www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 7 July 2015, Page No. 13369-13387

Programmable Direct Memory Access Controller Design Using Advanced High Performance Bus Protocol

Ms. Sarojini Hoogar S, Mrs. Sapna M K
Dept. of ECE
SET Jain University
Bangalore, India
saisarojinihs@gmail.com
Dept. of ECE SET Jain University
Bangalore, India

ABSTRACT- Direct memory access is process, which allows certain peripheral devices to access system memory without intervention of CPU. When CPU is performing I/O operation, it will occupy the entire duration of IOR/IOW operations and hence it is unavailable to do other operations. The DMAC is mainly helps in efficient utilization of hardware resources.

The DMAC has 8 channels and may be increased to any number of channels by cascading additional number of controller chips and each channel can be individually programmed. Instead of ASP bus protocol the AHB bus protocol is used to increase the performance of the DMAC. Different transfer modes such as single, block or burst-block transfer modes are allowed in latest DMAC. Edge-sensitive and Level-sensitive triggers are used to initialize the transfer. EOP is active low bidirectional signal which is used to interrupt the transfer externally. The channel priority is set using round robin scheduling algorithm.

The DMAC operates in two cycles, idle cycle and active cycle. In idle cycle the DMAC is inactive and it sample DREQ & CS signal continuously for every clock cycle, to find any channel is requesting for DMA service. In active cycle the DMA will initialize the data transfer until the TC over rolls zero to FFFFh. The design of 8 channels DMA Controller is simulated using Modalism tool, and synthesized using Xilinx tool.

I

INTRODUCTION TO 8237A: The peripheral interface circuit 8237A DMA controller is designed to boost system performance by permitting external devices to access the system memory (1). It also permits memory to memory transfer. Basically it's having 4 channels (1) and by cascading the8237A DMA controller number of channels can be increased to 'n'. Each channel can be activated independently following an active low EOP signal. Each channel has 64K address. The 8237A is 40 pins package chip (1) and one pin is reserved (pin5) and it should be always high. The DMAC operates

in two modes; one is SLAVE mode and MASTER mode. In SLAVE mode the control of the system bus is occupied by the CPU and in MASTER mode the system bus control is occupied by DMAC.
The DMA operates in 2 major cycles, i.e. idle cycle and Active cycle (1). When there is no channel is requesting any service, the 8237A will enter the idle cycle. In Idle cycle the 8237A will sample the DREQ signal for every clock cycle to find any channel is seeking for service. In active cycle the modes of transfer are different such as (1) single transfer mode, block transfer mode, demand transfer mode and cascade mode. Priority of the channel is decided by Rotating scheduling algorithm. (1) The registers used in 8237A are current address register, current word register, base address and base word count registers, command register, mode register, request register, status register, mask register and temporary register. Each register used to do unique operations needed to transfer data from peripherals to system memory and vice-versa.

In DMA controller 8237A has been improved with lots of different features. Mainly the different addressing mode and the different transfer are modes are used. The data block size is increased to 65535 bits. Each channel is configured separately. The special data transfer mode is Burst-Block transfer mode. In Block- burst transfer mode the data is transferred in block of size 65535 bits (2^16) and transferred repeatedly until the peripheral data is exhausted.

II

THE 8237 ARCHITECTURE AND PIN DIAGRAM: This is the complete block diagram of the 8237A. The major blocks of this architecture are timing and control block, buffers, priority encoder and rotating priority block and registers.

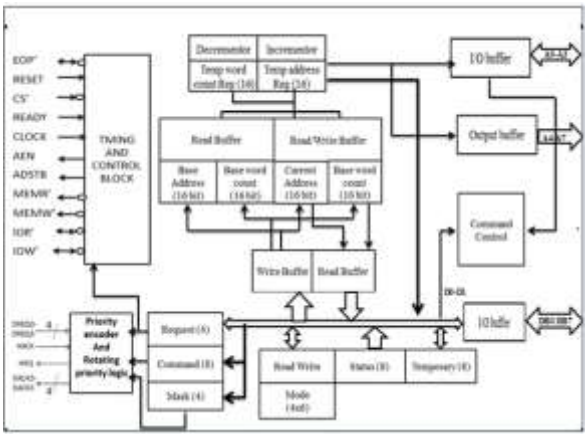


Figure: 1 Architecture of 8237A

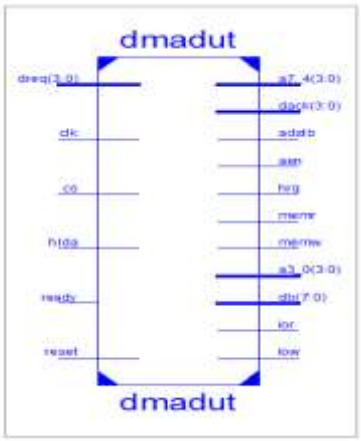


Figure: 2 pin diagram of 8237A

III

REGISTER DESCRIPTION: The 8237A DMAC is having different types of registers to perform specific functions. The 8237A contains 344 bits of internal memory in the form of registers. A details about registers are listed in the below table.

Name of Register	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Register	16 bits	1
Temporary Word Count Register	16 bits	1
Status Register	8 bits	1
Command Register	8 bits	1
Temporary Register	8 bits	1
Mode Registers	6 bits	4
Mask Register	4 bits	1
Request Register	4 bits	1

Table: 1 Register descriptions

IV

THE OPERATIONS OF THE REGISTERS:

- **Base address register**: This is the register used to store the original Address. These Addresses are used during auto-initialization to restore the register with original value. It's 16-bit register, during program condition these registers can't be read by processor. To store the initial address from where data transfer will take place.
- **Base word count register:** This is the register used to store the original Word Count. The Word Count is basically stands for the number of transfers to be performed. These Word Count is used during auto-initialization to restore the register with its original value. It's 16-bit register. During program condition these registers can't be read by processor.
- **Current address register:** It's 16 bit register, which is used to store the current address from where data is being transferred. The address is automatically incremented or decremented after each transfer. This register is read or written by processor during transfer. It can also be reinitialized be Auto-initialization back to its original or base value. The Auto-initialize can be performed only after EOP' or TC.
- **Current word count register:** its 16 bit register, which determines the number of transfers remaining to be performed. The Word Count is automatically decremented after each transfer. When the register value rolls over zero to FFFFH, a TC will be generated. This register is read or written by processor during transfer. It can also be reinitialized be Auto-initialization back to its original or base value. The Auto-initialize can be performed only after EOP' or TC.

- **Temporary address register:** To hold address of data during memory-to-memory transfer. The Temporary register always holds the last byte transferred in previous memory-to-memory operation, until it's cleared by a Reset.
- **Temporary word count register:** To hold number of transfers to be performed in memory-to-memory transfer. The Temporary register always holds the last byte transferred in previous memory-to-memory operation, until it's cleared by a Reset.
- **Mode register:** 6-bit register which stores the channel to be used, the operating mode, i.e. the transfer mode, and other transfer parameters.

  - ❖ Bits 6 and 7 are used to select the transfer mode: 00b = Demand mode, 01b = Single mode, 10b = Block mode, 11b = Cascade mode.
  - ❖ Setting INC selects address decrement, clearing INC selects address increment.
  - ❖ Setting AI enables auto-initialization.

- Bits 2 and 3 are used to select the transfer type: 00b = Verify, 01b = Write to memory, 10b = Read from memory, 11b = Undefined.
- Bits 0 and 1 are used to select the channel: 00b = channel 0, 01b = channel 1 10b = channel 2 and 11b = channel 3.

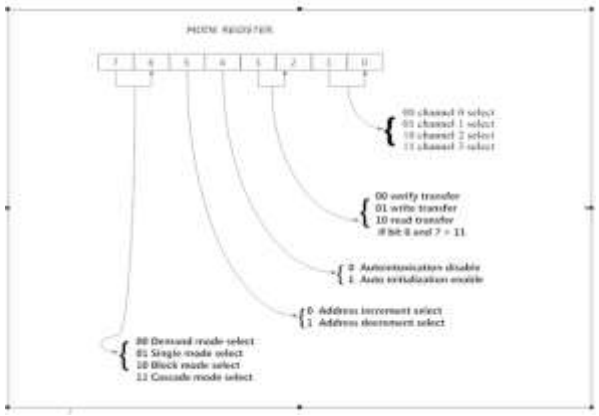


Figure: 3 mode register

- **Command register:** 8-bit register which controls the operation of the 8237A. This initializes the channel to be used for data transfer. The about the bits of the register is shows in the below figure.

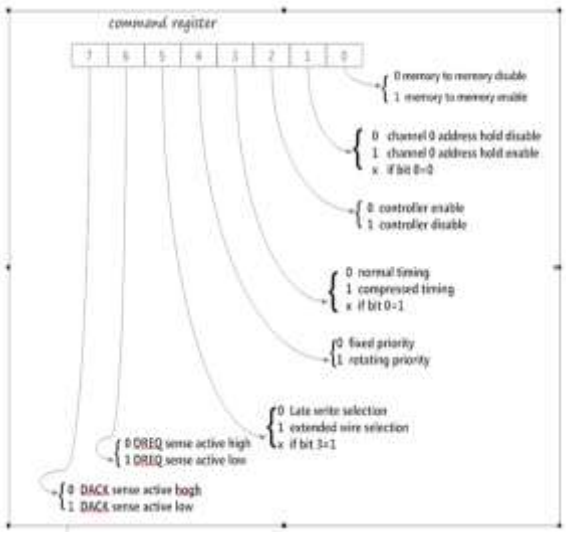


Figure: 4 Command register

- **Request register:** 8-bit register used to indicate which channel is requesting for data transfer. Each channel having request bit which can be se & reset. And two bits are used to select channel using 4 possible patterns such as 00,01,10,11. This is the non-mask able register and subjected to the prioritization by priority encoder circuit.

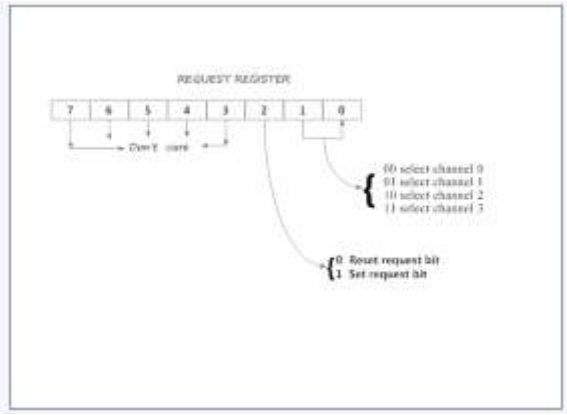


Figure: 5 Request register

- **Mask register:** 8-bit register used to mask a particular channel from requesting for DMA service. It consists of mask bit, which can be set or reset. This disables the he DMA requests until mask bit is cleared.

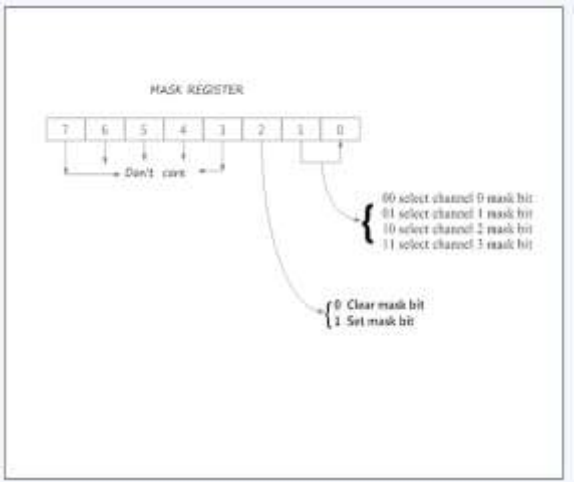


Figure: 6 Mask register

- **Status register:** 8-bit register used to indicate which channel is currently under DMA service and some other parameters. This register also contained the information about which channel has reached the TC and which channel have the pending requests.

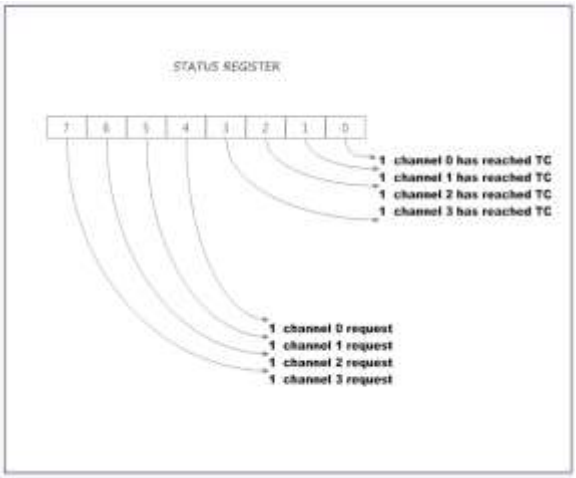


Figure: 7 status register

V

FUNCTIONAL DISCRIPTION OF 8237A:

The Timing control block, Command control block and Priority encoder block are the important blocks of 8237A DMAC. External & internal control signals are generated by the timing control block. The priority encoder block is used to resolve problem between DMA channels requesting service simultaneously. The decoding various commands given to 8237A DMAC and also decoding of the mode word used to control the type of the DMA during service is done by the program command control block. The timing control derives internal timing from the clock input.

8237A DMA OPERATIONS: The 8237A is designed to operate in two major cycles. These are called Idle and Active cycles. Each device cycle is made up of a number of states. The 8237A can assume seven separate states, each composed of one full clock period.

- State I (SI) stand for the inactive state. The 8237A will enter the SI when there is no valid DMA request pending. While in SI, the DMA controller is inactive but may be in the Program Condition, being programmed by the processor.

- State S0 (S0) is the first state of a DMA service. The 8237A has requested a hold but the processor has not yet returned an acknowledge. The 8237A may still be programmed until it receives HLDA from the CPU. An acknowledge from the CPU will signal that DMA transfers may begin. The DREQ must be maintained high until the DACK has arrived.

- The S1, S2, S3 and S4 are the working states of the DMA service, in which the actual data transfer occurs. If more time is needed to complete a transfer than is available with normal timing, wait states (SW) can be inserted between S2 or S3 and S4 by the use of the Ready line on the 8237A.

- **MEMORY-TO-MEMORY**: Transfers require a read-from and a write-to-memory to complete each transfer. The states, which resemble the normal working states, use two digit numbers for identification. Eight states are required for a single transfer. The first four states (S11, S12, S13, S14) are used for the read from- memory half and the last four states (S21, S22, S23, S24) for the write-to-memory half of the transfer.

IDLE CYCLE: When no channel is requesting service, the 8237A will enter the Idle cycle and perform ``SI" states. In this cycle the 8237A will sample the DREQ lines every clock cycle to determine if any channel is requesting a DMA service. The device will also sample CS, looking for an attempt by the microprocessor to write or read the internal registers of the 8237A. When CS is low and HLDA is low, the 8237A enters the Program Condition.

Special software commands can be executed by the 8237A in the Program Condition. These commands are decoded as sets of addresses with the CS and IOW. The commands do not make use of the data bus. Instructions include Clear First/Last Flip-Flop and Master Clear.

ACTIVE CYCLE: Whenever the 8237A determines the service request from peripherals, it will send HREQ signal to CPU to release the system bus. When DMA will receive the HACK then it will start the actual data transfer in one of four modes. The arbitration process before the data transfer is shown in below diagram.

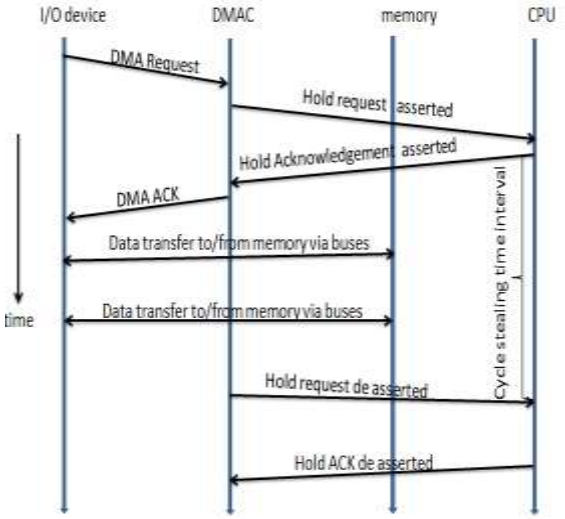


Figure: 8 DMA arbitration

DATA TRANSFER MODES OF 8237A:

1) **SINGLE TRANSFER MODE**: In single transfer mode the device is programmed to make one transfer only at a time. After each transfer the word count is decreased & address is incremented or decremented. When the word count "rolls over" to FFFFH from Zero, the terminal count (TC) will cause the auto-initialize, if the channel has been programmed to do so. When the word count "rolls over" from FFFFH to Zero, the terminal count (TC) will cause the auto- termination of transfer, which has been programmed to do so. It's only suitable to transfer small amount of data. If we use single transfer mode to transfer large amount of data then it leads latency in transfer.

   The DREQ must hold high until DACK becomes high. If DREQ is held active until the complete transfer then, HRQ must be in-active and releases the bus to the system. Again HREQ will go high upon receiving receipt of new HLD, another single transfer can be performed.

2) **BLOCK TRANSFER MODE:** In Block transfer mode the block of data has been transferred upon receiving DREQ to continue making transfers during the service until a TC, caused by word count going to FFFFH or an external End of process

(EOP) is encountered. This is used to transfer the large amount of data.

3) **DEMAND TRANSFER MODE:** In Demand Transfer mode the device is programmed to continue making transfer until the external active low EOP or TC is encountered or until DREQ goes inactive. Thus transfer may continue until the I/O device has exhausted its data capacity. Only active low EOP can cause the termination of transfer and EOP' is generated by TC.

4) **CASCADE MODE:** This is used to expand the system by cascading more than one 8237A together. The HREQ & HLDA signals from the additional 8237A are connected to the DREQ & DACK signals of a channel of the initial 8237A.

TRANSFER TYPE:

Each of the three active transfer modes can perform three different types of transfers. These are Read, Write and Verify. Write transfers move data from an I/O device to the memory by activating MEMW and IOR. Read transfers move data from memory to an I/O device by activating MEMR and IOW. Verify transfers are pseudo transfers. The 8237A operates as in Read or Write transfers generating addresses, and responding to EOP, etc. However, the memory and I/O control lines all remain inactive. The ready input is ignored in verify mode.

MEMORY-TO-MEMORY: The 8237A includes a memory-to-memory transfer feature. Programming a bit in the Command register selects channels 0 and 1 to operate as memory-to memory transfer channels. The transfer is initiated by setting the software DREQ for channel 0. The 8237A requests a DMA service in the normal manner. After HLDA is true, the device, using four state transfers in Block Transfer mode, reads data from the memory. The channel 0 Current Address register is the source for the address used and is decremented or incremented in the normal manner. The data byte read from the memory is stored in the 8237A internal Temporary register. Channel 1 then performs a four-state transfer of the data from the Temporary register to memory using the address in its Current Address register and incrementing or decrementing it in the normal manner. The channel 1 current Word Count is decremented. When the word count of channel 1 goes to FFFFH, a TC is generated causing an EOP output terminating the service.

Channel 0 may be programmed to retain the same address for all transfers. This allows a single word to be written to a block of memory. The 8237A will respond to external EOP signals during memory-to-memory transfers. Data comparators in block search schemes may use this input to terminate the service when a match is found. Memory-to-memory operations can be detected as an active AEN with no DACK outputs.

AUTOINITIALIZE: By programming a bit in the Mode register, a channel may be set up as an Auto initialize channel. During Auto initialize initialization, the original values of the Current Address and Current Word Count Registers are automatically restored from the Base Address and Base Word count registers of that channel following EOP. The base registers are loaded simultaneously with the current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not altered when the channel is in Auto initialize. Following Auto initialize the channel is ready to perform another DMA Service, without CPU intervention, as soon as a valid DREQ is detected. In order to Auto initialize both channels in a memory-to-memory transfer, both word counts should be programmed identically. If interrupted externally, EOP pulses should be applied in both bus cycles.

PRIORITY: The 8237A has two types of priority encoding available as software selectable options.

ROTATING PRIORITY: The first is Fixed Priority which fixes the channels in priority order based upon the descending value of their number. The channel with the lowest priority is 3 followed by 2, 1 and the highest priority channel, 0. After the recognition of any one channel for service, the other channels are prevented from interfering with that service until it is completed. After completion of a service, HRQ will go inactive and the 8237A will wait for HLDA to go low before activating HRQ to service another channel. The second scheme is Rotating Priority. The last channel to get service becomes the lowest priority channel with the others rotating accordingly. The rotating priority is applied to all channels to resolve the problems.

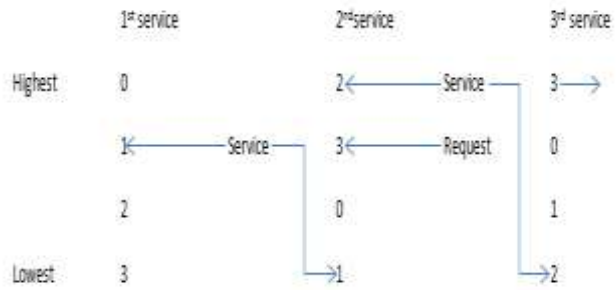


Figure: 10 rotating priority

ROUND ROBIN SCHEDULING: Round robin is the scheduling algorithm used by the CPU during execution of the process. Round robin is designed specifically for time sharing systems. It is similar to first come first serve scheduling algorithm but the pre-emption is the added functionality to switch between the processes. A small unit of time also known as time slice or quantum is set/defined. The ready queue works like circular queue .All processes in this algorithm are kept in the circular queue also known as ready queue. Each New process is added to the tail of the ready/circular queue. By using this algorithm, CPU makes sure, time slices (any natural number) are assigned to each process in equal portions and in circular order, dealing with all process without any priority.

SPECIAL COMMANDS AND OPERATION:

Compressed Timing: If the system characteristics permit then the even greater throughput, the 8237A can compress the transfer time to two clock cycles. The state S3 is used to extend the access time. But in compressed timing the S3 is removed, the read pulse width is made equal to write pulse width. Then the transfer consist only S2 & S4. S2 is used to change address and S4 is used to perform read/write. The S1 is used for updating the A8-A15.

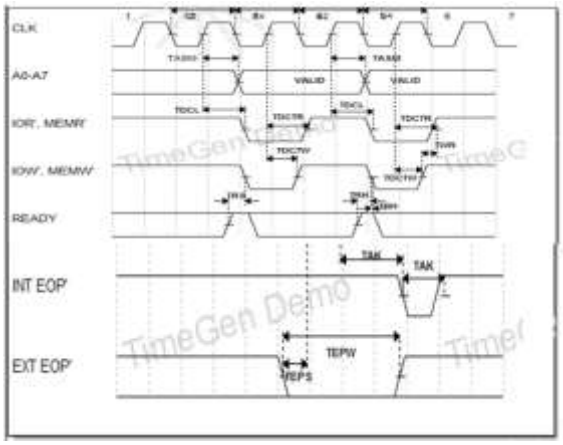


Figure: 12 compressed timing diagram

Address Generation: The address used in 8237A is of 16 bit. So 16 pins should be used for address, which lead to increase in number of pins unnecessarily on package. The address generation is used to reduce number of pins. The 8 bit LSB are used as pins & the 8237A multiplexes 8 bit MSB on data lines. The state S1 is used to output 8-bit MSB on external latch from which they may be placed on address bus. The falling edge of the address strobe (ASTB) is used to load these bits from data line to latch. Address enable (AEN) is used to enable the address bits onto the address bus through a three state enable. The lower order bits are output by the 8237A directly. The lines A0-A7 are connected to the address bus. The timing relationship between the CLK, ASTB, AEN, DB0-DB7 and A0-A7 is shows in timing diagram figure 31.

During Block and Demand Transfer mode services, which include multiple transfers, the addresses generated will be sequential. For many transfers the data held in the external address latch will remain the same. This data need only change when a carry or borrow from A7 to A8 takes place in the normal sequence of addresses. To save time and speed transfers, the 8237A executes S1 staonly when updating of A8±A15 in the latch is necessary. This means for long services, S1 states and Address Strobes may occur only once every 256 transfers, a savings of 255 clock cycles for each 256 transfers.

Software Commands: There are many special software commands which can be executed in the program condition for special operation. There are three software commands are clear First/Last Flip-Flop, Master Clear and Clear Mask Register.

1) **Clear first/last flip-flop:** This command is used for reading or writing new address or word count to the 8237A. This command initilizes the flip-flop to the known state so that subsequent accesses to register contentents by the processor will address upper and lower bytes in the correct sequence.

2) **Master Clear register:** It acts as hardware reset. The command, status, temporary, and internal First/Last Flip-Flop registered are cleared and the Mask register is set to block the channel. The 8237A will enter the idle cycle.

3) **Clear Mask Register:** To enable the four channels to accept the request to provide service by using the Clear Mask Register.

SIGNALS						
IOW'	**IOR'**	**A3**	**A2**	**A1**	**A0**	**Function**
1	0	1	0	0	0	Read status register
0	1	1	0	0	0	Write command register
1	0	1	0	0	1	Illegal
0	1	1	0	0	1	Write request register
1	0	1	0	1	0	Illegal
0	1	1	0	1	0	Write single mask register bit
1	0	1	0	1	1	Illegal
0	1	1	0	1	1	Write mode register
1	0	1	1	0	0	Illegal
0	1	1	1	0	0	Clear byte pointer flip-flop
1	0	1	1	0	1	Read temporary register
0	1	1	1	0	1	Master clear
1	0	1	1	1	0	Illegal
0	1	1	1	1	0	Clear mask register
1	0	1	1	1	1	Illegal
0	1	1	1	1	1	Write all mask register bits

Table: 3 codes for software commands

These codes specify the special function occurs based on codes given above. By using above codes of A0-A3 and active low IOW & IOR the read or write of the internal register of 8237A. Some of the codes are illegal, which are not used actually in reality.

The Word Count and Address Register Command Codes are given in the below table. This table gives the details about when to write or read the Base and current address, current address, Base and current word count and current word count.

PROPOSED DMAC: The 8237A has only four channels, it can provide service to only four peripheral requests. In proposed DMA Controller the number of channels is increased from four to eight, so that it can provide service to 8 peripheral requests at a time. The proposed DMAC is designed by using some special features compared to 8237A, which can helps in fast data transfer more

efficiently. The special features of the proposed DMC are given below.

I. The eight independent channels. This can be programed and configured independently.
II. The channel priorities are decided by using fixed priority or Round Robin Scheduling algorithm.
III. For each transfer two machine clock cycles are required.
IV. The block size used in proposed DMAC in 65535.
V. Triggering selection.
VI. Selectable edge or level-triggered transfer.
VII. And four different addressing modes are used.
VIII. Three different transfer modes are used, i.e. single, block or burst-block transfer modes.
IX. Byte or word and mixed byte and word transfer capability.

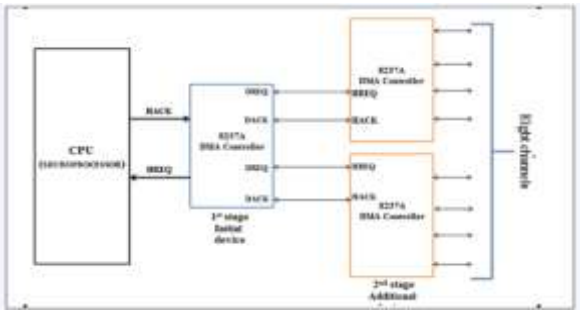


Figure: 13 eight channel DMAC

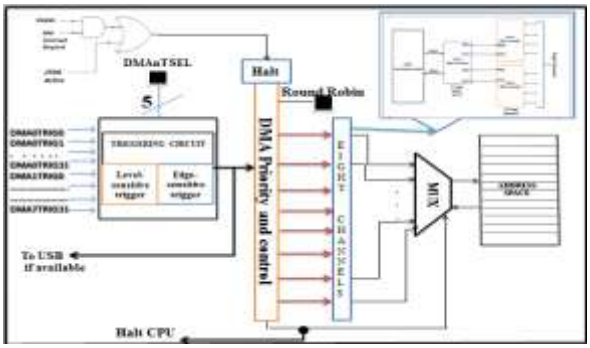


Figure: 14 complete block diagram of proposed DMAC

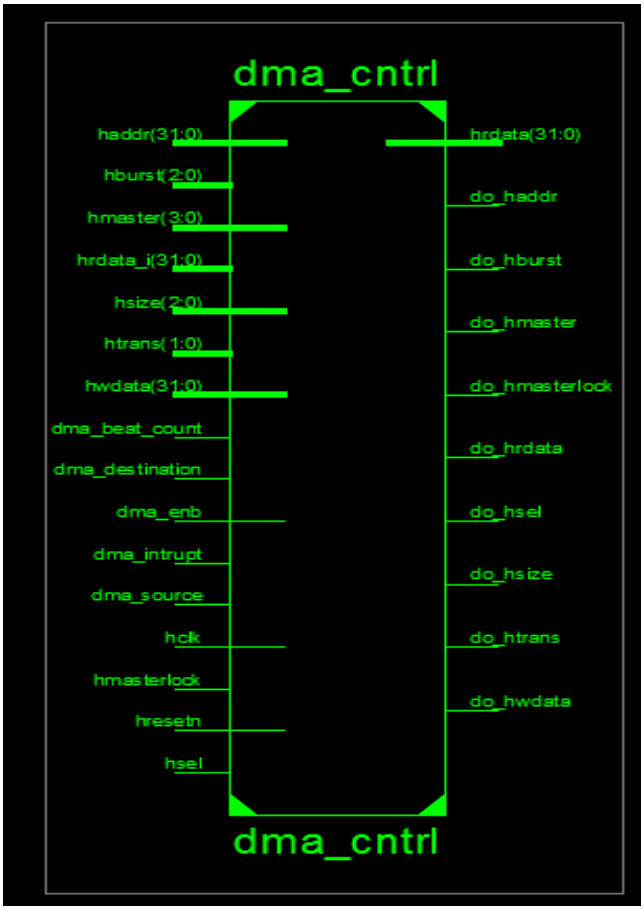


Figure: 15 proposed DMAC pin diagram

DMA Addressing Modes:

There are four different addressing modes. The each DMA channels is independently configurable for addressing modes. For example, channel 0 may transfer between two fixed addresses, while channel 1 may transfer between two blocks of addresses. The addressing modes are configured with the DMASRCINCR (DMA source increment register) and DMASRCDECR (DMA source decrement register) control bits, which are used to decide the address, should be incremented or decremented after each transfer. The DMADSTINCR (DMA destination increment register) control bit, which is used to decide the destination address, should be incremented, decremented or unchanged after each transfer. The transfer may be byte to byte, word to word, or word to byte, byte to word. When transferring byte to word, the higher order bits are padded with zeros and then transferred. When transferring word to byte, the lower byte is transferred of the source word transfer.

1) **Fixed address to fixed address:** The data is transferred from single address to single address only.

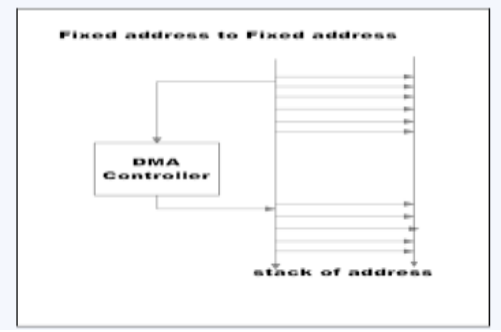


Figure: 15 fixed to fixed address mode

2) **Fixed address to block of addresses:** The data is transferred from single or fixed address to block of address only.

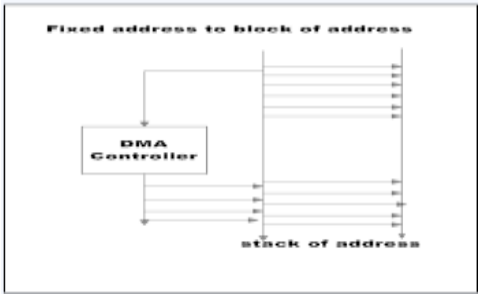


Figure: 16 fixed to block of address mode

3) **Block of addresses to fixed address:** The data is transferred from block of address to single address only.

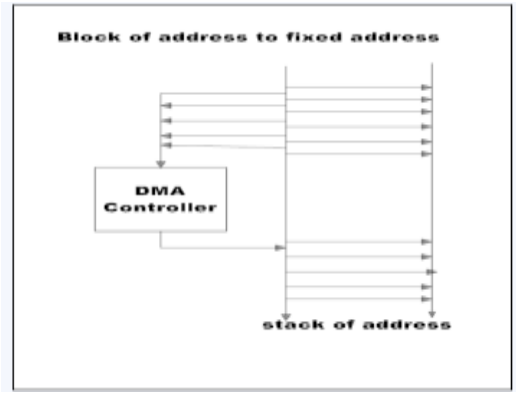


Figure: 17 block of address to fixed address mode

4) **Block of addresses to block of addresses:** The data is transferred from block of address to block of address only.

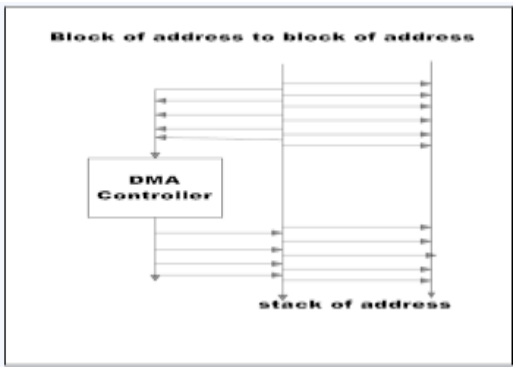


Figure: 18 block of address to block of address mode

6.3 DMA Transfer modes: The DMAC uses six transfer mode by the DMADT bits, which are listed in the below table. Each channel can be configured individually for transfer modes. For example, the channel 0 is configured to use single transfer mode, the channel 1 can be configured to block transfer mode and the channel 2 can be configured to burst-block transfer mode.

The transfer mode is configured independently from the addressing modes. Any transfer mode can be used with any addressing mode. Two type of data can be transferred is selectable by DMAxCTL DSTBYTE AND SRCBYTE fields. The source and/or destination location can be either byte or word data.

ALGORITHMS FOR SINGLE TRANSFER MODE: In single transfer mode each word /byte requires a separate trigger. This below flow diagram shows the complete operation of single transfer mode.

The DMAxSZ register is used to store the information about how many number of transfers to be made. After ach transfer the source and destination address are incremented or decremented is decided by DMADSTINCR and DMASRCINCR bits. If DMAxSZ=0, then no transfer occurs.

The DMAxSA, DMAxDA, and DMAxSZ registers are copied into temporary registers. The temporary values of DMAxSA and DMAxDA are incremented or decremented after each transfer. The DMAxSZ register is decremented after each transfer. When the DMAxSZ register decrements to zero, it is reloaded from its temporary register and the corresponding DMAIFG flag is set. When DMADT = {0}, the DMAEN bit is cleared automatically when DMAxSZ decrements to zero and must be set again for another transfer to occur. In repeated single transfer mode, the DMA controller remains enabled with DMAEN = 1, and a transfer occurs every time a trigger occurs.

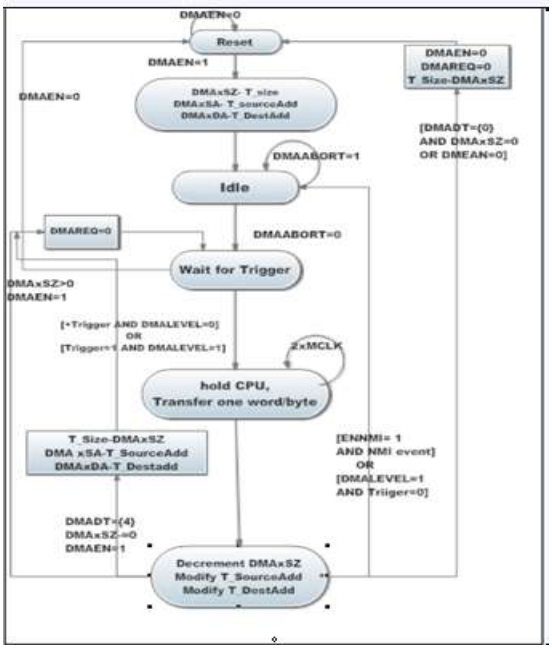
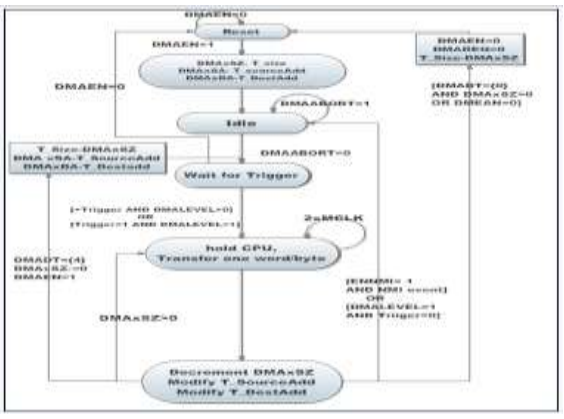


Figure: 19 DMA single transfer state

ALGORITHMS FOR BLOCK TRANSFER MODE: In block transfer mode, a transfer of a complete block of data occurs after one trigger. When DMADT = {1}, the DMAEN bit is cleared after the completion of the block transfer and must be set again before another block transfer can be triggered. After a block transfer has been triggered, further trigger signals occurring during the block transfer are ignored. The block transfer state diagram is shown in Figure 20.

The DMAxSZ register is used to define the size of the block, and the DMADSTINCR and DMASRCINCR bits select if the destination address and the source address are incremented or decremented after each transfer of the block. If DMAxSZ = 0, no transfers occur. The DMAxSA, DMAxDA, and DMAxSZ registers are copied into temporary registers. The temporary values of DMAxSA and DMAxDA are incremented or decremented after each transfer in the block. The DMAxSZ register is decremented after each transfer of the block and shows the number of transfers remaining in the block. When the DMAxSZ register decrements to zero, it is reloaded from its temporary register and the corresponding DMAIFG flag is set.



DMADT	Transfer Mode	Description
000	Single transfer	Each transfer requires a trigger. DMAEN is automatically cleared when DMAxSZ transfers have been made.
001	Block transfer	A complete block is transferred with one trigger. DMAEN is automatically cleared at the end of the block transfer.
010, 011	Burst-block transfer	CPU activity is interleaved with a block transfer. DMAEN is automatically cleared at the end of the burst-block transfer.
100	Repeated single transfer	Each transfer requires a trigger. DMAEN remains enabled.
101	Repeated block transfer	A complete block is transferred with one trigger. DMAEN remains enabled.
110, 111	Repeated burst-block transfer	CPU activity is interleaved with a block transfer. DMAEN remains enabled.

Figure: 20 DMA Block transfer state

During a block transfer, the CPU is halted until the complete block has been transferred. The block transfer takes 2 × MCLK × DMAxSZ clock cycles to complete. CPU execution resumes with its previous state after the block transfer is complete.

In repeated block transfer mode, the DMAEN bit remains set after completion of the block transfer. The next trigger after the completion of a repeated block transfer triggers another block transfer.

ALGORITHMS FOR BURST-BLOCK TRANSFER MODE: In burst-block mode, transfers are block transfers with CPU activity interleaved. The CPU executes two MCLK cycles after every four byte/word transfers of the block, resulting in 20% CPU execution capacity. After the burst-block, CPU execution resumes at 100% capacity and the DMAEN bit is cleared. DMAEN must be set again before another burst-block transfer can be triggered.

After a burst-block transfer has been triggered, further trigger signals occurring during the burst-block transfer are ignored. The burst-block transfer state diagram is shown in Figure 21. The DMAxSZ register is used to define the size of the block, and the DMADSTINCR and DMASRCINCR bits select if the destination address and the source address

are incremented or decremented after each transfer of the block. If DMAxSZ = 0, no transfers occur.

The DMAxSA, DMAxDA, and DMAxSZ registers are copied into temporary registers. The temporary values of DMAxSA and DMAxDA are incremented or decremented after each transfer in the block. The DMAxSZ register is decremented after each transfer of the block and shows the number of transfers remaining in the block. When the DMAxSZ register decrements to zero, it is reloaded from its temporary register and the corresponding DMAIFG flag is set.

In repeated burst-block mode, the DMAEN bit remains set after completion of the burst-block transfer and no further trigger signals are required to initiate another burst-block transfer. Another burst-block transfer begins immediately after completion of a burst-block transfer. In this case, the transfers must be stopped by clearing the DMAEN bit, or by an (non) maskable interrupt (NMI) when ENNMI is set. In repeated burst-block mode the CPU executes at 20% capacity continuously until the repeated burst-block transfer is stopped.

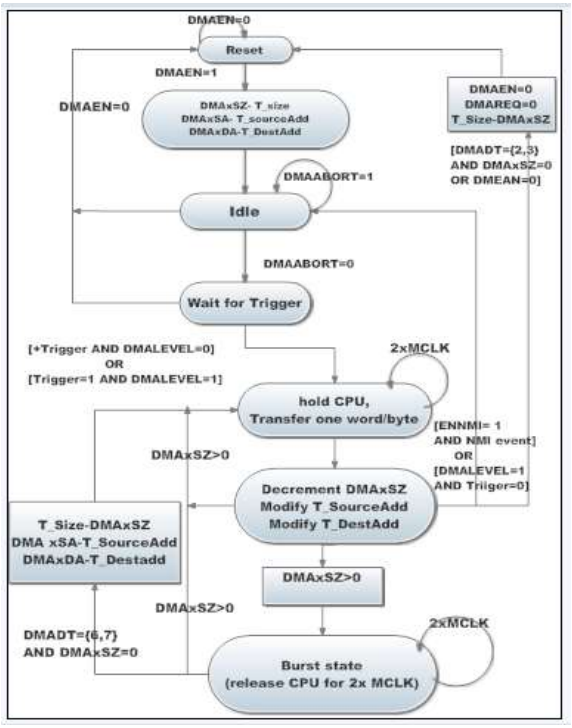


Figure: 21 DMA Burst transfer state

VI

INITIATING DMA TRANSFERS: Each DMA channel is independently configured for its trigger source with the DMAxTSEL. The DMAxSA bits should be modified only when the DMACTLx DMAEN bit is 0. Otherwise, unpredictable DMA triggers may occur. When selecting the trigger, the trigger must not have already occurred, or the transfer does not take place.

EDGE-SENSITIVE TRIGGERS: When the DMA LEVEL=0, Edge- Sensitive Trigger is used. In single transfer mode each transfer requires separate trigger. In block or burst-block modes only one trigger at the starting point of transfer is enough.

LEVEL-SENSITIVE TRIGGERS: When the DMA LEVEL=1, the level sensitive triggers are used. For proper operation, level-sensitive triggers can only be used when external trigger DMAE0 is selected as the trigger. DMA transfers are triggered as long as the trigger signal is high and the DMAEN bit remains set.

The trigger signal must remain high for a block or burst-block transfer to complete. If the trigger signal goes low during a block or burst-block transfer, the DMA controller is held in its current state until the trigger goes back high or until the DMA registers are modified by software. If the DMA registers are not modified by software, when the trigger signal goes high again, the transfer resumes from where it was when the trigger signal went low.

When DMALEVEL = 1, transfer modes selected when DMADT = {0, 1, 2, 3} are recommended because the DMAEN bit is automatically reset after the configured transfer.

HALTING EXECUTING INSTRUCTION FOR DMA TRANSFER: The DMARMWDIS bit controls when the CPU is halted for DMA transfers. When DMARMWDIS = 0, the CPU is halted immediately and the transfer begins when a trigger is received. In this case, it is possible that CPU read-modify-write operations can be interrupted by a DMA transfer. When DMARMWDIS = 1, the CPU finishes the currently executing read-modify-write operation before the DMA controller halts the CPU and the transfer begins.

STOPPING DMA TRANSFERS: There are two ways to stop DMA transfers in progress:

• A single, block, or burst-block transfer may be stopped with an NMI, if the ENNMI bit is set in register DMACTL1.

• A burst-block transfer may be stopped by clearing the DMAEN bit.

DMA TRANSFER CYCLE TIME:

The DMA controller requires one or two MCLK clock cycles to synchronize before each single transfer or complete block or burst-block transfer. Each byte/word transfer requires two MCLK cycles after synchronization, and one cycle of wait time after the transfer. Because the DMA controller uses MCLK, the DMA cycle time is dependent on the MSP430 operating mode and clock system setup.

If the MCLK source is active but the CPU is off, the DMA controller uses the MCLK source for each transfer, without re-enabling the CPU. If the MCLK source is off, the DMA controller temporarily restarts MCLK, sourced with DCOCLK, for the single transfer or complete block or burst-block transfer. The CPU remains off and after the transfer completes, MCLK is turned off. The maximum DMA cycle time for all operating modes is shown in Table 6.

CPU Operating mode clock source	Maximum DMA Cycle Time
Active mode MCLK = DCOCLK	4 MCLK cycles
Active mode MCLK = LFXT1CLK	4 MCLK cycles
Low-power mode LPM0/1 MCLK = DCOCLK	5 MCLK cycles
Low-power mode LPM3/4 MCLK = DCOCLK	5 MCLK cycles + 5 µs(1)
Low-power mode LPM0/1 MCLK = LFXT1CLK	5 MCLK cycles
Low-power mode LPM3 MCLK = LFXT1CLK	5 MCLK cycles
Low-power mode LPM4 MCLK = LFXT1CLK	5 MCLK cycles + 5 µs(1)

Table: 6 Maximum Single-Transfer DMA Cycle Time

USING DMA WITH SYSTEM INTERRUPTS:

DMA transfers are not interruptible by system interrupts. System interrupts remain pending until the completion of the transfer. NMIs can interrupt the DMA controller if the ENNMI bit is set. System interrupt service routines are interrupted by DMA transfers. If an interrupt service routine or other routine must execute with no interruptions, the DMA controller should be disabled prior to executing the routine.

DMA CONTROLLER INTERRUPTS:

Each DMA channel has its own DMAIFG flag. Each DMAIFG flag is set in any mode when the corresponding DMAxSZ register counts to zero. If the corresponding DMAIE and GIE bits are set, an interrupt request is generated.

All DMAIFG flags are prioritized, with DMA0IFG being the highest, and combined to source a single interrupt vector. The highest-priority enabled interrupt generates a number in the DMAIV register. This number can be evaluated or added to the program counter (PC) to automatically enter the appropriate software routine. Disabled DMA interrupts do not affect the DMAIV value.

Any access, read or write, of the DMAIV register automatically resets the highest pending interrupt flag. If another interrupt flag is set, another interrupt is immediately generated after servicing the initial interrupt. For example, assume that DMA0 has the highest priority. If the DMA0IFG and DMA2IFG flags are set when the interrupt service routine accesses the DMAIV register, DMA0IFG is reset automatically. After the RETI instruction of the interrupt service routine is executed, the DMA2IFG generates another interrupt.

DMA REGISTERS:

The DMA module registers are listed in table 7. The base addresses can be found in the –device-specific data sheet. Each channel starts at its respective base address. The address offsets are listed in Table 7.

Offset	Register name	Type	Access
00h	DMA Control 0	Read/Write	Word
02h	DMA Control 1	Read/Write	Word
04h	DMA Control 2	Read/Write	Word
06h	DMA Control 3	Read/Write	Word
08h	DMA Control 4	Read/Write	Word
0Eh	DMA Interrupt vector	Read only	Word
00h	DMA Channel 0 Control	Read/Write	Word
02h	DMA Channel 0 source address	Read/Write	Word, Double word
06h	DMA Channel 0 destination address	Read/Write	Word, Double word
0Ah	DMA channel 0 transfer size	Read/Write	Word
00h	DMA Channel 1 Control	Read/Write	Word
02h	DMA Channel 1 source address	Read/Write	Word, Double word
06h	DMA Channel 1 destination address	Read/Write	Word, Double word
0Ah	DMA channel 1 transfer size	Read/Write	Word
00h	DMA Channel 2 Control	Read/Write	Word
02h	DMA Channel 2 source address	Read/Write	Word, Double word
06h	DMA Channel 2 destination address	Read/Write	Word, Double word
0Ah	DMA channel 2 transfer size	Read/Write	Word
00h	DMA Channel 3 Control	Read/Write	Word
02h	DMA Channel 3 source address	Read/Write	Word, Double word
06h	DMA Channel 3 destination address	Read/Write	Word, Double word
0Ah	DMA channel 3 transfer size	Read/Write	Word
00h	DMA Channel 4 Control	Read/Write	Word
02h	DMA Channel 4 source address	Read/Write	Word, Double word
06h	DMA Channel 4 destination address	Read/Write	Word, Double word
0Ah	DMA channel 4 transfer size	Read/Write	Word
00h	DMA Channel 5 Control	Read/Write	Word
02h	DMA Channel 5 source address	Read/Write	Word, Double word
06h	DMA Channel 5	Read/Write	Word,

	destination address		Double word
0Ah	DMA channel 5 transfer size	Read/Write	Word
00h	DMA Channel 6 Control	Read/Write	Word
02h	DMA Channel 6 source address	Read/Write	Word, Double word
06h	DMA Channel 6 destination address	Read/Write	Word, Double word
0Ah	DMA channel 6 transfer size	Read/Write	Word
00h	DMA Channel 7 Control	Read/Write	Word
02h	DMA Channel 7 source address	Read/Write	Word, Double word
06h	DMA Channel 7 destination address	Read/Write	Word, Double word
0Ah	DMA channel 7 transfer size	Read/Write	Word

Table: 7 Register of each channel

DMACTL n Register: DMA Control n Register

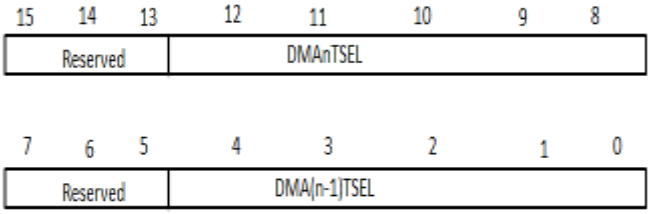


Figure: 22 DMACTLn Register

DMACTL4 Register: DMA Control 4 Register

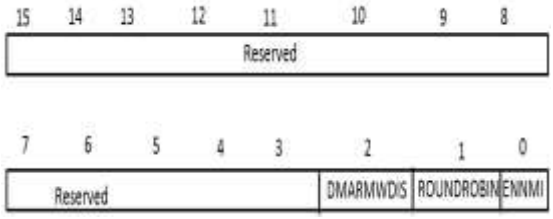


Figure: 23 DMACTL4 Register

Bit	Type	Field	Reset	Description
15-3	R	0h	Reserved	Reserved. Always reads as 0.
2	RW	0h	DMARMWDIS	Read-modify-write disables. When set, this bit inhibits any DMA transfers from occurring during CPU read-modify-write operations. 0b = DMA transfers can occur during read-modify-write CPU operations. 1b = DMA transfers inhibited during read-modify-write CPU operations
1	RW	0h	ROUNDROBIN	Round robin. This bit enables the round-robin DMA channel priorities. 0b = DMA channel priority is DMA0-DMA1-DMA2- ..... -DMA7. 1b = DMA channel priority changes with each transfer.
0	RW	0h	ENNMI	Enable NMI. This bit enables the interruption of a DMA transfer by an NMI. When an NMI interrupts a DMA transfer, the current transfer is completed normally, further transfers are stopped and DMAABORT is set. 0b = NMI

				does not interrupt DMA transfer. 1b = NMI interrupts a DMA transfer.

Table: 9 DMACTL4 Register Descriptions

8.1.3 DMAxCTL Register: DMA Channel x Control Register

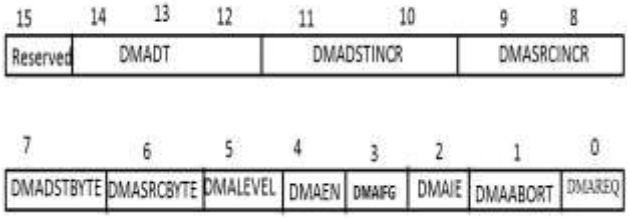


Figure: 24 DMAxCTL Register

Bit	Field	Type	Description
15	Reserved	R	Reserved. Always reads as 0.
14-12	DMADT	RW	DMA transfer mode 000b = Single transfer 001b = Block transfer 010b = Burst-block transfer 011b = Burst-block transfer 100b = Repeated single transfer 101b = Repeated block transfer 110b = Repeated burst-block transfer 111b = Repeated burst-block transfer
11-10	DMADSTINCR	RW	DMA destination increment. This bit selects automatic incrementing or decrementing of the destination address after each byte or word transfer. When DMADSTBYTE = 1, the destination address increments/decrements by one. When DMADSTBYTE = 0, the destination address increments/decrements by two. The DMAxDA is copied into a temporary register and the temporary register is incremented or decremented. DMAxDA is not incremented or decremented. 00b = Destination address is unchanged. 01b = Destination address is unchanged. 10b = Destination address is decremented. 11b = Destination address is incremented.
9-8	DMASRCINCR	RW	DMA source increment. This bit selects automatic incrementing or decrementing of the source address for each byte or word transfer. When DMASRCBYTE = 1, the source address increments/decrements by one. When DMASRCBYTE = 0, the source address increments/decrements by two. The DMAxSA is copied into a temporary register and the temporary register is incremented or decremented. DMAxSA is not incremented or decremented. 00b = Source address is unchanged. 01b = Source address is unchanged. 10b = Source address is decremented. 11b = Source address is incremented.
7	DMADSTBYTE	RW	DMA destination byte. This bit selects the destination as a byte or word. 0b = Word 1b = Byte
6	DMASRCBYTE	RW	DMA source byte. This bit selects the source as a byte or word. 0b = Word 1b = Byte
5	DMALEVEL	RW	DMA level. This bit selects between edge-sensitive and level-sensitive triggers. 0b = Edge sensitive (rising edge) 1b = Level sensitive (high level)
4	DMAEN	RW	DMA enable 0b = Disabled

			1b = Enabled
3	DMAIFG	RW	DMA interrupt flag 0b = No interrupt pending 1b = Interrupt pending
2	DMAIE	RW	DMA interrupt enable 0b = Disabled 1b = Enabled
1	DMAABORT	RW	DMA abort. This bit indicates if a DMA transfer was interrupt by an NMI. 0b = DMA transfer not interrupted 1b = DMA transfer interrupted by NMI
0	DMAREQ	RW	DMA request. Software-controlled DMA start. DMAREQ is reset automatically. 0b = No DMA start 1b = Start DMA

Table: 10 DMAxCTL Register Descriptions

8.1.4 DMAxSA Register: DMA Channel x Source Address Register

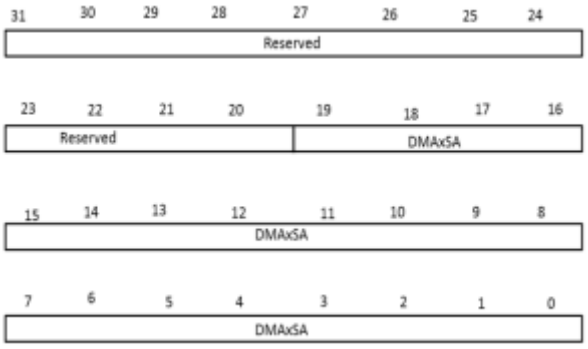


Figure: 25 DMAxSA Register

Bit	Field	Type	Description
31-20	Reserved	R	Reserved. Always reads as 0.
19-0	DMAxSZ	RW	DMA source address. The source address register points to the DMA source address for single transfers or the first source address for block transfers. The source address register remains unchanged during block and burst-block transfers. There are two words for the DMAxSA register. Bits 31-20 are reserved and always read as zero. Reading or writing bits 19-16 requires the use of extended instructions. When writing to DMAxSA with word instructions, bits 19-16 are cleared.

Table: 11 DMAxSA Register Descriptions

8.1.5 DMAxDA Register: DMA Channel x Destination Address Register

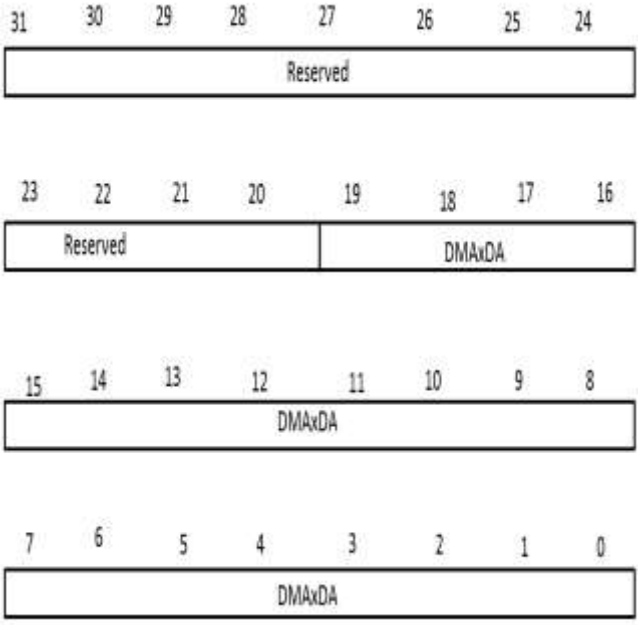


Figure: 26 DMAxDA Register

Bit	Field	Type	Description
31-20	Reserved	R	Reserved. Always reads as 0.
19-0	DMAxDA	RW	DMA destination address. The destination address register points to the DMA destination address for single transfers or the first destination address for block transfers. The destination address register remains unchanged during block and burst-block transfers. There are two words for the DMAxDA register. Bits 31-20 are reserved and always read as zero. Reading or writing bits 19-16 requires the use of extended instructions. When writing to DMAxDA with word instructions, bits 19-16 are cleared.

Table: 12 DMAxDA Register Descriptions

8.1.6 DMAxSZ Register: DMA Channel x Size Address Register

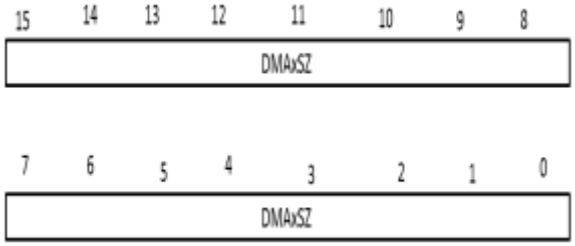


Figure: 27 DMAxSZ Register

Bit	Field	Type	Description
15-0	DMAxSZ	RW	DMA size. The DMA size register defines the number of byte/word data per block transfer. DMAxSZ register decrements with each word or byte transfer. When DMAxSZ decrements to 0, it is immediately and automatically reloaded With its previously initialized value. 00000h = Transfer is disabled. 00001h = One byte or word is transferred. 00002h = Two bytes or words are transferred. ⋮ 0FFFFh = 65535 bytes or words are transferred.

Table: 13 DMAxSZ Register Descriptions

8.1.7 DMAIV Register: DMA Interrupt Vector Register

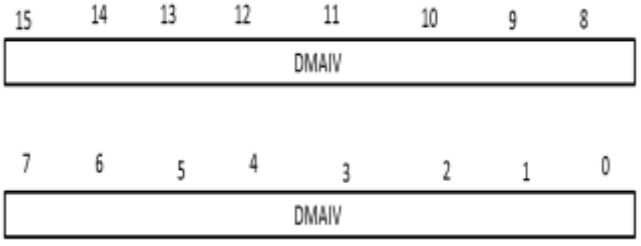


Figure: 28 DMAIV Register

Bit	Field	Type	Description
15-0	DMAIV	R	DMA interrupt vector value 00h = No interrupt pending 02h = Interrupt Source: DMA channel 0; Interrupt Flag: DMA0IFG; Interrupt Priority: Highest 04h = Interrupt Source: DMA channel 1; Interrupt Flag:DMA1IFG 06h = Interrupt Source: DMA channel 2; Interrupt Flag: DMA2IFG 08h = Interrupt Source: DMA channel 3; Interrupt Flag: DMA3IFG 0Ah = Interrupt Source: DMA channel 4; Interrupt Flag: DMA4IFG 0Ch = Interrupt Source: DMA channel 5; Interrupt Flag: DMA5IFG 0Eh = Interrupt Source: DMA channel 6; Interrupt Flag: DMA6IFG 10h = Interrupt Source: DMA channel 7; Interrupt Flag: DMA7IFG; Interrupt Priority: Lowest

Table: 14 DMAIV Register Descriptions

VII

SIMULATION RESULTS: The simulation results of 8237A DMAC are shown below for each operation separately.

SLAVE MODE WRITE TIMING DIAGRAM: In slave mode the DMA is having the control of the system bus and it acts as Master and CPU will act as Slave.

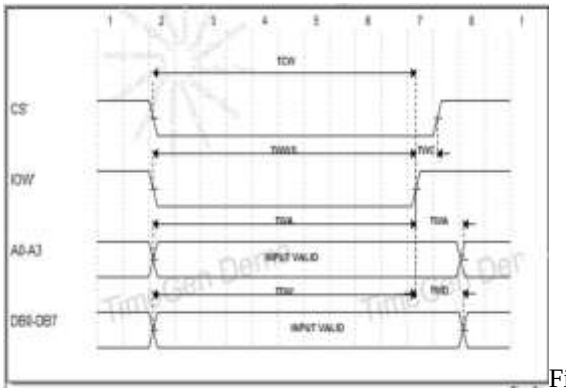


Figure: 29 Slave Mode Write

SLAVE MODE READ TIMING: In slave mode the CPU is having the control of the system bus and it acts as Master and DMA will act as Slave.

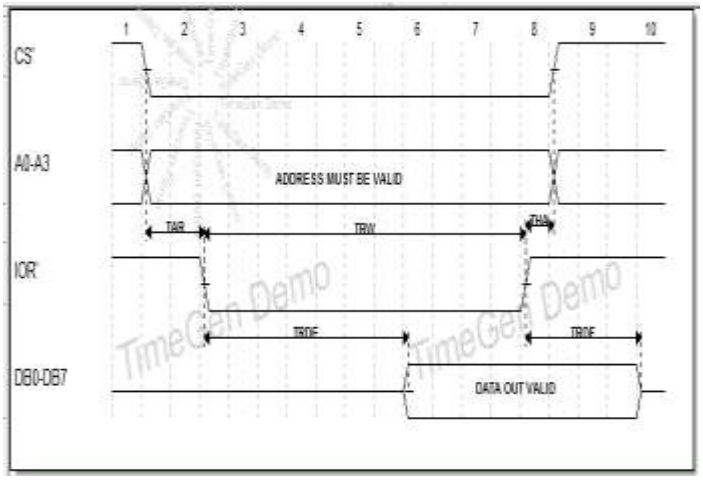


Figure: 30 Slave Mode Read

DMA TRANSFER TIMING: In transfer timing the actual data is transferred. DREQ should be held active until DACK is returned. This below figure shows the information about the actual data transfer timing.

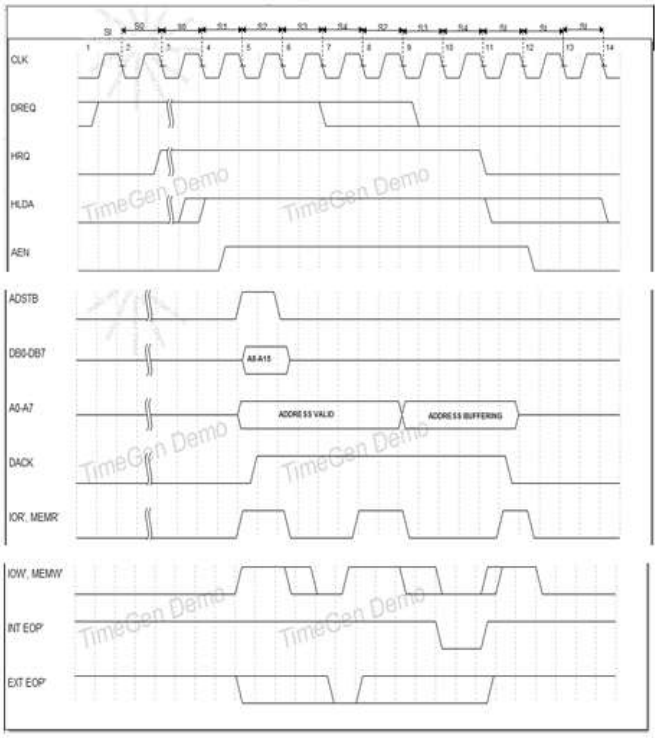


Figure: 31 DMA Transfer timing

MEMORY TO MEMORY TIMING: In memory to memory transferred between two memory elements. This figure shows the timing diagram of the memory to memory data transfer. It uses 8 states and active low EOP for terminating the transfer.

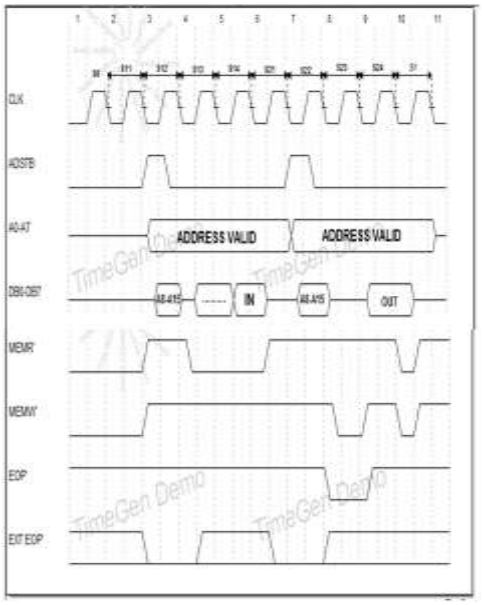


Figure: 32 Memory-to-Memory Transfer timing

READY TIMING:

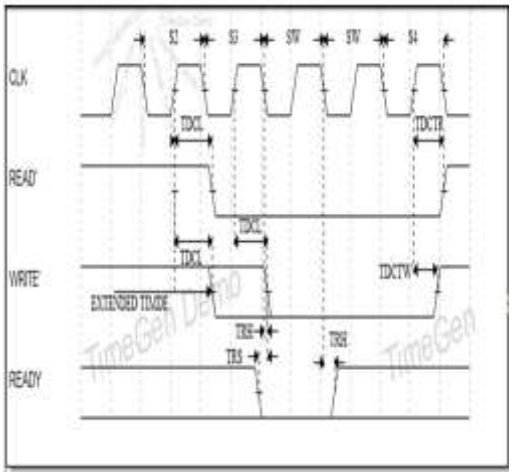


Figure: 33 Ready Timing

COMPRESSED TRANSFER TIMING: In compressed timing the state S3 & S1 are removed and delay in data transfer is removed. S1 is used only when there is need of updating higher order 8 bits of address.

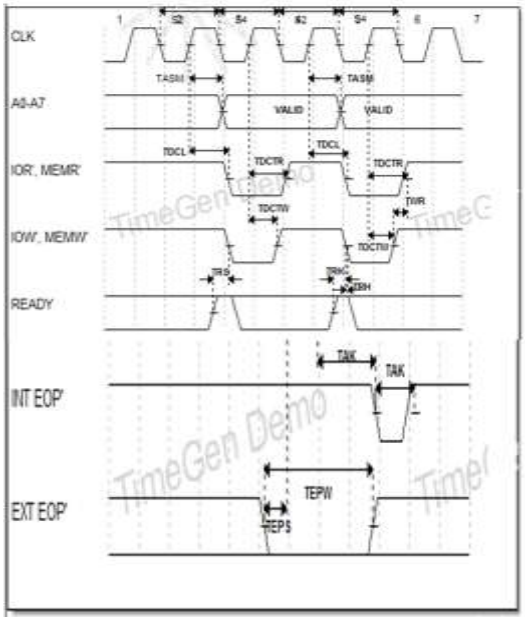


Figure: 34 Compressed Transfer timing

RESET TIMING: When reset is set then the all IOR AND IOW will go low.

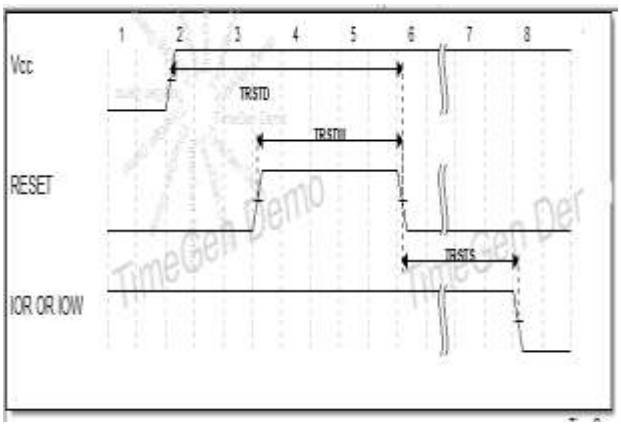


Figure: 35 Reset timing

PROPOSED DMA CONTROLLER SIMULATION RESULTS:

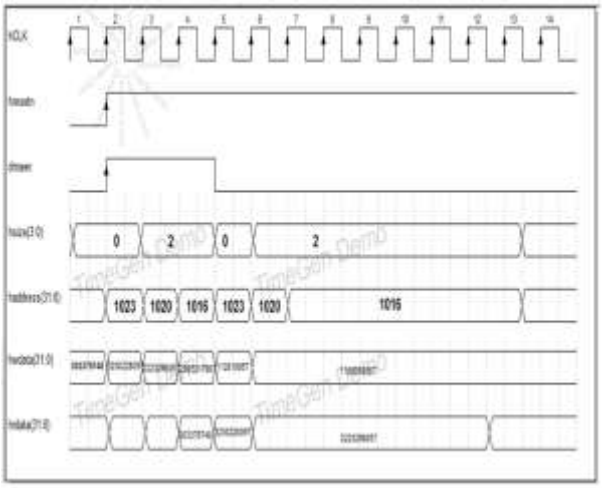


Figure: 36 final DMAC timing diagrams

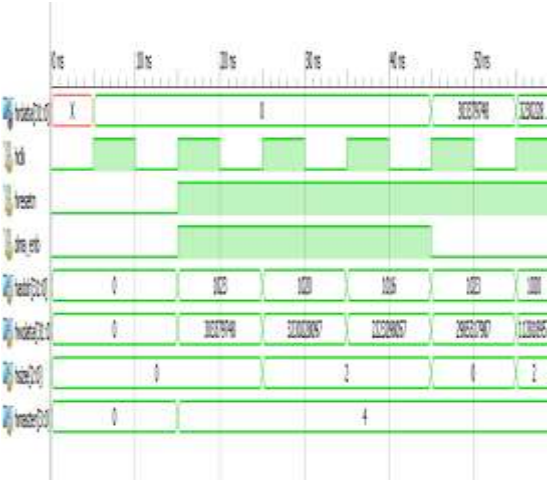


Figure: 37 proposed DMAC data transfer timing diagrams

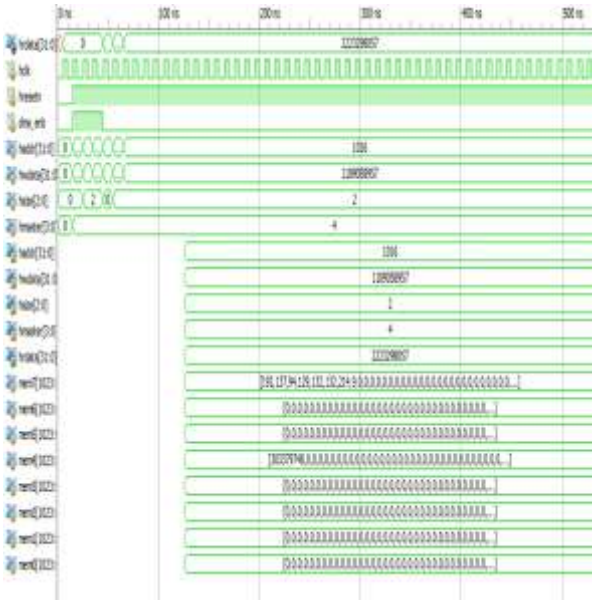


Figure: 38 simulation result showing memory contents

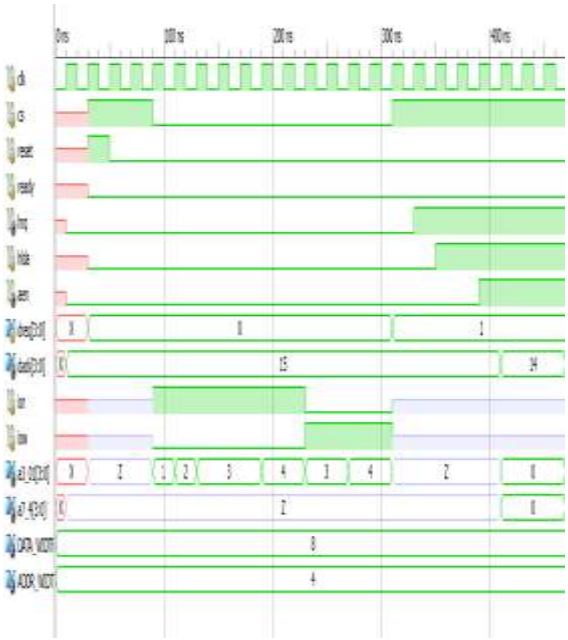


Figure: 38 8237A DMA data transfer timing

COMPARISON: The comparison between 8237A and Proposed DMAC. Here the area of 8237A DMAC and proposed DMAC are compared. The utilization of available resource is good in proposed compared to 8237A. The speed of the proposed is better compared to 8237A.

ONCLUSION AND FURE SCOPRE: From this project I conclude that, by using 8 channels direct memory access controller, it can provide service to 8 I/O requests at a time. By using burst –block transfer mode the latency in the data transfer is reduced. The priority problem between the channels is resolved by using round robin scheduling and fixed or rotating priority. The block size of 65535 bits can be transferred at a time by using block or burst-block transfer mode. In future the number of channels can be increased to n number. The priority of the channel can be resolved by using different or best scheduling algorithm. The block size can be changed in future

Device	Speed/ timing constraints	Area			
8237A DMA	4.875ns	Logic Utilization	Used	Available	Utilization
		Number of Slice Registers	106	4800	2%
		Number of Slice LUTs	250	2400	10%
		Number of fully used LUT-FF pairs	98	260	37%
		Number of bonded IOBs	35	102	34%
		Number of BUFG/BUFGCTRLs	1	16	6%
Proposed DMAC	3.856ns	Logic Utilization	Used	Available	Utilization
		Number of Slice Registers	16416	4800	342%
		Number of Slice LUTs	89153	2400	3714%
		Number of fully used LUT-FF pairs	13391	92178	14%
		Number of bonded IOBs	104	102	101%
		Number of BUFG/BUFGCTRL	1	16	6%

REFERENCES:

[1] Direct Memory Access (DMA) Controller Module by Texas instruments SLAU395C–August 2012–Revised May 2014.

[2] SLAU395D–August 2012–Revised May 2015 Direct Memory Access (DMA) Controller Module (http://www.ti.com/lit/pdf/slau208.)

[3] PrimeCell Single Master DMA Controller(PL081) Revision:r1p2 Technical Reference Manual.

[4] 8237A-5 Programmable dam controllers by Intel.

[5] Yuwen Xis, "Verilop HDL Design Technology of Complex Digital Circuit and System," Bcijing University of Aeronautics and Astronautics, August 1998.

[6] Bt8474 (Multichannel Synchronous Communications Controller) Data Sheet. Rockwell Semiconductors Systems.

[7] Liu Haihua, Chen Xinhao. Weighted Priority Rotational Algorithm on PCI Bus Arbitration. Computer Engineering and Applications 2003.36

[8] AMBA Specification (Rev 2.0).

[9] R. Yu, C. Liu, C. Kang, T. Wang, et al. An efficient DMA Controller for Multimedia Application in MPU Based Soc, International Conference on Multimedia and Expo, IEEE, 2007, pp. 80- 83.

[10] Aghdasi, F., "Self-Clocked Asynchronous Controllers", University of Zimbabwe

Publications, Harare, 1996, pp 34-58.

AUTHORS:

1} Sarojini hoogar s is pursuing her M. Tech from SBMJCE, Bangalore in the Department of Electronics & Communications Engineering (ECE) with Specialization in Signal Processing & VLSI

2) Sapna M K is assistant professor in SBMJCE, Bangalore in the Department of Electronics & Communications Engineering (ECE).