once more the reset sequence. The highest interrupt, marked in priority and active register, is deleted from the pending and the active register with RESET. CLOCK1 and CLOCK2 update the priority and the active register as shown in Figure 10.

Figure 10. After Return from INT2

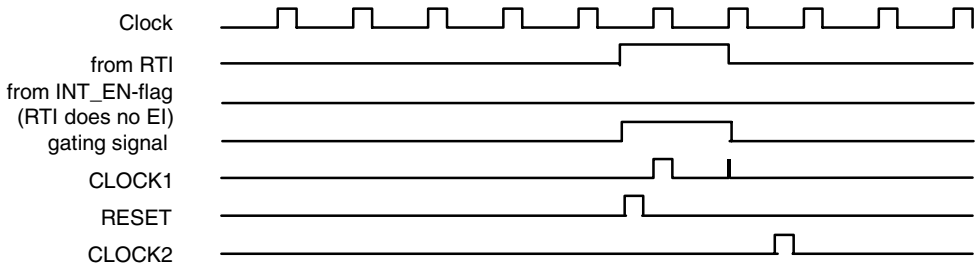


This time it is not the interrupt that met the DI (INT4), but the one that has already been active before (INT2) that blocks further interrupts of the same or lower priority.

Enhancement

The described function is caused by an undetected influence to the timing of the interrupt controller's reset sequence. To solve this issue, Atmel will correct the timing. First, the MTP-parts ATAM893 and ATAM894 will incorporate this enhancement. The modification under test conditions is shown in Figure 11.

Figure 11. Modified Timing



Moving the RESET in front of the first CLOCK1 solves the issue of the missing second CLOCK1. Now, the reset sequence starts with the RESET of the active and pending register, then CLOCK1 updates the priority register and CLOCK2 updates the active register. To ensure proper operation under all conditions, intensive simulations with different software configurations are ongoing. The hardware revisions of the corrected MTPs will be named ATAM893T and ATAM894R. They will be available in Q3/2004.

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

4814A–4BMCU–06/04