Eureka Technology

EP660 DMA Controller

FEATURES

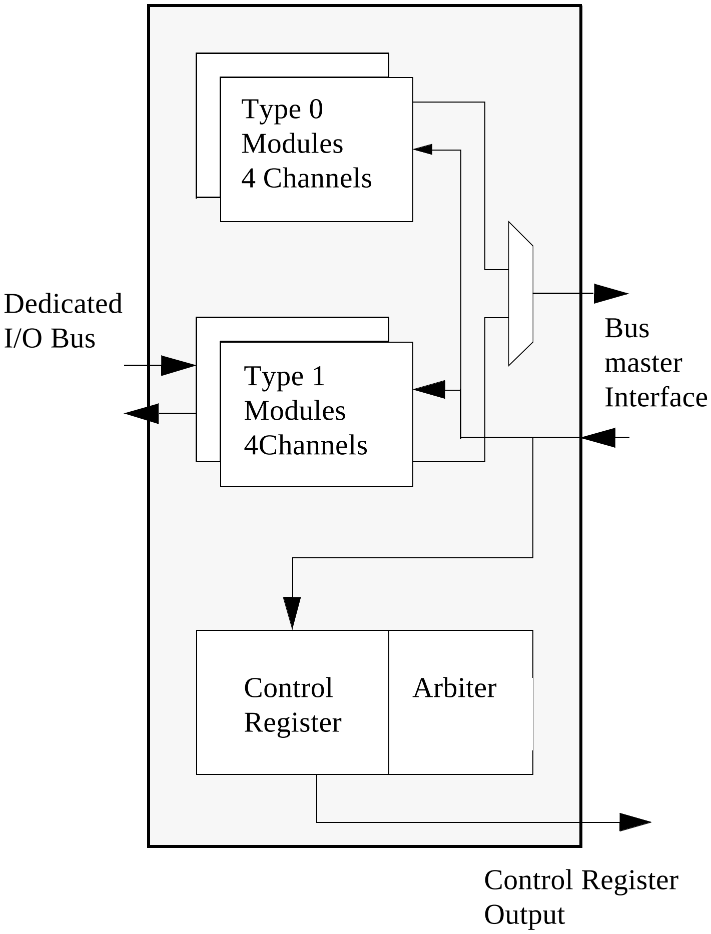
- Multiple independent DMA modules
- Up to 16 DMA channels supported
- Designed for ASIC or FPGA implementations in various system environments
- Two types of DMA modules for memory - memory and memory - I/O data transfer.
- Supports both hardware initiated transfer and software initiated block transfer.
- Interrupt generation on transfer completion.
- Two levels of transfer priority for individual DMA channels.

DESCRIPTIONS

The EP660 is a general purpose DMA controller designed for data transfer in different situations. Two type of modules are provided and the user can choose the number of modules for each type. Type 0 module is designed for transferring data that reside on the same memory bus. For example, it can transfer between memory to memory that both reside on the processors host bus. Type 1 module is designed for transferring data between the memory bus and dedicated I/O bus. Each I/O channel has dedicated interface to its I/O device. This type of module is particularly useful when the DMA core is integrated with I/O devices on the same chip.

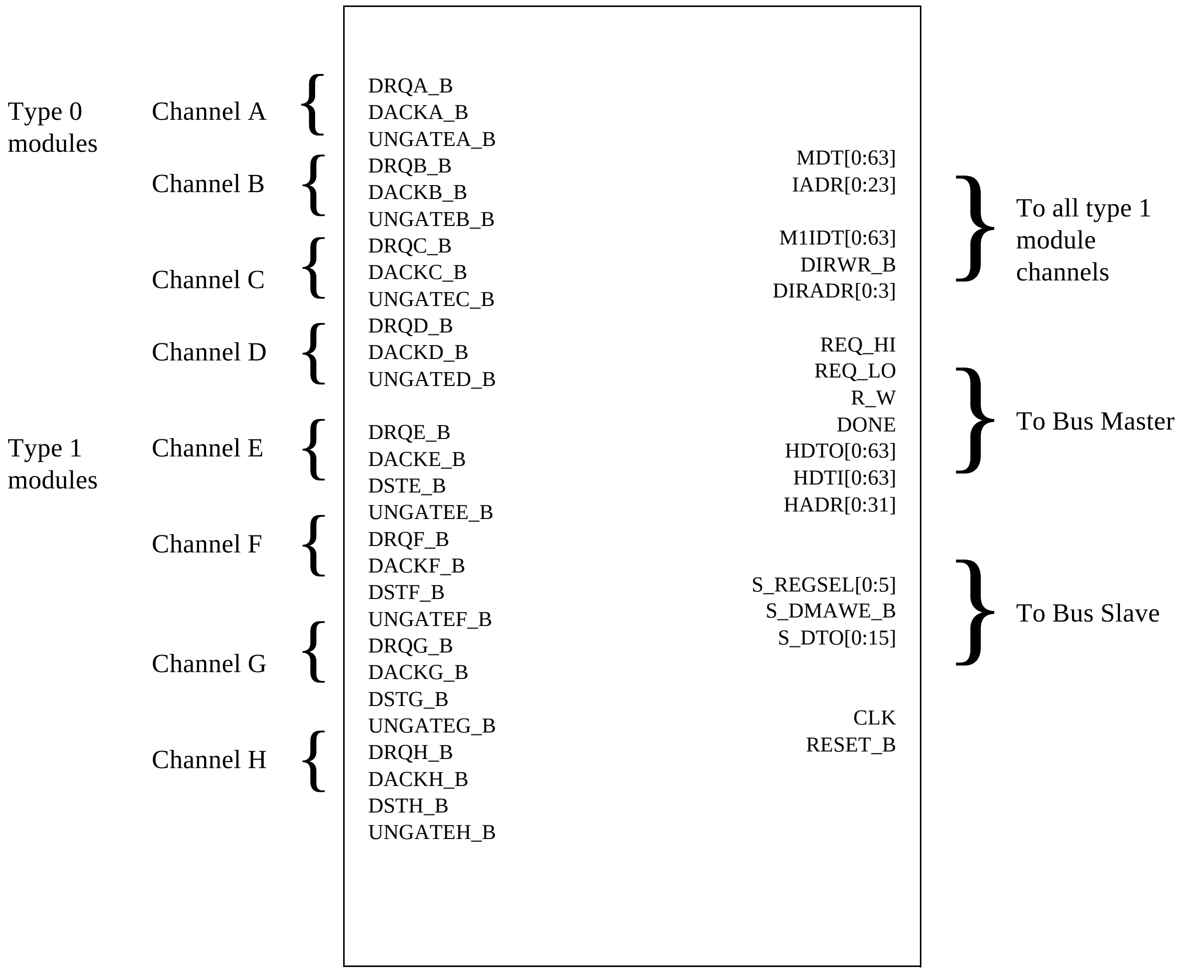
Each module supports up to 4 channels and up to 16 channels are supported by the DMA. Each channel can be programmed independently and has dedicated DMA request line.

Block Diagram



R 1.0S

1 Pin Definition



2 Signal Definition

All signal or bus names ended with "_B" are active low signals, otherwise it is active high.

Table 1: Interface to individual I/O devices

Name	Type	Description
DACK(A-H)_B	O	DMA acknowledge. One for each DMA channel. This signal is asserted by the DMA controller at the completion of a DMA transfer.
DRQ (A-H)_B	I	DMA request. One for each DMA channel. This signal is asserted by an I/O device to request DMA service. Once asserted, the I/O device must keep it asserted until the request is acknowledged.
DST(E-H)_B	O	DMA start. This signal is asserted by the DMA controller to indicate to the I/O device of the beginning of an unsynchronized DMA transfer. Channel A to D does not have DST_B signal since these channels are type 0 channels.
UNGATE(A-H)_B	I	Unsynchronized gate. This signal is asserted by the I/O device to indicate the it can accept an unsynchronized DMA transfer. UNGATE_B must be asserted before DST_B can be asserted by the controller. The DMA transfer is suspended if it is de-asserted while block transfer is running
M1IDT[0:63] M2IDT[0:63]	I	Input data from I/O for Type 1 Module. This type of has 4 channels and only the first 2 channels are used for transferring data from I/O to memory.
DIRWR_B	I	Direct write. When this signal is asserted, an external device is writing directly to the control register of channel F and G. Input data is received on M2IDT and the control register location is selected by DIRADR

Table 1: Interface to individual I/O devices

Name	Type	Description
DIRADR[0:3]	I	Direct write address. Select one of the 12 control register from channel F and G to be written directly. This signal is sampled when DIRWR_B is asserted. DIRADR[0:3] value: 0000: Memory address, high order, channel F 0001: Memory address, low order, channel F 0010: I/O address, high order, channel F 0011: I/O address, low order, channel F 0100: Channel control, channel F 0101: Transfer count, channel F 1000: Memory address, high order, channel G 1001: Memory address, low order, channel G 1010: I/O address, high order, channel G 1011: I/O address, low order, channel G 1100: Channel control, channel G 1101: Transfer count, channel G

.

Table 2: Interface common to all I/O devices

Name	Type	Description
IADR[0:23]	O	I/O address. Indicates the I/O address for DMA transfer. This bus is shared by all I/O devices
MDT[0:63]	O	Memory data to be transfer from memory to I/O devices

.

Table 3: Interface to system bus master

Name	Type	Description
REQ_HI	O	Request signal to bus master indicating high priority DMA request.
REQ_LO	O	Request signal to bus master indicating low priority DMA request
DONE	I	Request complete. Generated by the bus master to indicate that the request is done.
R_W	O	Read/write request indicator to the bus master. High indicates read and low indicates write

Table 3: Interface to system bus master

Name	Type	Description
HDTO[0:63]	O	Output data to the memory bus. This bus would be driven through data muxes and tri-state buffer on to the host bus. Controls to the muxes and tri-state buffer are generated by the host bus master.
HDTI[0:63]	I	Input data from the memory bus. This signal is captured by the DMA controller during read cycle on the host bus. This bus also provides the input to the control register during control register write.
HADR[0:31]	O	Output to the host bus for read and write. This bus would be driven through tri-state buffer on to the host bus. Enable signal to the tri-state buffer is generated by the host bus master.

.

Table 4: Interface to system bus slave

Name	Type	Description
S_DMAWE_B	I	Control register write enable. When this signal is active, data from HDTI is written into the control register selected by S_REGSEL[0:5]. This signal is active for only one cycle during control register write.
S_REGSEL[0:5]	O	Control register select. This bus select one of the control register to be accessed by the bus slave. Data of the selected register is multiplexed on to S_DTO immediately regardless of the state of the DMA controller.
S_DTO[0:15]	O	Control register output data. This bus represent the value of the control register selected by S_REGSEL.

.

Table 5: Common signals

Name	Type	Description
CLK	I	System clock. All synchronous signal are sampled at the rising edge of CLK or driven out from the rising edge of the clock.
RESET_B	I	System reset. This signal is synchronous with CLK and RESET_B must be sampled by at least one rising clock edge in order to reset the system.

3 Functional Description

The DMA controller has twelve channels organized in five modules. Each channel can accept DMA requests from hardware or request through software.

Two type of module are provided. Type 0 module is designed for transferring data between memory or I/O locations that resides on the same bus. For example, if the DMA is connected to the processor's host bus through a bus master, it can be used to transfer between memory to memory that are on the processor's host bus. Alternatively, if the bus master is connected to the PCI bus, it can be used to transfer data between memory or I/O locations that are accessible through the PCI bus. The DMA is mostly independent of the system bus type that the bus master is connected to.

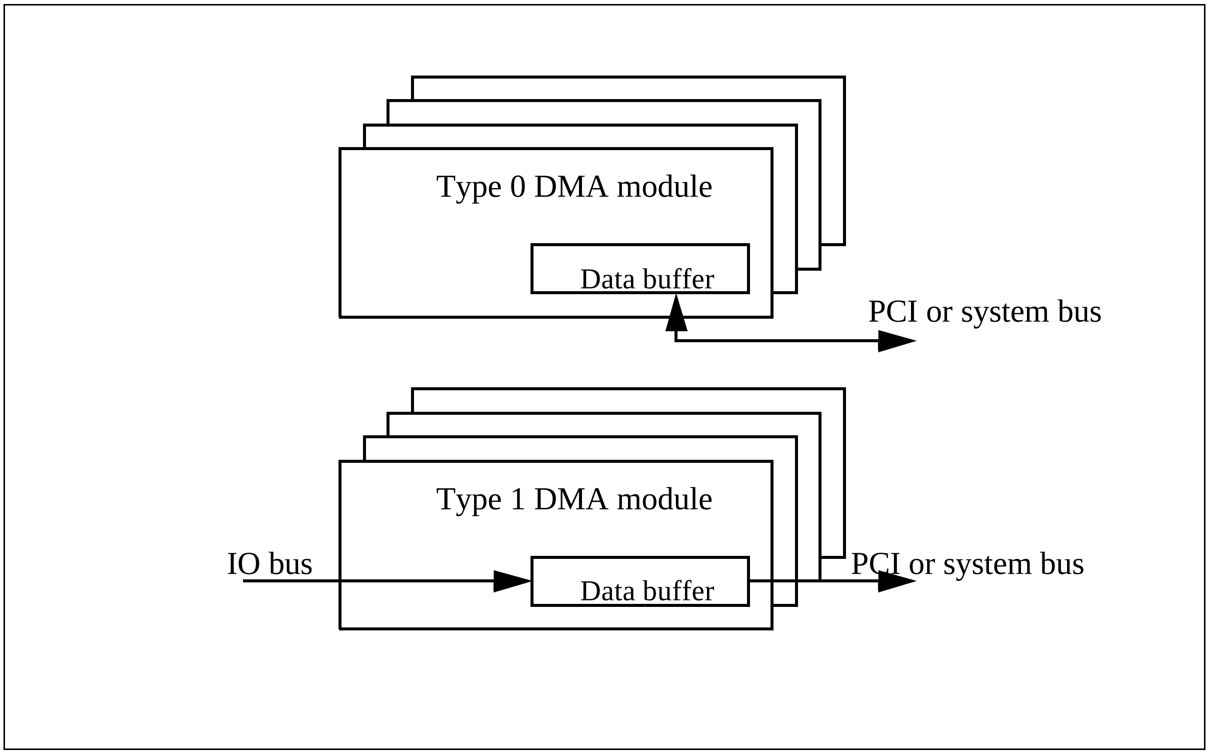
Type 1 module is designed for transferring data between the bus master and dedicated I/O bus. Each I/O device has dedicated buses goes directly into the DMA controller for data transfer. Type 1 module is more efficient in terms of memory bandwidth because each transfer access the system bus only once. This type of module is especially useful when the DMA core is integrated with the I/O devices on the same chip.

Each type of module supports 4 DMA channels and the DMA controller supports up to 4 DMA modules. More modules or channels can be added if necessary. Contact Eureka Technology for more detail.

All channels share the same interface to the system memory and the controller arbitrates the system memory access among the modules.

In the following sections, the function of Type 1 module is described first followed by the description of Type 0 module.

.



3.1 Synchronized DMA, Type 1 Module

DMA request can be initiated by the I/O device for each word transfer or can be initiated by software for block transfer. The control register for each channel can be programmed for either types of transfer.

Request initiated by hardware is called synchronous transfer. The DMA controller responds to the I/O requests only when the start bit (STRT) for the particular channel is set.

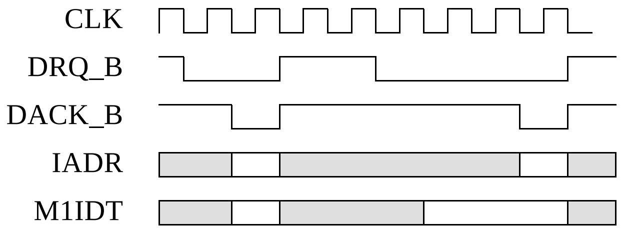
When an external device initiates DMA request, it must be ready to provide the read data or ready to accept the write data. A DMA request is completed on the I/O device when acknowledge is received. The timing between DMA request and acknowledge various depending on the availability of the system memory bus and other DMA request in progress.

The following diagram shows the timing of a DMA transfer from I/O to memory. At the first cycle, DRQ_B is asserted by the I/O device requesting DMA service. If there is no other DMA request running at the same time, DACK_B is asserted in the next cycle to terminate the transfer. The I/O device requesting DMA must have data available one cycle after DRQ_B is asserted and data must remain valid until DACK_B is asserted. The I/O address, IA, is also valid at the cycle when DACK_B is asserted.

The following diagram also shows a second DMA request when the DMA controller is busy. DACK_B is not asserted until a few cycles later. The I/O device must keep data valid until DACK_B is asserted.

Once DRQ_B is asserted, it must remains asserted until DACK_B. If the device intends to perform only one data transfer, it must de-assert DRQ_B in the clock cycle after DACK_B is asserted. Otherwise the DMA controller will assume that the device wants to perform another DMA request immediately after the current transfer.

DMA From IO to Memory

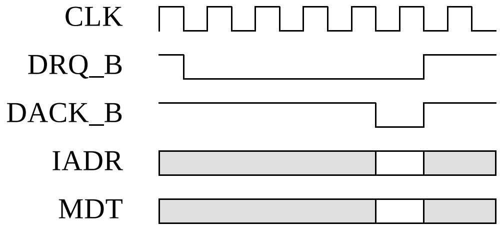


The following diagram shows a DMA transfer from memory to I/O device. DRQ_B is asserted by the requesting device at cycle 1. This type of transfer would take at least a few clock cycles to complete. Data must be read from the system memory by the system bus master and then returned back to the I/O device. The exact number of clock cycle depends on the speed of the memory device and the arbitration time on the system bus.

Similar to the previous transfer, the I/O device must keep DRQ_B asserted until DACK_B is asserted. Data is valid for one clock cycle when DACK_B is asserted so that the I/O device must be ready to received the return data.

It is possible for synchronous DMA to perform block data transfer by keeping DRQ_B asserted for continuously. This will cause the DMA channel to perform multiple DMA transfers until DRQ_B is de-asserted. The I/O device must be responsible for de-asserting DRQ_B after the intended amount of data has been transferred. During the transfer, other lower priority DMA channels will not receive DMA service.

DMA From Memory to I/O



3.2 Memory and I/O address Pointers

Each I/O channel maintains a 32-bit memory address point (the lower 3 bits fixed at zero) and a 24-bit I/O address pointer. This allows each DMA channel to access 4-Gbytes of memory space and 16-Mbytes I/O space.

Memory and I/O address pointers can be programmed for auto-increment, auto-decrement or unchanged after each transfer. If they are programed for increment or decrement, the increment and decrement amount is 8, i.e. the lower 3 bits of each pointer is fixed at zero.

All transfers involving I/O devices are assumed to be 64-bit wide. If the actual device bus width is less the 64 bits, the valid data must be aligned with bit zero and ends at bit N, where N is bus width minus 1.

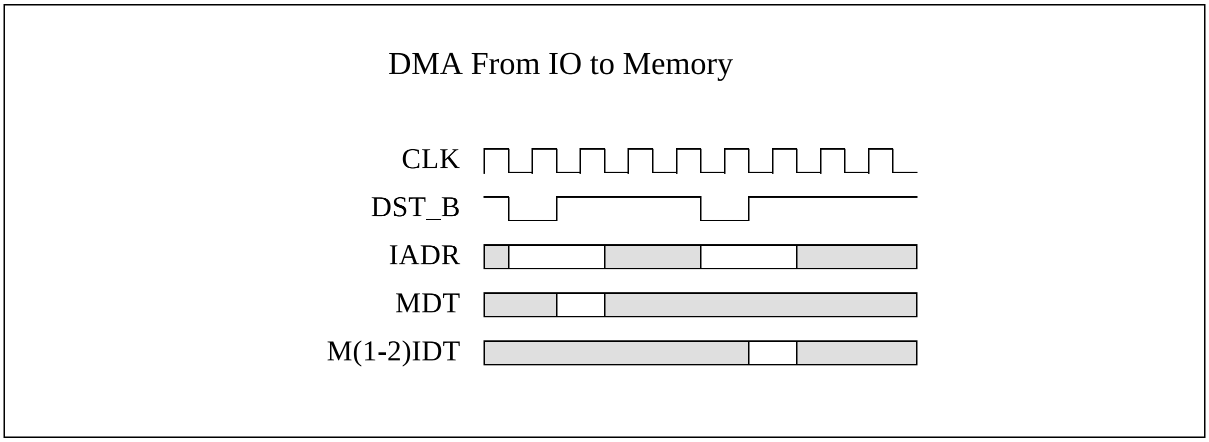
3.3 Unsynchronized Transfer

DMA transfers can be initiated by system software by selecting unsynchronized transfers in the control register. Transfer starts when the start bit (STRT) is in the control register is set and when the UNGATE_B signal is asserted (low). Once started, transfer continue until the transfer count reaches zero or when UNGATE_B is de-asserted.

The UNGATE_B signal can be generated by external logic or connected to the output of an I/O register. It is designed for external control of DMA start and suspend. UNGATE_B is ignored when the channel is programmed for synchronized transfer or when STRT bit is cleared.

The following diagram shows the timing of unsynchronized transfer. DST_B for the DMA channel is asserted by the DMA controller to indicate that data transfer is required. At the same time, the I/O address is driven by the DMA controller. The first transfer is an example of a memory to I/O transfer, data is present at the MDT in the following clock cycle and the I/O device must be able to captured the data at this cycle. The second transfer is an example of an I/O to memory transfer. The I/O device must place the valid data at M1IDT in the following clock cycle. The direction of the transfer is implied because each DMA channel is dedicated for one direction transfer only.

Since unsynchronized transfer can be used to transfer a block of data, DST_B can be asserted again to transfer another data. It is guaranteed that DST_B is de-asserted for at least two clock cycles before it can be asserted again.



This is a partial data sheet of the EP660 DMA controller.

For the complete data sheet or additional information about the operations of the EP660 DMA controller, please contact Eureka Technology Inc. at 1 650 960 3800 or send email to info@eurekatech.com