iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** X **Month of publication:** October 2014

DOI:

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Direct Memory Access

Amrendra Kumar Upadhyay[1], Ankit Bhatt[2], Anil Pilaniya[3]

Dronacharya College of Engineering, Gurgaon

Sec-5, haryana-122506

Abstract: - Direct memory access (DMA) is a vital part of many high-end systems. It allows additional bus masters to read or write system memory independently of the CPU(s). DMA channels can transfer blocks of data to or from devices with no CPU overhead.

The CPU manages DMA operations via a DMA controller unit. While the DMA transfer is in progress, the CPU can continue executing code. When the DMA transfer is completed, the DMA controller will signal the CPU with an interrupt.

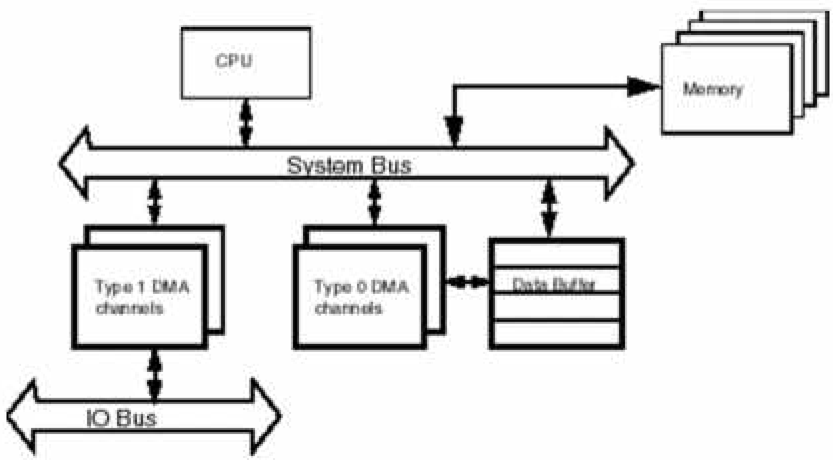
Typical scenarios of block memory copy where DMA can be useful are network packet routing and video streaming. DMA is a particular advantage in situations where the blocks to be transferred are large or the transfer is a repetitive operation that would consume a large portion of potentially useful CPU processing time.

Keywords: - Count register, cycle stealing, bus mastering, interleaved, memory addresses

I. INTRODUCTION

A DMA controller can generate memory addresses and initiate memory read or write cycles. It contains several processor registers that can be written and read by the CPU. These include a memory address register, a byte count register, and one or more control registers. The control registers specify the I/O port to use, the direction of the transfer (reading from the I/O device or writing to the I/O device), the transfer unit (byte at a time or word at a time), and the number of bytes to transfer in one burst. To carry out an input, output or memory-to-memory operation, the host processor initializes the DMA controller with a count of the number of words to transfer, and the memory address to use. The CPU then sends commands to a peripheral device to initiate transfer of data. The DMA controller then provides addresses and read/write control lines to the system memory. Each time a byte of data is ready to be transferred between the peripheral device and memory, the DMA controller increments its internal address register until the full block of data is transferred. DMA transfers can either occur one byte at a time or all at once in burst mode. If they occur a byte at a time, this can allow the CPU to access memory on alternate bus cycles – this is called cycle stealing since the DMA controller and CPU contend for memory access. In burst mode DMA, the CPU can be put on hold while the DMA transfer occurs and a full block of possibly hundreds or thousands of bytes can be moved. When memory cycles are much faster than processor cycles, an interleaved DMA cycle is possible, where the DMA controller uses memory while the CPU cannot.

In a bus mastering system, both the CPU and peripherals can be granted control of the memory bus. Where a peripheral can become bus master, it can directly write to system memory without involvement of the CPU, providing memory address and control signals as required. Some measure must be provided to put the processor into a hold condition so that bus contention does not occur.



II. OPERATION MODES

A. Transparent modes

The transparent mode takes the most time to transfer a block of data, yet it is also the most efficient mode in terms of overall system performance. The DMA controller only transfers data when the CPU is performing operations that do not use the system buses. It is the primary advantage of the transparent mode that the CPU never stops executing its programs and the DMA transfer is free in terms of time. The disadvantage of the transparent mode is that the hardware needs to determine when the CPU is not using the system buses, which can be complex.

B. Burst mode

An entire block of data is transferred in one contiguous sequence. Once the DMA controller is granted access to the system bus by the CPU, it transfers all bytes of data in the data block before releasing control of the system buses back to the CPU, but renders the CPU inactive for relatively long periods of time. The mode is also called "Block Transfer Mode". It is also used to stop unnecessary data.

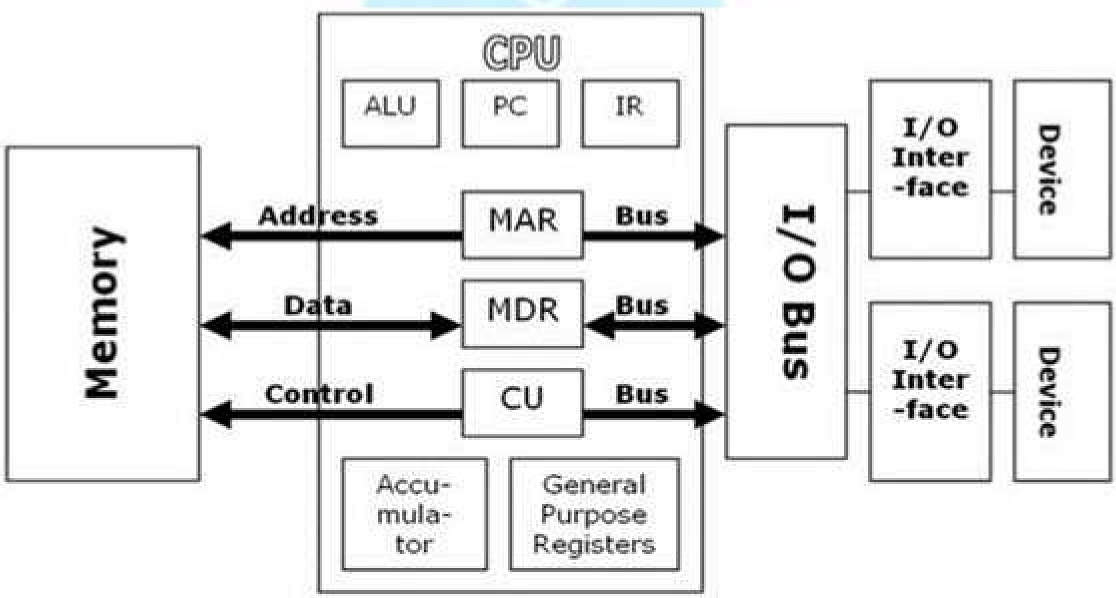
C. Cycle Stealing Mode

The cycle stealing mode is used in systems in which the CPU should not be disabled for the length of time needed for burst transfer modes. In the cycle stealing mode, the DMA controller obtains access to the system bus the same way as in burst mode, using BR (Bus Request) and BG (Bus Grant) signals, which are the two signals controlling the interface between the CPU and the DMA controller. However, in cycle stealing mode, after one byte of data transfer, the control of the system bus is deasserted to the CPU via BG. It is then continually requested again via BR, transferring one byte of data per request, until the entire block of data has been transferred. By continually obtaining and releasing the control of the system bus, the DMA controller essentially interleaves instruction and data transfers. The CPU processes an instruction, then the DMA controller transfers one data value, and so on. On the one hand, the data block is not transferred as quickly in cycle stealing mode as in burst mode, but on the other hand the CPU is not idled for as long as in burst mode. Cycle stealing mode is useful for controllers that monitor data in real time

III. IMPLEMENTATION

To implement the direct transfer of data between memory and input output a hardware DMA controller is added to the system.

This hardware works like a bypass for a town, allowing large blocks of data to bypass the CPU without snarling it up.



In this scenario large blocks of data are transferred using the DMA hardware.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

IV. FUTURE SUGGESTION

The impact of multiprocessor, multiple, DMA channels for DMA cache

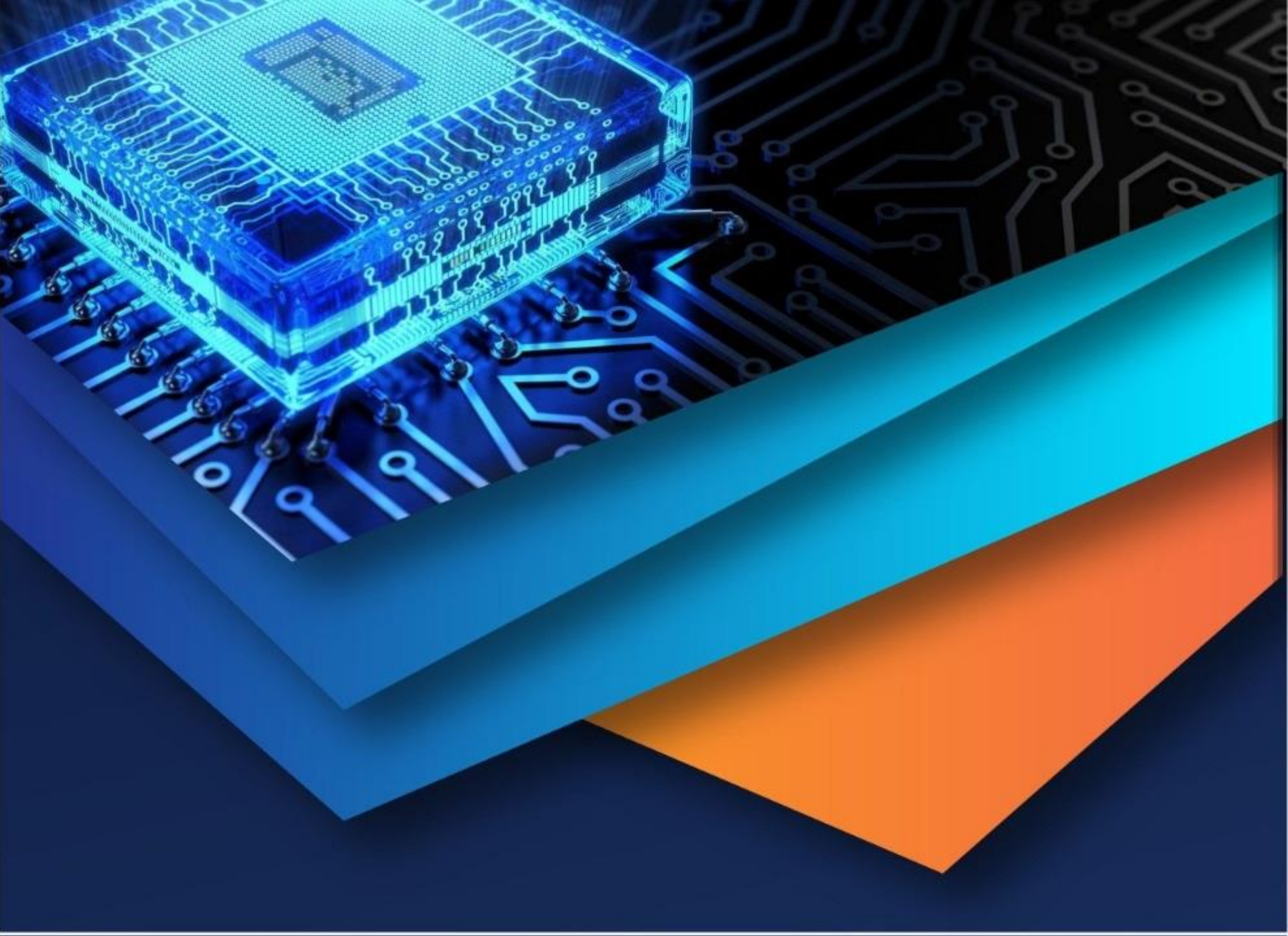
A shared cache with an intelligent replacement policy can achieve the effect of DMA cache scheme.

V. CONCLUSION

By understanding the dynamics of DMA capabilities and being aware of their strengths and weaknesses, the user can enhance performance criteria for his or her PC whether used for personal purposes or for business. The information provided above can applied to the specific needs and requirements of the user for improvement of system performance and efficiency.

REFERNCES

[1] www.youtube.com
[2] www.wikipedia.org
[3] www.google.co.in



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY