Welcome to E-XFL.COM

Understanding Embedded - DSP (Digital Signal Processors)

Embedded - DSP (Digital Signal Processors) are specialized microprocessors designed to perform complex mathematical computations on digital signals in real-time. Unlike general-purpose processors, DSPs are optimized for high-speed numeric processing tasks, making them ideal for applications that require efficient and precise manipulation of digital data. These processors are fundamental in converting and processing signals in various forms, including audio, video, and communication signals, ensuring that data is accurately interpreted and utilized in embedded systems.

Applications of Embedded - DSP (Digital Signal Processors)

Details	
Product Status	Obsolete
Type	Floating Point
Interface	AIF, I²C, I2S, SPI, TDM, UART
Clock Rate	260MHz
Non-Volatile Memory	Boot ROM (32kB)
On-Chip RAM	676 kB
Voltage - I/O	1.80V
Voltage - Core	1.20V
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	49-WFBGA, WLCSP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/cirrus-logic/wm0011ecs-r

CODE DATA DOWNLOAD

The Software Code download comprises one Code Header packet (as described above), followed by multiple Code Data packets.

The Code Packet definition for the Code Data is:

- CMD = 0x03
- LEN = Data Length (in bytes)
- ADDR = Start Address for code data
- DAT = Data words

On completion of the full set of Code Data packet downloads, the $\overline{IRQ}$ output is de-asserted and the WM0011 will commence execution of the downloaded software.

Note that completion of the Code Data packets is determined by the code image length that is contained within the Code Header packet (DAT). Software execution commences at the start address (ADDR) - also contained in the Code Header packet.

Boot status and error codes are reported via the UART interface, and via the SPI interface during the Code Data download - see "Boot Status and Error Reporting".

PLL CONFIGURATION DOWNLOAD

The PLL Configuration download operation requires a single Code Packet to be sent to the WM0011. The download may be achieved via the SPI Slave or SPI Master (eg. Flash Memory) interfaces. If the Custom fuses are not programmed, then only the SPI (Slave) download method is possible.

The Code Packet definition for PLL Configuration download is:

- CMD = 0x04
- LEN = 0x00_0018
- ADDR = 0x0000_0000
- DAT = PLL Configuration Data

The "DAT" portion must comprise 24 bytes, corresponding to the intended contents of the clocking configuration registers listed below. The CCM_CLK_CTRL1 register is transmitted first.

- CCM_CLK_CTRL1 (4 bytes, see Table 19)
- CCM_CLK_CTRL2 (4 bytes, see Table 20)
- CCM_CLK_CTRL3 (4 bytes, see Table 21)
- CCM_PLL_LOCK_CTRL (4 bytes, see Table 22)
- UART_BAUD_LSW (1 byte, see Table 118)
- UART_BAUD_MSW (1 byte, see Table 119)
- Padding (2 bytes)
- SPI_SCLKDIV (4 bytes, see Table 123)

On receipt of a valid PLL Configuration packet, the control registers noted above will be updated with the received data, and the new clocking configuration will become effective.

Note that the SPI_SCLKDIV register on the WM0011 is only updated if the selected boot method is SPI Master. In all other cases, the SPI_SCLKDIV portion of the PLL download is ignored and discarded. Note that the SPI Master boot method is only possible via the Custom fuse data settings.

Boot status and error codes are reported via the UART interface, and via the SPI interface during the PLL Configuration download - see "Boot Status and Error Reporting".

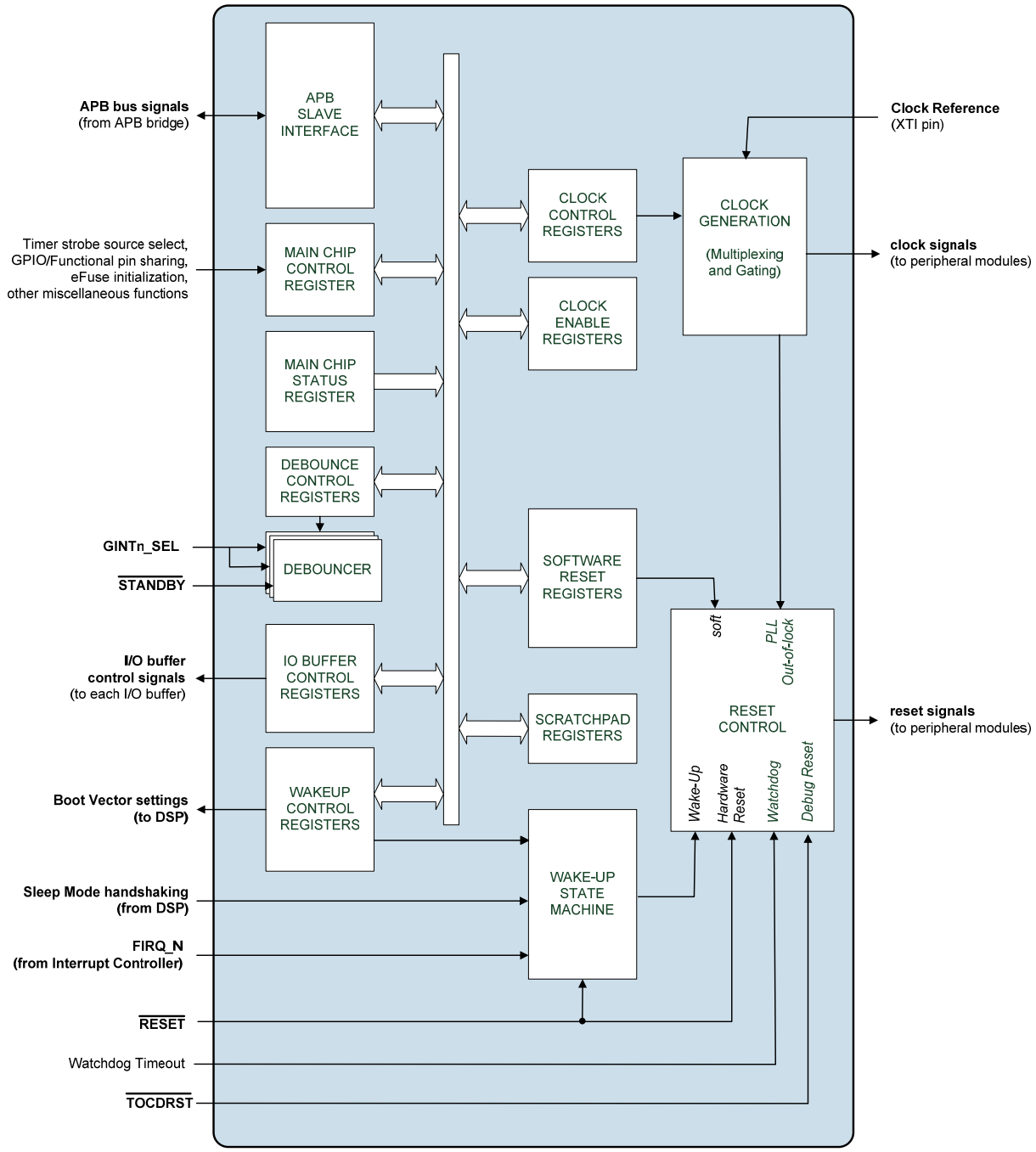


Figure 20 Chip Configuration Module (CCM) Block Diagram

CLOCKING CONTROL

The CCM registers allow full configuration of the WM0011 clocking options, including clock dividers, clock multiplexers and the 2-stage Phase Locked Loop (PLL). Individual clock enable registers are provided for each peripheral module.

RESET CONTROL

The CCM registers allow flexible control of the Warm Reset functions. The Warm Reset conditions are individually maskable, and status readback is also provided. Software Reset control registers allow each peripheral to be reset individually.

TIMER (TMR) MODULES

TIMER 1 - BASE ADDRESS 0xF001_0000

TIMER 2 - BASE ADDRESS 0xF001_0020

TIMER 3 - BASE ADDRESS 0xF001_0040

TIMER DESCRIPTION

The WM0011 provides three timers, which count up from 0, or count down from TMR_MAX_CNT. The counters are enabled using the TMR_ENA bit, and count direction is selected using the TMR_DIR bit (see Table 49).

The number of APBCLK clock cycles per count is determined by the TMR_PRESCALE register. When TMR_PRESCALE = 00h, the module will count at the APBCLK clock rate.

The TMR_MODE bit enables a selectable external trigger to be used to start the timer count. The TMC_INC bit configures the external trigger either as a 'start' trigger or as an alternate 'clock' signal. When TMR_MODE=1 and TMR_INC=0, the count rate is controlled only by the external trigger (ie. not by APBCLK).

The TMR_1SHOT bit selects whether the Timer automatically re-starts after the 'end of count' condition has been reached.

Note that the timer clock enable bit (TMR_CLK_ENA) is on the CCM_CLK_ENA register, and the timer reset bit (TMR_SOFTRST_N) is in the CCM_SOFTRST register. Note that these signals are common to all three Timer (TMR) modules.

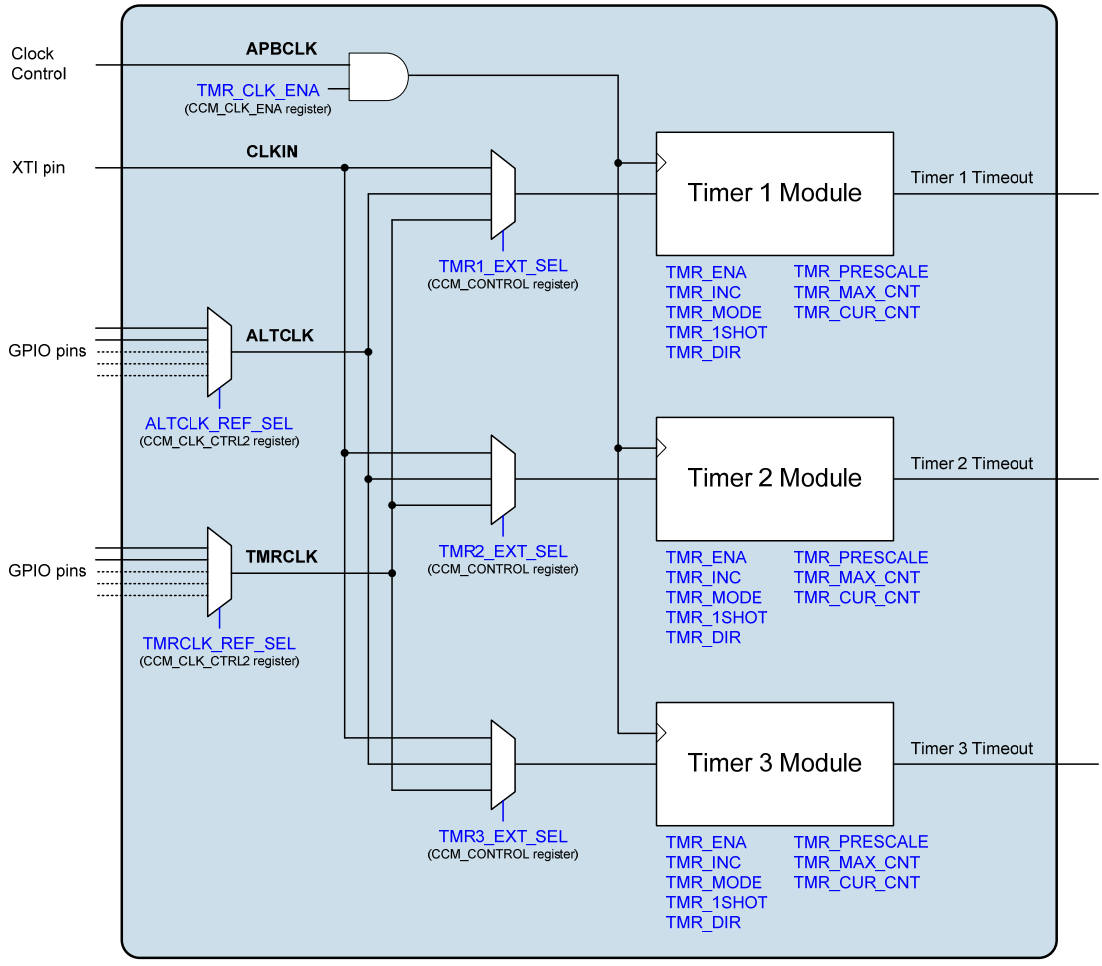


Figure 23 Timer (TMR) Modules Block Diagram

I2C INTERRUPTS

The I2C module can generate an interrupt when any of the conditions described in the I2C_STATUS register occurs. The interrupt conditions provide status indications of the I2C bus transactions.

The I2C interrupt control registers are illustrated in Figure 30.

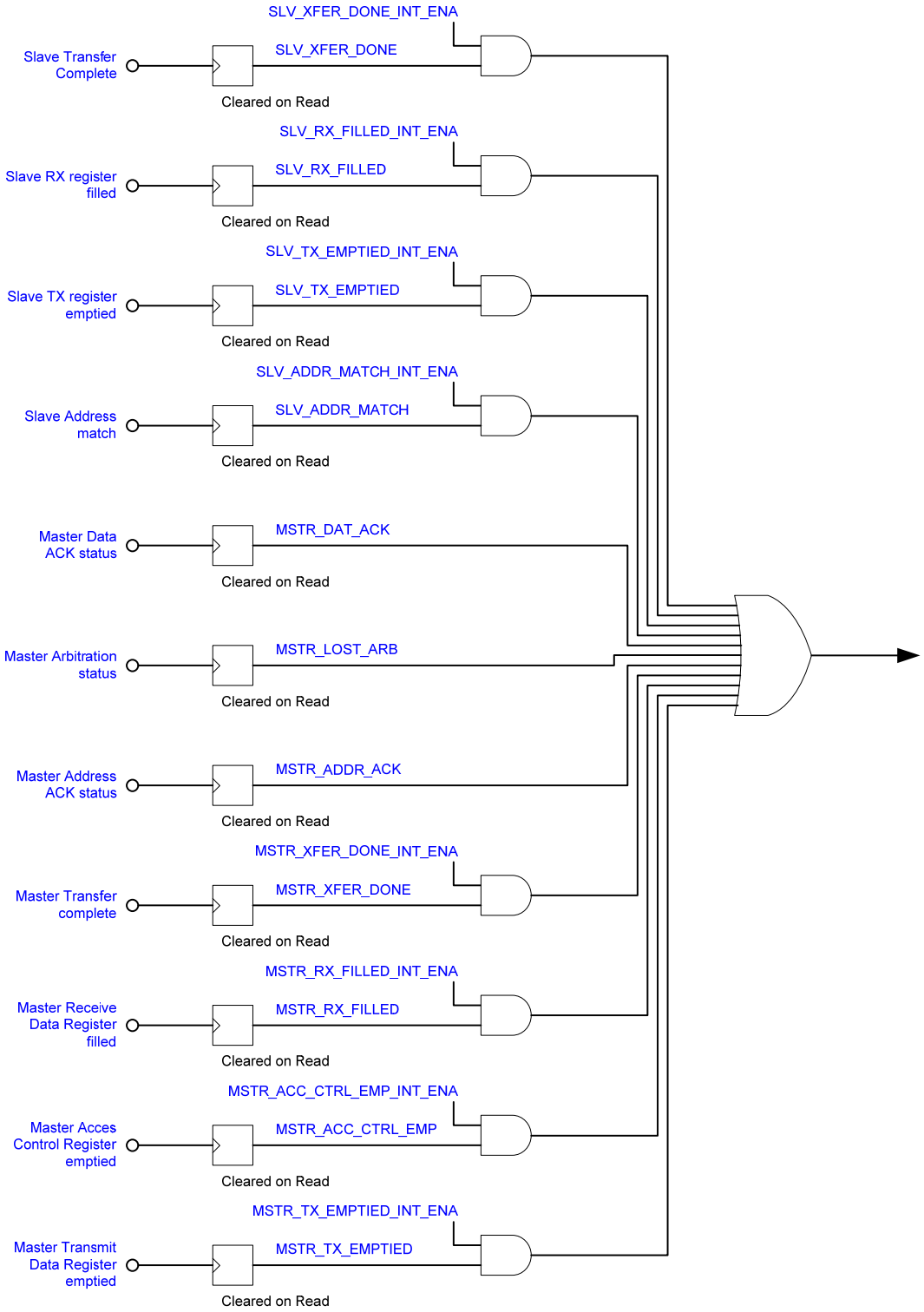


Figure 30 I2C Interrupts

I2C_SLV_ADDR – I²C SLAVE ADDRESS REGISTER

The I²C slave address is held in the SLV_ADDR register. Note that this is the 7-bit Device Address, and does not include the Read/Write bit. The equivalent 8-bit Device Address comprises the 7 bits of SLV_ADDR, and the R/W bit in the LSB position. This is illustrated by an example in Table 62.

I²C ACTION	SLV_ADDR	8-BIT DEVICE ADDRESS
Write	0x38 (hex), 011 1000 (binary)	0x70 (hex), 0111 0000 (binary)
Read		0x71 (hex), 0111 0001 (binary)

Table 62 Illustration of 7-bit SLV_ADDR compared with 8-bit Device Address

I2C_SLV_ADDR
I2C SLAVE RECEIVE DATA REGISTER

Address = 0xF002_0078 **Default value = 0x0000_0000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
31:7	Reserved		0x000_0000	
6:0	SLV_ADDR[6:0]	RW	0x00	This field holds the 7 bit I²C slave address. (Note this does not include the R/W bit.)

Table 63 I2C_SLV_ADDR Register

INTERRUPT CONTROLLER (IRQC) MODULE

BASE ADDRESS 0xF005_0000

INTERRUPT CONTROLLER (IRQC) FEATURES

- 10 interrupt inputs from peripheral modules, including cascaded GPIO input
- De-bounced input from the $\overline{\text{STANDBY}}$ pin
- Register control of the $\overline{\text{IRQ}}$ output pin
- 2 register-controlled software interrupts
- Configurable interrupt logic using edge or level detection
- Individual Mask control for each interrupt
- Configurable FIRQ_N output to the Wake-Up FSM
- Configurable IRQ_N and FIRQ_N outputs to the Wake-Up FSM and HiFi EP™ DSP Core

The IRQC module supports 11 inputs, comprising Interrupt signals from peripheral modules (eg. I2C Module), the cascaded input from the GPIO module, and also the de-bounced input from the $\overline{\text{STANDBY}}$ pin.

Any of the inputs may be selected as interrupt sources for the IRQC module, and used to generate the IRQ_N and FIRQ_N outputs to the HiFi EP™. A priority-encoded readback is available on the occurrence of an IRQ_N or FIRQ_N interrupt.

The FIRQ_N ('Fast Interrupt) signal is also an input to the CCM module, providing a configurable 'Wakeup' control signal.

Note that many of the peripheral module interrupt signals are also independently provided as direct inputs to the HiFi EP™.

The IRQC module provides software capability to generate user-defined interrupts to the HiFi EP™ and also to directly control the $\overline{\text{IRQ}}$ output pin logic level.

The inputs and outputs of the IRQC module are illustrated in Figure 15.

INPUT / OUTPUT CONTROL

Each signal described in the IRQC_DIR must be configured as an input or as an output. The software interrupts (bits [15:14] and the $\overline{\text{IRQ}}$ output (bit [0]) should be configured as outputs. All other bits should be configured as inputs.

The logic level each input is observable after de-metastability logic and inversion logic by reading IRQC_IN. When input inversion is selected (using IRQC_INV), value read from IRQC_IN will be the opposite logic level from the signal source.

In the case of output signals, these are controlled by the respective IRQC_OUT register bits. Note that these outputs are not affected by the IRQC_INV bits.

LEVEL/EDGE INTERRUPT CONTROL

The interrupt inputs are configured as level sensitive or edge sensitive using the IRQC_EDGE1 and IRQC_EDGE0 registers.

In Level-sensitive configuration, the interrupt is asserted when the applicable logic level is detected. Active High is selected when IRQC_INV=0; Active Low is selected when IRQC_INV=1. Note that the interrupt cannot be cleared whilst the Active logic level is present (and unmasked) on the respective IRQC input.

In Edge-sensitive configuration, the interrupt is asserted when the applicable logic transition is detected. This may be the rising (leading) edge, falling (trailing) edge, or both edges. The active edge(s) will cause the respective IRQC_IRQ_STS and/or IRQC_FIRQ_STS bit to be set (as controlled by the respective mask bits). Note that the active edge(s) are inverted when IRQC_INV=1.

IRQC_EDGE0 – IRQ EDGE DETECTION 0 REGISTER

IRQC_EDGE0 IRQ EDGE DETECTION 0 REGISTER																															
Address = 0xF005_0010																Default value = 0x0000_0000															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
31:16	Reserved		0x0	
15		RW	0x0	Reserved - set to 0 only
14		RW	0x0	Reserved - set to 0 only
13	Reserved		0x0	
12	Reserved		0x0	
11	TMR2_INT_EDGE0	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE1
10	TMR1_INT_EDGE0	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE1
9	DMA_INT_EDGE0	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE1
8	WDT_INT_EDGE0	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE1
7	STBY_INT_EDGE0	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE1
6	I2C_INT_EDGE0	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE1
5	AIF2_INT_EDGE0	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE1
4	AIF1_INT_EDGE0	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE1
3	UART_INT_EDGE0	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE1
2	SPI_INT_EDGE0	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE1
1	GPIO_INT_EDGE0	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE1
0		RW	0x0	Reserved - set to 0 only

Table 85 IRQC_EDGE0 Register

IRQC_EDGE1 – IRQ EDGE DETECTION 1 REGISTER

IRQC_EDGE1 IRQ EDGE DETECTION 1 REGISTER																															
Address = 0xF005_0014																Default value = 0x0000_0000															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
31:16	Reserved		0x0	
15		RW	0x0	Reserved - set to 0 only
14		RW	0x0	Reserved - set to 0 only
13	Reserved		0x0	
12	Reserved		0x0	
11	TMR2_INT_EDGE1	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE0
10	TMR1_INT_EDGE1	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE0
9	DMA_INT_EDGE1	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE0
8	WDT_INT_EDGE1	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE0
7	STBY_INT_EDGE1	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE0
6	I2C_INT_EDGE1	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE0
5	AIF2_INT_EDGE1	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE0
4	AIF1_INT_EDGE1	RW	0x0	Selects Level or Edge interrupt detection, depending on *EDGE0

IRQC_FIRQ_STS – IRQ FAST INTERRUPT STATUS REGISTER

IRQC_FIRQ_STS IRQ FAST INTERRUPT STATUS REGISTER																															
Address = 0xF005_0034																Default value = 0x0000_0000															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
31:16	Reserved		0x0	
15		RO	0x0	Reserved - reads back 0 only
14		RO	0x0	Reserved - reads back 0 only
13	Reserved		0x0	
12	Reserved		0x0	
11	TMR2_FIRQ_STS	RO	0x0	TMR2 (Timer 2) Interrupt Status
10	TMR1_FIRQ_STS	RO	0x0	TMR1 (Timer 1) Interrupt Status
9	DMA_FIRQ_STS	RO	0x0	DMA Interrupt Status
8	WDT_FIRQ_STS	RO	0x0	WDT (Watchdog Timer) Interrupt Status
7	STBY_FIRQ_STS	RO	0x0	$\overline{\text{STANDBY}}$ Interrupt Status
6	I2C_FIRQ_STS	RO	0x0	I2C Interrupt Status
5	AIF2_FIRQ_STS	RO	0x0	AIF2 Interrupt Status
4	AIF1_FIRQ_STS	RO	0x0	AIF1 Interrupt Status
3	UART_FIRQ_STS	RO	0x0	UART Interrupt Status
2	SPI_FIRQ_STS	RO	0x0	SPI Interrupt Status
1	GPIO_FIRQ_STS	RO	0x0	GPIO Interrupt Status
0		RO	0x0	Reserved - reads back 0 only

Table 94 IRQC_FIRQ_STS Register

WDT_MAX_CNT WATCHDOG MAXIMUM COUNT REGISTER																															
Address = 0xF007_0008																Default value = 0x0000_FFFF															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
31:0	WDT_MAX_CNT	RW	0x0000_FFFF	Count value (measured in APBCLK cycles) before watchdog times out.

Table 107 WDT_MAX_CNT Register

WDT_CUR_CNT – WATCHDOG CURRENT COUNT REGISTER

WDT_CUR_CNT WATCHDOG CURRENT COUNT REGISTER																															
Address = 0xF007_000C																Default value = 0x0000_FFFF															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
31:0	WDT_CUR_CNT	RO	0x0000_FFFF	The current counter value of the watchdog timer. When WDT_CUR_CNT reaches the value held in WDT_MAX_CNT, watchdog is activated. After de-assertion of $\overline{\text{RESET}}$ and whenever the watchdog timer is disabled (WDT_ENA = 0), the WDT_CUR_CNT value will reflect the value of the WDT_MAX_CNT register. This is a read only register

Table 108 WDT_CUR_CNT Register

WDT_RST_LEN – WATCHDOG RESET PULSE LENGTH REGISTER

WDT_RST_LEN controls the duration (pulse length) of the Watchdog Reset signal. This field represents the number of APBCLK cycles for which the Watchdog Reset signal is asserted (Active Low output to the Reset Controller).

WDT_RST_LEN WATCHDOG RESET PULSE LENGTH REGISTER																															
Address = 0xF007_0010																Default value = 0x0000_00FF															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
31:8	Reserved		0x00_0000	
7:0	WDT_RST_LEN	RW	0x0000_00FF	Number of APBCLK cycles for which the Watchdog Timer Reset signal is asserted (Active Low output to the Reset Controller).

Table 109 WDT_RST_LEN Register

SERIAL PERIPHERAL INTERFACE (SPI) MODULE

BASE ADDRESS 0xF030_0000

SPI FEATURES

- Configurable Data/Clock phase and Clock polarity
- Data word length can be on 8, 16, 24, 32 or 64 bits
- Selectable data bit ordering (LSB first or MSB first)
- Polarity selection for the Slave Select ($\overline{\text{SPISS}}$) signal
- Programmable soft reset capability
- Selectable “auto-retransmit” mode
- Selectable “early-tx-data transition” mode
- Byte-packing options
- Multiple Transfer mode allowing multiple data words per $\overline{\text{SPISS}}$ assertion
- Master Mode Slave Select “shaping” (configurable $\overline{\text{SPISS}}$ set-up, hold and wait times)

SPI MASTER MODE

The SPI_MISO pin direction is Input.

The $\overline{\text{SPISS}}$, SPISCLK, and SPIMOSI pins are driven as Outputs, but only during an actual data transfer. After a master data transfer has completed, these signals are tri-stated. This allows for lower power usage, and for usage in a multi-master SPI scenario.

Note that the above behaviour can be adjusted using the SPI_MM_MODE register (see Table 121), which allows constant driving of these master mode output signals whenever the SPI block is enabled.

The SPI Master mode is selected by setting SPI_MODE=0. The user should configure the desired SPISCLK, $\overline{\text{SPISS}}$, and MISO/MOSI parameters, and lastly set SPI_ENA=1 to enable the SPI module.

The SPI module will then be in Master mode, and will initiate a SPI data transfer when data is written to the SPI_DAT data register. The outgoing SPI_DAT data is double-buffered, allowing for the queuing of the “next word” to be transferred, while the current word is being shifted out.

SPI SLAVE MODE

The $\overline{\text{SPISS}}$, SPISCK, and SPIMOSI pin direction is input.

The SPIMISO pin is driven as Output, but only during an actual data transfer. After a slave data transfer has completed (i.e. de-assertion of $\overline{\text{SPISS}}$ by the master), this signal is tri-stated. This allows for usage in a multi-slave SPI scenario.

The SPI Slave mode is selected by setting SPI_MODE=1. The user should configure the desired SPISCLK, $\overline{\text{SPISS}}$, and MISO/MOSI parameters, and lastly set SPI_ENA=1 to enable the SPI module.

The SPI module will then be in Slave mode, and will wait for a SPI data transfer from an external master. Once initiated, the incoming data bits are shifted in until one word is received. The incoming data word is placed in a holding register, allowing for the reception of the serial bits of a “new current word”, while the previous word is being queued for transfer to the AHB system side.

SPISCLK (CLOCK) CONFIGURATION

In SPI Master mode, the SPI Clock Divisor register SPI_SCLKDIV is used to control the frequency of SPISCLK. The register stores a 16-bit parameter that supplies the initial value for the clock generator counter. The derived frequency for SPISCLK is:

[AHBCLK frequency] / (SPI_SCLKDIV+1) * 2

In SPI Master mode, the maximum supported SPISCLK frequency is [AHBCLK frequency] / 8.

SPI DMA OPERATIONS

DMA operations associated with the SPI interface are controlled by the SPI_DMA_CTRL register.

For DMA handshake in Master or Slave modes, the SPI_DMA_CTRL register bits must be set for the desired operation:

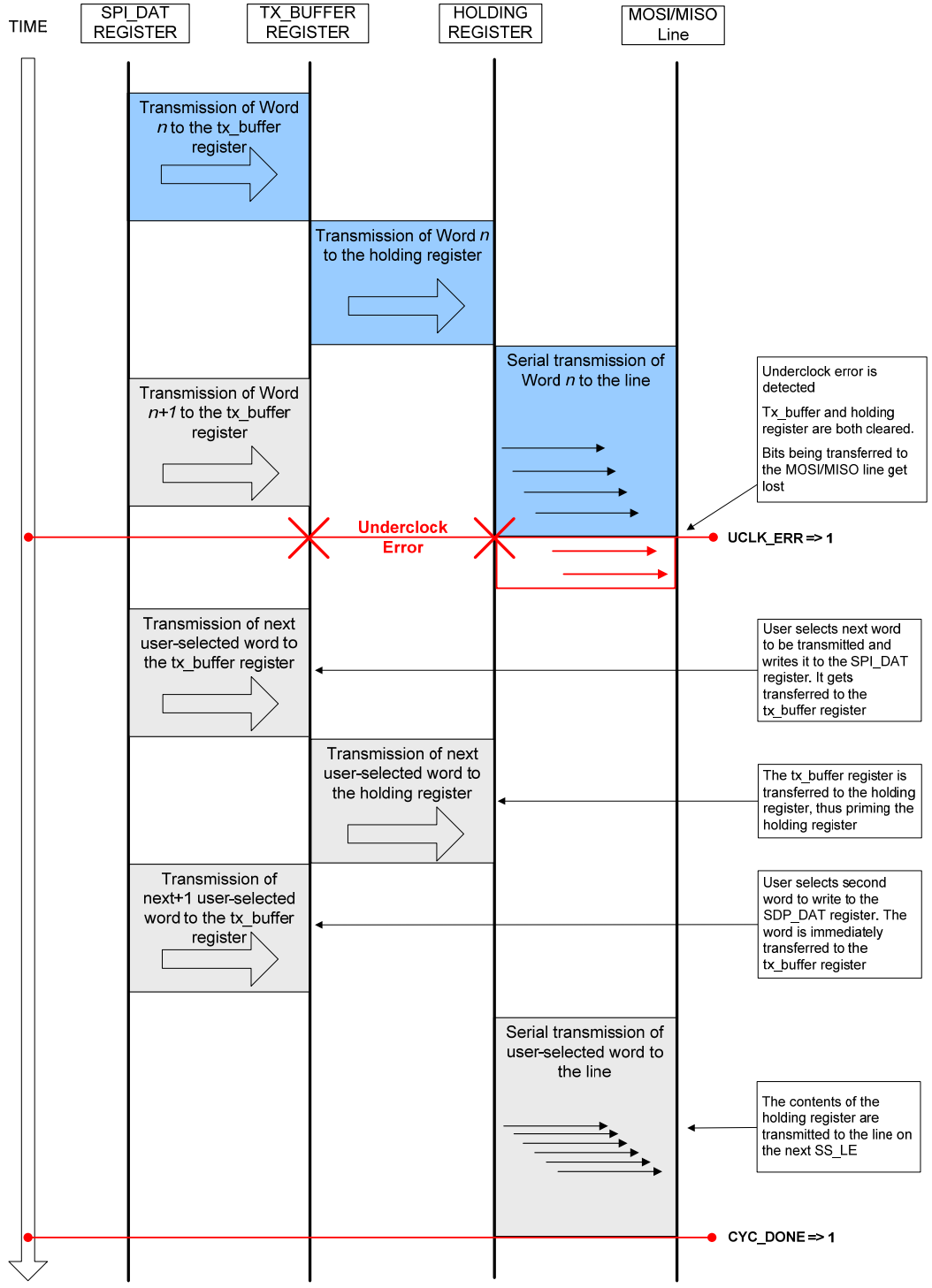
The WR_RQST_ENA bit enables the DMA Write request handshake, which indicates the transmit buffer is empty and ready for more data.

The RD_RQST_ENA bit enables the DMA Read request handshake, which indicates the receive buffer is full and needs to be read.

In the case where byte-packing is disabled (BP_EN=0), the DMA requests are based on the normal buffer status (empty, full) – ie. mimics the function of the CYC_DONE status.

When byte-packing is enabled (BP_EN=1), the DMA requests are based on the packed BP 64-bit buffer status (empty, full), OR on the determination that the byte-packed transfer is done – ie. it mimics the function of the BP_DONE status.

The CYC_DONE and BP_DONE registers are held within the SPI_STATUS register.



Underclock error - **UCLK_ERR = 0**

Following an Underclock Error, the user determines what data to send next. Both tx_buffer and the holding register must be primed before the next CYC_DONE. It is recommended that this is done during the UCLK_ERR interrupt.

Figure 42 SPI Transmission with Underclock Error, UCLK_ERR=0

SPI_STATUS
SPI STATUS REGISTER

Address = 0xF030_0028 | **Default value = 0x0000_0000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
31:15	Reserved		0	
14:12	SPI_CURRENT_STS	RO	0x0	Raw status of the SPI master state machine's "current_state" register: 000 = IDLE 001 = CSSETUP 010 = TRANSFER 011 = CSHOLD 100 = CSWAIT 101 = CKWAIT 110 = MTRANS 111 = CSBEGIN
11	Reserved		0x0	
10	RX_BUF_FULL	RO	0x0	Raw indicator of Rx incoming holding register status 0 = No data in holding register 1 = Holding register contains a valid data word
9	TX_BUF_FULL	RO	0x0	Raw indicator of Tx outgoing holding register status 0 = Holding register ready for new data word 1 = Tx Buffer is full
8	TX_UFL_ERR	R/W1C	0x0	Write Underflow Error indication: indicates that the outgoing data buffer did not get loaded with new data since the last transmission, and another transmission began. 0 = No Write Underflow since this bit was cleared 1 = Write Underflow detected This bit is cleared by writing a '1' to it. This bit is a sticky status bit, and will set upon meeting the condition regardless of the state of its corresponding interrupt enable TX_UFL_ERR_INT_ENA.
7	BP_DONE	R/W1C	0x0	Byte Packing Transfer Done: indicates a request for more packed data. This bit is set when the Byte Packing Holding Register is full (RX) and empty (TX), or when the current transfer of multiple byte-packed words is complete. 0 = Byte Packing Transfer is not complete 1 = Byte Packing Transfer is complete Only valid if byte packing is enabled (BP_EN = 1). This bit is cleared by writing a '1' to it. This bit is a sticky status bit, and will set upon meeting the condition regardless of the state of its corresponding interrupt enable BP_DONE_INT_ENA.
6	SS_TE	R/W1C	0x0	Slave Select Trailing Edge Detect: 0 = no $\overline{\text{SPISS}}$ de-assertion detected since this bit was cleared 1 = the $\overline{\text{SPISS}}$ de-assertion has been detected This bit is cleared by writing a '1' to it. This bit is a sticky status bit, and will set upon meeting the condition regardless of the state of its corresponding interrupt enable SS_TE_INT_ENA.
5	CYC_DONE	R/W1C	0x0	Cycle Done: this bit will set when the current transfer of word-length "WL" bits is complete. It indicates that "WL" bits were sent on the transmit port and "WL" bits were sampled on the receive port. This bit is cleared by writing a '1' to it. This bit is a sticky status bit, and will set upon meeting the condition regardless of the state of its corresponding interrupt enable CYC_DONE_INT_ENA.

DMA_PRI_DST REGISTER

DMA_PRI_DST_n DMA PRIMARY DESTINATION ADDRESS REGISTER																															
Address = 0xF040_0104 + (n * 0x40) (n = DMA Channel, valid from 0 to 31)																Default value = 0x0000_0000															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
31:0	DMA_PRI_DST_n	RW	0x0000_0000	DMA Destination Address (Channel 'n') Each register holds base address for the destination of the DMA Transfer for its respective channel. Unaligned Address bits for all SRC, DST, and LINK_ADDR registers are ignored. For example, when DMA_SRC_HSIZE=0x3, bits [2:0] are ignored. Not to be written while Linked List chaining is enabled (DMA_LINK_ENA=1).
Note that 'n' represents the DMA channel number, ie. 0, 1, 2 … 31.				

Table 148 DMA_PRI_DST_n Register

DMA_PRI_LEN REGISTER

DMA_PRI_LEN_n DMA PRIMARY TRANSFER LENGTH																															
Address = 0xF040_0108 + (n * 0x40) (n = DMA Channel, valid from 0 to 31)																Default value = 0x0000_0000															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
31:0	DMA_PRI_LEN_n	RW	0x0000_0000	DMA Transfer Length (Channel 'n') Contains the required number of transfer bytes for the channel. The value in this register is not affected by the transfer occurring. The number of bytes must be aligned to the programmed DMA_SRC_HSIZE ("unaligned" bits are ignored). Not to be written while Linked List chaining is enabled (DMA_LINK_ENA=1).
Note that 'n' represents the DMA channel number, ie. 0, 1, 2 … 31.				

Table 149 DMA_PRI_LEN_n Register

DMA_CTRL1_n
DMA CONTROL 1 REGISTER

Address = 0xF040_0128 + (n * 0x40)
(n = DMA Channel, valid from 0 to 31)

Default value = 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
19	DMA_LINK_ENA	RW	0x0	Enable Linked List DMA chaining. 0 = Disabled 1 = Enabled When enabled, the DMA_LINK_ADDR register is used to direct the DMA to the SRC, DST, LEN and LINK_ADDR registers for each linked transfer. The linked list terminates when it reaches a LINK_ADDR equal to 0x0000_0000. Note that, when Linked List DMA chaining is enabled, the data word size must be 32 bits, and Double-Buffering must be enabled (DMA_DWB_ENA=1).
18:17	DMA_AHB_MAX_BURST	RW	0x0	Maximum burst size/type used by the DMA master controller when Burst Data transfer is enabled (DMA_AHB_BURST_ENA=1). 00 = SINGLE 01 = INCR4 10 = INCR8 11 = Reserved The highest setting (10) is recommended in all cases.
16	DMA_AHB_BURST_ENA	RW	0x0	Enable AHB burst transfers. 0 = Disabled 1 = Enabled Burst Data transfers must be enabled for Low-Priority DMA channels. Burst Data transfers must be disabled for High-Priority DMA channels. Accordingly, DMA_AHB_BURST_ENA and DMA_CH_PRI_LOW_ENA must always be set to the same value.
15	DMA_SHA_XFER_ENA	RW	0x0	Enable SHA data transfer. 0 = Disabled 1 = Enabled Note that the SHA data transfer is via a dedicated FIFO. The SHA data transfer is enabled in addition to any 'normal' transfer to the DMA_PRI_DST_n address.
14	Reserved		0x0	
13:11	DMA_ENDIAN_SWAP_LEN	RW	0x0	Endian Byte Swap select. 000 = 16-bit word size 001 = 24-bit word size (pad LS Byte) 010 = 24-bit word size (pad MS Byte) 011 = 32-bit word size 100 = 64-bit word size 101 to 111 = Reserved for future implementations Only valid when DMA_ENDIAN_SWAP_ENA=1
10	Reserved		0x0	
9	DMA_ENDIAN_SWAP_ENA	RW	0x0	Endian Byte Swap enable 0 = Disabled 1 = Enabled
8	DMA_AHB_ARB_SET	RW	0x0	AHB Master priority select Controls which module has priority, in the event of conflicting demands for accessing the AHB bus. 0 = DMA controller has higher priority 1 = DSP Core has higher priority

EXAMPLE 3: LINKED LIST DMA OPERATION

This example describes a mechanism to transfer 3 packets of 32-bit data words from memory to the AIF2 module. The packets are defined in a list of descriptor registers, with the first packet descriptor at memory address 0x6007_0000.

The first packet comprises 256 words (1024 bytes) read from base address 0x6000_0000. The DMA descriptors for this part of the transfer are located at address 0x6007_0000.

The second packet comprises 256 words (1024 bytes) read from base address 0x6001_0000. The DMA descriptors for this part of the transfer are located at address 0x6007_0010.

The third packet comprises 512 words (2048 bytes) read from base address 0x6002_0000. The DMA descriptors for this part of the transfer are located at address 0x6007_0020.

The DMA descriptors for each of the packet transfers are contained in the memory configuration described in Table 162.

ADDRESS	VALUE	DESCRIPTION
Packet 1 definition		
0x6007_0000	0x6000_0000	Selects 0x6000_0000 as the source
0x6007_0004	0xF080_0020	Selects AIF_TX_DAT as the destination
0x6007_0008	0x400	Selects a transfer length of 256 bytes
0x6007_000C	0x6007_0010	Identifies the next packet descriptors address
Packet 2 definition		
0x6007_0010	0x6001_0000	Selects 0x6001_0000 as the source
0x6007_0014	0xF080_0020	Selects AIF_TX_DAT as the destination
0x6007_0018	0x400	Selects a transfer length of 256 bytes
0x6007_001C	0x6007_0020	Identifies the next packet descriptors address
Packet 3 definition		
0x6007_0020	0x6002_0000	Selects 0x6002_0000 as the source
0x6007_0024	0xF080_0020	Selects AIF_TX_DAT as the destination
0x6007_0028	0x1000	Selects a transfer length of 512 bytes
0x6007_002C	0x0000_0000	Terminates the Linked List chain

Table 162 DMA Example 3 - Linked List Memory configuration

The data words will be written to a fixed address in the AIF2 module; the AIF_TX_DAT register address for AIF2 is 0xF080_0020.

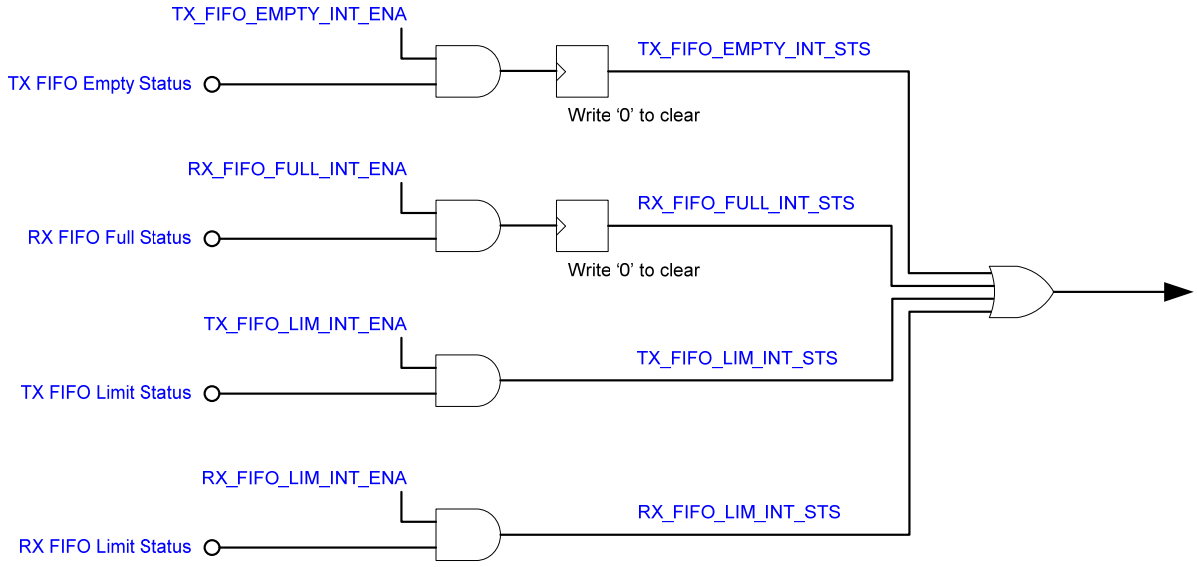
The AIF2 TX path requires the use of DMA channel 9. The handshake configuration must be Destination Data Phase ACK. (These requirements are described in Table 131.)

The required register settings for this transfer are noted in Table 163. Note that the default setting is assumed for any register fields that are not quoted here.

AIF INTERRUPTS

The AIF module can generate an interrupt when any of the conditions described in the AIF_INT_CTRL register occurs. The interrupt conditions provide status indications of the AIF TX and RX data buffers.

The AIF interrupt control registers are illustrated in Figure 56.



The interrupt control functions are replicated for each of the 3 AIF modules.

Figure 56 AIF Interrupts

AIF_RX_CH_ID – AIF RECEIVE CHANNEL ID REGISTER

This register indicates the channel number of the last audio sample read from the AIF_RX_DAT register.

AIF_RX_CH_ID
AIF RECEIVE CHANNEL ID REGISTER

Address = 0xF070_0004 (AIF 1)
Address = 0xF080_0004 (AIF 2)
Address = 0xF090_0004 (AIF 3)

Default value = 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
31:8	Reserved		0x00_0000	
7:0	RX_SLOT_ID	RO	0x00	Slot ID RX Slots are identified by an integer from 0 to [N-1]

Table 166 AIF_RX_CH_ID Register

AIF_RX_STS – AIF RECEIVE FIFO STATUS REGISTER

This register indicates the number of samples currently in the RX FIFO.

AIF_RX_STS
AIF RECEIVE FIFO STATUS REGISTER

Address = 0xF070_0008 (AIF 1)
Address = 0xF080_0008 (AIF 2)
Address = 0xF090_0008 (AIF 3)

Default value = 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
31:29	Reserved		0x0	
28	RX_EMPTY_STS	RO	0x0	RX FIFO Empty/Full indication. 0 = not Empty, 1= Empty.
27:0	RX_FIFO_SAMPLES	RO	0x000 0000	Number of samples in the RX FIFO

Table 167 AIF_RX_STS Register

AIF_TX_LIMIT – AIF TRANSMIT FIFO LOWER LIMIT REGISTER

This register holds the TX FIFO Lower Limit value.

When the number of samples in the TX FIFO is less than the Lower Limit value, the TX_FIFO_LIM_INT_STS interrupt will be asserted (if enabled by the TX_FIFO_LIM_INT_ENA bit in the AIF_INT_CTRL register).

The DMA handshake to the TX FIFO is also triggered by the same Lower Limit value (when enabled by TX_FIFO_LIM_DMA_ENA). The DMA operation will not execute while the number of samples in the buffer is greater than or equal to AIF_TX_LIMIT.

The TX FIFO buffer size is 64 samples. Therefore, to support the functionality described above, the AIF_TX_LIMIT is valid from 1 to 64.

AIF_TX_LIMIT
AIF TRANSMIT FIFO LOWER LIMIT REGISTER

Address = 0xF070_0030 (AIF 1) **Address = 0xF080_0030 (AIF 2)** **Address = 0xF090_0030 (AIF 3)**	**Default value = 0x0000_0000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BITS	FIELD NAME	S/W ACCESS	RESET VALUE	FIELD DESCRIPTION
31:0	AIF_TX_LIMIT	RW	0x00	TX FIFO Lower Limit value. When the number of samples in the TX FIFO is less than the Lower Limit value, the TX_FIFO_LIM_INT_STS interrupt and TX_FIFO_LIM_DMA_STS handshake will be asserted (if enabled by the respective bits in the AIF_INT_CTRL register). To support Interrupt or DMA Handshake functionality, AIF_TX_LIMIT is valid from 1 to 64.

Table 172 AIF_TX_LIMIT Register

AIF_DATA_CFG – AIF DATA CONFIGURATION REGISTER

The AIF data format comprises a sequence of data words corresponding to as many data slots as are configured. The number of slots, number of bits per slot, and audio sample size are configurable. Each audio sample may be Left or Right justified within the allocated time slots. Each audio sample is transmitted/received MSB-first. The first sample can be delayed relative to the leading edge of the Frame Sync (LRCLK) signal using the AIF_DATA_DLY control field.

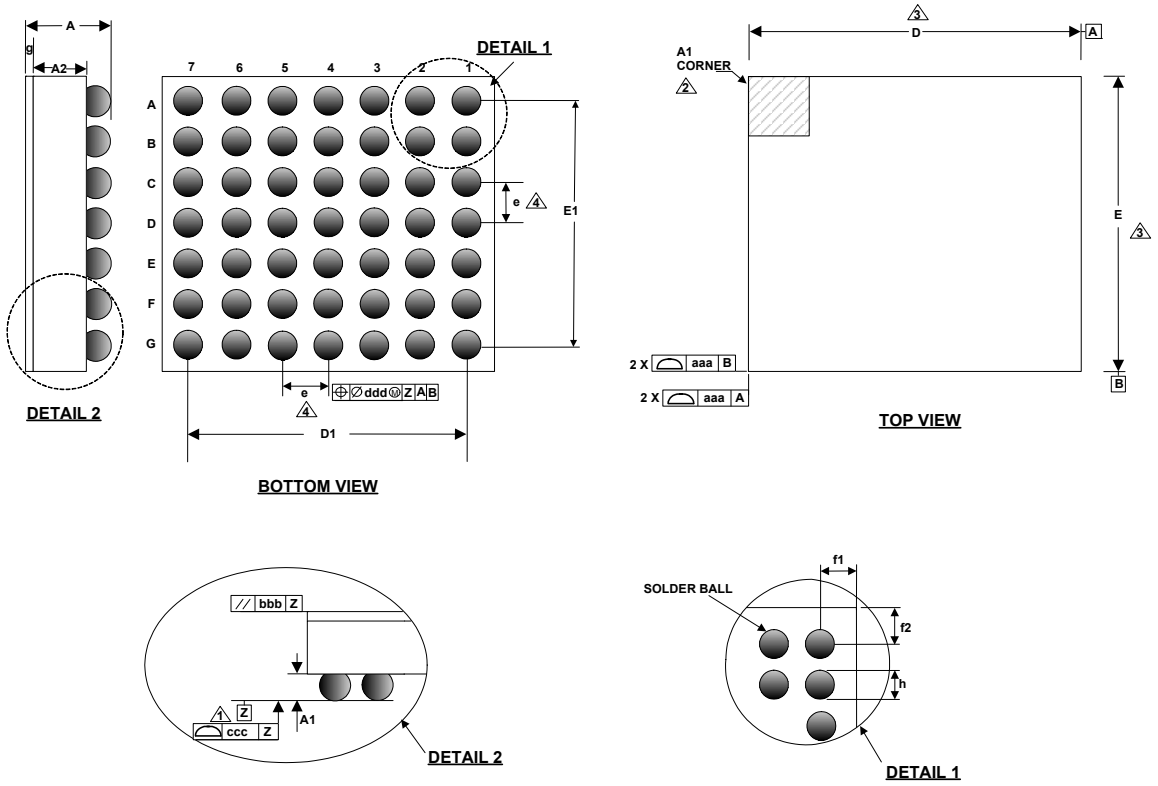
In Dual-Phase mode (AIF_DUAL_PHASE=1), the sequence comprises two phases, where each phase is independently configurable. This allows, for example, 'n' channels of 24-bit samples to be followed by 'm' channels of 16-bit samples in an efficient manner. Phase 1 is transmitted/received before Phase 2.

The timing and polarity of the Frame Sync (LRCLK) signal is configurable, as described in the AIF_CLK_CFG register (see Table 174).

The AIF data format is highly flexible, supporting I2S, Left-Justified, Right-Justified, DSP Mode-A, DSP Mode-B formats, and many others.

PACKAGE DIMENSIONS

B: 49 BALL W-CSP PACKAGE 3.055 x 3.055 x 0.636mm BODY, 0.40 mm BALL PITCH **DM114.B**



Symbols	Dimensions (mm)			
	MIN	NOM	MAX	NOTE
A	0.592	0.636	0.681	
A1	0.175	0.190	0.205	
A2	0.381	0.406	0.432	
D	3.000	3.055	3.080	
D1		2.400 BSC		
E	3.000	3.055	3.080	
E1		2.400 BSC		
e		0.400 BSC		4
f1	0.300	0.328		Bump centre to die edge
f2	0.300	0.328		Bump centre to die edge
h	0.216	0.270	0.324	
g	0.036	0.040	0.044	
aaa		0.10		
bbb		0.10		
ccc		0.03		
ddd		0.015		

NOTES:
1. PRIMARY DATUM -Z- AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
2. A1 CORNER IS IDENTIFIED BY INK/LASER MARK ON TOP PACKAGE.
3. BILATERAL TOLERANCE ZONE IS APPLIED TO EACH SIDE OF THE PACKAGE BODY.
4. 'e' REPRESENTS THE BASIC SOLDER BALL GRID PITCH.
5. THIS DRAWING IS SUBJECT TO CHANGE WITHOUT NOTICE.
6. FOLLOWS JEDEC DESIGN GUIDE MO-211-C.