Multi-Channel DMA Controller

User's Guide

February 2006
ipug11_04.0

Introduction

The Multi-Channel Direct Memory Access (MCDMA) Controller is designed to improve microprocessor system performance by allowing external devices to transfer information directly from the system memory and vice versa. Memory-to-memory transfer capability is also supported.

The MCDMA Controller core supports two modes of operation: 8237 and non-8237 modes. When the 8237 mode is selected, the core is functionally compatible with the Intel 8237A DMA Controller device with a few variations. These variations are listed in the Compatibility Differences with the 8237 Intel Device section of this document. The 8237 and non-8237 modes are detailed later in this document to provide a clearer description of each mode.

Differences Between 8237 Mode and Non-8237 Mode MCDMA

While the 8237 and non-8237 modes share some commonality, they also have differences. Table 1 shows the differences between the two modes.

Table 1. Feature Differences Between the 8237 and Non-8237 Modes

Feature	8237 Mode	Non-8237 Mode
Multiple independent channels	4	1-16
Parameterized address bus	Fixed 16 bits	16, 24 or 32 bits
Parameterized data bus	Fixed 8 bits	8, 16, 32 or 64 bits
Parameterized word count register	Fixed 16 bits	8, 16, 24 or 32 bits
Auto-initialization	Supported	Supported
Compressed timing	Supported	Not supported
Cascade mode	Not supported	Not supported
DMA transfer configuration for each channel	Not supported	Supported
Priority request mode	Rotating/fixed priority mode	Fixed priority mode
DMA request active state	High/low	High
Software reset	Supported	Not supported

Compatibility Differences with the 8237 Intel Device

When the MCDMA core is configured for the 8237 mode, it differs from the Intel 8237A core in the following ways:

- The bi-directional ports are split into separate input and output ports.
- MCDMA does not support the cascade mode of operation.
- The latch that holds the upper byte of the address is internal and the address strobe signal ADSTB is not generated.

The slave's write cycle in the MCDMA core is synchronous.

Features

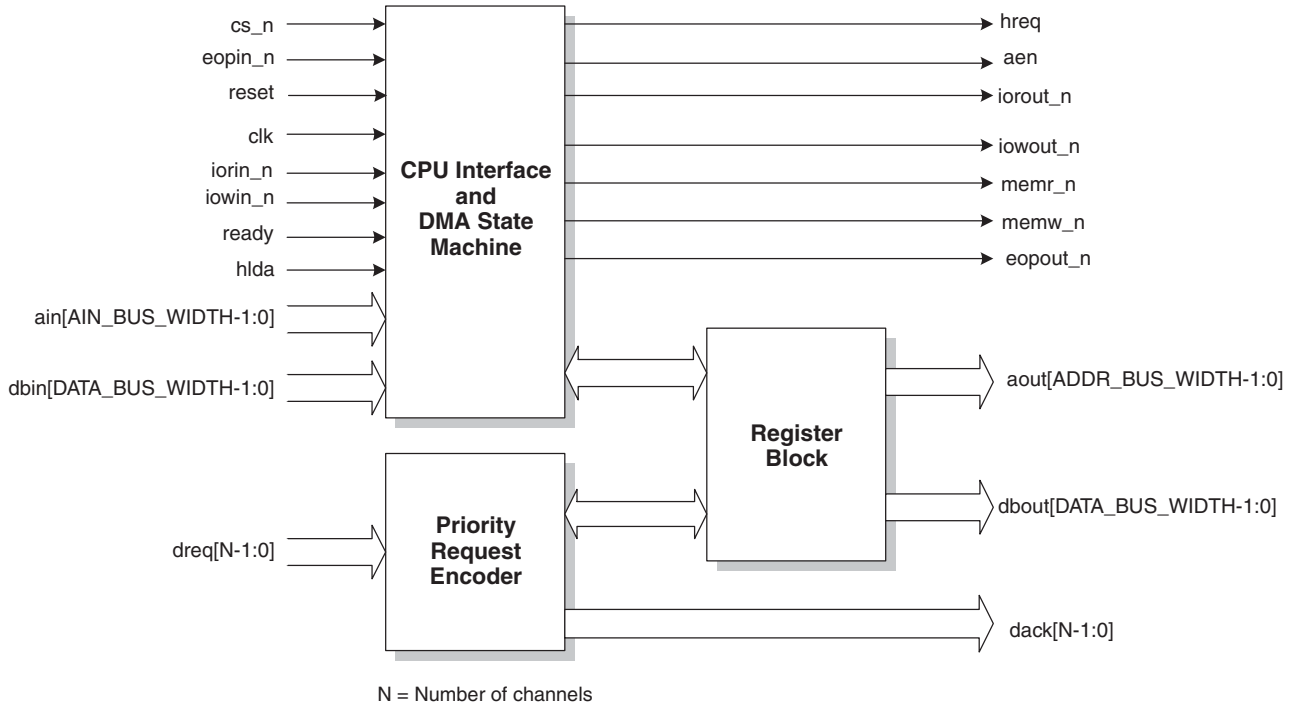
- Selectable 8237 mode
- Configurable up to 16 independent DMA channels for non-8237 mode
- Configurable data width of 8-, 16-, 32- or 64-bits for non-8237 mode
- Configurable address width of 16-, 24- or 32-bits for non-8237 mode
- Configurable Word Count register width for non-8237 mode
- Independent auto-initialization of all channels
- Memory-to-memory transfers on single, block, and demand transfer mode
- Memory block initialization

- Software DMA requests

Block Diagram

Figure 1 shows the block diagram of this core.

Figure 1. Block Diagram of MCDMA Core



Functional Description

The MCDMA contains three basic blocks of control logic: CPU Interface (Data and Control Blocks), the DMA State Machine, and the Priority Request Encoder

CPU Interface Control

This explanation applies mainly to the non-8237 mode because most of the programmability lies in this mode. However, the concepts are also applicable to the 8237 mode.

The CPU Interface Control block first decodes the `ain` bus. It then generates the enable signals to the selected registers or to a subset of the selected registers when byte enables are present during the write cycle. When the registers are read, it provides a select signal to the multiplexer that routes the appropriate register contents onto the data bus.

CPU Interface Data

The CPU Interface Data block contains all the configuration registers. It includes all the routing logic required to transfer either the selected register's contents (during the register read cycle) or the temporary register contents (during the memory write cycle of a memory-to-memory transfer).

DMA Finite State Machine

The DMA FSM (Finite State Machine) module initiates data transfers and generates control signals for various transfer modes. It also generates the address and address-enable signals (`aen`). The FSM exchanges signals with the CPU interface block and priority encoder block. The state machine in the 8237 mode is similar to the non-8237 mode except a few additional FSM branches in 8237 mode that incorporate:

- Illegal I/O to memory transfer mode bits
- Compressed timing mode

FSM Operation

When a software or hardware request is received and is found to be valid (having passed the polarity, mask and mode checks), the DMA FSM in SI (Idle) state transmits a request signal, `hreq` to the CPU and transitions to S0 and waits for the `hlda` signal. If the request drops (`dreq` is de-asserted) or the mode register of the request in hand is in cascade mode (unsupported), the FSM returns to SI (idle). Otherwise, the FSM remains in S0. Once the request is acknowledged by the assertion of `hlda` signal, the FSM transitions to S1/S11 (S1 and S11 are synonymous).

The FSM also determines the transfer type based on the Command and Mode register that is received from the CPU interface. If memory-to-memory transfer is enabled, the FSM transitions through states S12, S13, S14, S21, S22, S23 and S24.

If the next transfer should continue for the same request, the path from S11 to S24 is repeated. If memory-to-I/O or I/O-to-memory is enabled, the FSM goes through states S2, S3 (eliminated in 8237's compressed timing mode), and S4. If the transfer continues, the state machine repeats the loop from S2 through S4. This goes on until the Word Count register gets an overflow or a termination from an external input occurs. External inputs that can terminate a transfer includes an `eopin_n` or `hlda` drop or a request drop during demand transfer. (Note: In case of a non-demand transfer, the request can be dropped after `dack` is received.). The `eopout_n` signal is generated on the falling clock edge of S4 or S24 by the end of the transfer. All transfer read/write signals are generated on the falling edge of clock.

In the state machine described in Figure 2, "input signals" refer to the signals that the state machine samples. These signals affect the state machine's transition logic. "Output signals" refer to signals that will be asserted or de-asserted (transition) while in that state.

Figure 2. MCDMA Finite State Machine for 8237 and Non-8237 Modes

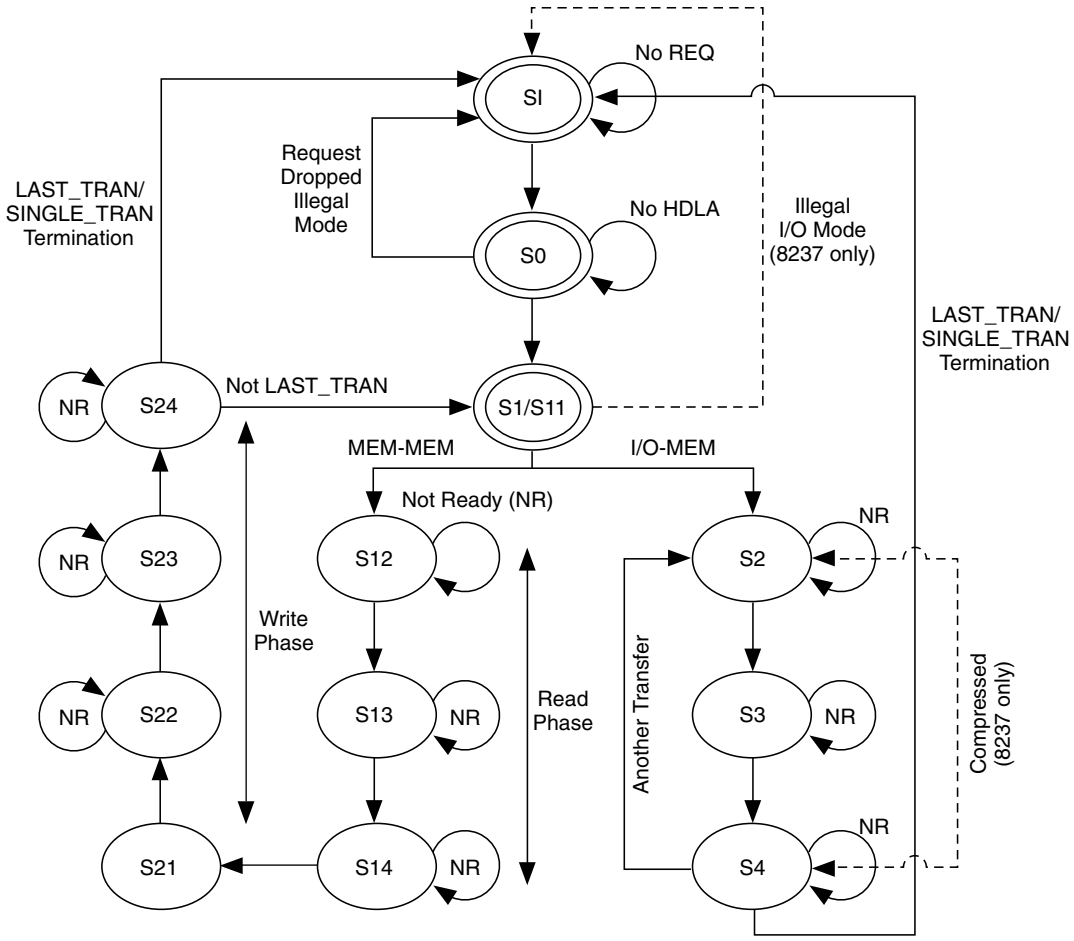


Table 2. State Descriptions

State	Description
Idle State - SI	Upon reset, the state machine enters the idle state, SI. The CPU can program the core's internal registers while it is in this state. The device stays in this state until an unmasked DMA request is detected; at which point the state machine asserts the hreq signal then transitions to state S0. While in state SI, all the outputs of the state machine are in their inactive states. **Input Signals:** hardware reset, software reset (only for 8237 mode), unmasked `dreq` signal **Asserted Output Signals:** `hreq` **Possible State Transitions:** SI, S0
Acquire Bus State - S0	The device stays in this state until the `hlda` signal from the CPU is sampled asserted. The internal registers can still be programmed while in this state. Once the state machine samples and asserts the `hlda` signal, it transitions to the state S1 for regular I/O-to-memory or memory-to-I/O DMA transfers. For memory-to-memory transfers, the state machine transitions to state S11. The criteria for detecting a memory-to-memory transfer are different in the 8237 and non-8237 modes. **8237 Mode:** In this mode, a memory-to-memory transfer is detected if the memory-transfer enable bit in the Command register is set and the `dreq[0]` signal is asserted. The `dackout` signal generated by the priority encoder is used to check if Channel 0 has the highest priority at that time. The DMA priority scheme will be described more in the priority request encoder section. **Non-8237 Mode:** In this mode, memory-to-memory transfer is detected if bit zero of the current channel's mode register is set. If the current channel's `dreq` signal is de-asserted in this state and no other requests are pending, the state machine transitions to state SI. If other requests are pending or the `dreq` signal remains asserted, the state transitions to either S1 or S11. **Input Signals:** `hlda`, `command[0]` or `mode[0]`, `dackout` **Asserted Output Signals:** `hreq` **Possible State Transitions:** SI, S1, S11
Memory-to-Memory Read Transfer State One - S11	This is the first state of the memory-to-memory transfer. The absence of the `dack` signal characterizes this transfer. The aen signal is asserted. In the 8237 mode, the address from Channel 0 of the current address register is placed on the address bus. In the non-8237 mode, the contents of the source address register are placed on the address bus. The `memr_n` and `memw_n` signals are de-asserted. During each of the eight states of the memory-to-memory transfer, the state machine responds to external `eopin_n` signal and stops the DMA transfer service as soon as the current cycle is completed. The state machine transitions to state S12. **Input Signals:** `eopin_n` **Asserted Output Signals:** `aen`, `address` **Possible State Transitions:** S12
Memory-to-Memory Read Transfer State Two - S12	This is the second state of memory-to-memory transfer. The `memr_n` signal is asserted. The state transitions to state S13. **Input Signals:** `eopin_n` **Asserted Output Signals:** `memr_n, address` **Possible State Transitions:** S13

Table 2. State Descriptions (Continued)

State	Description
Memory-to-Memory Read Transfer State Three - S13	This is the third state of the memory-to-memory transfer. The state machine samples the ready signal and stays in this state as long as it is asserted. The machine transitions to state S14 when the ready signal is de-asserted. **Input Signals:** eopin_n, ready **Asserted Output Signals:** memr_n, address **Possible State Transitions:** S13, S14
Memory-to-Memory Read Transfer State Four - S14	This is the fourth stage of the memory-to-memory transfer. The state machine de-asserts memr_n signal and asserts an enable signal to flop the incoming data into the temporary register. The state machine transitions to state S21, which is the first state of the memory-to-memory write transfer stage. **Input Signals:** eopin_n **Asserted Output Signals:** none **Possible State Transitions:** S21
Memory-to-Memory Write Transfer State One - S21	This is the fifth stage of the memory-to-memory transfer mode. In the 8237 mode, the content of the current address register on Channel 1 is put on the address bus. In the non-8237 mode, the content of the destination address register on the channel being serviced is put on the address bus. The memr_n and memw_n signals are de-asserted. The state machine transitions to state S22. **Input Signals:** eopin_n **Asserted Output Signals:** address **Possible State Transitions:** S22
Memory-to-Memory Write Transfer State Two - S22	This is the fifth state of memory-to-memory transfer. The state transitions to state S23. **Input Signals:** eopin_n **Asserted Output Signals:** address **Possible State Transitions:** S23
Memory-to-Memory Write Transfer State Three - S23	This is the seventh state of the memory-to-memory transfer. The memw_n signal is asserted, and the content of the temporary register is placed on the data bus. The state machine samples the ready signal and stays in this state as long as it is asserted. Then the machine transitions to state S24. **Input Signals:** eopin_n, ready **Output Signals:** memw_n **Possible State Transitions:** S23, S24

Table 2. State Descriptions (Continued)

State	Description
Memory-to-Memory Write transfer state four - S24	This is the eighth and final stage of the memory-to-memory transfer. The state machine de-asserts the memw_n signal. In the 8237 mode, Channel 1's current word register is decremented. In the non-8237 mode, the word count register of the channel being serviced is decremented. If the counter rolls over from 0xFFFF to 0x0000, the eopout_n signal is asserted and the state machine transitions to state SI. Otherwise, the state machine transitions to state S11 and starts a new memory-to-memory transfer. **Input Signals:** eopin_n **Asserted Output Signals:** eopout_n (in case the counter rolls over) **Possible State Transitions:** SI, S11
Active DMA State One - S1	This is the first state of DMA transfer. The aen signal is asserted in this state while a valid address is placed on the address bus. If dreq continues to be asserted, the state machine transitions to state S2. If dreq is de-asserted, the state machine transitions to state SI. DMA requests must be held active until the dack signal is asserted. **Input Signals:** dreq **Asserted Output Signals:** aen, address **Possible State Transitions:** S2, SI
Active DMA state two - S2	This is the second state of the DMA transfer. The dack signal is asserted. The dreq signal does not need to be held asserted after this state if block or single transfer mode is selected. memr_n or iorout_n is asserted depending on the direction of the transfer. **8237 Mode:** memw_n or iowout_n is asserted if the extended write option is selected in the command register. The state machine will skip state S3 and transition to state S4 if the compressed timing option is selected. This will result both read and write pulses being asserted for just a single cycle. **Non-8237 Mode:** memw_n or iowout_n is asserted, and the state machine transitions to state S3. While in state S2, S3, or S4, the state machine will terminate a block or demand transfer if the eopin_n signal is sampled asserted. **Input Signals:** eopin_n, dreq **Asserted Output Signals:** dack, memw_n, memr_n, iorout_n, iowout_n **Possible State Transitions:** S3, S4
Active DMA state three - S3	This is the third state of the DMA transfer. In the 8237 mode, the memw_n or iowout_n signal is asserted if extended write is not selected. The state machine is sensitive to the eopin_n and dreq signals for demand transfers. The state machine transitions to state S4, when ready signal is sampled de-asserted. The machine stays in state S3 as long as ready is sampled asserted. **Input Signals:** eopin_n, dreq, ready **Asserted Output Signals:** memw_n, iowout_n **Possible State Transitions:** S3, S4

Table 2. State Descriptions (Continued)

State	Description
Active DMA state four - S4	This is the last stage of the DMA transfer. The memw_n or iowout_n signal is de-asserted, depends on the operation (I/O-to-memory or memory-to-I/O). The same thing happens to the memr_n or iorout_n signal for only one of them being de-asserted. The eopout_n signal is asserted if the current word register rolls over from 0xFFFF to 0x0000. This causes the state machine to transition to state SI. If block or demand transfer mode is selected, the counter has not rolled over, and DMA hasn't satisfied the transfer complete-conditions, the state machine transitions to a state where DMA transfers will continue. This next state depends upon the mode of operation. **8237 Mode:** The state machine transitions to state S2 as long as the higher order address remains the same. If the higher order address for the new transfer changes, the state machine transitions to state S1 to enable the external latch to update its latched value. **Non-8237 Mode:** The state machine transitions to state S2. **Input Signals:** eopin_n **Asserted Output signals:** eopout_n **Possible State Transitions:** SI, S1, S2
Compressed Timing Mode	This feature is only available in the 8237 mode MCDMA. The purpose of this mode is to allow MCDMA to achieve greater throughput by compressing the memory-to-I/O (or I/O-to-memory) transfer time to two clock cycles. In this mode, state S3 is removed from the state machine. This causes the read pulse-width to equal the write pulse-width. Thus, the transfer only has state S2 to change the address and state S4 to perform a read/write operation.

Priority Request Encoder

This block prioritizes the DMA request and asserts the dack signal for the winning request. In the 8237 mode, the arbitrating scheme is user programmable and available in either a fixed priority or rotating priority mode. In non-8237 operation, the arbitrating scheme is restricted to the fixed priority mode.

In the fixed priority mode, dreq[0] has the highest priority and dreq[n] has the lowest priority. In the 8237 mode, n (channel number) is fixed to 3 while in the non-8237 mode, n is user selectable (up to 16).

The rotating priority mode assigns the lowest priority to the channel that has been serviced most recently. This mode ensures that all devices will be serviced fairly and prevents any one channel from monopolizing the system. The maximum wait time for a channel to be serviced is the time taken to service all the other channels.

The main function of this block is to generate the dack[x] signal. Each channel has an associated priority register that indicates the channel's priority. In the fixed priority mode, the values in these registers will never change, while in the rotating priority mode, their values change every time a channel is serviced.

DMA Operation

In the non-8237 mode, each channel can be programmed to perform DMA operation between memory locations or from I/O-to-memory. Each channel has a dedicated source address register that holds the address of the targeted read memory location and a destination address register, which points to the targeted write memory location.

Memory locations are addressable, but I/O locations are not addressable. The source address register in the non-8237 mode holds the memory location address during DMA transfers between an I/O device and memory.

The functionality of the core in the non-8237 mode is very similar to that of the 8237 mode. However, the two modes have totally different sets of programmable control registers. This increases the programmability features in the non-8237 mode. Some of the features available only in the non-8237 mode are:

- Multiple channels
- Configurable channels for DMA transfers between memory-and-memory or I/O-and-memory
- Parameterized address output and data bus width
- Parameterized word-count register

The microprocessor programs a number of registers to ensure the DMA controller functions properly. The internal registers are accessed when the cs_n signal is asserted and the address of the register is placed on the ain bus. When the iowin_n signal is low, the registers are over written with the data on dbin bus. When the iorin_n signal is asserted, the registers are read and their contents are placed on the dbout bus. The least significant eight bits of the data bus can access the internal registers irrespective of the data bus width.

To prevent erroneous behavior, the DMA registers should be programmed only when the controller is in the Idle State (SI) or before it receives the hlda signal from the microprocessor. Not adhering to these rules will cause the DMA controller to function in a non-deterministic way. The registers visible to the microprocessor are different for the 8237 and non-8237 modes.

The MCDMA controller core is a fully synchronous machine that runs off the positive edge of the clock. However, the DMA request signals, dreq[N-1:0], are asynchronous with respect to the clock. These signals have to be synchronized within the core.

DMA Initialization

Once the Command and Mode registers are programmed and the controller is enabled, DMA transfers can be initiated either by asserting the unmasked channel's dreq signal or by requesting to DMA by programming the request register. The internal registers of the core are accessible during the idle state (SI) when no channel is requesting service and no DMA transfers are in progress.

The core operates in two cycles: Idle and Active. The core remains in the idle cycle as long as it is not performing any DMA transfers and none of the unmasked channels have a pending request. In this state, the microprocessor can program the device. Once an unmasked channel requests DMA service, the device asserts the hreq signal to take control of the bus and enters the first active cycle state S0. The device can still be programmed in the S0 state until it receives the hlda signal from the bus arbiter. For DMA transfers between memory and I/O, the active cycles go from states S1 through S4. When memory-to-memory DMA transfers are performed, the active cycles go through states S11, S12, S13, S14 for the memory read operation and states S21, S22, S23, S24 for the memory write operation. Wait states are introduced whenever the slave device is not ready for the transfer.

MCDMA Transfer Modes

When the MCDMA is in the active cycle, the DMA service takes place in one of the following three modes:

- **Single Transfer Mode:** In single transfer mode, the device is programmed to make one transfer only. Following each transfer, the Word Count decrements and the address decrements or increment, depending on what is selected in the Mode register. To be recognized, the dreq signal must be held active until dack becomes active. If dreq is held active throughout the single transfer, hreq goes inactive and releases the bus to the system. After the transfer, hreq will go active again. When the controller receives a new hlda, another single transfer will be performed.
- **Block Transfer Mode:** In block transfer mode, dreq signals the device to continue making transfers during the service until a terminal count is encountered (generation of eopout_n).This occurs when the word count goes to 0xFFFF, or an external End of Process (eopin_n) is encountered. The dreq signal must be held active until dack becomes active. Auto-initialization occurs at the end of the service if the channel has been programmed for it.

- **Demand Transfer Mode:** In demand transfer mode, the device is programmed to continue making transfers until a Terminal Count or external `eopin_n` is encountered or until `dreq` goes inactive. Thus, transfers continue until the I/O device has exhausted its data capacity. After the I/O device has had a chance to catch up, the DMA service is reestablished by `dreq`.

Parameter Descriptions

Table 3 lists the parameters used to configure the MCDMA core. The values of these parameters must be set prior to functional verification. The parameters in parenthesis can be configured using the IPexpress™ software tool, included with the ispLEVER® design tools.

Table 3. MCDMA Parameters

Parameter	Description	Supported Values
`DMA Mode (MODE_8237)`	Defines the DMA mode. If it is TRUE, the DMA will be in 8237 mode, otherwise it will be in non-8237 mode.	TRUE/FALSE
`Number of Channel (NUM_CHANNELS)`	Sets the number of channels. In 8237 mode it is fixed to 4 channels. In non 8237 it can be set for 1 to 16 channels	4 (8237) 1-16 (non 8237)
`Data Width (DATA_BUS_WIDTH)`	Sets the size of data buses and the temporary register.	8 (8237) 8/16/32/64 (non 8237)
`Address Width (ADDR_BUS_WIDTH)`	Sets the size of DMA output address. In the 8237 mode, it sets the size of the current and base address register. In the non-8237 mode, it sets the size of the source address register	16 (8237) 16/24/32 (non 8237)
`Word Count Width (WORD_COUNT_WIDTH)`	Sets the size of the Word Count Register.	16 (8237) 8/16/24/32 (non 8237)
`Internal Address Width (AIN_BUS_WIDTH)`	Value is set automatically based in the number of channels parameter (NUM_CHANNELS or N).	4 (8237) 3 when N = 1 (non 8237) 4 when N = 2 (non 8237) 5 when 3 ≤ N ≤ 4 (non 8237) 6 when 5 ≤ N ≤ 8 (non 8237) 7 when 9 ≤ N ≤ 16 (non 8237)

Signal Descriptions

Table shows the input and output ports of the MCDMA core that apply for both 8237 and non-8237 modes.

Table 4. Signal Definitions of the MCDMA Controller

Port Name	Type	Active State	Description
clk	Input	Rising Edge	**Clock**. This signal controls and synchronizes the operations of the MCDMA.
cs_n	Input	Low	**Chip Select**. This is an active low signal used to select the MCDMA.
reset	Input	High	**Reset**. This is an active high signal that clears the internal registers. After reset, the device is placed in the Idle state and the DMA requests are masked.
ready	Input	High	**Ready**. This is an active high signal used to extend the memory read and write pulses from the MCDMA. This is most of often used to accommodate slow memories.
hlda	Input	High	**Hold Acknowledge**. This active high signal generated by the CPU indicates the CPU has relinquished control of the system buses.
eopin_n	Input	Low	**End Of Process Input**. This active low input permits the external termination of the current DMA service.
iorin_n	Input	Low	**I/O Read Input**. This is an active low signal when asserted along with cs_n. Thus, it permits the CPU to read the internal registers of MCDMA.
iowin_n	Input	Low	**I/O Write Input**. This is an active low signal. When asserted along with cs_n, it permits the CPU to write into the internal registers of the MCDMA.
ain [AIN_BUS_WIDTH-1:0]	Input	N/A	**Address**. This signal selects one of the internal registers. In the 8237 mode, ain is 4 bits wide. In the non-8237 mode, the bus width depends on the number of channels selected.
dbin [DATA_BUS_WIDTH-1:0]	Input	N/A	**Data Bus Input**. The CPU writes to the internal registers through this data bus.
dreq[N-1:0]	Input	High/Low (8237) High (Non-8237)	**DMA Request**. These programmable parity signals are asynchronous signals generated by peripherals requesting DMA service. A device reset initializes dreq to active high. In 8237 mode, these parity signals are programmable to be active high or low. In non-8237, these signals are always active high.
hreq	Output	High	**Hold Request**. This is an active high signal sent to the CPU to request control over the system bus.
eopout_n	Output	Low	**End of Process Out**. This active low signal indicates normal termination of a DMA service.
iorout_n	Output	Low	**I/O Read Output**. This active low signal is used to access data from a peripheral during a DMA Write transfer.
dbout [DATA_BUS_WIDTH-1:0]	Output	N/A	**Data Bus Output**. This bus contains the value of the internal register when read by the CPU. In the write-to-memory phase of the memory-to-memory DMA operation, the dbout data bus transmits the data from the temporary register.
iowout_n	Output	Low	**I/O Write Output**. This active low signal is used to load data to a peripheral during a DMA Read transfer.
memw_n	Output	Low	**Memory Write**. This active low signal is used to indicate that data is being written to the selected memory location during a DMA Write or a memory-to-memory transfer.
memr_n	Output	Low	**Memory Read**. This active low signal is used to indicate that data is being read from the selected memory location during a DMA Read or a memory-to-memory transfer.

Table 4. Signal Definitions of the MCDMA Controller (Continued)

Port Name	Type	Active State	Description
aen	Output	High	**Address Enable**. This active high signal enables the 8-bit latch that contains the upper 8 address bits onto the system address bus.
aout [ADDR_BUS_WIDTH-1:0]	Output	N/A	**Address Output**. These lines are enabled only during active DMA transfer and contain the memory address.
dack[N-1:0]	Output	High/Low	**DMA Acknowledge**. This signal is used to notify the requesting peripheral that it has been granted a DMA cycle. The polarity of this signal is programmable. A device reset initializes all dack signals to active low.

Note: N = number of channels

Timing Specifications

Figure 3 illustrates the waveform for a write operation into one of the internal registers in the MCDMA core. Clock edges 1, 2 and 3 indicate the point where the data is written into the registers. Clock edges 2 and 3 indicate a back-to-back write operation into the same register. The iorin_n signal is held high during the entire write operation.

Figure 3. Processor Write Timing Waveform

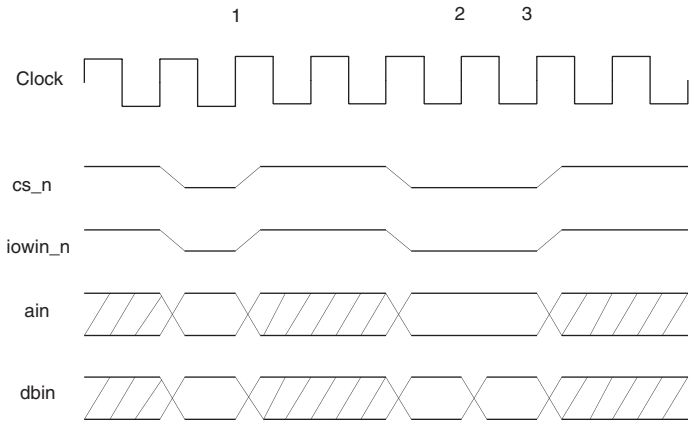


Figure 4 shows the processor read timing waveform. Data is available on the following clock edge after cs_n and iorin_n are asserted. Clock edges 1, 2, and 3 indicate the edges on which the data is valid.

Figure 4. Processor Read Timing Waveform

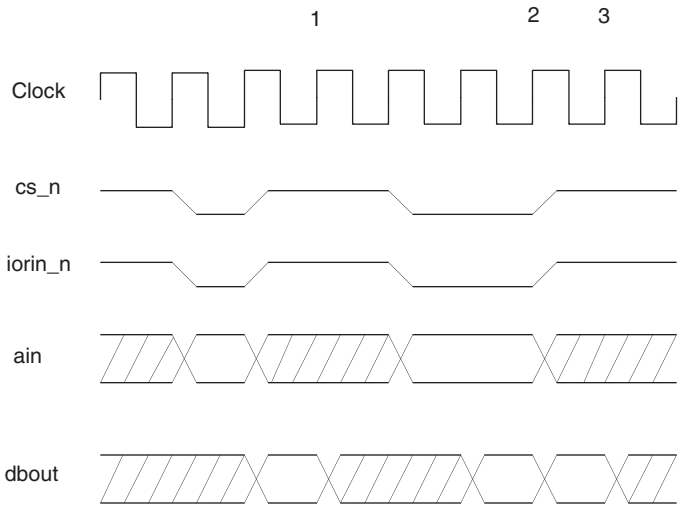
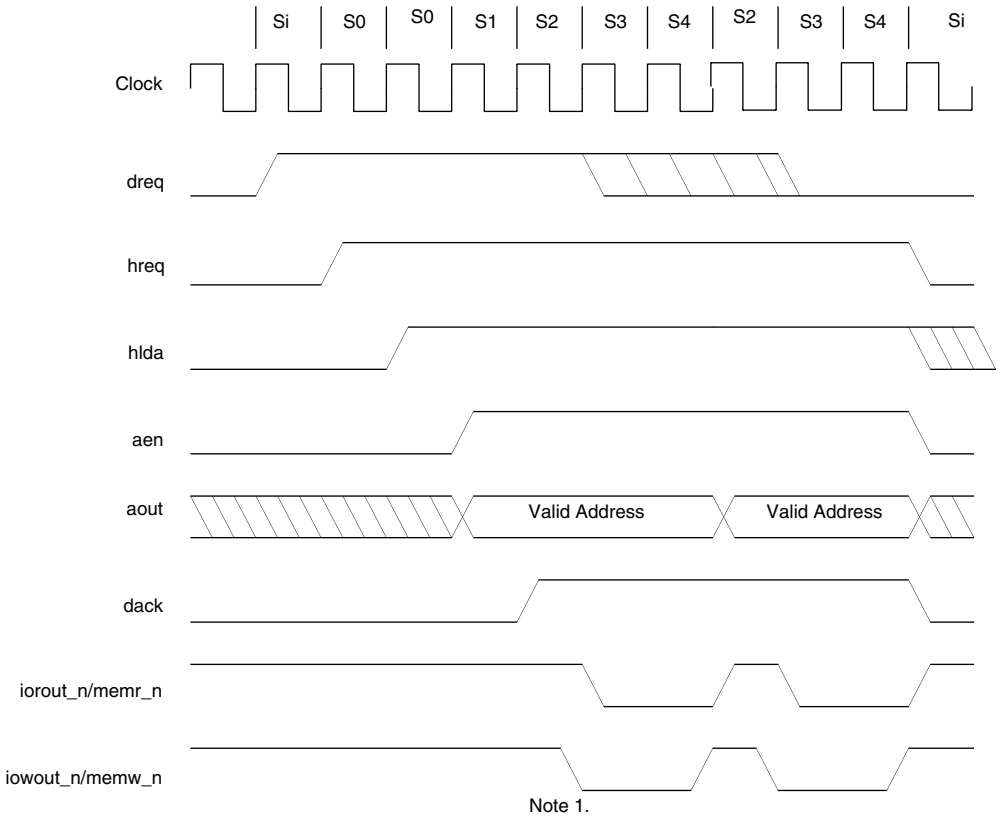


Figure 5 shows the timing waveform for two words DMA transfer.

Figure 5. Two Word DMA Transfer Timing Waveform



Note 1. This timing diagram demonstrates the extended write operation. In the 8237 mode, when normal write operation is selected, `iowout_n` or the `memw_n` is asserted one clock cycle later.

If compressed timing is selected, the state S3 is bypassed, making the read and write pulses of equal width. This is only applicable in 8237 mode.

The `iowout_n` and `memw_n` signals are generated off the falling clock edge. This ensures the address is held at least for half a cycle after the rising edge of the write signal.

Register Descriptions

The 8237 and non-8237 modes of the MCDMA Controller have different types and number of internal registers. The 8237 mode has ten types of internal registers that are visible to the microprocessor while the non-8237 mode has seven types of internal registers that are visible to the microprocessor.

8237 Mode Internal Registers

Table 5. Internal Registers in 8237 Mode

Name	Size in Bits	Number of Registers
Base Address Registers	16	4
Base Word Count Registers	16	4
Current Address Registers	16	4
Current Word Count Registers	16	4
Command Register	8	1
Status Register	8	1
Temporary Register	8	1
Mode Register	8	4
Mask register	8	1
Request Register	4	1

Current Address Register
This register is only available in the 8237 mode. Each of the four channels has a 16 -bit wide Current Address Register that holds the value of the address used during DMA transfers. The address is automatically incremented or decremented after each transfer. The microprocessor loads the Current Address Register simultaneously with the Base Address Register. If Auto-Initialization is enabled, the MCDMA reloads the base address value at the end of the DMA cycle. This register has to be written in two consecutive cycles after clearing the byte pointer.

Current Word Count Register
This register is only available in the 8237 mode. Each channel has a 16-bit Current Word Count register that determines the number of transfers to be performed. The actual number of transfers is one more than the value programmed into this register. The Current Word Count is decremented after each transfer. If Auto-Initialization is enabled, the value in the Base Word Count register is reloaded at the end of the DMA service. When the value in the register goes from zero to 0xFFFF, a terminal count (`eopout_n`) signal is generated. If Auto-Initialization is not enabled, this register has a count of 0xFFFF at the end of DMA service.

Base Address Register
This register is only available in the 8237 mode. Each channel has a 16-bit Base Address Register. This register stores the starting address for the transfer. In the idle state or program condition, the microprocessor simultaneously writes to the Base Address register and to the Current Address register. The microprocessor cannot read this register.

Base Word Count Register
This register is available only in the 8237 mode. Each channel has a 16-bit Base Word Count register that stores the starting word count for DMA transfers. During Auto-Initialization, this value is used to restore the current word count register. The microprocessor cannot read this register.

Command Register
This register controls the operation of the core. In the 8237 mode, this register is 8 bits wide. In non-8237 mode, it is 4 bits wide. When DMA is in state idle, the microprocessor programs this register. A reset or master clear clears the register. Table 6 lists the function of this register.

Mode Register
Each channel has a 6-bit wide register. During a write operation by the microprocessor when MCDMA is in idle state, the least two significant bits (bit 0 and 1) of the data bus determine which channel mode register is being accessed. A reset or a master clear clears the mode registers. Table 7 lists the format of a mode register in the 8237 mode.

Mask Register
This register is only visible in the 8237 mode. Each channel has a bit associated with it that is used to mask a hardware DMA request (disable the incoming `dreq`). All four bits of this register can be accessed at once, or the CPU can program each of the bits separately. Each mask bit is set when its associated channel produces an `eopout_n` signal. A reset or master clear sets all four bits and masks all the channels. Table 8 and Table 9 list the mask register format for the 8237 mode.

Request Register
This register is only visible in 8237 mode. The request register allows software DMA requests. The values in the mask register mask the hardware request (`dreq`). Software requests generated from the request register are non-maskable. Individual bits of this register can be accessed with the channel number supplied on the two least significant bits of the data bus. A reset or master clear clears this register. The channel must be in block mode in order to make a software request. Table 10 lists the request register format in 8237 Mode.

Status Register
This register is only available in the 8237 mode. The microprocessor can read the status register, which contains information about the status of the device. This information includes which of the channels have completed their DMA service and which channels have a DMA request pending. The Status Register is reset upon a hardware reset or a master clear. Bits 0 through 3, which indicate which channel has reached Terminal Count, are cleared every time the Status register is read. Bits 4 through 7 are set when their corresponding channel is requesting service. Table 11 shows the status register format.

Temporary Register
This register holds data during memory-to-memory transfers. A reset or master clear command clears this register. The microprocessor can read this register in the 8237 mode while the MCDMA is in the idle state. This register yields the last data transferred during the most recent memory-to-memory transfer.

Table 6. Command Register - 8237 Mode

Bit		Description
0		Memory-to-memory disable Memory-to-memory enable
1	 X	Channel 0 address hold disable Channel 0 address hold enable if bit0 = 0
2		Controller enable Controller disable
3	 X	Normal timing Compressed timing if bit0 = 1
4		Fixed Priority Rotating Priority
5	 X	Late Write Extended Write if bit3 = 1
6		`dreq` active high `dreg` active low
7		`dack` active low `dack` active high

Table 7. Mode Register - 8237 Mode

Bit	Description
1:0	00 Channel 0 select 01 Channel 1 select 10 Channel 2 select 11 Channel 3 select
3:2	00 Verify transfer 01 Write transfer 10 Read transfer 11 Illegal xx If bits 6 & 7 are 11
4	0 Auto-initialization disable 1 Auto-initialization enable
5	0 Address increment 1 Address decrement
7:6	10 Demand mode select 01 Single mode select 10 Block mode select 11 Cascade mode (unsupported)

Table 8. Mask Register: Access All Bits - 8237 Mode

Bit	Description
0	0 Channel 0 unmasked 1 Channel 0 masked
1	0 Channel 1 unmasked 1 Channel 1 masked
2	0 Channel 2 unmasked 1 Channel 2 masked
3	0 Channel 3 unmasked 1 Channel 3 masked

Table 9. Mask Register: Access One Bit - 8237 Mode

Bit	Description
1:0	00 Select channel 0 mask bit 01 Select channel 1 mask bit 10 Select channel 2 mask bit 11 Select channel 3 mask bit
2	0 Clear mask bit 1 Set mask bit
7:3	Don't care

Table 10. Request Register: Access One Bit - 8237 Mode

Bit	Description
1:0	00 Select channel 0 request bit 01 Select channel 1 request bit 10 Select channel 2 request bit 11 Select channel 3 request bit
2	0 Clear request bit 1 Set request bit
7:3	Don't care

Table 11. Status Register: Access One Bit – 8237 Mode

Bit	Description
0	Channel 0 terminal count
1	Channel 1 terminal count
2	Channel 2 terminal count
3	Channel 3 terminal count
4	Channel 0 request
5	Channel 1 request
6	Channel 2 request
7	Channel 3 request

Table 12. Non-8237 Internal Registers

Name	Size in Bits	Number of Registers
Source Address Register	16, 24 or 32[1]	N[4]
Word Count Register	8, 16, 24 or 32[2]	N
Destination Address Register	16, 24 or 32[1]	N
Command Register	4	1
Temporary Register	8,16,32 or 64[3]	1
Mode Register	8	N
Channel Control Register	3	N

1. Based on the width of the address bus selected
2. Based on the width of the word count register selected
3. Based on the width of the data bus selected
4. N = Number of channels selected

Source Address Register

This register is only available in the non-8237 mode. Each channel has a Source Address Register whose width matches with the address bus width. This register stores the value of the source or memory address used during DMA transfers. The address is automatically incremented or decremented by 1, 2, or 4 after each transfer, depending on the respective data bus width of 8, 16 or 32 bits. This register has to be written in consecutive cycles after clearing the byte pointer. The number of cycles taken to access this register depends on the size of the address bus. During a DMA transfer between an I/O location and memory, this register holds the address of the memory location. During a memory-to-memory transfer, this register stores the address of the location that is read from. The user must always program the register with the address that is aligned with the width of the data bus.

Destination Address Register

This register is only available in the non-8237 mode. Each channel has a Destination Address Register whose width matches with the address bus width. The address is automatically incremented or decremented by 1, 2, or 4 after each transfer depending on the respective data bus width of 8, 16 or 32 bits. This register has to be written in consecutive cycles after clearing the byte pointer. The number of cycles taken to access this register depends on the size of the address bus. During memory-to-memory transfers, this register stores the address of the memory location that is written into. This register is not used during DMA transfers between memory and I/O. The user must always program the register with the address that is aligned with the width of the data bus.

Word Count Register

This register is only available in non-8237 mode, and its width is configurable. This register determines the number of transfers to be performed. The actual number of transfers is one more than the value programmed into this register. The current word count is decremented after each transfer. When the value in the register goes from zero to 0xFFFF, a Terminal Count (`eopout_n`) signal is generated. At the end of a DMA transfer, this register has a value of 0xFFFF if auto-initialization is not enabled for the channel.

Command Register
This register controls the operation of the core. This register is 4 bits wide in the non-8237 mode. A reset or master clear clears the register. Table 13 lists the functions of this register for the non-8237 mode.

Mode Register (Non- 8237 Mode)
Each channel has an 8-bit mode register. Table 14 lists the format of the mode register in the non-8237 mode. Programming the corresponding increment and decrement bits to the same value can hold the source or destination address constant.

Channel Control Register
This register is visible in the non-8237 mode. Each channel has one channel control register. Table 15 lists the register's format. A reset or a master clear resets the request and sets the mask. This masks the channel's hardware requests. Auto-initialization is disabled upon a reset.

Table 13. Command Register – Non-8237 Mode

Bit	Description
0	0 Controller enable 1 Controller disable
1	Reserved. This bit is always 0
2	Reserved. This bit is always 0
3	0 dack active low 1 dack active high

Table 14. Mode Register – Non-8237 Mode

Bit	Description
0	0 Memory-to-memory disable 1 Memory-to-memory enable
1	0 Write transfer 1 Read transfer X If bit0=1
2	0 Increment Source Address disable 1 Increment Source Address enable
3	0 Decrement Source Address disable 1 Decrement Source Address enable
4	0 Increment Destination Address disable 1 Increment Destination Address enable
5	0 Decrement Destination Address disable 1 Decrement Destination Address enable
7:6	00 Demand mode select 01 Single mode select 10 Block mode select 11 Illegal

Note: Bits 2 and 3 are mutually exclusive. They cannot be enabled at the same time since the address will either increment or decrement. This also applies to Bits 4 and 5.

Table 15. Channel Control Register – Non-8237 Mode

Bit	Description
0	0 Clear Request bit 1 Set Request bit
1	0 Channel unmasked 1 Channel masked
2	0 Auto Initialization disable 1 Auto Initialization enable

Register Address Map

The 8237 and non-8237 modes of the MCDMA Controller decode and use different numbers of `ain` bit input signals. The `ain` signal is used for mapping the register address and decoding software command.

Table 16. Register Address Map and Software Command of 8237 Mode

ain3	ain2	ain1	ain0	Channel	Write iorin_n = 1, iowin_n = 0	Read iorin_n = 0, iowin_n = 1
0	0	0	0	0	Base and current Address reg	Current DMA address reg
0	0	0	1		Base and current Word Count reg	Current Word Count reg
0	0	1	0	1	Base and current Address reg	Current DMA address reg
0	0	1	1		Base and current Word Count reg	Current Word Count reg
0	1	0	0	2	Base and current Address reg	Current DMA address reg
0	1	0	1		Base and current Word Count reg	Current Word Count reg
0	1	1	0	3	Base and current Address reg	Current DMA address reg
0	1	1	1		Base and current Word Count reg	Current Word Count reg
1	0	0	0	X	Command Register	Read Status Register
1	0	0	1	X	Single Request bit command	Illegal
1	0	1	0	X	Single Mask bit command	Illegal
1	0	1	1	X	Mode Register	Illegal
1	1	0	0	X	Clear Byte Pointer command	Illegal
1	1	0	1	X	Master clear command	Read temporary reg
1	1	1	0	X	Clear Mask Register command	Illegal
1	1	1	1	X	Mask register	Illegal

In the 8237 mode, the additional software commands are: Clear Byte Pointer, Master Clear and Clear Mask Register.

Table 17. Register Address Map and Software Command of Non-8237 Mode

ain2	ain1	ain0	Write: iorin_n = 1, iowin_n = 0 Read: iorin_n = 0, iowin_n = 1
0	0	0	Command register
0	0	1	Source Address register
0	1	0	Word Count register
0	1	1	Destination Address register
1	0	0	Mode register
1	0	1	Channel Control register
1	1	0	Master Clear command
1	1	1	Clear Byte Pointer command

Programming the MCDMA Controller

Programming the core to achieve the desired functionality is very similar in both the 8237 and non-8237 modes. The differences lie in the address map and the name of the registers

For the 8237 mode, the steps to program the core are:

- Disable the controller through the command register
- Program the Mode Register
  - Select the channel to be programmed
  - Set the features on the channel to programmed, such as transfer mode, read/write/verify transfer, address increment/decrement, etc.
- Write into the Address Registers and Word Count Register
  - Set the base source address register
  - Set the number of transfer to be performed in the Word Count Registers.
- Enable the controller

For the non-8237 mode, the steps to program the core are:

- Disable the controller
- Set Input ain
  - The first 3 bits of `ain`, `ain[2:0]`, selects which registered to be programmed. The rest of the bits select which channel to be programmed.
- Program the Mode and Channel control registers of all the channels
- Write into the Address registers and Word Count register
- Enable the controller

The core remains in the idle state (SI) as long as DMA transfers are not requested. While in state SI, the `dreq` signals are sampled. When an unmasked DMA request is presented, the core enters the active state. The request can be either a software or hardware request.

Although the internal registers can be programmed in state S0, or before the `hlda` signal is asserted, it is a good practice to program the internal registers only while the core is in the Idle state (SI). The functionality of the controller is not guaranteed to be deterministic if the internal registers are accessed during an active DMA cycle.

Reference Information

- *ispLEVER Software User Manual*, Lattice Semiconductor Corporation
- *8237A High Performance Programmable DMA Controller,* Intel Corporation, September 1993.

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Appendix for ORCA® Series 4 FPGAs

Table 18. Performance and Resource Utilization[1]

Mode	Name of Parameter File	LUTs	ORCA 4 PFUs[2]	Registers	sysMEM™ EBRs	I/O	f_{MAX} (MHz)
8237	dma_mc_o4_2_001.lpc	1258	200	524	N/A	59	58
Non-8237	dma_mc_o4_2_002.lpc	2661	499	1187	N/A	125	66

1. Performance and utilization characteristics are generated using OR4E02-2PBGAM680-DE in Lattice's ispLEVER 3.0 SP1 software. Synthesized using Synplicity® Synplify®7.03. When using this IP core in a different density, package, speed, or grade within the ORCA family, performance may vary.
2. PFU is a standard logic block of some Lattice devices. For more information, check the data sheet of the device.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all configurations of this core in ORCA Series 4 devices is DMA-MC-O4-N2. Table 19 lists the Lattice-specific netlists that are available in the Evaluation Package, which can be downloaded from the Lattice web site at www.latticesemi.com.

Table 19. Core Configuration

Name of Parameter File	Number of Channels	Data Bus Width	Address Bus Width	Word Count Width
8237 Mode				
dma_mc_o4_2_001.lpc	4	8	16	16
Non-8237 Mode				
dma_mc_o4_2_002.lpc	4	32	32	16

Appendix for ispXPGA® FPGAs

Table 20. Performance and Resource Utilization[1]

Mode	Name of Parameter File	LUT4[2]	ispXPGA PFUs[2]	Registers	sysMEM EBRs	I/O	f_{MAX} (MHz)
8237	dma_mc_xp_2_001.lpc	1450	432	562	N/A	58	58
Non-8237	dma_mc_xp_2_002.lpc	3487	1072	1181	N/A	124	66

1. Performance and utilization characteristics are generated using LFX1200B-05F900C in Lattice ispLEVER 3.x software. The evaluation version of this IP core only works on this specific device density, package, and speed grade.
2. PFU is a standard logic block of some Lattice devices. For more information, check the data sheet of the device.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all configurations of this core in ispXPGA devices is DMA-MC-XP-N2. Table 21 lists the Lattice-specific netlists that are available in the Evaluation Package, which can be downloaded from the Lattice web site at www.latticesemi.com.

Table 21. Core Configuration

Name of Parameter File	Number of Channels	Data Bus Width	Address Bus Width	Word Count Width
8237 Mode				
dma_mc_xp_2_001.lpc	4	8	16	16
Non-8237 Mode				
dma_mc_xp_2_002.lpc	4	32	32	16

You can use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

Appendix for LatticeECP™ and LatticeEC™ FPGAs

Table 22. Performance and Resource Utilization[1]

Mode	Name of Parameter File	SLICEs	LUTs	sysMEM EBRs	Registers	I/O	f_{MAX} (MHz)
8237	dma_mc_e2_3_001.lpc	710	1087	0	551	59	72
Non-8237	dma_mc_e2_3_002.lpc	1633	2249	0	1181	125	86

1. Performance and utilization characteristics are generated using LFEC20E-4F672C in Lattice ispLEVER 4.1 software. When using this IP core in a different density, package, or speed grade, performance may vary.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all configurations of this core in LatticeEC devices is DMA-MC-E2-N3. Table 23 lists the Lattice-specific netlists that are available in the Evaluation Package, which can be downloaded from the Lattice web site at www.latticesemi.com.

Table 23. Core Configuration

Name of Parameter File	Number of Channels	Data Bus Width	Address Bus Width	Word Count Width
8237 Mode				
dma_mc_e2_3_001.lpc	4	8	16	16
Non-8237 Mode				
dma_mc_e2_3_002.lpc	4	32	32	16

You can use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

Appendix for LatticeXP™ FPGAs

Table 24. Performance and Resource Utilization[1]

Mode	Name of Parameter File	SLICEs	LUTs	sysMEM EBRs	Registers	I/O	f_{MAX} (MHz)
8237	dma_mc_xm_3_001.lpc	746	1287	0	555	59	71
Non-8237	dma_mc_xm_3_002.lpc	1794	3084	0	1179	125	80

1. Performance and utilization characteristics are generated using LFXP10E-4F388C in Lattice ispLEVER 5.0 software. When using this IP core in a different density, package, or speed grade, performance may vary.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all configurations of this core in LatticeXP devices is DMA-MC-XM-N3. Table 25 lists the Lattice-specific netlists that are available in the Evaluation Package, which can be downloaded from the Lattice web site at www.latticesemi.com.

Table 25. Core Configuration

Name of Parameter File	Number of Channels	Data Bus Width	Address Bus Width	Word Count Width
8237 Mode				
dma_mc_xm_3_001.lpc	4	8	16	16
Non-8237 Mode				
dma_mc_xm_3_002.lpc	4	32	32	16

You can use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

Appendix for LatticeSC™ FPGAs

Table 26. Performance and Resource Utilization[1]

Mode	Name of Parameter File	SLICEs	LUTs	sysMEM™ EBRs	Registers	I/Os	f_{MAX} (MHz)
8237	dma_mc_sc_3_001.lpc	717	1249	0	534	59	>100
Non-8237	dma_mc_sc_3_002.lpc	1744	2864	0	1179	125	>100

1. Performance and utilization characteristics are generated using LFSC3GA25E-5F900C in Lattice ispLEVER 5.1 SP2 software. When using this IP core in a different density, package, or speed grade, performance may vary.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all configurations of this core in LatticeSC devices is DMA-MC-SC-N3. Table 27 lists the Lattice-specific netlists that are available in the Evaluation Package, which can be downloaded from the Lattice web site at www.latticesemi.com.

Table 27. Core Configuration

Name of Parameter File	Number of Channels	Data Bus Width	Address Bus Width	Word Count Width
dma_mc_sc_3_001.lpc	4	8	16	16
dma_mc_sc_3_002.lpc	4	32	32	16

You can use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.