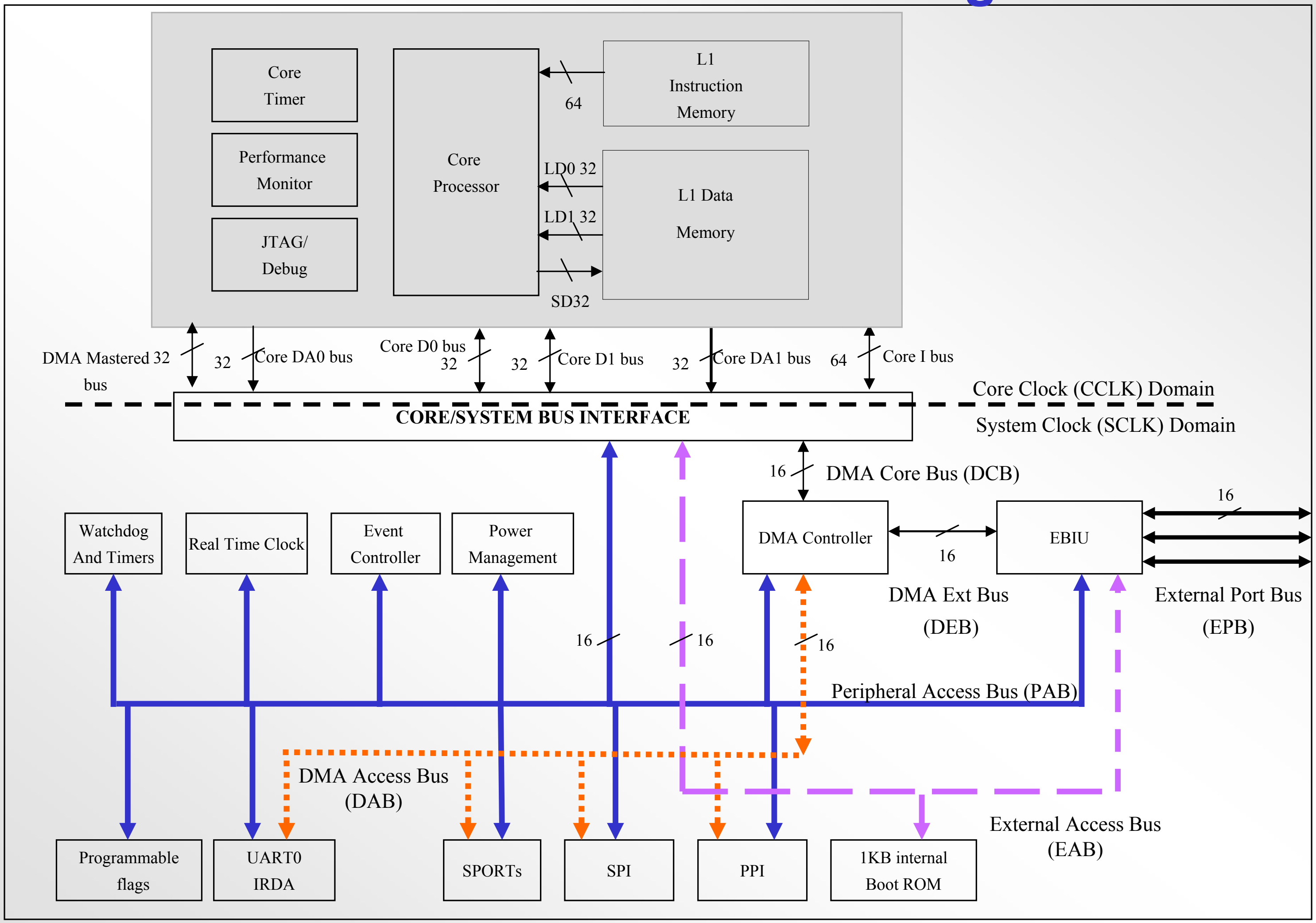
Section 11

Direct Memory Access (DMA)

ADSP-BF533 Block Diagram



KAZTEK ENGINEERING

ANALOG DEVICES

BF533 DMA Overview

- The ADSP-BF533 DMA controller allows data transfer operations without processor intervention
  - Core sets up registers or descriptors
  - Core responds to interrupts when data is available
- Types of data transfers

Internal or External Memory	↔	Internal or External Memory
Internal or External Memory	↔	Serial Peripheral Interface (SPI)
Internal or External Memory	↔	Serial Port
Internal or External Memory	↔	UART Port
Internal or External Memory	↔	Parallel Port Interface (PPI)

Overview (cont.)

- **The ADSP-BF533 system includes 6 DMA-capable peripherals, including the Memory DMA controller (MemDMA) with 12 DMA channels and bus masters that support these devices:**

SPORT0 RCV DMA Channel	**UART RCV Channel**
SPORT1 RCV DMA Channel	**UART XMT Channel**
SPORT0 XMT DMA Channel	**PPI DMA Channel**
SPORT1 XMT DMA Channel	**4 Memory DMA Channels**
SPI DMA Channel	**- Equates to 2 DMA Streams**

KAZTEK ENGINEERING

ANALOG DEVICES

BF533 DMA Buses

- The DMA Access Bus (DAB) provides a means for DMA channels to be accessed by the peripherals.
- The DMA External Bus (DEB) provides a means for DMA channels to gain access to off-chip memory.
  - The core processor has priority over the DEB on the External Port Bus (EPB) for off-chip memory.
- The DMA Core Bus (DCB) provides a means for DMA channels to gain access to on-chip memory.
  - The DCB has priority over the core processor on arbitration into L1 memory configured as SRAM.

BF533 DMA Priority

The ADSP-BF533 processor uses the following priority arbitration policy on the DAB.

DMA Channel	Default Peripheral Mapping	Comments
0 – highest	PPI	Re-assignable
1	SPORT0 RX	Re-assignable
2	SPORT0 TX	Re-assignable
3	SPORT1 RX	Re-assignable
4	SPORT1 TX	Re-assignable
5	SPI	Re-assignable
6	UART RX	Re-assignable
7	UART TX	Re-assignable
8	Memory DMA Stream 0 TX (destination)	Fixed
9	Memory DMA Stream 0 RX (source)	Fixed
10	Memory DMA Stream 1 TX (destination)	Fixed
11 - lowest	Memory DMA Stream 1 RX (source)	Fixed

Peripheral Map Register

DMAx_PERIPHERAL_MAP / MDMA_yy_ PERIPHERAL_MAP

(x = 0,1,2,…, or 7 ;   yy = 8, 9, 10, or 11)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Reset = 0x0000

PMAP[3:0] (Peripheral Mapped to this Channel)

0000 – PPI
0001 – SPORT0 RX
0010 – SPORT0 TX
0011 – SPORT1 RX
0100 – SPORT1 TX
0101 – SPI
0110 – UART RX
0111 – UART TX

CTYPE (DMA Channel Type) – RO

0 – Peripheral DMA
1 – Memory DMA

- **The Peripheral Map Register allows the user to map a peripheral to a specific channel thus programming the priority of each peripheral.**

KAZTEK ENGINEERING

ANALOG DEVICES

DMA Initialization

- To initiate a DMA transfer, certain parameters need to be defined before the DMA engine can start a DMA sequence. These parameters are:
  - Configuration
    - describes certain characteristics of the DMA transfer such as data size, transfer direction, etc..
  - Start Address
    - Specifies the address where the DMA transfer will start from.
  - Count
    - Specifies the number of elements the DMA Engine will transfer.
  - Modify
    - Specifies the address increment after every element transfer

DMA Schemes

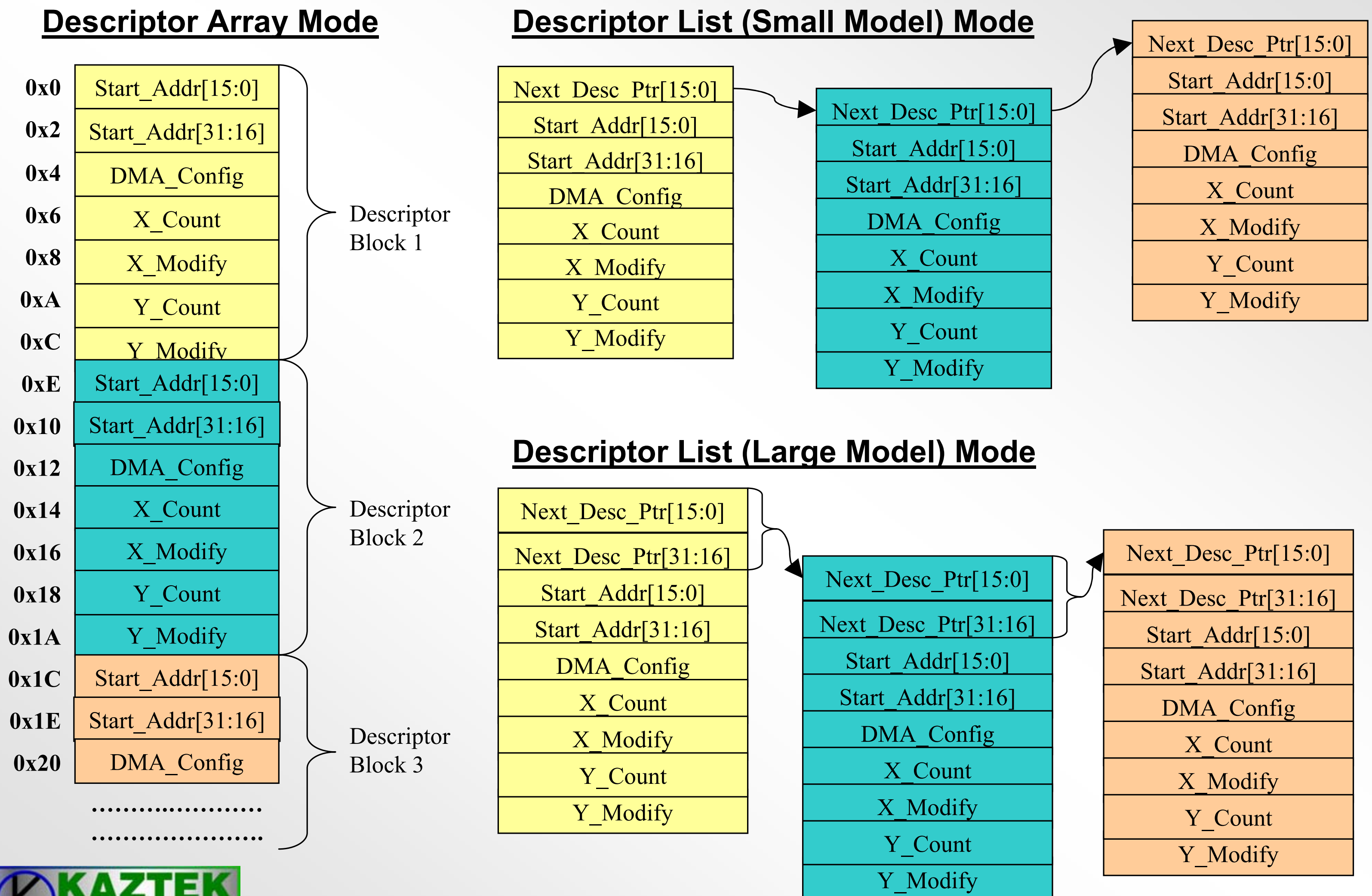
Two Types of DMA transfers available on the ADSP-BF533/BF561

- **Descriptor-based DMA transfers**
  - **Requires a set of parameters stored within memory to initiate a DMA sequence. These parameters are transferred to DMA control registers upon a start of a DMA transfer.**
  - **Supports chaining of multiple DMA transfers.**
- **Register-based DMA transfers**
  - **Allows the user to program the DMA control registers directly to define and initiate a DMA sequence.**
  - **Upon DMA completion, depending on certain bits with the Configuration Register:**
    - **Control registers are automatically updated with their original setup values (Autobuffer Mode).**
    - **Or the DMA Channel gracefully shuts off (Stop Mode).**

KAZTEK ENGINEERING

ANALOG DEVICES

Descriptor Blocks



Transfer Modes

- The Transfer Mode is controlled by 3 bits called the FLOW[2:0] bits within the DMA Configuration Register.
  - Stop Mode (FLOW = 0x0).
    - When the current DMA transfer completes, the DMA channel stops automatically, after signaling an interrupt if enabled.
  - Autobuffer Mode (FLOW = 0x1).
    - DMA is performed in a continuous circular-buffer fashion based on user-programmed DMAx MMR settings. On completion of the DMA transfer, the Parameter registers are reloaded into the Current registers, and DMA resumes immediately with zero overhead. Autobuffer mode is stopped by a user write of 0 to the DMA enable bit in the DMAx_DMA_Config Register.
  - Descriptor Array Mode (FLOW = 0x4).
    - In this mode, the Descriptor Block does not include the NEXT_DESC_PTR parameter. Descriptor Blocks are placed one after the other within memory like an array.
  - Descriptor List (Small Model) Mode (FLOW = 0x6).
    - In this mode, the Descriptor Block does not include the upper 16 bits of the NEXT_DESC_PTR parameter. The upper 16 bits are taken from the upper 16 bits of the NEXT_DESC_PTR register, thus confining all descriptors to a specific 64K page in memory.
  - Descriptor List (Large Model) Mode (FLOW = 0x7).
    - In this mode, Descriptor Block includes all 32 bits of the NEXT_DESC_PTR parameter, thus allowing maximum flexibility in locating descriptors in memory.

Descriptor Block Structures

- **Depending on the Descriptor Mode used, the following lists the order of the Descriptor Block Parameters stored within memory:**

Descriptor Offset	Descriptor Array Mode (FLOW = 0x4)	Small Descriptor List Mode (FLOW = 0x6)	Large Descriptor List Mode (FLOW = 0x7)
0x0	START_ADDR[15:0]	NEXT_DESC_PTR[15:0]	NEXT_DESC_PTR[15:0]
0x2	START_ADDR[31:16]	START_ADDR[15:0]	NEXT_DESC_PTR[31:16]
0x4	DMA_CONFIG	START_ADDR[31:16]	START_ADDR[15:0]
0x6	X_COUNT	DMA_CONFIG	START_ADDR[31:16]
0x8	X_MODIFY	X_COUNT	DMA_CONFIG
0xA	Y_COUNT	X_MODIFY	X_COUNT
0xC	Y_MODIFY	Y_COUNT	X_MODIFY
0xE		Y_MODIFY	Y_COUNT
0x10			Y_MODIFY

NOTE: Not all of the Parameters need to be initialized within the Descriptor Block depending on the NDSIZE value within the DMA Configuration Register. The NDSIZE value is the number of Parameters that the DMA engine will fetch for the next Descriptor Block

DMA Register Setup

To start DMA operation, some or all of the DMA Parameter Registers must first be initialized depending on the 'Next Descriptor Size'(NDSIZE) and 'FLOW' bits in the DMA Configuration Register. After Initialization, DMA operation begins by writing a 1 to the DMA Enable bit in the DMA Configuration Register.

1) **FLOW = 0x0 (Stop Mode), NDSIZE = 0x0:**

   Initialize all of the following:

   - START_ADDR
   - X_COUNT
   - X_MODIFY
   - Y_COUNT (if 2D DMA)
   - Y_MODIFY (if 2D DMA)
   - DMA_CONFIG

2) **FLOW = 0x1 (Autobuffer Mode), NDSIZE = 0x0:**

   Initialize all of the following:

   - START_ADDR
   - X_COUNT
   - X_MODIFY
   - Y_COUNT (if 2D DMA)
   - Y_MODIFY (if 2D DMA)
   - DMA_CONFIG

3) **FLOW = 0x4 (Descriptor Array Mode), NDSIZE = 0x0 – 0x7:**

   Initialize at least:

   - CURR_DESC_PTR[31:16]
   - CURR_DESC_PTR[15:0]

4) **FLOW = 0x6 (Small Descriptor List Mode), NDSIZE = 0x0 – 0x8:**

   Initialize at least:

   - NEXT_DESC_PTR[31:16]
   - NEXT_DESC_PTR[15:0]

5) **FLOW = 0x7 (Large Descriptor List Mode), NDSIZE = 0x0 – 0x9:**

   Initialize at least:

   - NEXT_DESC_PTR[31:16]
   - NEXT_DESC_PTR[15:0]

How to Stop DMA Transfers

- **FLOW = 0x0 (Stop Mode):**
  - **DMA stops automatically after the DMA transfer is complete.**

- **FLOW = 0x1 (Autobuffer Mode):**
  - **Write a 0 to the DMA Enable bit in the DMA Configuration Register. A write of 0x0 to the entire register will always terminate DMA gracefully (without DMA Abort).**

- **FLOW = 0x4, 0x6, 0x7 (Array / List Mode):**
  - **Set the final DMA_CONFIG Register with FLOW = 0x0 setting to gracefully stop the DMA channel. If the DMA_CONFIG Parameter is not included within the Descriptor Block, use the FLOW = 0x1 method above to end the DMA.**

Memory DMA (MemDMA)

- Allows memory-to-memory DMA transfers between the various ADSP-BF533 memory spaces
- A single MemDMA transfer requires a pair of DMA channels:
  - One to specify the Source block of memory
  - One to specify the Destination block of memory
- ADSP-BF533 consists of four MemDMA channels which allows setup for 2 memory-to-memory DMA transfers at the same time
  - Two Source DMA Channel – used to read from memory
  - Two Destination DMA Channel – used to write to memory
- Both the Source and Destination DMA Channels share a 8-entry, 16-bit FIFO (32-bit FIFOs on the BF561)
  - Source DMA Channel fills the FIFO
  - Destination DMA Channel empties the FIFO

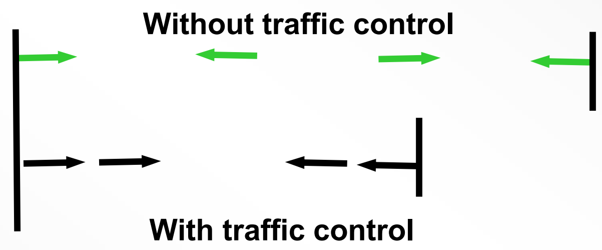
Memory DMA (MemDMA)

- Each DMA transfer sequence requires two sets of Descriptor Blocks within memory
  - One for the Source DMA Channel
  - One for the Destination DMA Channel
  - Both sets of Descriptor Blocks must be configured to have the same transfer count and data size but they can have different modify values.
  - The DMA Configuration Register for the source channel must be written before the DMA Configuration Register for the destination channel. When the destination DMA Configuration Register is written, MemDMA operations starts after a latency of 3 SCLK cycles
- It is preferable to activate interrupts on only one channel
  - Eliminates ambiguity when trying to identify the channel (either source or destination) that requested the interrupt

Prioritization and Traffic Control

- Traffic can be independently controlled for each of the three buses (DAB, DCB, and DEB) with simple counters
  - alternation of transfers between MDMA streams can also be controlled
- Using the traffic control features, the DMA system preferentially grants data transfers on the DAB or memory buses (DCB and DEB), which are going in the same read/write direction as the previous transfer, until either the traffic control counter times out, or until traffic stops or changes direction on its own.
- When a count field in TC_CNT expires, it is automatically reloaded with the appropriate value programmed in TC_PER (ie period value).
- When a DAB, DEB, or DCB counter decrements from 1 to 0, the opposite-direction DAB, DCB, or DEB access is preferred,
  - This may result in a direction change.
- When the MDMA counter decrements from 1 to 0, the next available MDMA stream is selected.
  - If the MDMA period is set to 0, then MDMA is scheduled by fixed priority.
  - If the MDMA period is set between 1 <= p <= 31, the two MDMA streams are granted bus access in alternate bursts of up to p data transfers.

Traffic Control (cont'd)



Arrows represent transfers in and out of SDRAM

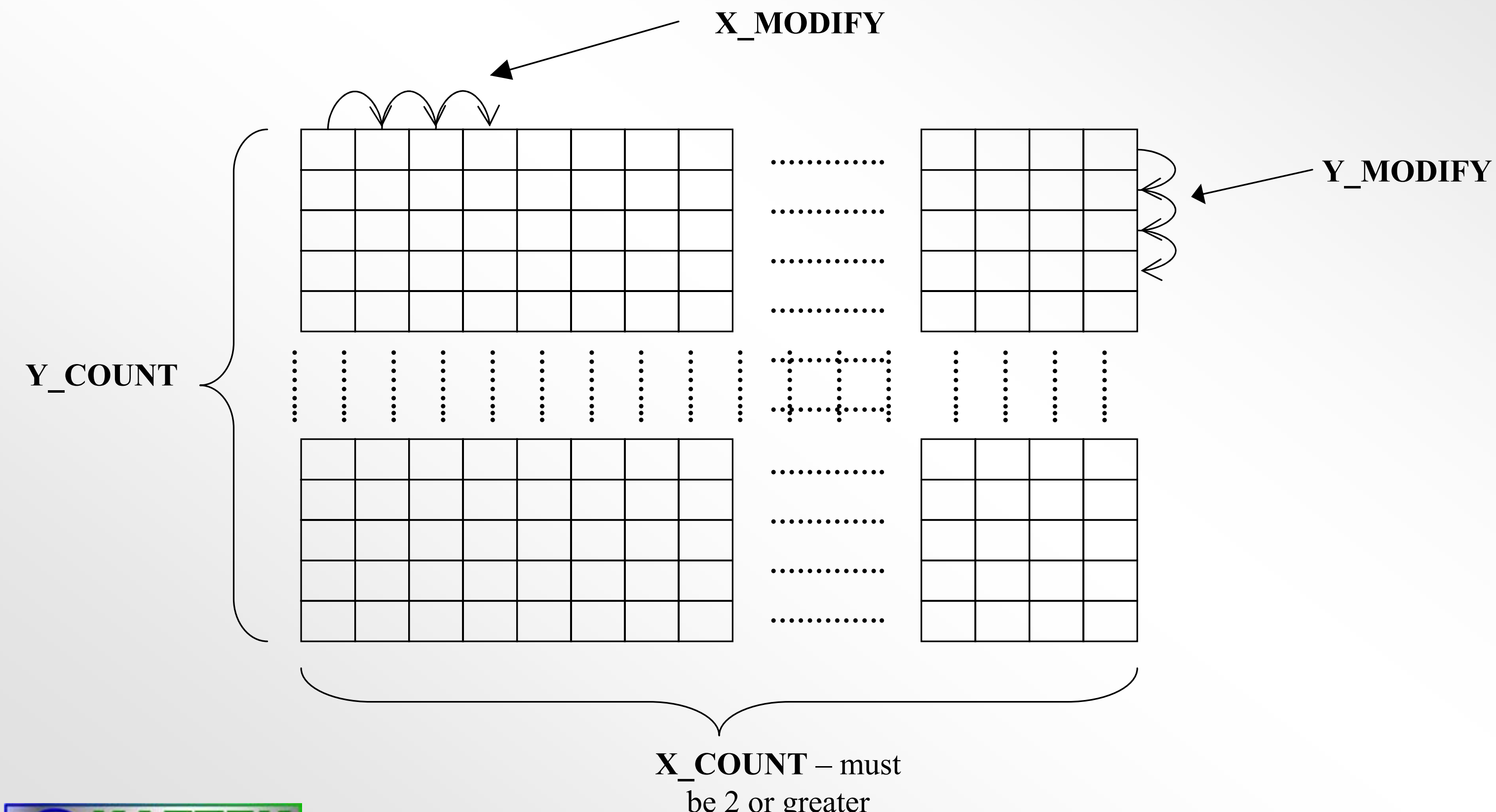
Important Register: Allows the definition of transfer sizes in a given direction on DMA busses

Max values usually yield best performance but it is application dependent

2 Reads and 2 writes are more efficient with traffic control

Two-Dimensional DMA (2D DMA)

Supports arbitrary row and column sizes up to 64K x 64K elements. X_Count = row size and Y_Count = column size.



Two-Dimensional DMA (2D DMA)

- X_Modify
  - is the byte-address increment applied after each transfer that decrements Curr_X_Count.
  - is not applied when the inner loop (row) count is ended by decrementing Curr_X_Count from 1 to 0.
- Y_Modify
  - is the byte-address increment applied after each decrement of Curr_Y_Count.
  - is not applied to the last element in the array on which the outer loop (column) count, Curr_Y_Count, also expires by decrementing from 1 to 0.
- After the last transfer completes,
  - Curr_Y_Count = 1
  - Curr_X_Count = 0
  - Curr_Addr is equal to the last item's address plus X_Modify.
- In Autobuffer Mode, these registers are reloaded from X_Count, Y_Count, and Start_Addr upon the first data transfer.

BF533 MMRs for Peripheral DMA

MMR Name	Description
DMAx_NEXT_DESC_PTR	Link pointer to next descriptor
DMAx_START_ADDR	Start address of DMA buffer
DMAx_DMA_CONFIG	DMA configuration register
DMAx_X_COUNT	Inner loop count
DMAx_X_MODIFY	Inner loop address increment, in bytes
DMAx_Y_COUNT	Outer loop count (2D DMA only)
DMAx_Y_MODIFY	Outer loop address increment, in bytes
DMAx_CURR_DESC_PTR	Current Descriptor Pointer
DMAx_CURR_ADD	Current DMA Address
DMAx_IRQ_STATUS	Interrupt Status Register contains completion and error interrupt status information
DMAx_PERIPHERAL_MAP	Priority mapping register
DMAx_CURR_X_COUNT	Current count (1D) or intra-row X count (2D)
DMAx_CURR_Y_COUNT	Current row count (2D DMA only)

Parameter Registers

Current Registers

Control / Status Registers

BF533 MMRs for Memory DMA

MMR Name (yy = S0, S1, D0, D1)	Description
MDMA_yy_NEXT_DESC_PTR	Link pointer to next descriptor
MDMA_yy_START_ADDR	Start address of DMA buffer
MDMA_yy_DMA_CONFIG	DMA configuration register
MDMA_yy_X_COUNT	Inner loop count
MDMA_yy_X_MODIFY	Inner loop address increment, in bytes
MDMA_yy_Y_COUNT	Outer loop count (2D DMA only)
MDMA_yy_Y_MODIFY	Outer loop address increment, in bytes
MDMA_yy_CURR_DESC_PTR	Current Descriptor Pointer
MDMA_yy_CURR_ADD	Current DMA Address
MDMA_yy_IRQ_STATUS	Interrupt Status Register contains completion and error interrupt status information
MDMA_yy_PERIPHERAL_MAP	Priority mapping register (read only)
MDMA_yy_CURR_X_COUNT	Current count (1D) or intra-row X count (2D)
MDMA_yy_CURR_Y_COUNT	Current row count (2D DMA only)

Parameter Registers

Current Registers

Control / Status Registers

KAZTEK ENGINEERING

ANALOG DEVICES

Next Descriptor Pointer Register

DMAx_NEXT_DESC_PTR / MDMA_yy_NEXT_DESC_PTR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Next Descriptor Pointer [31:16]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

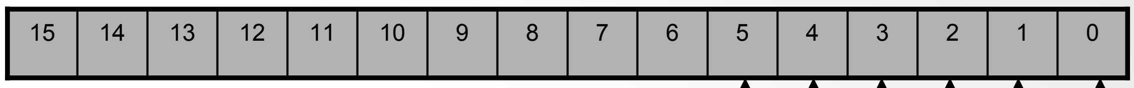
Next Descriptor Pointer [15:0]

Reset = 0x0000 0000

- **Specifies the location of the Next Descriptor Block when the current DMA transfer finishes. Used only in Small and Large Descriptor List Modes. Contents of this register are copied into the Curr_Desc_Ptr Register at the start of a descriptor block fetch. Disregarded in Stop, Autobuffer, and Descriptor Array Mode.**

DMA Configuration Register

DMAx_CONFIG / MDMA_yy_CONFIG



DMA Buffer Clear
0 = Retain DMA FIFO data between DMA transfers
1 = Discard DMA FIFO before beginning DMA transfer

DMA Mode
0 = Linear
1 = 2D DMA

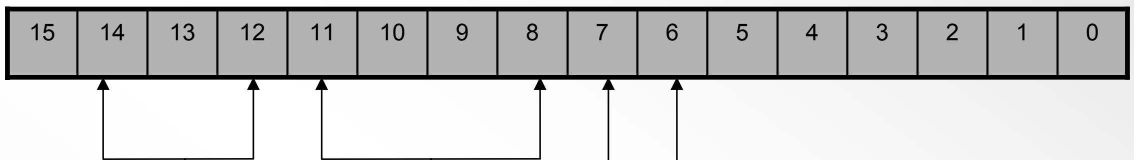
Transfer Word Size
00 = 8-bit transfers
01 = 16-bit transfers
10 = 32-bit transfers
11 = reserved

Transfer Direction
0 = Memory Read
1 = Memory Write
(Bit 1 cannot be modified for some peripherals and MemDMA)

DMA Enable
0 = Disabled
1 = Enabled

DMA Configuration Register (cont.)

DMAx_CONFIG / MDMA_yy_CONFIG



<u>Interrupt Timing Select</u>
0 = Interrupt after completing whole buffer
1 = Interrupt after completing each row (inner loop), 2D only

<u>FLOW (Next Operation)</u>
0x0 = Stop
0x1 = Autobuffer Mode
0x4 = Descriptor Array
0x6 = Descriptor List (small model)
0x7 = Descriptor List (large model)

<u>Interrupt Enable</u>
0 = Do not allow completion of DMA transfer to generate an interrupt
1 = Allow completion of DMA transfer to generate an interrupt

<u>NDSIZE (Next Descriptor Size)</u>
0000 = Required if Stop or Autobuffer Mode
0001 – 1001 = Descriptor Size
1010 – 1111 = Reserved

KAZTEK ENGINEERING

ANALOG DEVICES

Start Address Register

DMAx_START_ADDR_PTR / MDMA_yy_START_ADDR_PTR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA Start Address [31:16]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA Start Address [15:0]

Reset = 0x0000 0000

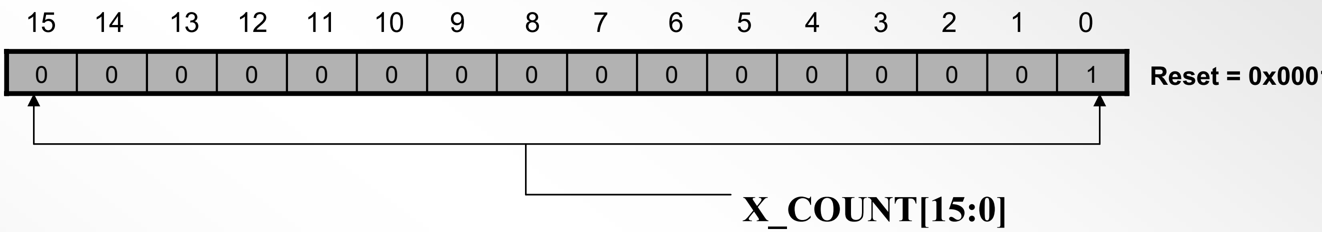
- **Specifies the address of the data buffer currently targeted for DMA. Contents of the Start_Addr_Ptr Register is copied into the Curr_Start_Addr Register at the start of a DMA transfer.**

KAZTEK ENGINEERING

ANALOG DEVICES

X Count Register

DMAx_X_COUNT / MDMA_yy_X_COUNT



- **For 2D DMA, the X_Count Register contains the inner loop count. For 1D DMA, it specifies the number of elements (8-, 16-, or 32-bit) to read in. A value of 0x0 in X_Count corresponds to 65,536 elements.**

X Address Increment Register

DMAx_X_MODIFY / MDMA_yy_X_MODIFY

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Reset = 0x0002

X_MODIFY[15:0]

- **This register contains a signed, 2's compliment byte-address increment. In 1D DMA, this increment is the stride that is applied after transferring each element.**
- **In 2D DMA, this increment is applied after transferring each element in the inner loop, up to but not including the last element in each inner loop. After the last element in each inner loop, Y_Modify is applied instead.**

KAZTEK ENGINEERING

ANALOG DEVICES

Outer Loop Count Register

DMAx_Y_COUNT / MDMA_yy_Y_COUNT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Reset = 0x0001

Y_COUNT[15:0]

- **For 2D DMA, the Y_Count Register contains the outer loop count. This register contains the number of rows in the outer loop of a 2D DMA sequence.**
- **It is not used in 1D DMA.**

KAZTEK ENGINEERING

ANALOG DEVICES

Outer Loop Address Increment Register

DMAx_Y_MODIFY / MDMA_yy_Y_MODIFY

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Reset = 0x0002

Y_MODIFY[15:0]

- **This register contains a 2's compliment byte-address increment. In 2D DMA, this increment is applied after each decrement of Curr_Y_Count except for the last item in the 2D array on which the Curr_Y_Count also expires.**
- **The value is the offset between the last word of one "row" and the first word of the next "row"**

Current Descriptor Pointer Register

DMAx_CURR_DESC_PTR / MDMA_yy_CURR_DESC_PTR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Current Descriptor Pointer [31:16]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

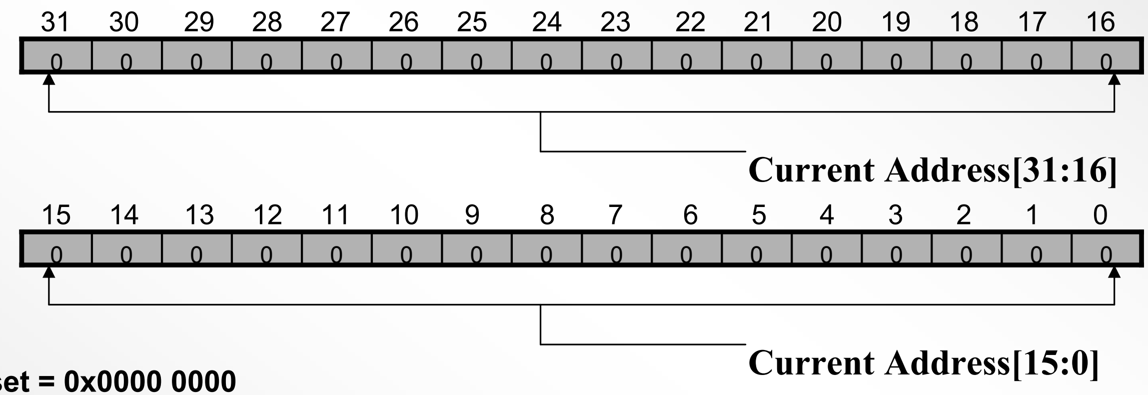
Current Descriptor Pointer [15:0]

Reset = 0x0000 0000

- **Contains the memory address of the next descriptor element to be loaded. Curr_Desc_Ptr Register increments as each descriptor element is read in. For Descriptor Array Mode, the Curr_Desc_Ptr Register must be programmed, not the Next_Desc_Ptr Register, to initiate a DMA transfer.**

Current Start Register

DMAx_CURR_ADDR / MDMA_yy_CURR_ADDR



Reset = 0x0000 0000

- **Contains the current DMA transfer address. At the start of a DMA transfer, the Curr_Addr Register is loaded from the Start_Addr Register and it is incremented as each transfer occurs.**

Current X Count Register

DMAx_CURR_X_COUNT / MDMA_yy_CURR_X_COUNT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**Reset = 0x0000**

CURR_X_COUNT[15:0]

- **This register is loaded by X_Count at the beginning of each DMA transfer.**
- **It is decremented each time an element is transferred.**
- **For 2D DMA, Curr_X_Count is reloaded after the end of DMA for each row.**
- **Expiration of the count in this register signifies that DMA is complete. In 2D DMA, this register is 0 only when the entire transfer is complete.**

KAZTEK ENGINEERING

ANALOG DEVICES

Current Outer Loop Count Register

DMAx_CURR_Y_COUNT / MDMA_yy_CURR_Y_COUNT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Reset = 0x0000

CURR_Y_COUNT[15:0]

- **This register is loaded by Y_Count at the beginning of each 2D DMA transfer.**
- **Not used for 1D DMA.**
- **This register is decremented each time that the Curr_X_Count Register expires during 2D DMA (1 to X_Count or 1 to 0 transition), signifying completion of an entire row transfer.**
- **After 2D DMA is complete, Curr_Y_Count = 1 and Curr_X_Count = 0**

KAZTEK ENGINEERING

ANALOG DEVICES

Interrupt Status Register

DMAx_IRQ_STATUS / MDMA_yy_ IRQ_STATUS

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Reset = 0x0000

DMA_RUN (DMA Channel Running) – RO
This bit is set to 1 automatically when the DMA_CONFIG register is written
0 – This DMA channel is disabled, or it is enabled but paused
1 – This DMA channel is enabled and operating, either transferring data or fetching a DMA descriptor

DFETCH (DMA Descriptor Fetch) – RO
This bit is set to 1 automatically when the DMA_CONFIG register is written with FLOW = 0x4 – 0x7
0 – This DMA channel is disabled, or it is enabled but stopped
1 – This DMA channel is enabled and presently fetching a DMA descriptor

DMA_DONE (DMA Completion Interrupt Status) – W1C
0 – No interrupt is being asserted for this channel
1 – DMA transfer has completed, and this DMA channel's interrupt is being asserted

DMA_ERR (DMA Error Interrupt Status) – W1C
0 – No DMA error has occured
1 – A DMA error has occured, and the global DMA error interrupt is being asserted.

KAZTEK ENGINEERING

ANALOG DEVICES

DMA Traffic Control Counter Period Register

TC_PER

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MDMA_ROUND_ROBIN_PERIOD[4:0]
Max. length of MDMA round-robin bursts. If not zero, any MDMA stream which receives a grant is allowed up to that number of DMA transfers, to the exclusion of the other MDMA streams.

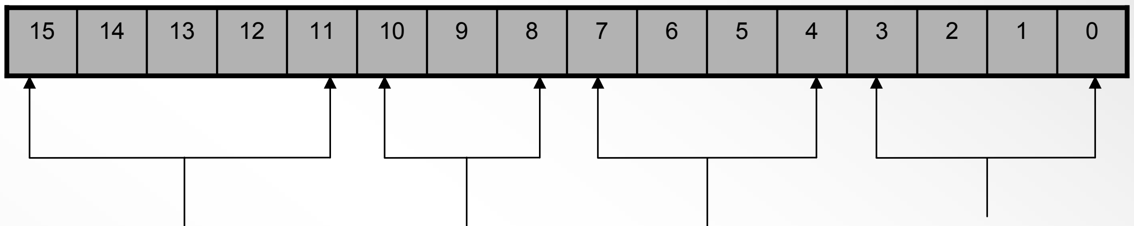
DAB_TRAFFIC_PERIOD[2:0]
000 = No DAB bus transfer grouping performed
Other = Preferred length of unidirectional bursts on the DAB bus between the DMA and the peripherals.

DEB_TRAFFIC_PERIOD[3:0]
000 = No DEB bus transfer grouping performed
Other = Preferred length of unidirectional bursts on the DEB bus between the DMA and external memory.

DCB_TRAFFIC_PERIOD[3:0]
000 = No DCB bus transfer grouping performed
Other = Preferred length of unidirectional bursts on the DCB bus between the DMA and internal L1 memory

DMA Traffic Control Counter Register

TC_CNT - RO



MDMA_ROUND_ROBIN_COUNT[4:0]
Current cycle count remaining in the MDMA round robin period

DCB_TRAFFIC_COUNT[3:0]
Current cycle count remaining in the DCB traffic period

DEB_TRAFFIC_COUNT[3:0]
Current cycle count remaining in the DEB traffic period

DAB_TRAFFIC_COUNT[2:0]
Current cycle count remaining in the DAB traffic period

KAZTEK ENGINEERING

ANALOG DEVICES