INTERRUPTS(Polled & Daisy chain) & DMA

BY

-A SAI SUSHANTH - 120101001

-VASIREDDY GOPICHAND - 120101076

(SCRIBE DATE-10-04-2014)

REVIEW:

Interrupt:

- An **interrupt** is a signal to the processor emitted by hardware or software indicating an event that needs immediate attention.

Working:

- An interrupt alerts the processor to a high-priority condition requiring the interruption of the current code the processor is executing (the current *thread*). The processor responds by suspending its current activities, saving its state, and executing a small program called an *interrupt handler* (or interrupt service routine, ISR) to deal with the event. This interruption is temporary, and after the interrupt handler finishes, the processor resumes execution of the previous thread.

INTRODUCTION

There are two types of interrupts:

- Hardware Interrupts
- Software Interrupts

HARDWARE INTERRUPTS:

- A **hardware interrupt** is an electronic alerting signal sent to the processor from an external device, either a part of the computer itself such as a disk controller or an external peripheral.
- Hardware interrupts are asynchronous and can occur in the middle of instruction execution, requiring additional care in programming. The act of initiating a hardware interrupt is referred to as an interrupt request (IRQ).

SOFTWARE INTERRUPTS:

- A **software interrupt** is caused either by an exceptional condition in the processor itself, or a special instruction in the instruction set which causes an interrupt when it is executed. The former is often called a *trap* or *exception* and is used for errors or events occurring during program execution that are exceptional enough that they cannot be handled within the program itself.
- Software interrupt instructions function similarly to subroutine calls and are used for a variety of purposes, such as to request services from low level system software such as device drivers.

POLLED INTERRUPTS:

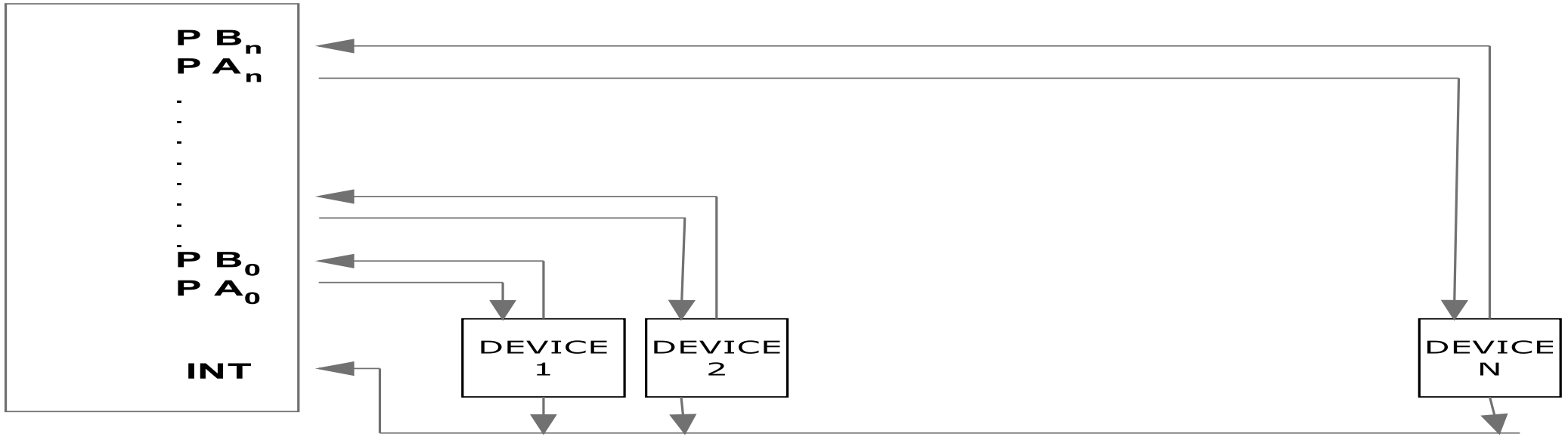
- Polled interrupts are handled by software.
- The processor responds to an interrupt by executing one general service routine for all devices
- The priority of devices are determined by the order in which the routine polls each device. The processor checks the status of each device in the general service routine, starting with the highest priority device, to service an interrupt.
- Once the processor determines the source of the interrupt, it branches to the service routine for the device.
- When one or more devices activate the INT line HIGH, the processor pushes the program counter and possibly some other registers on to the stack. It then branches to an address defined by the manufacturer of the processor.
- The user can write a program at this address to poll each device, starting with the highest priority device, to find the source of the interrupt.
- Polled interrupts are slow and for a large number of devices, the time required to poll each device may exceed the time to service the device.

DAISY CHAIN INTERRUPTS:

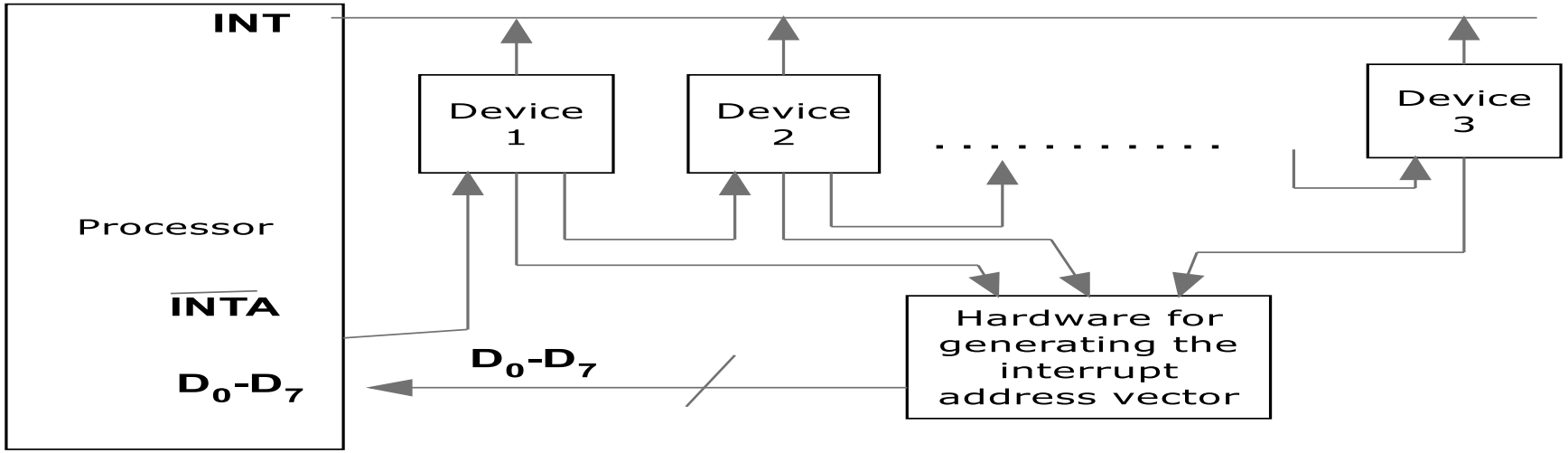
- The daisy chaining technique of building priority consists of a serial association of all devices that request an interrupt. The device with the very best priority is placed within the initial position, followed by lower priority devices up to the device with the bottom priority that is placed last within the chain.

Working:

- Suppose one or more devices interrupt the processor. In response, the processor pushes at least the Program Counter and generates an interrupt acknowledge($\overline{INTA}$) signal to the highest priority device ()
- If this device has generated the interrupt it will accept the INTA otherwise it will pass the INTA on to the next device until the INTA is accepted
- Once accepted the device provides a means for the processor to find the interrupt address vector by using external hardware.



Polled Interrupts



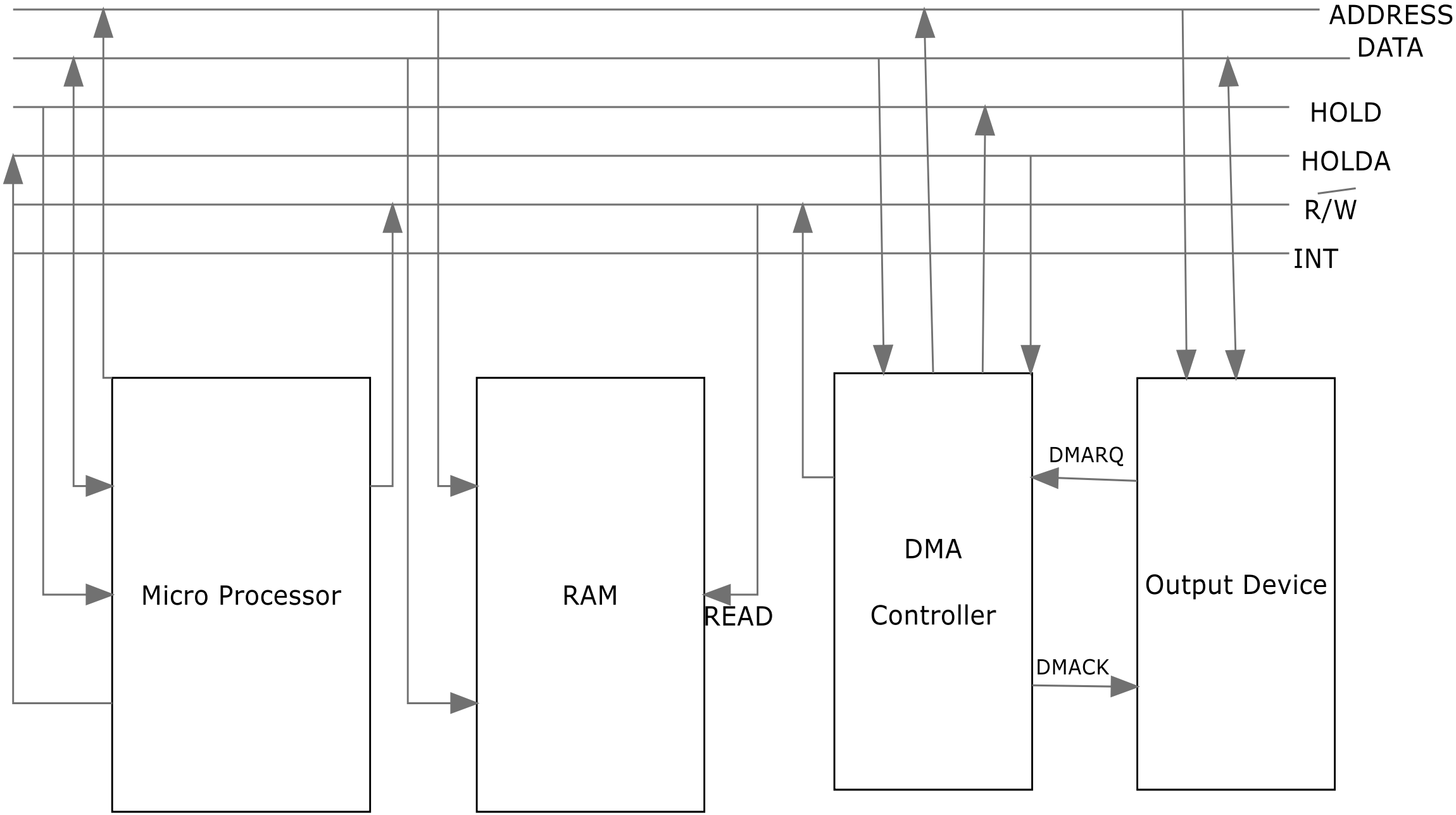
Daisy Chain Interrupt

Comparison between Polled and Daisy chain interrupts:

- The polled interrupt has a disadvantage that it consumes a lot of time, while the daisy chain is more efficient.
- The daisy chain has the disadvantage that the device nearest to the CPU would have highest priority. So, usually those devices which require higher priority would be connected nearer to the CPU.

DMA-DIRECT ACCESS MEMORY

- **Direct memory access** (**DMA**) is a feature of modern computers that allows certain hardware subsystems within the computer to access system memory independently of the central processing unit (CPU).
- Computers that have DMA channels can transfer data to and from devices with much less CPU overhead than computers without DMA channels.
- This is typically slower than copying normal blocks of memory since access to I/O devices over a peripheral bus is generally slower than normal system RAM.
- During this time the processor would be unavailable for any other tasks involving processor bus access. But it can continue to work on any work which does not require bus access.
- DMA transfers are essential for high performance embedded systems where large chunks of data need to be transferred from the input/output devices to or from the primary memory.


ADDRESS
DATA
HOLD
HOLDA
R/W
INT
Micro Processor
RAM
READ
DMA
Controller
DMARQ
DMACK
Output Device

TYPES OF DMA

The three basic types of DMA are

- block-transfer
- cycle-stealing
- inter-leaved DMA

Block transfer DMA:

- This is the most common type of DMA used with microprocessors. As mentioned before in this type of DMA the peripheral device request the DMA transfer via DMA request line , which is connected directly or through a DMA controller chip to the microprocessor . The microprocessor completes the current instruction and sends a DMACK to the peripheral device in order to indicate that the bus can be used for DMA operation. The DMA controller chip then completes the DMA transfer and transfers the control of the bus to the microprocessor.

Cycle stealing DMA:

- In this technique, the DMA controller transfers a byte of data between the memory and peripheral device by stealing a clock cycle of the microprocessor. The DMA controller will complete the transfer by passing the microprocessor and generating proper signals to complete the transfer. Since the microprocessor is operated by an external clock, it is quite simple to stop the microprocessor momentarily.
- This is accomplished by not providing the clock signal to the microprocessor. An INHIBIT signal is used for this purpose, which is normally HIGH and is logically AND with the system clock to generate the microprocessor clock, as shown in Fig.
- The DMA controller stops the microprocessor by lowering the INHIBIT signal to LOW. A timing diagram is shown in Fig. The DMA controller then takes over the control of the microprocessor system bus for the time that microprocessor is stopped. Using cycle stealing, data is transferred 1 byte at a time. The DMA controller requests the microprocessor for each byte to be transferred.

Interleaved DMA:

- Interleaved DMA is a more complex type of DMA operation using this technique, the DMA controller takes over the system bus when the microprocessor is not using it. For example, the microprocessor doesn't use the bus when it performs internal operations, such as decoding an instruction or ALU operations. The DMA controller takes advantage of those times in order to transfer data, and this called Interleaved DMA.
- One of the main characteristics of Interleaved DMA is that data transfer occurs without stopping the microprocessor. With Interleaved DMA, each data transfer includes 1 byte per instruction cycle.