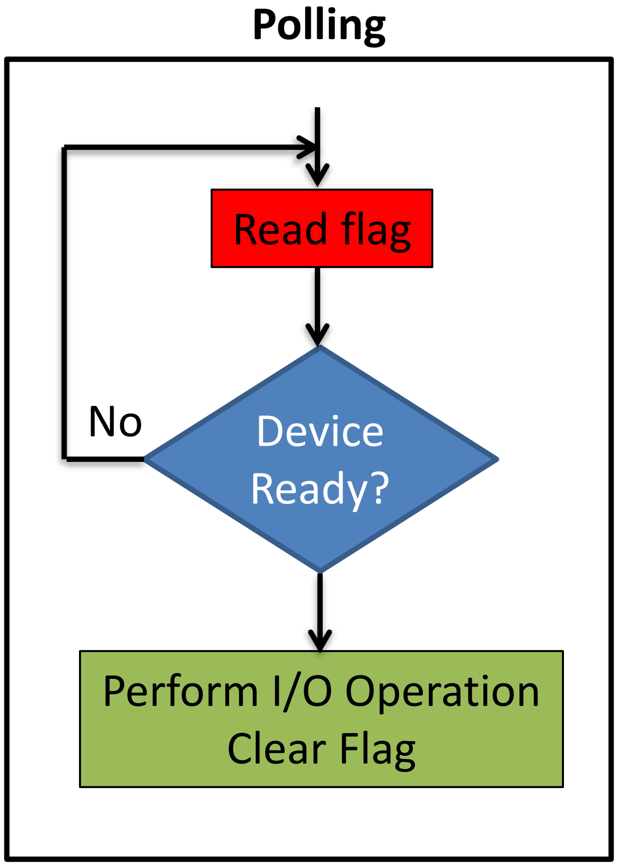
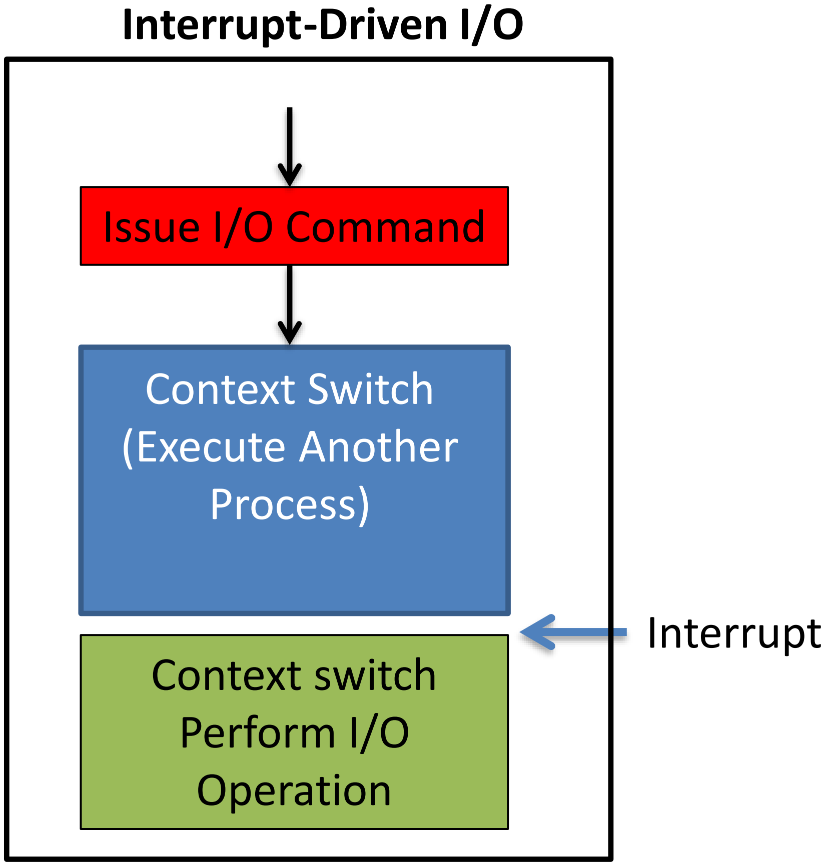
Limitation of Polling and Interrupt-Driven I/O

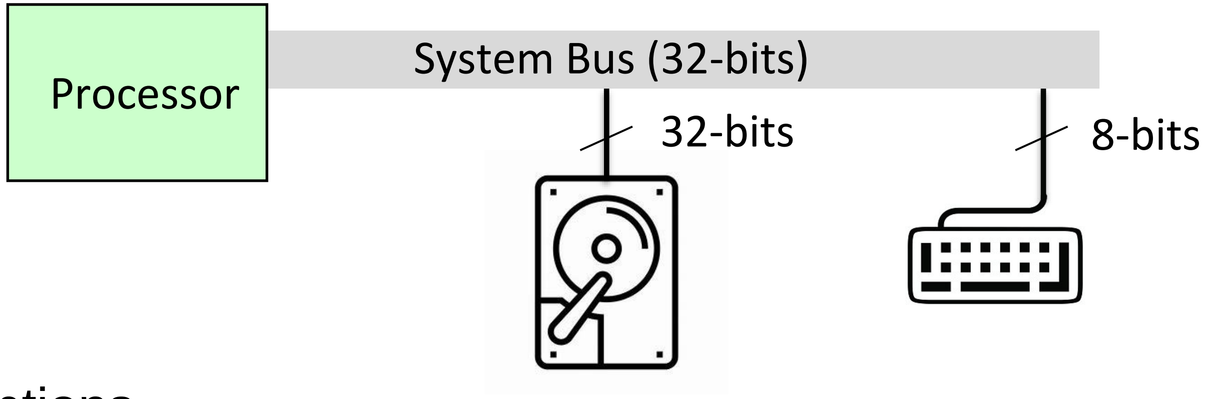




Very inefficient for data transfer to send/receive to/from memory via processor

Processor Time Allocated to Polling

- Determine the percentage of processor time required to poll each device.



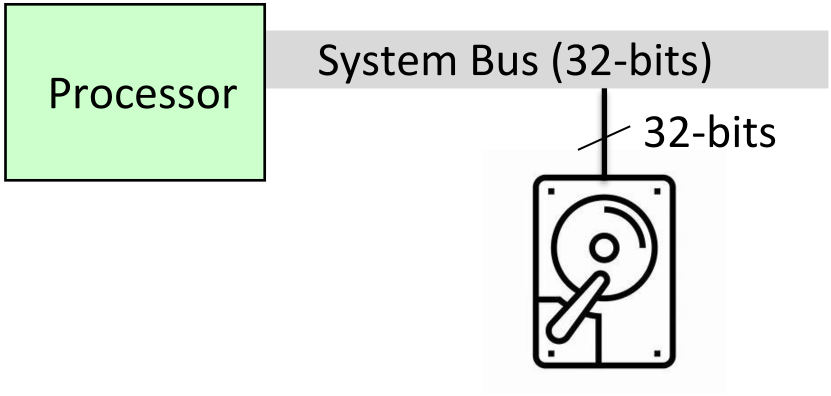
- Assumptions
  - Processor can execute 10 million instructions per second, and a 32-bit bus
  - Disk has a maximum transfer rate of 1 million bytes per second, and has a 32-bit bus
  - The keyboard is polled 30 times per second
  - The polling operation for each I/O device requires 20 instructions

Processor Time Allocated to Polling

- Determine the percentage of processor time required to poll each device.

Disk Transfer Time with Interrupts

- How many words can be transferred between the processor and the disk in one second?



- Assumptions
  - Processor has a 2.5 GHz clock
  - Context switch takes 1000 clock cycles
  - ISR takes 10,000 cycles to run

Processor Time Allocated to Polling

- How many words can be transferred per second between the processor and the disk?

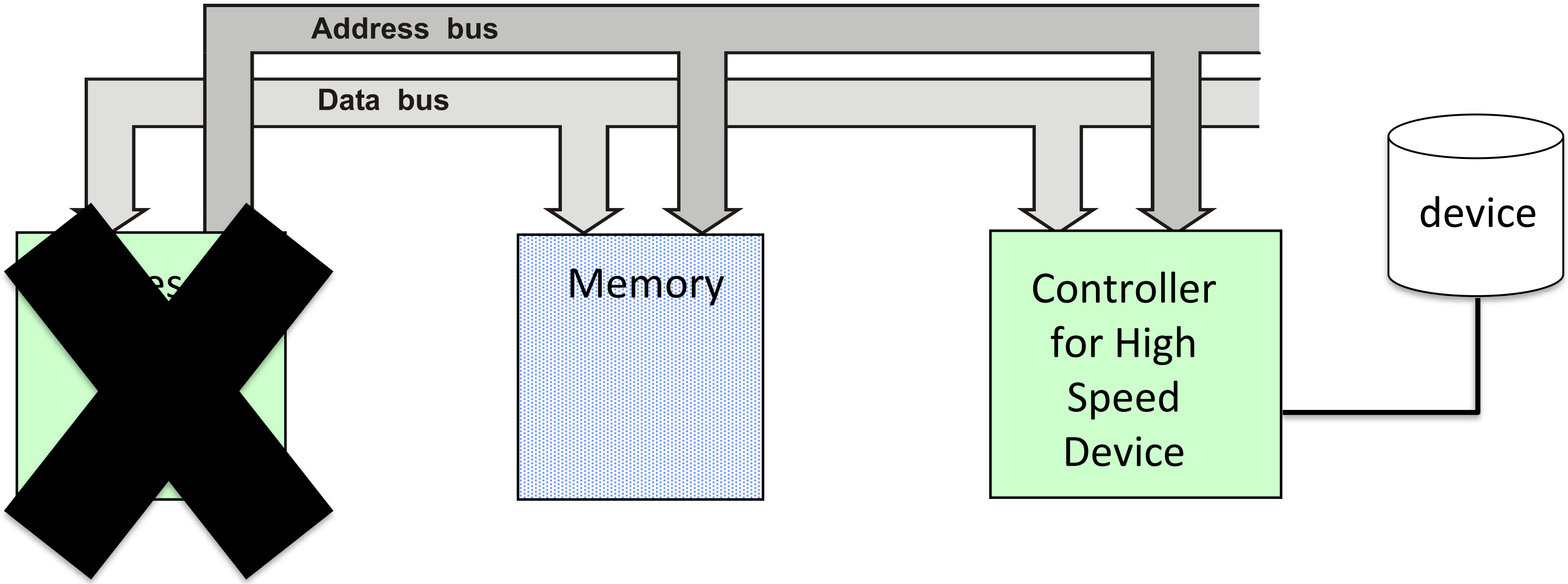
Motivation for Direct Memory Access

- Many streaming devices require large amounts of data to be transferred quickly between an I/O device and memory
  - If the processor handles the data transfer, the application may be blocked until the transfer is complete
  - Depending on the amount of data and transfer speed, the data transfer may be extremely slow

shutterstock.com • 1218330862

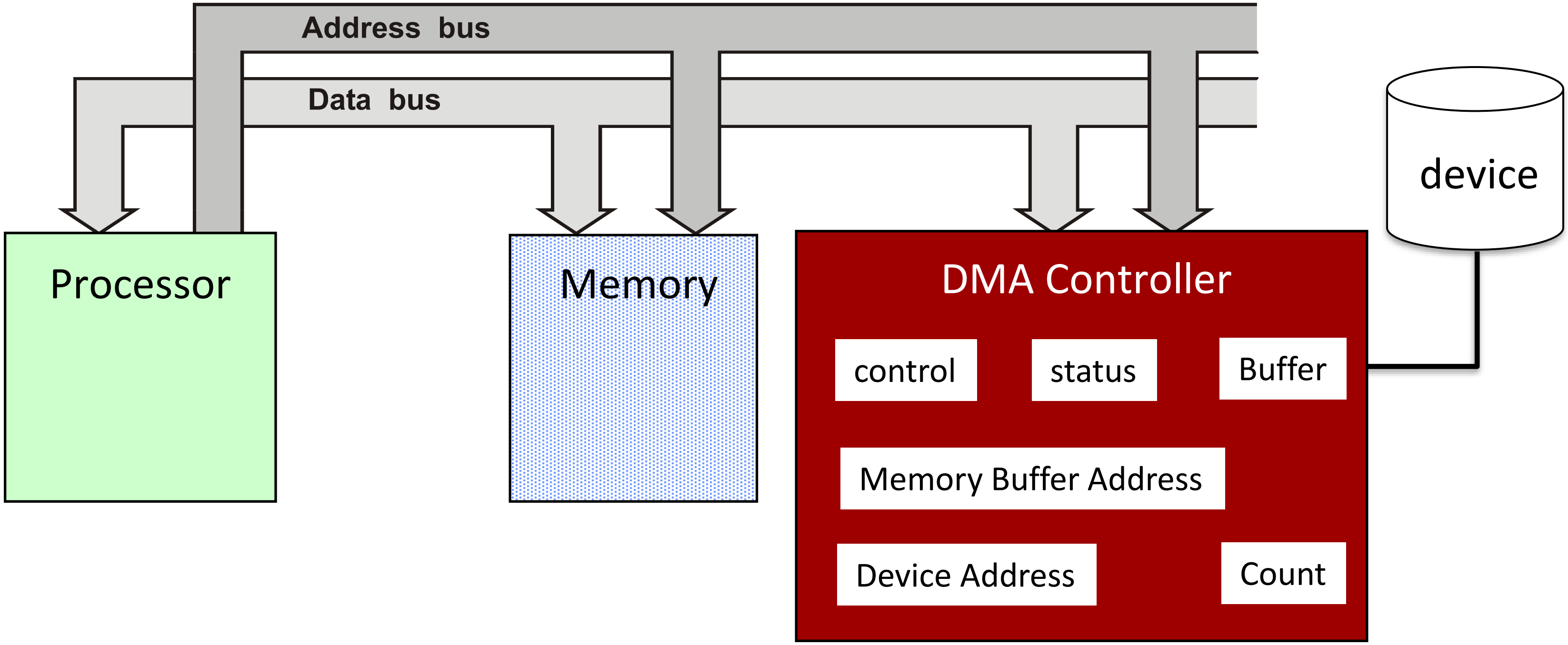
Direct Memory Access (DMA)

- Direct-Memory Access
  - Allow the peripheral to communicate directly with the memory bypassing the CPU



Direct Memory Access (DMA)

- Direct-Memory Access
  - Allow the peripheral to communicate directly with the memory bypassing the CPU

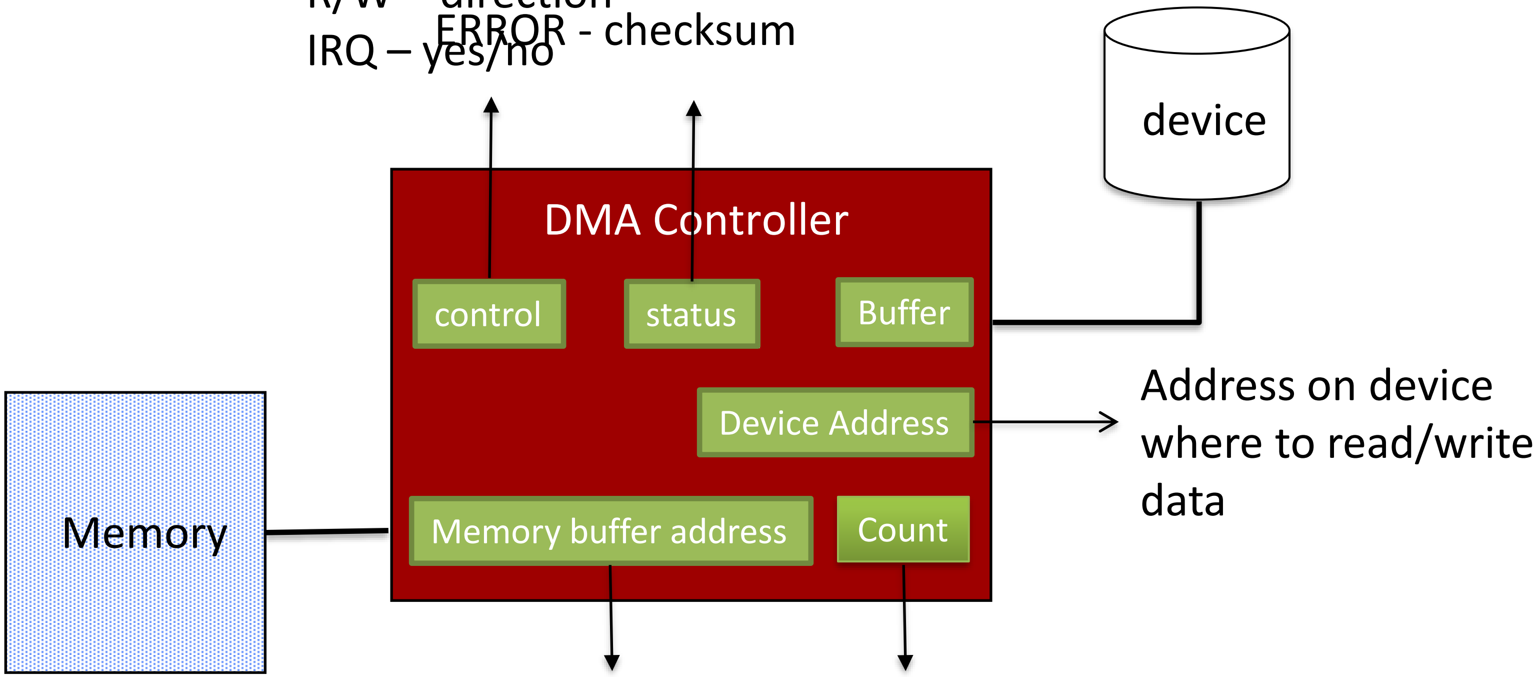


DMA Controller

- All of the registers are memory-mapped and can be accessed using simple load and store operations

GO – start transfer
R/W – direction
IRQ – yes/no

DONE – transfer complete
ERROR - checksum



Address on device where to read/write data

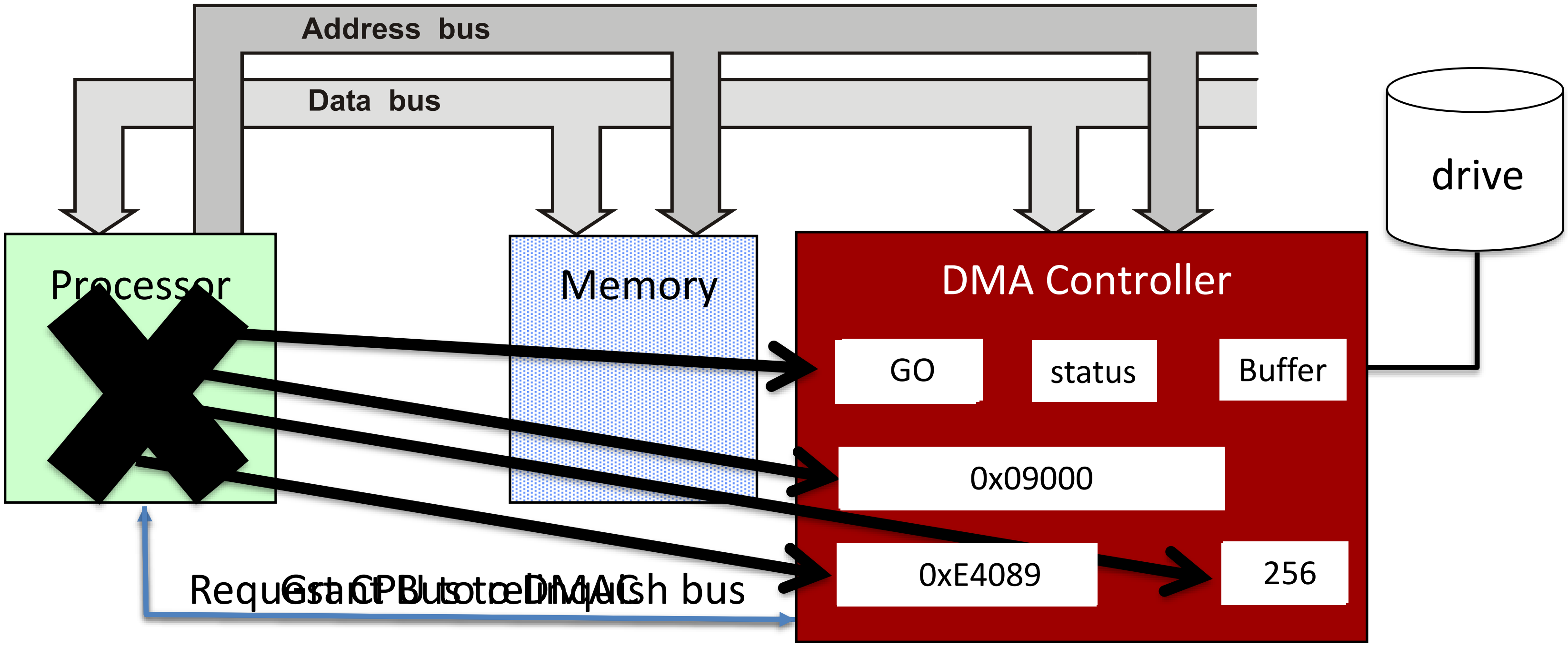
Location to read data from or write data to

Number of *bytes* of data to transfer

Slides are IP. Please do not copy, share, or post to website(s).

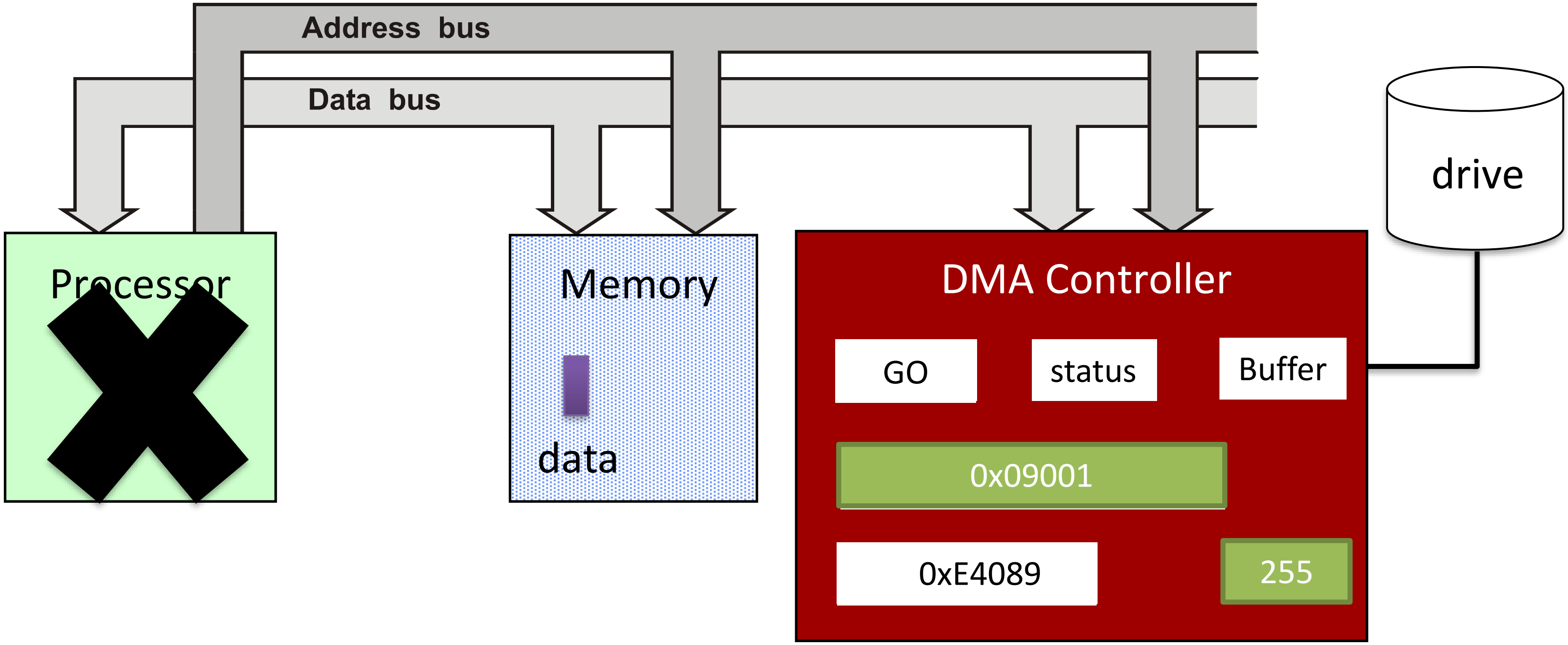
A Disk Write based on DMA

- Transfer 256 bytes of data from a memory buffer at address 0x09000 to the disk at address 0xE4089.



A Disk Write based on DMA

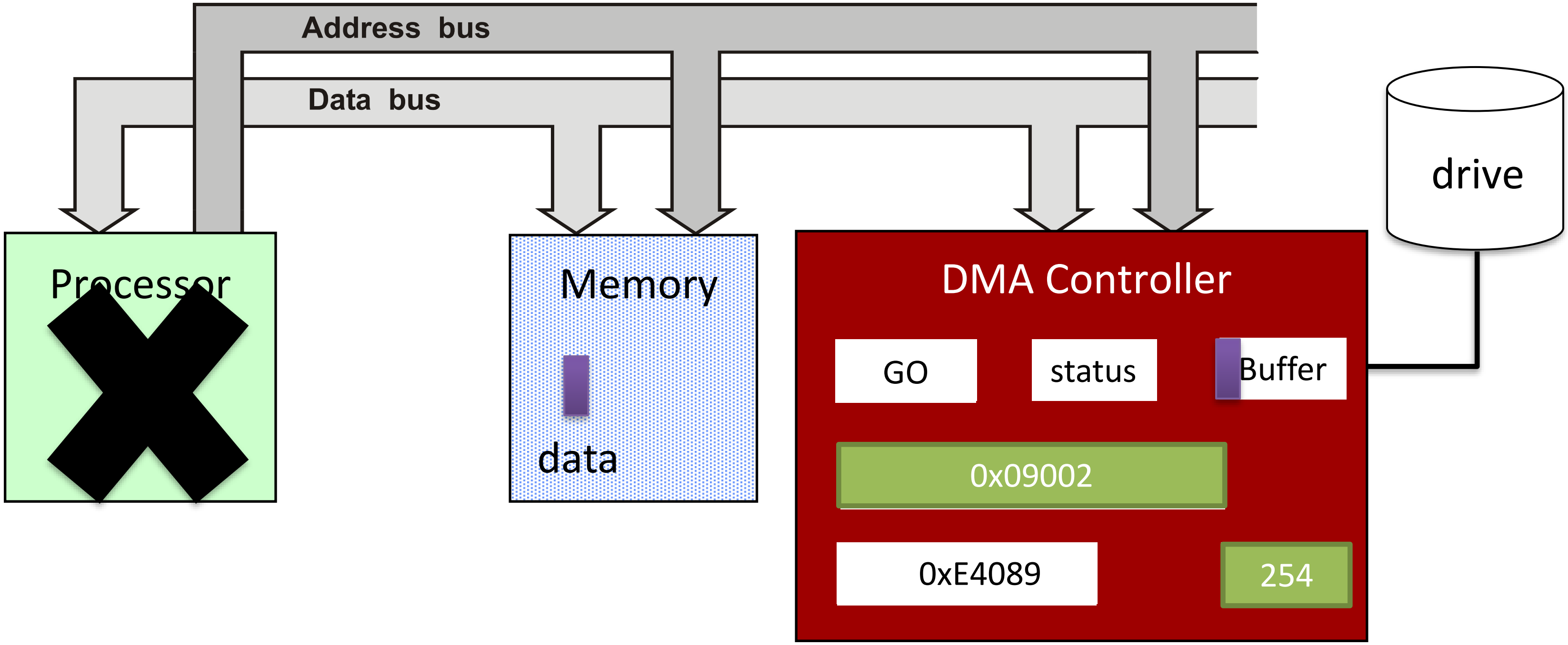
- Transfer 256 bytes of data from a memory buffer at address 0x09000 to the disk at address 0xE4089.



Slides are IP. Please do not copy, share, or post to website(s).

A Disk Write based on DMA

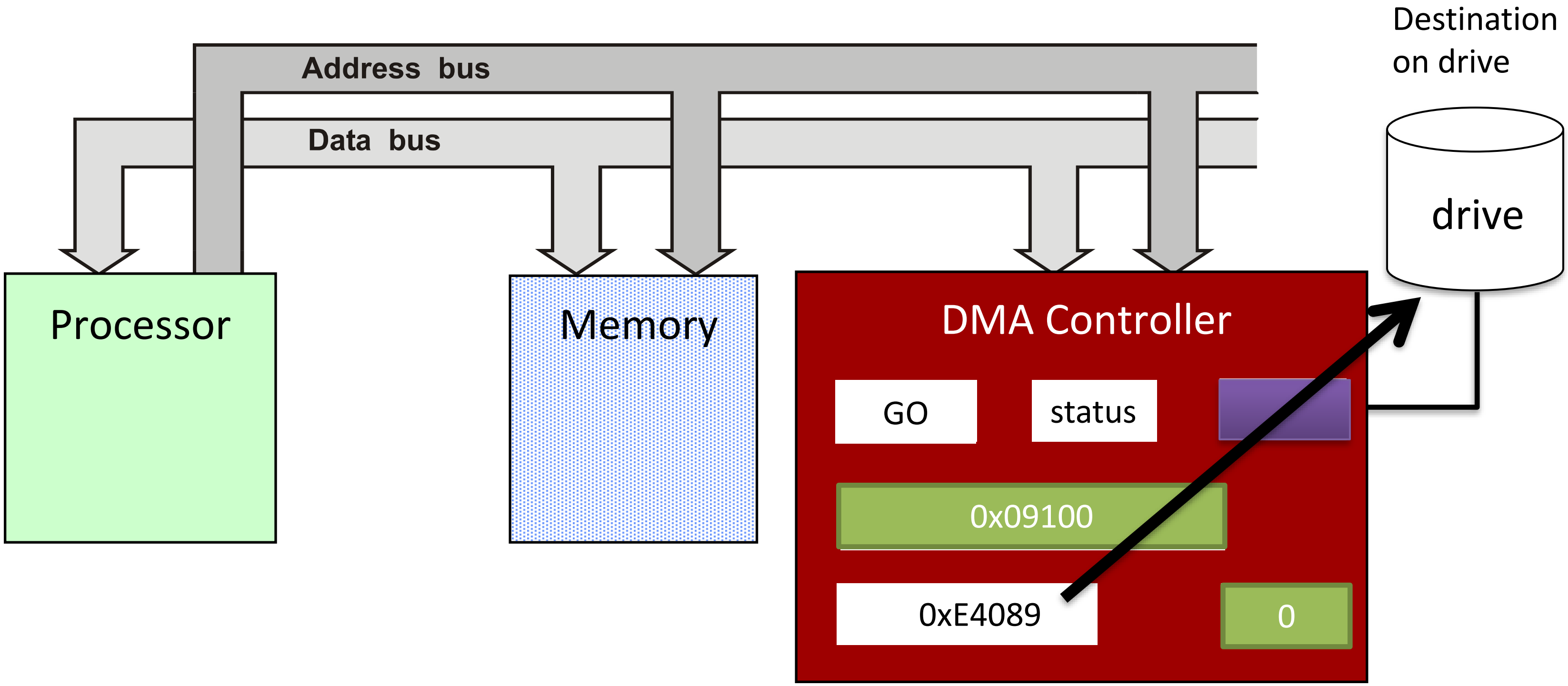
- Transfer 256 bytes of data from a memory buffer at address 0x09000 to the disk at address 0xE4089.



Slides are IP. Please do not copy, share, or post to website(s).

A Disk Write based on DMA

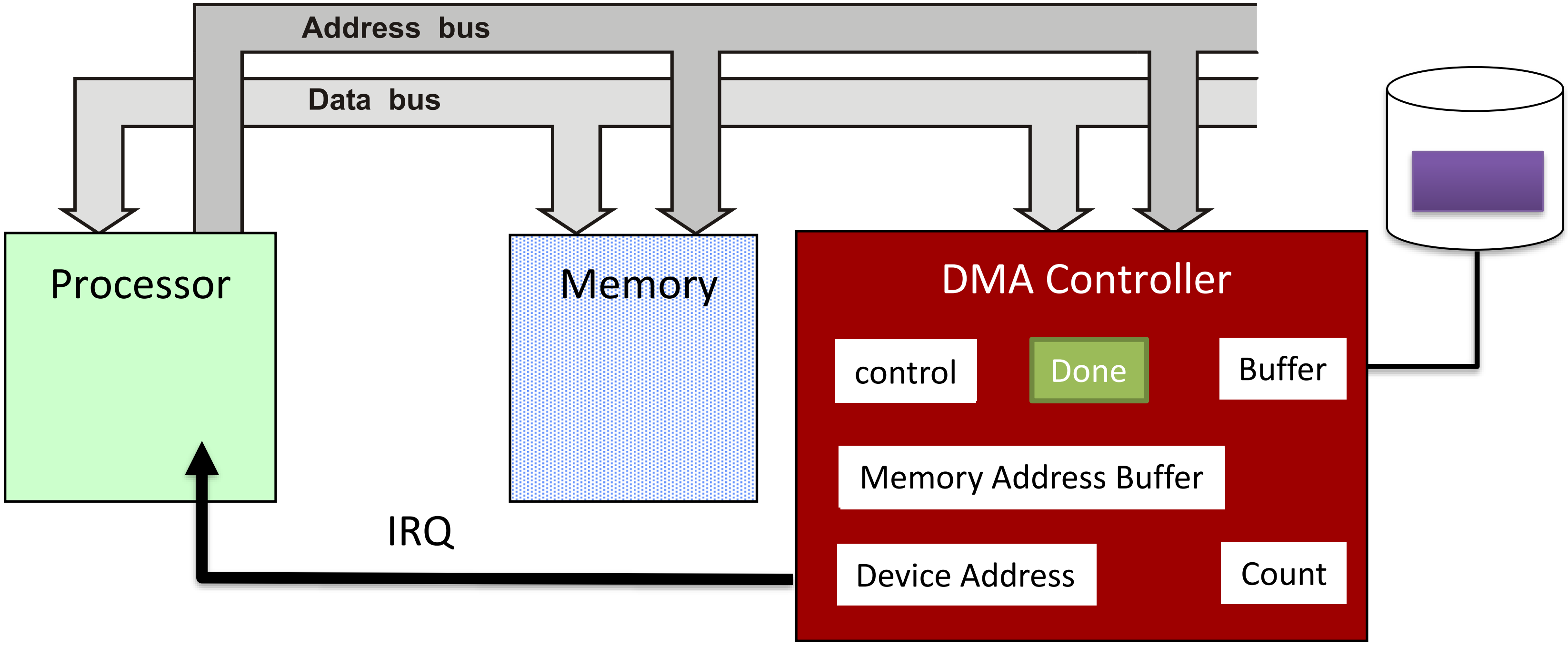
- Transfer 256 bytes of data from a memory buffer at address 0x09000 to the disk at address 0xE4089.



Slides are IP. Please do not copy, share, or post to website(s).

A Disk Write based on DMA

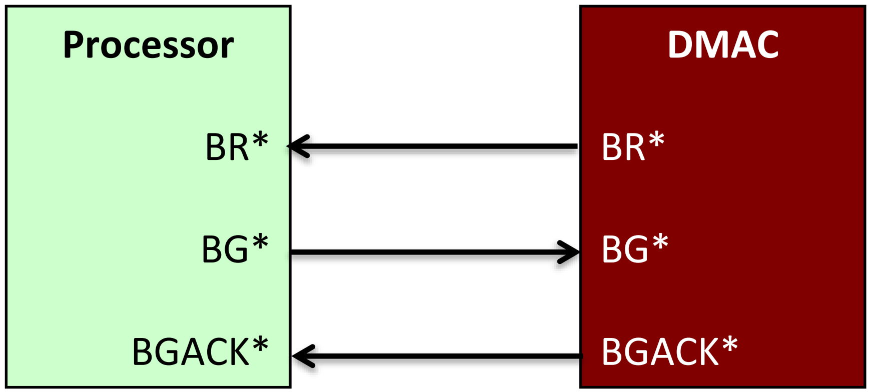
- Transfer 256 bytes of data from a memory buffer at address 0x09000 to the disk at address 0xE4089.



Slides are IP. Please do not copy, share, or post to website(s).

How can the DMAC become Bus Master?

- The DMAC must become bus master before any transfer can occur



Processor

BR*

BG*

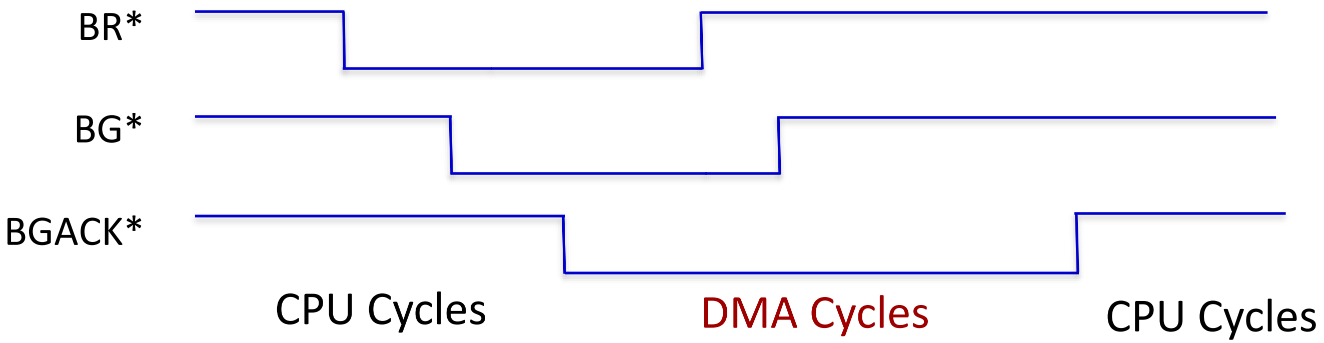
BGACK*

DMAC

BR*

BG*

BGACK*



BR*

BG*

BGACK*

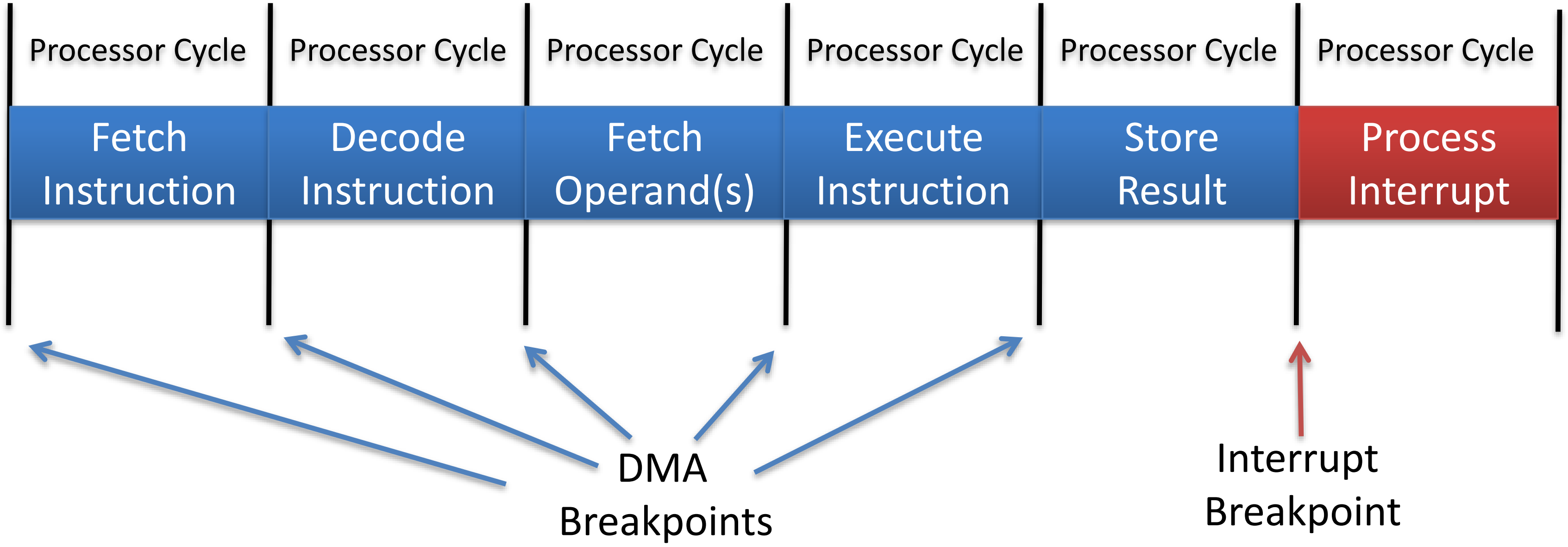
CPU Cycles

DMA Cycles

CPU Cycles

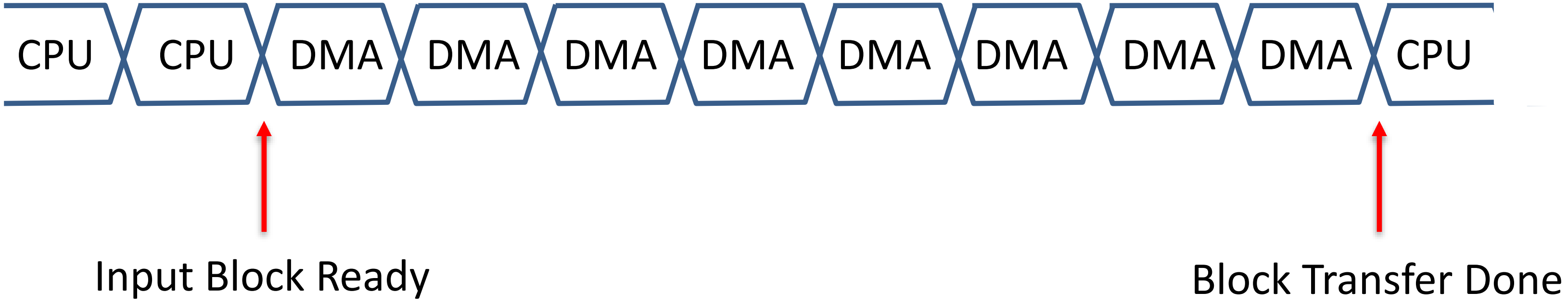
When can the DMAC become Bus Master?

- The DMAC can become master of the bus at almost any point in the instruction cycle!



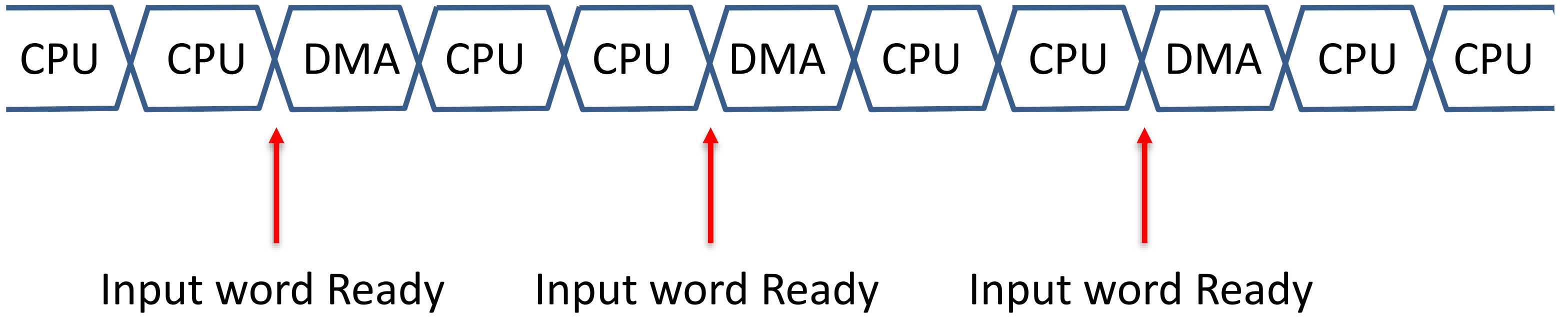
Transfer Mode: Burst

- Burst Mode
  - Bandwidth of I/O = bandwidth of bus
    - Device acquires bus
    - CPU remains idle for entire duration of the transfer
    - All data (block of words) is transferred between device and memory
    - Very fast transfers, but low CPU utilization



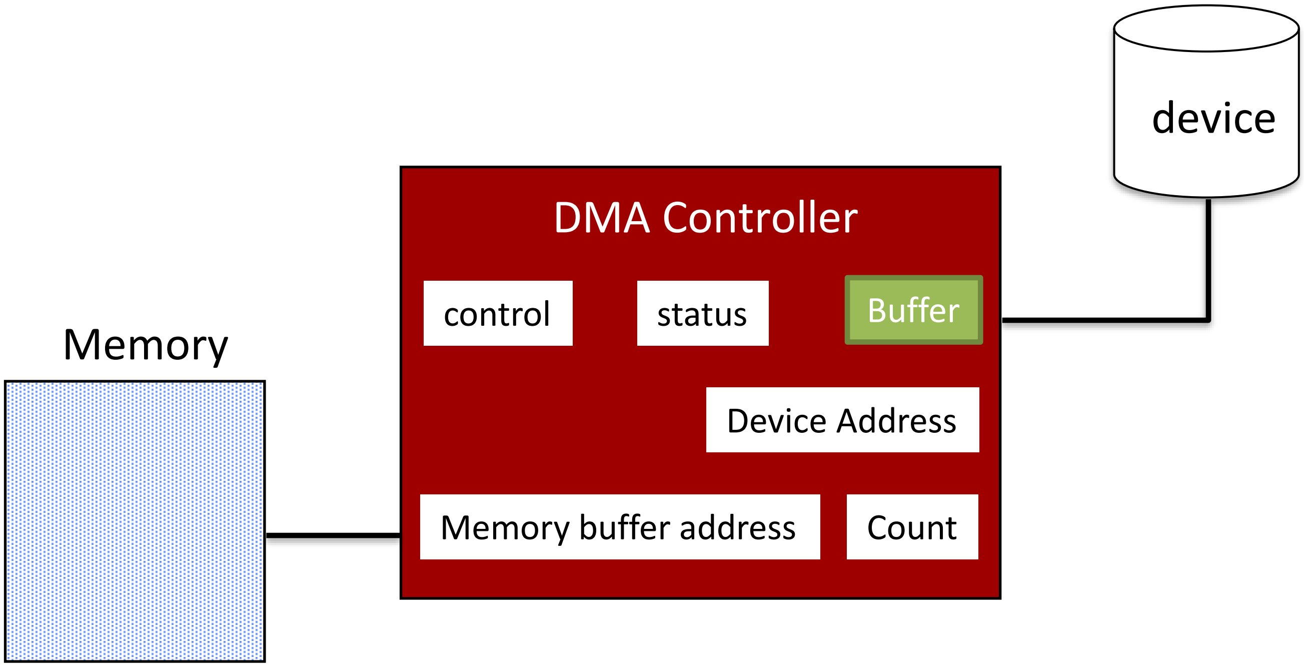
Transfer Mode: Cycle Stealing

- Cycle-Stealing Mode
  - Bandwidth of I/O < bandwidth of bus
    - Device acquires bus
    - CPU remains idle for entire duration of the transfer
    - One word is transferred between device and memory
    - slower transfers, but higher CPU utilization



I/O Buffering

- Buffering
  - Most I/O bandwidths are less than the memory bandwidth
  - I/O buffering in the DMAC can be used to promote the more frequent use of burst mode



Summary

- Polling and Interrupt-Driven I/O
  - Both involve the CPU in data transfers
  - This increases CPU utilization and limits data-transfer rates
- Direct-Memory Access
  - Removes CPU (mostly) from the I/O operation
    - Fewer CPU cycles required for each transfer
  - Data streamed between I/O device and Memory
    - Higher data transfer rates
  - Requires complex hardware (i.e., DMAC)
    - Data transfers can be transferred in bursts (faster transfers, lower CPU utilization) or one word at a time (slower transfers, higher CPU utilization) by stealing cycles from CPU
- DMA transfers can be problematic in systems that employ a memory hierarchy (i.e., multiple caches)