5 DMA

The ETRAX 100 DMA provides a high data transfer rate to and from the internal peripheral interfaces, or from one location in the external memory to another. The DMA consists of ten DMA channels, five in each direction. The ten DMA channels are served by a DMA controller which takes care of the data flow between the channels and the external memory.

5.1 DMA OPERATION

5.1.1 The overall architecture and operation

A simplified architecture overview of the DMA is shown in figure 5-1.

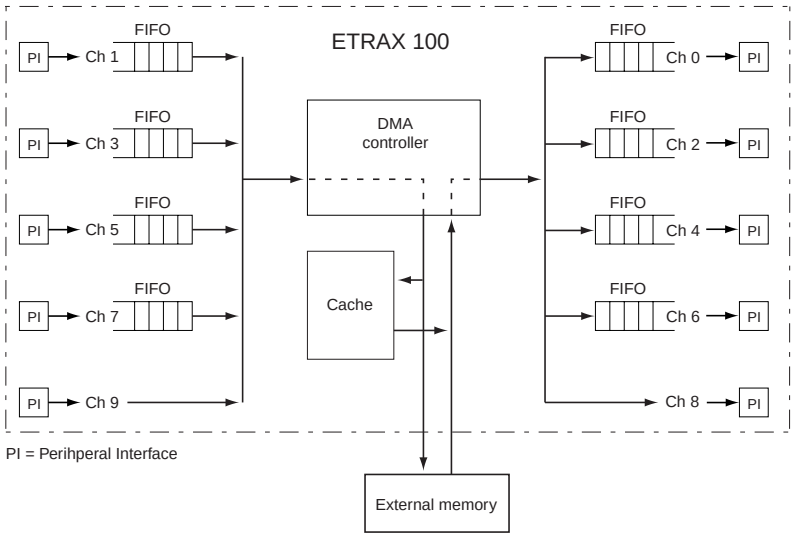


Figure 5-1 DMA internal overview

Figure 5-1 gives a schematic view of the flow of data. There are five input channels, and four of these have FIFO buffers, and there are correspondingly five output channels, four with FIFO buffers.

There are three cases of data transference: 1) from a peripheral interface to the external memory; 2) from the external memory to a peripheral interface; and 3) from one memory location to another.

1. Data transfer from a peripheral interface to the external memory.

When data is to be transferred from one peripheral interface, it is first stored into a 64 byte FIFO buffer. When this buffer fills up to half its size, i.e. 32 bytes, the DMA controller is notified and the data in the FIFO is being transferred to the external memory. If one of these memory addresses recently have been used by the CPU, the data is stored in the internal cache memory as well. The transfer of data from a FIFO to the memory is done in bursts, and the length of these bursts is either 16 or 32 bytes as chosen by the software programmer in an internal register.

2. Data transfer from the external memory to a peripheral interface.

When the DMA controller receives notice of that a FIFO buffer is becoming half empty, i.e. less than or equal to 32 bytes, it begins a transfer of data from the external memory. When data is transferred from the external memory it is read in bursts of 16 or 32 bytes (as chosen by the software programmer). If one of the memory addresses where the data is located has recently been in use by the CPU, the data is read from the internal cache memory.

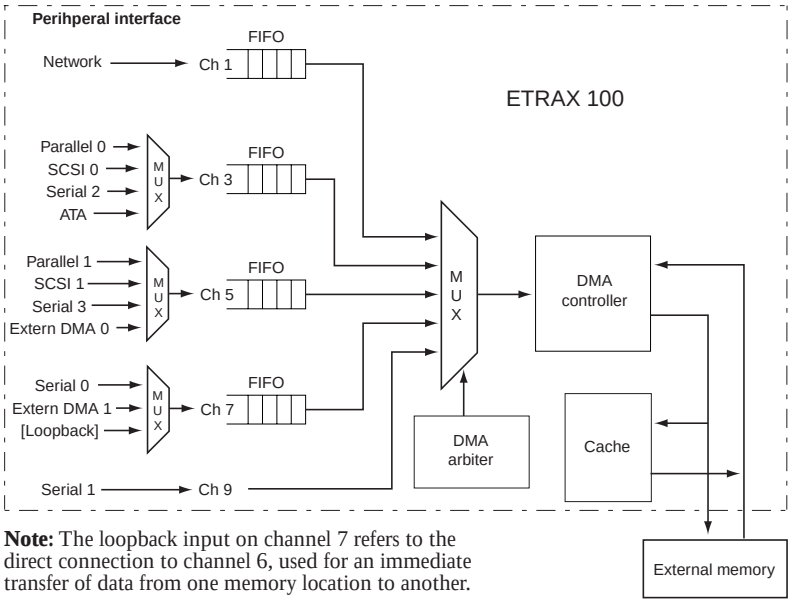
3. Data transfer from one external memory location to another.

In order to increase the performance of the DMA for this type of data transfer there are two channels specifically designed for this task, channel 6 and 7. The FIFO buffers of the two channels 6 and 7 can be set to connect directly to each other. As with previous transfers the data is read from the external memory when the FIFO buffer is becoming half empty. The data is either read from, or written to the cache - depending on the direction of the data flow - if one of the memory addresses have been in use by the CPU recently.

Apart from these three cases of data transference the ETRAX 100 DMA has two external DMA channels, DMA0 and DMA1, see section 5.3 "External DMA Channels" on page 54.

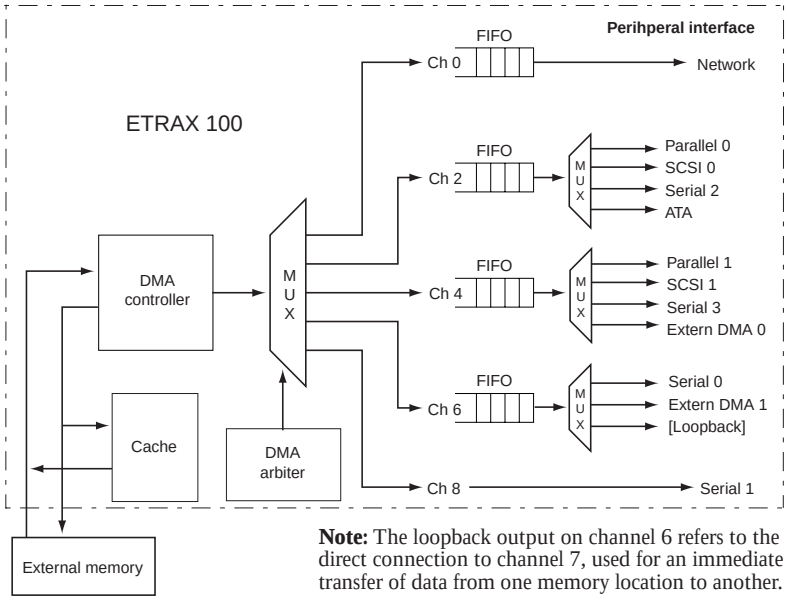
5.1.2 The DMA Channels

There are only a limited number of combinations in which the ten channels of the DMA can be used for interconnection between the internal peripheral interfaces. The reason for this is that some of the internal peripheral interfaces are multiplexed on the same package pins. The figure 5-2 and the figure Figure 5-3 on page 48 show how each peripheral interface are multiplexed on the DMA channels.



Note: The loopback input on channel 7 refers to the direct connection to channel 6, used for an immediate transfer of data from one memory location to another.

Figure 5-2 DMA input channels



Note: The loopback output on channel 6 refers to the direct connection to channel 7, used for an immediate transfer of data from one memory location to another.

Figure 5-3 DMA output channels

The choice of interface used is defined in an internal configuration register. The channels can be configured as shown in table 5-1.

DMA channel	I/O system(s) available				Direction	FIFO buffer
0	Network				out	64 bytes
1	Network				in	64 bytes
2	Parallel 0	SCSI0	Serial 2	EIDE/ATA-2	out	64 bytes
3	Parallel 0	SCSI0	Serial 2	EIDE/ATA-2	in	64 bytes
4	Parallel 1	SCSI1	Serial 3	DMA0, see note 1	out	64 bytes
5	Parallel 1	SCSI1	Serial 3	DMA0, see note 1	in	64 bytes
6	Serial 0	DMA1, see note 1	Memory transfer, see note 2		out	64 bytes
7	Serial 0	DMA1, see note 1	Memory transfer, see note 2		in	64 bytes
8	Serial 1				out	none
9	Serial 1				in	none

Table 5-1

Note 1: DMA0 and DMA1 are external DMA channels to be used between the external memory and an I/O device, see section 5.3 "External DMA Channels" on page 54.

Note 2: Memory-to-memory transfer. Channel 6 to channel 7 can be set for immediate connection and thus providing an efficient way of transferring data from one memory location to another.

5.1.3 DMA linked lists

The ETRAX 100 DMA stores data in the external memory in small buffers linked together with the use of a list descriptor. Each list descriptor contains a number of data fields which tells the DMA controller where to find next buffer in the list and how large it is.

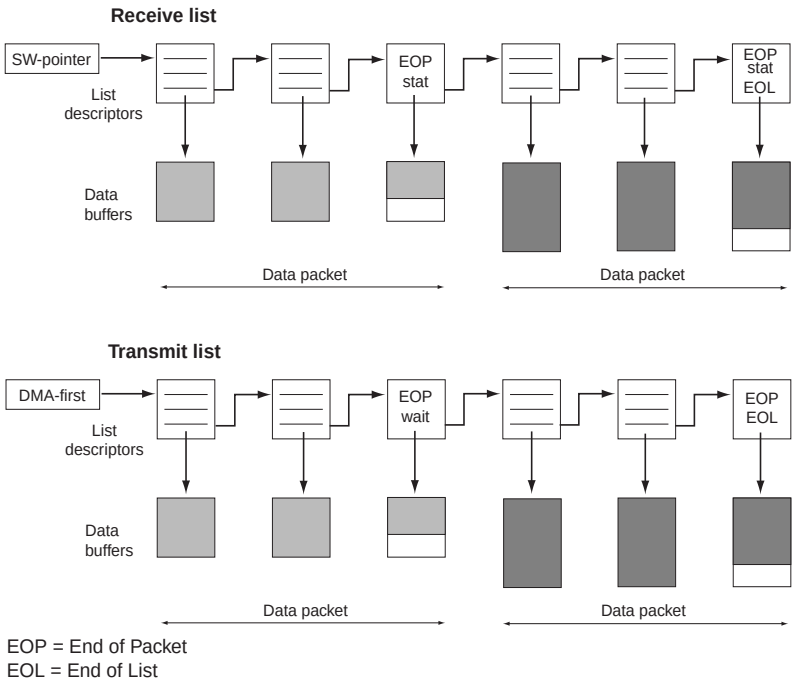


Figure 5-4 ETRAX 100 buffer structure

The list descriptors contains information of the location and size of the data buffer as well as a number of other status information. This organisation of data storage in the memory has the advantage of producing a small internal fragmentation, an efficient memory management, and a very flexible structure.

For a more detailed overview of the list descriptor see section 5.1.5 "DMA descriptor" on page 51.

5.1.4 DMA registers

There are a set of DMA registers, one set for each channel, which the DMA controller uses to handle the buffers, and to store information about the buffers in the external memory. These registers and a little about of their functions are shown in figure 5-5.

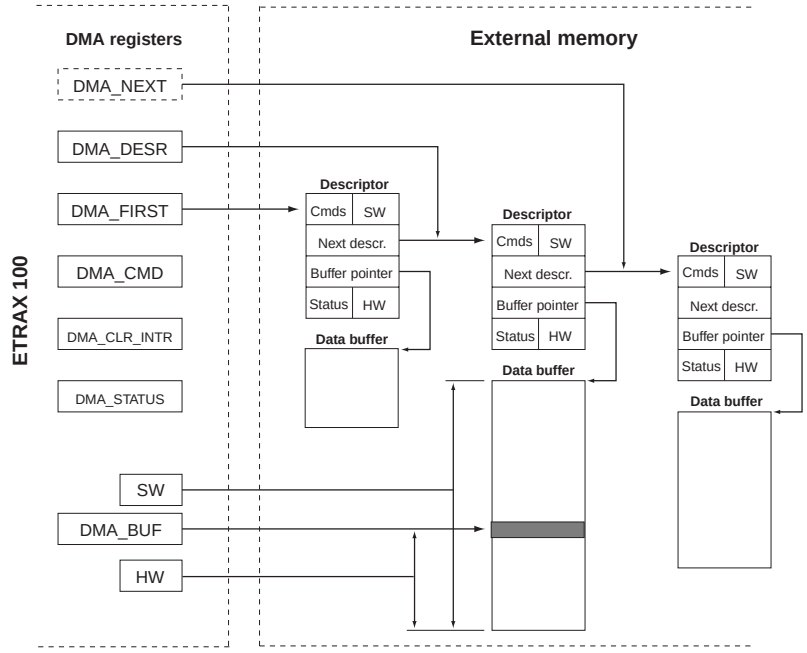


Figure 5-5 DMA registers and the structure of the linked list

Figure 5-5 shows the DMA registers and a simplified linked list in the external memory. When a DMA transfer is to start the DMA reads the linked list in the memory (see section 5.1.5 "DMA descriptor" on page 51).

The DMA_FIRST register points to a command field (Cmds in figure 5-5) in the first descriptor in the linked list. The DMA_DESCR register contains the address to the current descriptor, and the DMA_NEXT contains the address to the next descriptor.

Three registers manage the actual storage of data in the buffers: DMA_BUF register is a pointer to the next (byte) position in the buffer; the SW register gives the total length (in bytes) of the buffer, and the HW register gives the number of bytes left in the buffer.

5.1.5 DMA descriptor

The construction of a linked list is done by defining DMA descriptors. The DMA descriptor consists of four 32-bit fields. The contents of a descriptor is shown in figure 5-6 and table 5-2 on page 52.

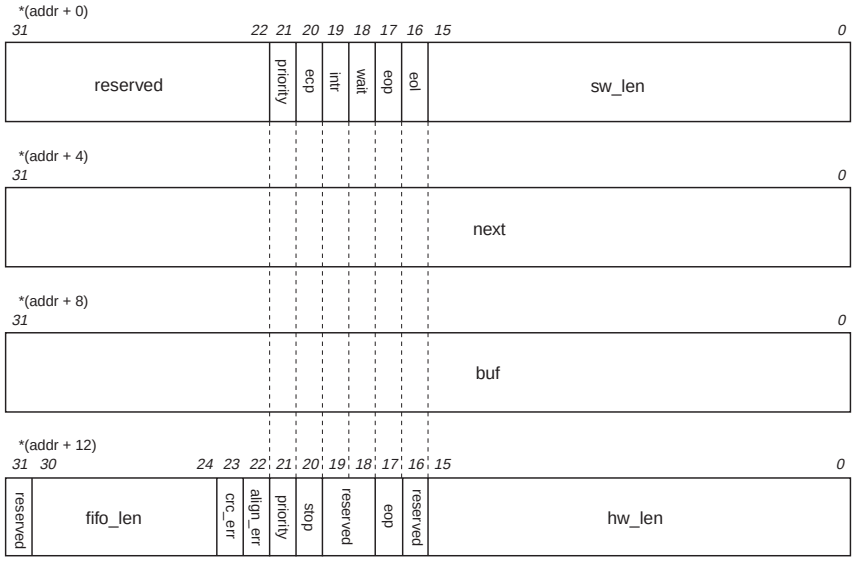


Figure 5-6 The DMA descriptor

The first 32-bit field of the descriptor gives the length of the buffer (sw_len) and a number of commands. The second 32-bit field gives the address to the next descriptor in the linked list (next), and the third 32-bit field gives the address to the data buffer (buf). The last 32-bit field contains status information written by the DMA controller.

Table 5-2 describes the contents of the DMA descriptor in more detail.

Bit	Name	Explanation
(addr + 0):		
31-22	reserved	Not written by DMA. See note.
21	reserved	
20	ecp_cmd	ECP command used by channel 2 and 4 when connected to a parallel port in ECP mode.
	tx_err	Force transmission error used by channel 0.
19	intr	Generate a descriptor interrupt when advancing to next descriptor.
18	wait	Wait until FIFO is empty before advancing to next descriptor (output channels only).
17	eop	Last descriptor in a packet.
16	eol	Last descriptor in the list.
15-0	sw_len	Length in bytes of data buffer. (If all bits are 0, the length is 2^{16})
(addr + 4):		
31-0	next	Pointer to next descriptor in list (no alignment restrictions). If eol == 1 next is not used.
(addr + 8):		
31-0	buf	Pointer to first byte in data buffer (no alignment restrictions).
(addr + 12):		
31	reserved	Always set to zero when written by the DMA.
30-24	fifo_len	If stop == 1; Number of bytes in FIFO. Output channels only.
23	crc_err	If eop == 1; Received packet has CRC error, used by channel 1.
22	align_err	If eop == 1; Received packet has alignment error, used by channel 1.
21	reserved	
20	stop	Output channel was stopped by I/O interface. Bit 20 and bit 17 are mutually exclusive.
19-18	reserved	Not read by DMA. Always set to zero when written.
17	eop	Last descriptor in a received packet. Bit 20 and bit 17 are mutually exclusive.
16	reserved	Not read by DMA. Always set to zero when written.
15-0	hw_len	If eop == 1; Number of bytes written to the data buffer. If stop == 1; Number of bytes read from the data buffer.

Table 5-2 Contents of the DMA descriptor

Note: For compatibility with future versions of ETRAX processors, reserved fields must be set to 0 before starting the DMA. When read no assumption can be made about their value.

5.2 DMA INTERRUPT

There are two interrupts per channel: descriptor interrupt and end-of-packet interrupt. Each channel has its own interrupt vector (for a list of all interrupt vector numbers see section 14.2.1 "Overview of internally generated interrupt vector numbers" on page 98), and each interrupt in a channel can be individually enabled or disabled.

For output channels the interrupts are generated if the intr-bit (descriptor interrupt) or eop-bit (end-of-packet interrupt) in the descriptor are set. The interrupts are generated after the DMA has read all data from the associated data buffer.

For input channels the descriptor interrupt is generated if the intr-bit is set in the descriptor. The end-of-packet interrupt is generated when the peripheral interface signals end-of-packet and the DMA has written all data to the associated data buffer.

Enable, disable, read and clear interrupts are done in internal registers.

5.3 EXTERNAL DMA CHANNELS

The external DMA channels DMA0 and DMA1 could be seen as an I/O interface with a "pseudo DMA" operation. Its purpose is to provide a DMA-like interface between the external memory and external I/O-devices, see figure 5-7.

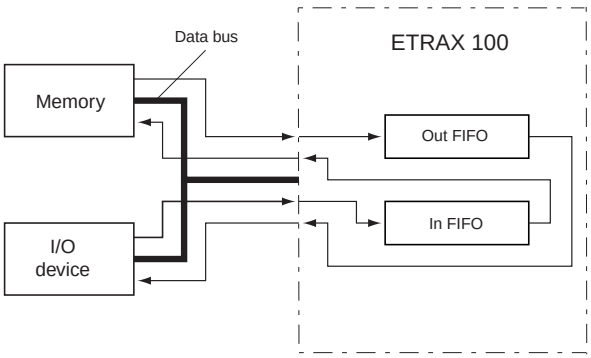


Figure 5-7 ETRAX100 external DMA channels' pseudo DMA principle

The two external DMA channels are bidirectional: DMA0 uses the internal DMA channels 4 and 5, and DMA1 uses the channels 6 and 7. The data bus width can be configured for 8, 16, or 32 bits. Burst or handshake modes can be used for the transfer of data, and the maximum data rate is 66 Mbyte/s in burst mode. In addition to these four internal DMA channels the external DMA uses two control signals, request and acknowledge, to enable a handshake procedure.

The pseudo DMA operation differs from an ordinary DMA operation in that the data from the memory or the I/O device passes through the FIFO buffer of the internal DMA channel (se figure 5-7). This fact has the advantage that the width of the I/O bus is independent of memory width, since a different bus width could be used between the I/O device and the FIFO than between the FIFO and the memory.

An internal configuration register is used to set the following modes of operation:

- width of transfer (8, 16 or 32 bits)
- direction (in or out)
- polarity of request signal
- polarity of acknowledge signal
- request/acknowledge mode (burst or handshake)

Burst mode

When burst mode is used for transferring data, the external I/O interface keeps the request signal active during the whole burst and a read or write is performed each

time the acknowledge signal is active. When the acknowledge signal is not active the data bus is available for other units.

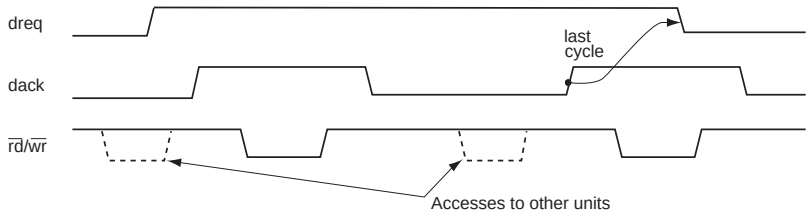


Figure 5-8 Timing diagram for bursts mode

Handshake mode

When handshake mode is used, the address bus is used to address the external interface (however, this could also be used in burst mode), and the address is set in an internal register. The two most significant bits of the address are always set to 10 (binary), so as not to access the internal cache memory or the DRAM (see section 3.2 "Address and Chip Selects" on page 30):

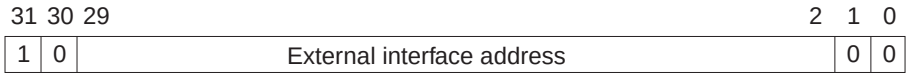


Figure 5-9 The address bus when addressing an external interface

A separate request/acknowledge handshake cycle with a valid external interface address is performed for each transfer of data.

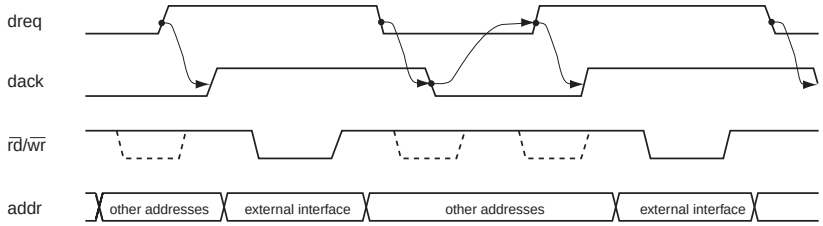


Figure 5-10 Timing diagram for handshaking mode

Transfer counter

There is also an internal counter register which can be used to transmit a certain number of data transfers. This is done by setting the counter to a value equal to the desired number of transfers and starting the transfer. The transfer will end when the counter has reached zero.