

```

*****
* Copyright (c) Texas Instruments Inc 2002, 2003, 2004, 2005, 2008
*
* Use of this software is controlled by the terms and conditions found in the
* license agreement under which this software has been supplied.
*****
;Module: INTC
;File: vectors.asm - vector table
;*****
;*****
    .mmregs ; included by Andrea
    .include "lpva200.inc" ; included by Andrea

    .C54CM_off
    .CPL_off
    .ARMS_off

;*****
;      predefined stack operation modes
;*****
;      USE_RET      : 2x16-bit fast return mode (RETA used)
;      NO_RET      : 2x16-bit slow return mode (RETA not used)
;      C54X_STK    : 32-bit fast return mode

;*****
    .ref _c_int00
    ;.ref _sarISR
    .def RST

;*****
    .sect "vectors"
    .align 256
    .global _VECSTART
;*****
_VECSTART:
RST:      .ivec    reset_isr, USE_RET      ; Reset / Software Interrupt #0

NMI:      .ivec    nmi_isr                ; Nonmaskable Interrupt

INT0:     .ivec    int0_isr                ; External User Interrupt #0

INT1:     .ivec    int1_isr                ; External User Interrupt #1

TINT:     .ivec    tim_isr                ; Timer #0 / Software Interrupt #4

PROG0:    .ivec    i2s0_mmc0_tx_isr        ; Programmable 0 Interrupt

UART:     .ivec    uart_isr                ; IIS #1 Receive Interrupt

PROG1:    .ivec    i2s0_mmc0_rx_isr        ; Programmable 1 Interrupt

DMA:      .ivec    dma_isr                ; DMA Interrupt

PROG2:    .ivec    i2s1_mmc1_tx_isr        ; Programmable 2 Interrupt

COPROCFIT: .ivec    coprocfft_isr          ; Coprocessor FFT Module Interrupt

PROG3:    .ivec    i2s1_mmc1_rx_isr        ; Programmable 3 Interrupt

LCD:      .ivec    lcd_isr                ; LCD Interrupt

SARADC:   .ivec    saradc_isr              ; SAR ADC Interrupt

XMT2:     .ivec    i2s2_tx_isr            ; I2S2 Tx Interrupt

RCV2:     .ivec    i2s2_rx_isr            ; I2S2 Rx Interrupt

XMT3:     .ivec    i2s3_tx_isr            ; I2S3 Tx Interrupt

```

```

RCV3:      .ivec    i2s3_rx_isr      ; I2S3 Rx Interrupt

RTC:       .ivec    rtc_isr         ; RTC interrupt

SPI:       .ivec    spi_isr         ; SPI Receive Interrupt

USB:       .ivec    usb_isr         ; USB Transmit Interrupt

GPIO:      .ivec    gpio_isr        ; GPIO Interrupt

EMIF:      .ivec    emif_error_isr   ; EMIF Error Interrupt

I2C:       .ivec    i2c_isr         ; IIC interrupt

BERR:      .ivec    berr_isr        ; Bus Error Interrupt

DLOG:      .ivec    dlog_isr        ; Emulation Interrupt - DLOG

RTOS:      .ivec    rtos_isr        ; Emulation Interrupt - RTOS

RTDXRCV:   .ivec    rtdxrcv_isr     ; Emulation Interrupt - RTDX receive

RTDXXMT:   .ivec    rtdxxmt_isr     ; Emulation Interrupt - RTDX transmit

EMUINT:    .ivec    emuint_isr      ; Emulation monitor mode interrupt

SINT30:    .ivec    sint30_isr      ; Software Interrupt #30

SINT31:    .ivec    sint31_isr      ; Software Interrupt #31

    .text ; included by Andrea
    .def reset_isr ; included by Andrea
    .ref _c_int00 ; included by Andrea

    .align 2 ; included by Andrea
reset_isr:
;    *port(#0x1C01) = #0x0          ; Clear idles
    bit (ST1, #11) = #1            ; Disable interrupts
    @IVPD_L = #(RST >> 8) || mmap()
    @IVPH_L = #(RST >> 8) || mmap()
    bit (ST3, #7) = #0              ; Clear bus error interrupts

    @#IFR0_L = #0xFFFF || mmap()   ; clear all pending interrupts
    @#IFR1_L = #0xFFFF || mmap()

; ----- THIS IS THE RELEVANT SECTION: -----
    *port(#0x0001) = #0x000E        ; take all clock domains out of idle
    (especially MPORT)
    nop_16                          ; flush the pipeline
    nop_16
    nop_16
    nop_16
    nop_16
    nop_16
    idle                            ; execute IDLE instruction to apply changes
; -----

; *****
;      Reset all peripherals
; *****
    *port(#0x1C04) = 0x20
    nop_16
    *port(#0x1C05) = 0x00FF ; Reset all peripherals
    nop_16
    goto    _c_int00

```

```

nmi_isr:      goto    nmi_isr
int0_isr:     goto    int0_isr
int1_isr:     goto    int1_isr
tim_isr:      goto    tim_isr
i2s0_mmc0_tx_isr: goto    i2s0_mmc0_tx_isr
uart_isr:     goto    uart_isr
i2s0_mmc0_rx_isr: goto    i2s0_mmc0_rx_isr
dma_isr:      goto    dma_isr
i2s1_mmc1_tx_isr: goto    i2s1_mmc1_tx_isr
coprocfft_isr: goto    coprocfft_isr
i2s1_mmc1_rx_isr: goto    i2s1_mmc1_rx_isr
lcd_isr:      goto    lcd_isr
saradc_isr:   goto    saradc_isr
i2s2_tx_isr:  goto    i2s2_tx_isr
i2s2_rx_isr:  goto    i2s2_rx_isr
i2s3_tx_isr:  goto    i2s3_tx_isr
i2s3_rx_isr:  goto    i2s3_rx_isr
rtc_isr:      goto    rtc_isr
spi_isr:      goto    spi_isr
usb_isr:      goto    usb_isr
gpio_isr:     goto    gpio_isr
emif_error_isr: goto    emif_error_isr
i2c_isr:      goto    i2c_isr
berr_isr:
                bit(ST3,#7) = #0    ; Clear bus error interrupts
                goto    berr_isr
dlog_isr:     goto    dlog_isr
rtos_isr:     goto    rtos_isr
rtdxrcv_isr:  goto    rtdxrcv_isr
rtdxxmt_isr:  goto    rtdxxmt_isr
emuint_isr:   goto    emuint_isr
sint30_isr:   goto    sint30_isr
sint31_isr:   goto    sint31_isr

```

.text

```

*****
** Name      : no_isr
**

```

**

```
**                                                                 **
** Purpose   : Spurious interrupt handler                       **
**                                                                 **
** Author    :                                                                 **
**                                                                 **
*****
no_isr:                goto no_isr
```