Analog and Digital Signal

An ***analog*** signal has continuous voltage levels.

If the chip uses operational amplifiers (op-amps), comparators, and integrators to "process" the analog signal, the chip is purely analog.

A ***digital*** signal changes between discrete voltage levels.

Modern processor chips operate in the digital domain, pushing its clock speed to enhance performance.

Serial and Parallel Data Transfer

Parallel data transfer allows more than one-bit of data to be transferred simultaneously between the computer and the external device. Typically synchronous, requiring clock information to govern the data transfer.

Since the data bits are typically adjacent to each other, they may cause serious **interference** at high frequency.

Parallel data transfer usually done between the processor and devices located **nearby** (within few meters).

Also, many wires are needed for parallel data transfer. There may be **synchronization issue**.

Serial data transfer allows one-bit of data to be transferred at a time through 2 or 3 wires.

UART is a circuit implemented in many integrated circuits to enable serial communication.

Serial data transfers, when ***compared*** with parallel data transfers are:

- Less expensive (less wires & connections)
- Typically slower than parallel, but new serial interfaces like USB 3.0 can support speeds up to 5 Gbits/s.
- More robust for long distance.

Data Interface Transfer Techniques

Polled/Programmed I/O:

The CPU **continuously pools**/asks/checks the I/O port for available data or **readiness** of the port.

CPU initiates, controls and transfers data.

Advantage of polled I/O

- Minimum hardware interface circuitry between I/O device and processor.
- Programmer has complete control over the entire process.
- Easiest method to test and debug

Disadvantage of Polled I/O

- Since the CPU waits in a loop, it cannot perform any other task until data transfer is completed.
- Inefficient use of CPU resources.
- program execution of CPU held up while waiting for I/O device to get ready.

Interrupt-driven I/O technique:

Data transfer **initiated by external** device.

An interrupt is an external hardware event.

This event causes the CPU to **interrupt** the current **instruction sequence** (suspend temporarily).

And execute a special routine written by the programmer called the **interrupt service routine (ISR)**.

When the device is READY for data transfer, it **sends an interrupt request** to the CPU. Accordingly, the CPU **executes its ISR**.

The ISR transfers data to/from the device. Typically the ISR is executed in a very short time, so the **main program is suspended for a brief time only.**

Advantage of Interrupt-driven I/O:

- Efficient use of CPU; promptly provide service at the request of the peripheral.
- CPU can continue with other tasks between interrupts.

Disadvantage of Interrupt-driven I/O:

- More hardware interface circuitry required between I/O device and processor.
- program is slightly more complex and difficult to debug.

Types of DMA Transfer

Flyby:

Data is transferred from the source to the destination **simultaneously** (same cycle).

Arbitration of system bus is controlled by DMA controller.

If DMA transfer is not mentioned, flyby technique is assumed as the default.

Fetch-and-Deposit:

Data is transferred from the source to a **temporary data register** internal to the DMA controller.

Data is then **written** to the destination in the **next cycle**.

DMA Modes of Operation

Block Transfer Mode:

The DMA controller gains control of the bus and transfer a block of data before returning control of the bus to the CPU.

When the DMA controller has control of the bus, the CPU is placed into a suspended/sleep state.

- This allows fast data transfer between I/O module and memory.
- Inefficient use of CPU as it is suspended for the duration of the transfer.
- Not suitable for real-time applications.

Cycle Stealing Mode:

The DMA controller steals a bus cycle from the CPU to transfer a word of data.

CPU is suspended just before it accesses bus..

- DMA transfer is interleaved with instruction execution allowing CPU to execute a program whilst DMA is doing the data transfer.
- The time it takes for the CPU to execute a program is increased as bus-cycles are inserted into instruction execution.

Transparent / Hidden-cycle Mode

The DMA controller transfers the data only when the CPU is not using the bus.

DMA gains control of the bus when CPU is decoding or executing the instructions.

- Unlike cycle-stealing, transparent mode does not add bus-cycles into CPU instructions.
- Suitable only when speed of data transfer by the I/O module and memory is much faster than the CPU processing time.

Volatile and Non-Volatile Memory

Volatile:

Data is lost when electric power is removed.

- Temporary.
- Typically used as system memory.
- Random Access Memories such as Static-RAM & Dynamic-RAM.

Non-volatile:

Data is retained even if electric power is removed.

- Permanent.
- Typically used as storage.
- FLASH, magnetic hard-disk and optical compact disc specifically.

RAM

Static Random Access Memory (SRAM)

- Data stored as long as power is applied.
- Large (4~6 transistors per cell)
- Fast.
- Differential.

Dynamic Random Access Memory (DRAM)

- Periodic refresh required.
- Small (1~3 transistors per cell)
- Slower.
- Single-ended.

Floating Gate Technologies

EPROM

- The earliest FAMOS transistors are implemented as EPROM devices.
- programmed using avalanche hot-electron-injection mechanism
- The device needs to be put under ultra-violet (UV) light to erase the programmed state.
- The UV light makes the polysilicon layer conductive so that the store charges may dissipate away.

E2PROM

- Advancement in process technologies allow the oxide thickness to be reduced, and the introduction of E2PROM devices.
- Electrons may travel to and from the floating gate by the Fowler-Nordheim Tunneling mechanism with the applied voltage.
- Can electrically program or erase device.

FLASH

- Reduce oxide thickness even further
- programmed using avalanche hot-electron-injection mechanism and erased using Fowler-Nordheim Tunneling.

Cache Mapping Schemes

Direct mapping

Many blocks of main memory map to a single block of cache. Each main memory address is divided into fields.

- A tag field distinguishes one cached memory block from another.
- A block field identifies the cache block.
- An offset field points to the desired data in the block.

In a direct mapped cache consisting of N blocks of cache, block X of main memory maps to cache block Y = X mod N.

Fully Associative Cache Mapping

- A memory address is partitioned into two fields: the tag and the word.
- During an access, all tags in the cache are searched in parallel to retrieve the data quickly.

Set Associative Cache Mapping

- An N-way set associative cache combines the ideas of direct mapped cache and fully associative cache. It is like direct mapped cache in that a memory address maps to a particular set in cache.
- In set associative cache mapping, a memory reference is divided into three fields: tag, set and offset.
- The offset field chooses the word within the cache block, and the tag field uniquely identifies the memory address.
- The set field determines the set to which the memory block maps.

Magnetic Hard Disk

- Platters on a common spindle,
- covered with thin magnetic film,
- rotating on spindle at constant rate.
- Data is stored on the surface of the platter in concentric rings called tracks.
- Tracks are divided into sectors, which are minimum data block size is a sector.

Seek Time (Ts): Time taken for the head to move to the correct track.

Rotational Delay (Tr =0.5/(RPM/60))): Time taken for the disk to rotate until the read/write head reaches the starting position of the target sector.

Access Time (Ta=Ts+Tr): Time from request to the time the head is in position

Transfer Time (Tt =N/(RPM/60*Dt*Ds)): Time required to transfer the required data after the head is positioned.

Ttotal=Taccess+Ttransfer