Generic AXI DMA

joe68.chou@msa.hinet.net

Revision History

Rev	Date	Description
0.0	2017/05/27	First release

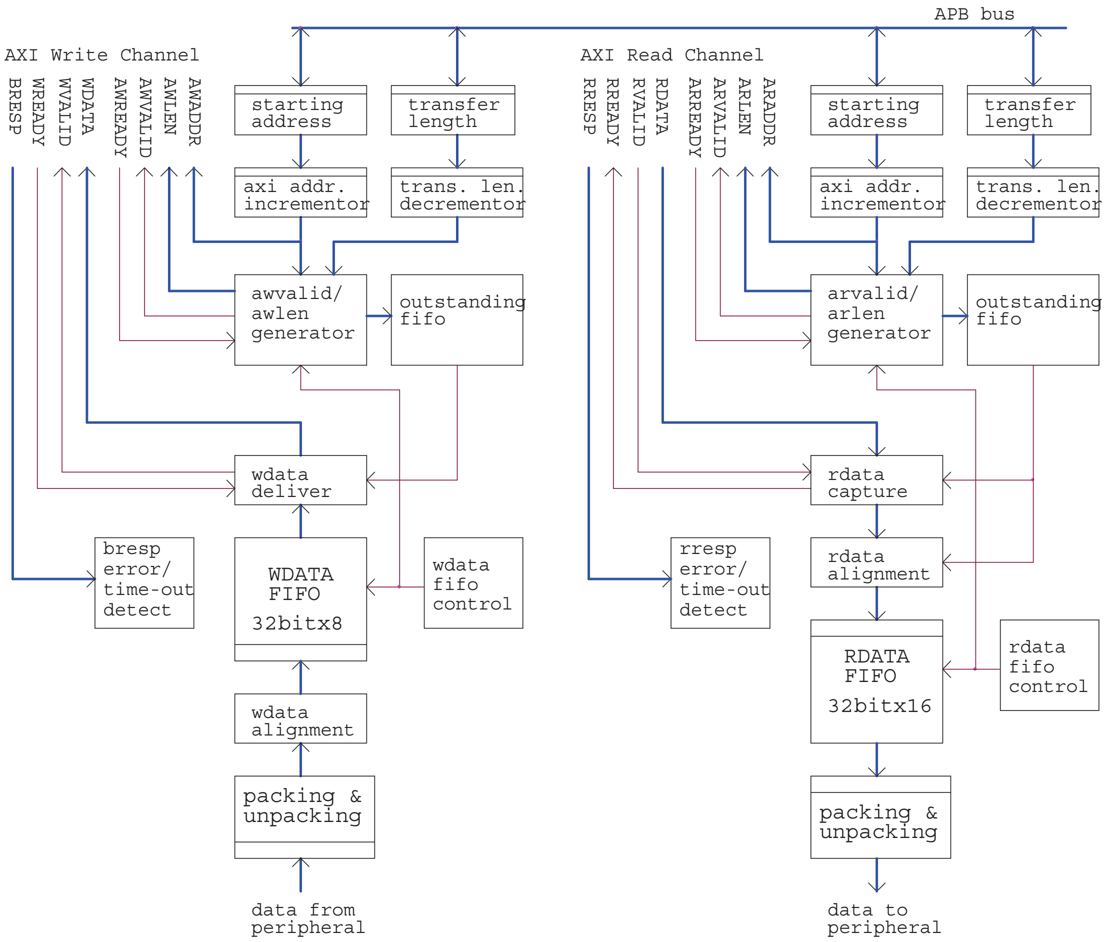
Description

An AXI DMA fits the peripheral (such as I2S, UART, SPI….) it needs DMA capability in SoC; the DMA will move data between the peripheral and system memory for efficient data transfer, also saving CPU computing power.

Overview

- AXI compliant; 32-bit data width.
- Bi-directional; independent read/write channel, and also read/write channel can be executed in parallel.
- Byte addressable; DMA starting address and transfer length could not have to be a multiple of 4.
- Maximum burst length is 8 words, and burst access won't cross 8-word boundary.
- Outstanding transaction support.
- Response error/time-out support.
- APB interface for register programing
- Peripheral interface protocol: AXI stream

Block Diagram



Hardware Description

- HW will break DMA transfer length into several AXI burst accesses. The maximum burst length is 8 words, and also burst access address will not cross 8-word boundary, and this is good for DDR efficiency if target slave is DDR. Up to DMA address and remaining DMA length situation the burst length could be smaller in order not to cross 8-word boundary while DMA starting and ending addresses are not located at 8-word boundary.
- HW will look ahead on internal data FIFO situation for issuing access to AXI. For read it will make sure FIFO has enough room for receiving next bursting RDATA before issuing read access; and for write it will make sure FIFO has bursting WDATA ready before issuing write access. This prevents data pending or waiting on AXI bus, which may cause AXI bus low efficiency.
- For supporting of outstanding access a FIFO is used to keep issued AxADDR and AxLEN information to handle RDATA receiving or WDATA delivering.
-

IO Interface Definition

General		
IO	name	description
input	clk	Clock input to DMA. The design is clocking by this clock only, and also only rising edge of the clock used.
input	rstn	Reset input to DMA, low active. The DMA used synchronous reset, the reset de-assertion needs to be synchronized with input clock.
output	dma_itr	DMA interrupt output; active high, level triggered. Registered output, glitch free. SW needs to clear the interrupt source within the design during ISR.
APB interface		
input	pselect	APB pselect input.
input	penable	APB penable input.
input	pwrite	APB pwrite input.
input	paddr[7:0]	APB address input
input	pwdata[31:0]	APB write data input.
output	prdata[31:0]	APB read data return.
AXI interface		
output	arvalid	AXI arvalid output
input	arready	AXI arready input
output	araddr[31:0]	AXI araddr input
output	arlen[2:0]	AXI arlen output. The value could be from 3’b111 to 3’b000.
output	arsize[2:0]	AXI arsize output. Fixed at 3’b010(32-bit)
output	arburst[1:0]	AXI arburst output. Fixed at 2’b01(INCR)
input	rvalid	AXI rvalid input.
output	rready	AXI rready output
input	rdata[31:0]	AXI rdata input.
input	rresp[1:0]	AXI rresp input
input	rlast	AXI rlast input
output	awvalid	AXI awvalid output
input	awready	AXI awready input
output	awaddr[31:0]	AXI awaddr input
output	awlen[2:0]	AXI awlen output.

		The value could be from 3’b111 to 3’b000.
output	awsize[2:0]	AXI awsize output. Fixed at 3’b010(32-bit)
output	awburst[1:0]	AXI awburst output. Fixed at 2’b01(INCR)
output	wvalid	AXI wvalid output
input	wready	AXI wready input
output	wdata[31:0]	AXI wdata output
output	wstrb[3:0]	AXI wstrb output
output	wlast	AXI wlast output
input	bvalid	AXI bvalid input
output	bready	AXI bready output
input	bresp[1:0]	AXI bresp input
Peripheral/RDMA interface (AXI stream protocol)		
output	rdma_tvalid	To indicate data output from RDMA is valid to peripheral.
input	rdma_tready	Peripheral is ready to receive the tdata.
output	rdma_tdata[31:0]	RDMA data output to peripheral.
output	rdma_tstrb[3:0]	tdata strobe indication; only combinations below. tstrb[3:0]=4’b0001 indicate tdata[7:0] is valid. tstrb[3:0]=4’b0011 indicate tdata[15:0] is valid. tstrb[3:0]=4’b0111: indicate tdata[23:0] is valid. tstrb[3:0]=4’b1111: indicate tdata[31:0] is valid.
output	rdma_tlast	To indicate the last data to peripheral of a DMA.
Peripheral/WDMA interface (AXI stream protocol)		
input	wdma_tvalid	To indicate data input to WDMA is valid from peripheral.
output	wdma_tready	WDMA is ready to receive the tdata.
input	wdma_tdata[31:0]	WDMA data input from peripheral.
input	wdma_tstrb[3:0]	tdata strobe indication; only combinations below allowed.. tstrb[3:0]=4’b0001 indicate tdata[7:0] is valid. tstrb[3:0]=4’b0011 indicate tdata[15:0] is valid. tstrb[3:0]=4’b0111: indicate tdata[23:0] is valid. tstrb[3:0]=4’b1111: indicate tdata[31:0] is valid.

Register Definition

Read DMA starting register / RDMA_SADR_REG	0x00

bit#	name	type	reset	Description
31:0	Read DMA Starting address	R/W	none	Read DMA starting address. To specify the source starting address for DMA transfer.

Read DMA length register / RDMA_LEN_REG				0x04
bit#	name	type	reset	Description
19:0	Read DMA Transfer length	R/W	0x0	Read DMA transfer length, unit: byte. A non-zero value written to this register will enable the read DMA transfer. This register will be cleared to zero by hardware while RDMA is started. Note: DMA starting address register and DMA transfer length register are double-buffered, and user may prepare next DMA starting address and transfer length while previous DMA start flag/interrupt has been set.
31:20	Reserved	R	0x0	Always zero while reading

Read DMA control register / RDMA_CTL_REG				0x08
bit#	name	type	reset	Description
0	Read DMA start interrupt enable	R/W	0x0	0: mask 1: enable interrupt
1	Read DMA finish interrupt enable	R/W	0x0	0: mask 1: enable interrupt
2	Read DMA RRESP error interrupt enable	R/W	0x0	0: mask 1: enable interrupt
3	Read DMA RRESP time out interrupt enable	R/W	0x0	0: mask 1: enable interrupt

4	Read DMA RVALID error interrupt enable	R/W	0x0	0: mask 1: enable interrupt
7:5	Reserved	R	0x0	Always zero while reading
8	Read DMA pause	R/W	0x0	0: play 1: pause; to disable DMA request to AXI bus.
9	Read DMA flush	R/W	0x0	To terminate on-going DMA operation. Set to 1 to terminate, and it will be auto-cleared by hardware while on-going AXI transaction has been finished.
31:10	Reserved	R	0x0	Always zero while reading

Write DMA starting register / WDMA_SADR_REG				0x10
bit#	name	type	reset	Description
31:0	Write DMA Starting address	R/W	none	Write DMA starting address. To specify the source starting address for write DMA transfer.

Write DMA length register / WDMA_LEN_REG				0x14
bit#	name	type	reset	Description
19:0	Write DMA Transfer length	R/W	0x0	Write DMA transfer length, unit: byte. A non-zero value written to this register will enable the write DMA transfer. This register will be cleared to zero by hardware while WDMA is started. Note: DMA starting address register and DMA transfer length register are double-buffered, and user may prepare next DMA starting address and transfer length while previous DMA start flag/interrupt has been set.
31:20	Reserved	R	0x0	Always zero while reading

Write DMA control register / WDMA_CTL_REG				0x18
bit#	name	type	reset	Description
0	Write DMA start	R/W	0x0	0: mask 1: enable interrupt

	interrupt enable			
1	Write DMA finish interrupt enable	R/W	0x0	0: mask 1: enable interrupt
2	Write DMA BRESP error interrupt enable	R/W	0x0	0: mask 1: enable interrupt
3	Write DMA BRESP time out interrupt enable	R/W	0x0	0: mask 1: enable interrupt
4	Write DMA BVALID error interrupt enable	R/W	0x0	0: mask 1: enable interrupt
7:5	Reserved	R	0x0	Always zero while reading
8	Write DMA pause	R/W	0x0	0: play 1: pause; to disable DMA request to AXI bus.
9	Write DMA flush	R/W	0x0	To terminate on-going DMA operation. Set to 1 to terminate, and it will be auto-cleared by hardware while on-going AXI transaction has been finished.
31:10	Reserved	R	0x0	Always zero while reading

DMA Status register / DMA_STT_REG				0x20
bit#	name	type	reset	Description
0	Read DMA start interrupt flag	R/W	0x0	Read DMA start interrupt flag. An interrupt is generated while the DMA is started. Software may clear it by writing 1 to this bit. 0: no interrupt 1: interrupt Writing 1 to clear this bit, 0 no effect.

1	Read DMA finish interrupt flag	R/W	0x0	Read DMA finish interrupt flag. An interrupt is generated while DMA is finished. Software may clear it by writing 1 to this bit. 0: no interrupt 1: interrupt Writing 1 to clear this bit, 0 no effect.
2	RRESP error interrupt flag	R/W	0x0	RRESP error received. Software may clear it by writing 1 to this bit. 0: no interrupt 1: interrupt Writing 1 to clear this bit, 0 no effect
3	RRESP time out interrupt flag	R/W	0x0	RRESP didn't get back to RDMA within specified time. Software may clear it by writing 1 to this bit. 0: no interrupt 1: interrupt Writing 1 to clear this bit, 0 no effect
4	RVALID error interrupt flag	R/W	0x0	Unexpected RVALID/RDATA received. Software may clear it by writing 1 to this bit. 0: no interrupt 1: interrupt Writing 1 to clear this bit, 0 no effect
7:5	Reserved	R	0x0	Always zero while reading
9:8	RRESP error message	R	0x0	00: OKAY 01: EXOKAY 10: SLVERR 11: DECERR See AXI spec for details.
10	Read DMA operating	R	0x0	To indicate Read DMA is operating 0: no 1: yes
15:11	Reserved	R	0x0	Always zero while reading
16	Write DMA start interrupt flag	R/W	0x0	Write DMA start interrupt flag. An interrupt is generated while the DMA is started. Software may clear it by writing 1 to this bit. 0: no interrupt 1: interrupt Writing 1 to clear this bit, 0 no effect.

17	Write DMA finish interrupt flag	R/W	0x0	Write DMA finish interrupt flag. An interrupt is generated while DMA is finished. Software may clear it by writing 1 to this bit. 0: no interrupt 1: interrupt Writing 1 to clear this bit, 0 no effect.
18	BRESP error interrupt flag	R/W	0x0	BRESP error received. Software may clear it by writing 1 to this bit. 0: no interrupt 1: interrupt Writing 1 to clear this bit, 0 no effect
19	BRESP time out interrupt flag	R/W	0x0	BRESP didn't get back to WDMA within specified time. Software may clear it by writing 1 to this bit. 0: no interrupt 1: interrupt Writing 1 to clear this bit, 0 no effect
20	BVALID error interrupt flag	R/W	0x0	Unexpected BVALID/BRESP received. Software may clear it by writing 1 to this bit. 0: no interrupt 1: interrupt Writing 1 to clear this bit, 0 no effect
23:21	Reserved	R	0x0	Always zero while reading
25:24	BRESP error message			00: OKAY 01: EXOKAY 10: SLVERR 11: DECERR See AXI spec for details.
26	Write DMA operating	R	0x0	To indicate write DMA is operating. 0: no 1: yes
31:27	Reserved	R	0x0	Always zero while reading

Response Time-out period register / RSP_TOUT_REG				0x24
bit#	name	type	reset	Description
15:0	Response time out period	R/W	0xFFFF	To specify AXI BRESP or RRESP time out period. If BRESP or RRESP didn't get back to DMA within specified time here, then a time-out

				interrupt will be generated. This is a down counter, will be reloaded by preset value while access issued or BRESP/RRESP received but another outstanding access still on-going. The counter is clocking by DMA input clock, an interrupt is generated while count reaches zero value. Value 0xFFFF is defined as a time-out counter disable state.
31:16	Reserved	R	0x0	Always zero while reading

Application Note

- The DMA operating is simple, SW user firstly may enable the interrupt wanted and set up DMA starting address, and then DMA transfer length. A non-zero value written to transfer length register will enable the DMA operation. After DMA is enabled, SW may wait for the interrupt for further operation.
- There are two interrupts from DMA operation; DMA start, and DMA finish. Note; DMA starting address and transfer length registers are double-buffered, while DMA is started, the software programmed DMA starting address and transfer length registers will be copied to internal, and also transfer length register will be cleared to zero, at this moment DMA starting address and transfer length registers are available for next DMA programming. While previous DMA is finished, HW will check DMA transfer length right away, if the content of transfer length register is not zero, then next DMA is started. Such mechanism will form a gapless transfer between two DMAs.
- SW user may select conventional way, wait for previous DMA finish, and then program DMA starting address and transfer length register to start next DMA.
- DMA pause function is to temporally disable DMA access to AXI.
- DMA flush function is to terminate on-going DMA as well as next DMA if it is already prepared in DMA transfer length register. The termination is executed after all outstanding transactions being finish to prevent AXI bus hang-up.