# 13 INTERRUPT CONTROLLER

The KS32C50100 interrupt controller has a total of 21 interrupt sources. Interrupt requests can be generated by internal function blocks and at external pins.

The ARM7TDMI core recongnizes two kinds of interrupts: a normal interrupt request (IRQ), and a fast interrupt request (FIQ). Therefore all KS32C50100 interrupts can be categorized as either IRQ or FIQ. The KS32C50100 interrupt controller has an interrupt pending bit for each interrupt source.

Four special registers are used to control interrupt generation and handling:

- Interrupt priority registers. The index number of each interrupt source is written to the pre-defined interrupt priority register field to obtain that priority. The interrupt priorities are pre-defined from 0 to 20.
- Interrupt mode register. Defines the interrupt mode, IRQ or FIQ, for each interrupt source.
- Interrupt pending register. Indicates that an interrupt request is pending. If the pending bit is set, the interrupt pending status is maintained until the CPU clears it by writing a "1" to the appropriate pending register. When the pending bit is set, the interrupt service routine starts whenever the interrupt mask register is "0". The service routine must clear the pending condition by writing a "1" to the appropriate pending bit. This avoids the possibility of continuous interrupt requests from the same interrupt pending bit.
- Interrupt mask register. Indicates that the current interrupt has been disabled if the corresponding mask bit is "1". If an interrupt mask bit is "0" the interrupt will be serviced normally. If the global mask bit (bit 21) is set to "1", no interrupts are serviced. However, the source's pending bit is set if the interrupt is generated. When the global mask bit has been set to "0", the interrupt is serviced.

## INTERRUPT SOURCES

The 21 interrupt sources in the KS32C50100 interrupt structure are listed, in brief, as follows:

**Table 13-1 KS32C50100 Interrupt Sources**

| Index Values | Interrupt Source |
|---|---|
| [20] | $I^2C$-bus interrupt |
| [19] | Ethernet controller MAC Rx interrupt |
| [18] | Ethernet controller MAC Tx interrupt |
| [17] | Ethernet controller BDMA Rx interrupt |
| [16] | Ethernet controller BDMA Tx interrupt |
| [15] | HDLC channel B Rx interrupt |
| [14] | HDLC channel B Tx interrupt |
| [13] | HDLC channel A Rx interrup |
| [12] | HDLC channel A Tx interrupt |
| [11] | Timer 1 interrupt |
| [10] | Timer 0 interrupt |
| [9] | GDMA channel 1 interrupt |
| [8] | GDMA channel 0 interrupt |
| [7] | UART1 receive & error interrupt |
| [6] | UART1 transmit interrupt |
| [5] | UART0 receive & error interrupt |
| [4] | UART0 transmit interrupt |
| [3] | External interrupt 3 |
| [2] | External interrupt 2 |
| [1] | External interrupt 1 |
| [0] | External interrupt 0 |

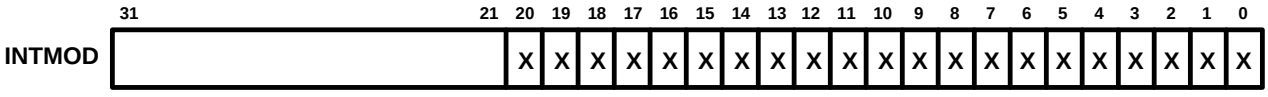
# INTERRUPT CONTROLLER SPECIAL REGISTERS

## INTERRUPT MODE REGISTER

Bit settings in the interrupt mode register, INTMOD, specify if an interrupt is to be serviced as a fast interrupt (FIQ) or a normal interrupt (IRQ).

**Table 13-2 INTMOD Register**

| Register | Offset Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INTMOD | 0x4000 | R/W | Interrupt mode register | 0x00000000 |




**[20:0] Interrupt mode bits**

**NOTE**: Each of the 21 bits in the interrupt mode enable register, INTMOD, corresponds to an interrupt source. When the source interrupt mode bit is set to 1, the interrupt is processed by the ARM7TDMI core in FIQ (fast interrupt) mode. Otherwise, it is processed in IRQ mode (normal interrupt). The 21 interrupt sources are mapped as follows:

[20] I²C interrupt
[19] Ethernet controller MAC Rx interrupt
[18] Ethernet controller MAC Tx interrupt
[17] Ethernet controller BDMA Rx i nterrupt
[16] Ethernet controller BDMA Tx interrupt
[15] HDLC channel B Rx interrupt
[14] HDLC channel B Tx interrupt
[13] HDLC channel A Rx interrupt
[12] HDLC channel A Tx interrupt
[11] Timer 1 interrupt
[10] Timer 0 interrupt
[9] GDMA channel 1 interrupt
[8] GDMA channel 0 interrupt
[7] UART1 receive & error interrupt
[6] UART1 transmit interrupt
[5] UART0 receive & error interrupt
[4] UART0 transmit interrupt
[3] External interrupt 3
[2] External interrupt 2
[1] External interrupt 1
[0] External interrupt 0

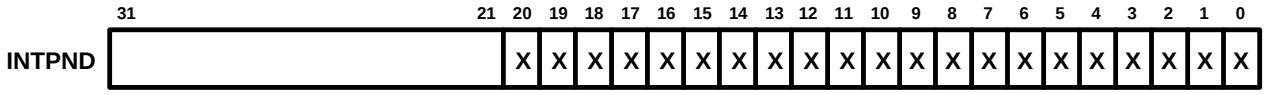
**Figure 13-1 Interrupt Mode Register (INTMOD)**

## INTERRUPT PENDING REGISTER

The interrupt pending register, INTPND, contains interrupt pending bits for each interrupt source. This register has to be cleared at the top of a interrupt service routine.

**Table 13-3 INTPND Register**

| Register | Offset Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INTPND | 0x4004 | R/W | Interrupt pending register | 0x00000000 |




**[20:0] Interrupt pending bits**

**NOTE**: Each of the 21 bits in the interrupt pending register, INTPND, corresponds to an interrupt source. When an interrupt request is generated, its pending bit is set to 1. The service routine must then clear the pending condition by writing a 1 to the appropriate pending bit at start. The 21 interrupt sources are mapped as follows:

[20] I $^2$C interrupt
[19] Ethernet controller MAC Rx interrupt
[18] Ethernet controller MAC Tx interrupt
[17] Ethernet controller BDMA Rx inter rupt
[16] Ethernet controller BDMA Tx interrupt
[15] HDLC channel B Rx interrupt
[14] HDLC channel B Tx interrupt
[13] HDLC channel A Rx interrupt
[12] HDLC channel A Tx interrupt
[11] Timer 1 interrupt
[10] Timer 0 interrupt
[9] GDMA channel 1 interrupt
[8] GDMA channel 0 interrupt
[7] UART1 receive & error interrupt
[6] UART1 transmit interrupt
[5] UART0 receive & error interrupt
[4] UART0 transmit interrupt
[3] External interrupt 3
[2] External interrupt 2
[1] External interrupt 1
[0] External interrupt 0

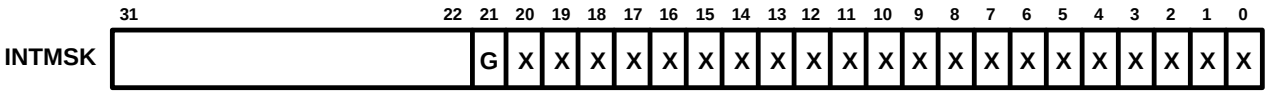
**Figure 13-2 Interrupt Pending Register (INTPND)**

## INTERRUPT MASK REGISTER

The interrupt mask register, INTMSK, contains interrupt mask bits for each interrupt source.

**Table 13-4 INTMSK Register**

| Register | Offset Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INTMSK | 0x4008 | R/W | Interrupt mask register | 0x003FFFFF |




**[20:0] Individual interrupt mask bits**

**NOTE**: Each of the 21 bits in the interrupt mask register, INTMSK, (except for the global mask bit, G) corresponds to an interrupt source. When a source interrupt mask bit is 1, the interrupt is not serviced by the CPU when the corresponding interrupt request is generated. If the mask bit is 0, the interrupt is serviced upon request. And if global mask bit (bit 21) is 1, no interrupts are serviced. (However, the source pending bit is set whenever the interrupt is generated.) After the global mask bit is cleared, the interrupt is serviced. The 21 interrupt sources are mapped as follows:

[20] I 2C interrupt
[19] Ethernet controller MAC Rx interrupt
[18] Ethernet controller MAC T x interrupt
[17] Ethernet controller BDMA Rx interrupt
[16] Ethernet controller BDMA Tx interrupt
[15] HDLC channel B Rx interrupt
[14] HDLC channel B Tx interrupt
[13] HDLC channel A Rx interrupt
[12] HDLC channel A Tx interrupt
[11] Timer 1 interrupt
[10] Timer 0 interrupt
[9] GDMA channel 1 interrupt
[8] GDMA channel 0 interrupt
[7] UART1 receive & error interrupt
[6] UART1 transmit interrupt
[5] UART0 receive & error interrupt
[4] UART0 transmit interrupt
[3] External interrupt 3
[2] External interrupt 2
[1] External interrupt 1
[0] External interrupt 0

**[21] Global interrupt mask bit**
0 = Enable interrupt requests
1 = Disable all interrupt requests

**Figure 13-3 Interrupt Mask Register (INTMSK)**

## INTERRUPT PRIORITY REGISTERS

The interrupt priority registers, INTPRI0–INTPRI5, contain information about which interrupt source is assigned to the pre-defined interrupt priority field. Each INTPRIn register value determines the priority of the corresponding interrupt source. The lowest priority value is priority 0, and the highest priority value is priority 20.

The index value of each interrupt source is written to one of the above 21 positions (see Figure 13-4). The position value then becomes the written interrupt's priority value. The index value of each interrupt source is listed in Table 13-1.

**Table 13-5 Interrupt Priority Register Overview**

| Registers | Offset Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INTPRI0 | 0x400C | R/W | Interrupt priority register 0 | 0x03020100 |
| INTPRI1 | 0x4010 | R/W | Interrupt priority register 1 | 0x07060504 |
| INTPRI2 | 0x4014 | R/W | Interrupt priority register 2 | 0x0B0A0908 |
| INTPRI3 | 0x4018 | R/W | Interrupt priority register 3 | 0x0F0E0D0C |
| INTPRI4 | 0x401C | R/W | Interrupt priority register 4 | 0x13121110 |
| INTPRI5 | 0x4020 | R/W | Interrupt priority register 5 | 0x00000014 |

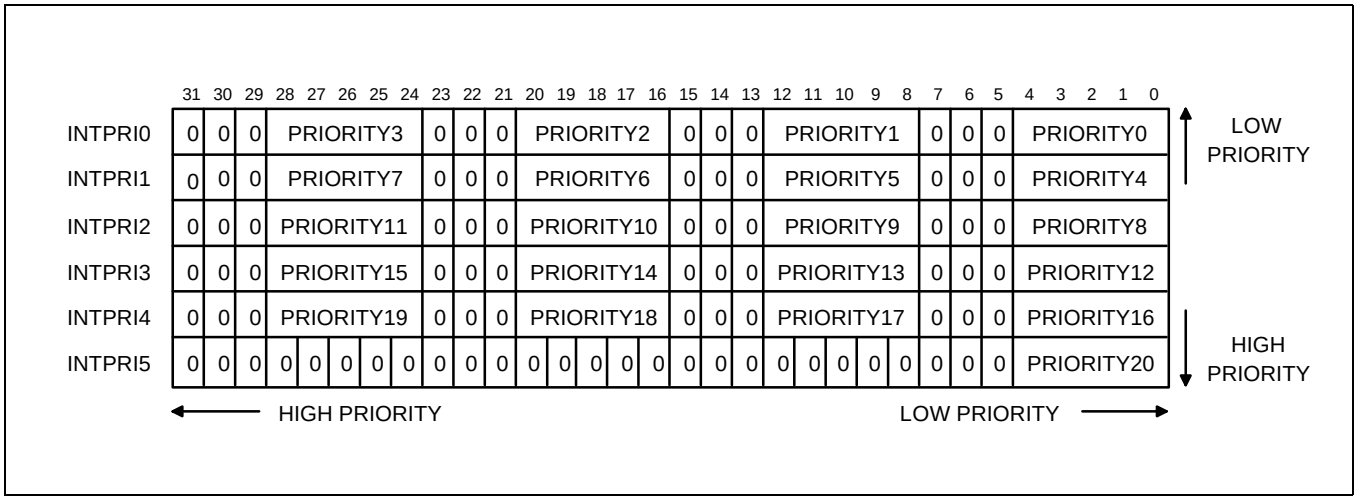



**Figure 13-4 Interrupt Priority Registers (INTPRIn)**

## INTERRUPT OFFSET REGISTER

The interrupt offset register, INTOFFSET, contains the interrupt offset address of the interrupt which has the highest priority among the pending interrupts. The content of the interrupt offset address is "bit position value of the interrupt source << 2".

If all interrupt pending bits are "0" when you read this register, the return value is "0x00000054".

This register is valid only under the IRQ or FIQ mode in the ARM7TDMI. In the interrupt service routine, you should read this register before changing the CPU mode.

INTOSET_FIQ/INTOSET_IRQ register can be used to get the highest priority interrupt without CPU mode change. Other usages are similar to INTOFFSET.

**NOTE**

If the lowest interrupt priority (priority 0) is pending, the INTOFFSET value will be "0x00000000". The reset value will, therefore, be changed to "0x00000054" (to be differentiated from interrupt pending priority 0).

**Table 13-6 INTOFFSET Register**

| Register | Offset Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INTOFFSET | 0x4024 | R | Interrupt offset register | 0x00000054 |
| INTOSET_FIQ | 0x4030 | R | FIQ Interrupt offset register | 0x00000054 |
| INTOSET_IRQ | 0x4034 | R | IRQ Interrupt offset register | 0x00000054 |

### INTERRUPT PENDING BY PRIORITY REGISTER

The interrupt pending by priority register, INTPNDPRI, contains interrupt pending bits which are re-ordered by the INTPRIn register settings. INTPNDPRI[20] is mapped to the interrupt source of whichever bit index is written into the priority 20 field of the INTPRIn registers.

This register is useful for testing. To validate the interrupt pending by priority value, you can obtain the highest priority pending interrupt from the interrupt offset register, INTOFFSET.

**Table 13-7 INTPNDPRI Register**

| Register | Offset Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INTPNDPRI | 0x4028 | R | Interrupt pending by priority | 0x00000000 |

### INTERRUPT PENDING TEST REGISTER

The interrupt pending test register, INTPNDTST, is used to set or clear INTPND and INTPNDPRI. If user writes data to this register, it is written into both the INTPND register and INTPNDPRI register. The interrupt pending test register, INTPNDTST, is also useful for testing.

For INTPND, the same bit position is updated with the new comming data. For INTPNDPRI, the mapping bit position by INTPRIn registers is updated with the new comming data to keep with the contents of the INTPND register.

**Table 13-8 INTPNDTST Register**

| Register | Offset Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INTPNDTST | 0x402C | W | Interrupt pending test register | 0x00000000 |

# NOTES