Module

1

Introduction

Lesson

4

Embedded Systems Components Part II

Overview on Components

Instructional Objectives

After going through this lesson the student would

- Overview of the following
    - Processors
    - Memory
    - Input/Output Devices

Pre-Requisite

Digital Electronics, Microprocessors

You are now almost familiar with the various components of an embedded system. In this chapter we shall discuss some of the general components such as

- Processors
- Memory
- Input/Out Devices

Processors

The central processing unit is the most important component in an embedded system. It exists in an integrated manner along with memory and other peripherals. Depending on the type of applications the processors are broadly classified into 3 major categories

1. General Purpose Microprocessors
2. Microcontrollers
3. Digital Signal Processors

For more specific applications customized processors can also be designed. Unless the demand is high the design and manufacturing cost of such processors will be high. Therefore, in most of the applications the design is carried out using already available processors in the market. However, the Field Programmable Gate Arrays (FPGA) can be used to implement simple customized processors easily. An FPGA is a type of logic chip that can be programmed. They support thousands of gates which can be connected and disconnected like an EPROM (Erasable Programmable Read Only Memory). They are especially popular for prototyping integrated circuit designs. Once the design is set, hardwired chips are produced for faster performance.

General Purpose Processors

A general purpose processor is designed to solve problems in a large variety of applications as diverse as communications, automotive and industrial embedded systems. These processors are generally cheap because of the manufacturing of large number of units. The NRE (Non-recurring Engineering Cost: Lesson I) is spread over a large number of units. Being cheaper the manufacturer can invest more for improving the VLSI design with advanced optimized architectural features. Thus the performance, size and power consumption can be improved. Most cases, for such processors the design tools are provided by the manufacturer. Also the supporting hardware is cheap and easily available. However, only a part of the processor capability may be needed for a specific design and hence the over all embedded system will not be as optimized as it should have been as far as the space, power and reliability is concerned.

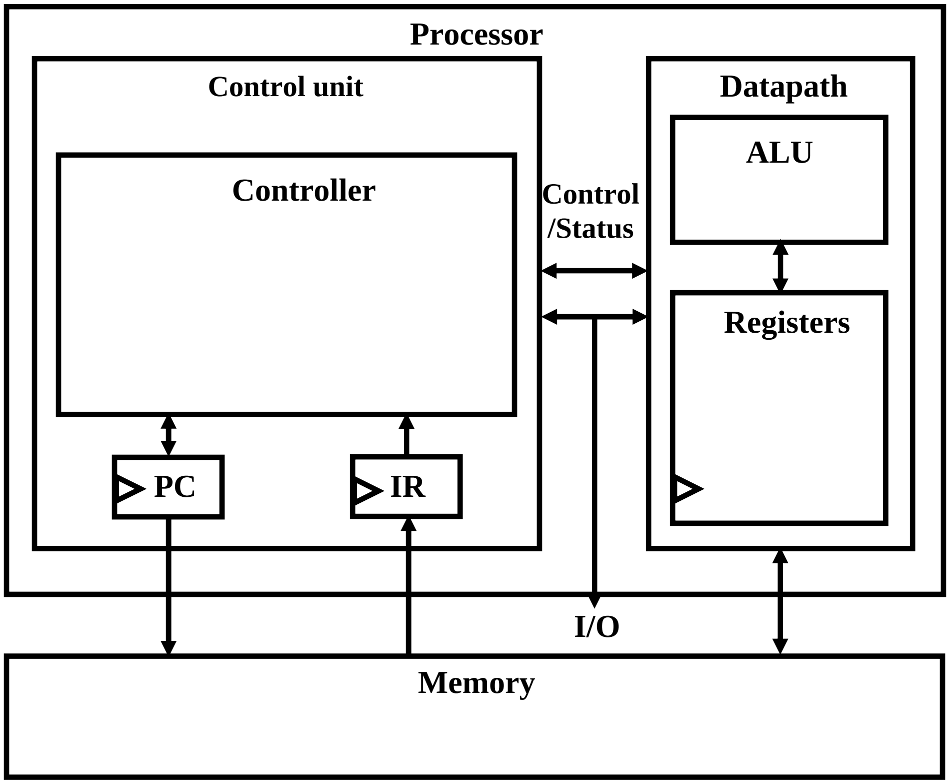


Fig. 4.1 The architecture of a General Purpose Processor

Pentium IV is such a general purpose processor with most advanced architectural features. Compared to its overall performance the cost is also low.
A general purpose processor consists of a data path, a control unit tightly linked with the memory. (Fig. 4.1)

The *Data Path* consists of a circuitry for transforming data and storing temporary data. It contains an arithmetic-logic-unit(ALU) capable of transforming data through operations such as addition, subtraction, logical AND, logical OR, inverting, shifting etc. The data-path also contains registers capable of storing temporary data generated out of ALU or related operations. The internal data-bus carries data within the data path while the external data bus carries data to and from the data memory. The size of the data path indicates the bit-size of the CPU. An 8-bit data path means an 8-bit CPU such as 8085 etc.

The *Control Unit* consists of circuitry for retrieving program instructions and for moving data to, from, and through the data-path according to those instructions. It has a program counter(PC) to hold the address of the next program instruction to fetch and an Instruction register(IR) to hold the fetched instruction. It also has a timing unit in the form of state registers and control logic. The controller sequences through the states and generates the control signals necessary to read instructions into the IR and control the flow of data in the data path. Generally the address size is specified by the control unit as it is responsible to communicate with the memory. For each instruction the controller typically sequences through several stages, such as fetching the instruction from memory, decoding it, fetching the operands, executing the instruction in the data path and storing the results. Each stage takes few clock cycles.

Microcontroller

Just as you put all the major components of a Desktop PC on to a Single Board Computer (SBC) if you put all the major components of a Single Board Computer on to a single chip it will be called as a Microcontroller. Because of the limitations in the VLSI design most of the input/output functions exist in a simplified manner. Typical architecture of such a microprocessor is shown in Fig. 4.2.

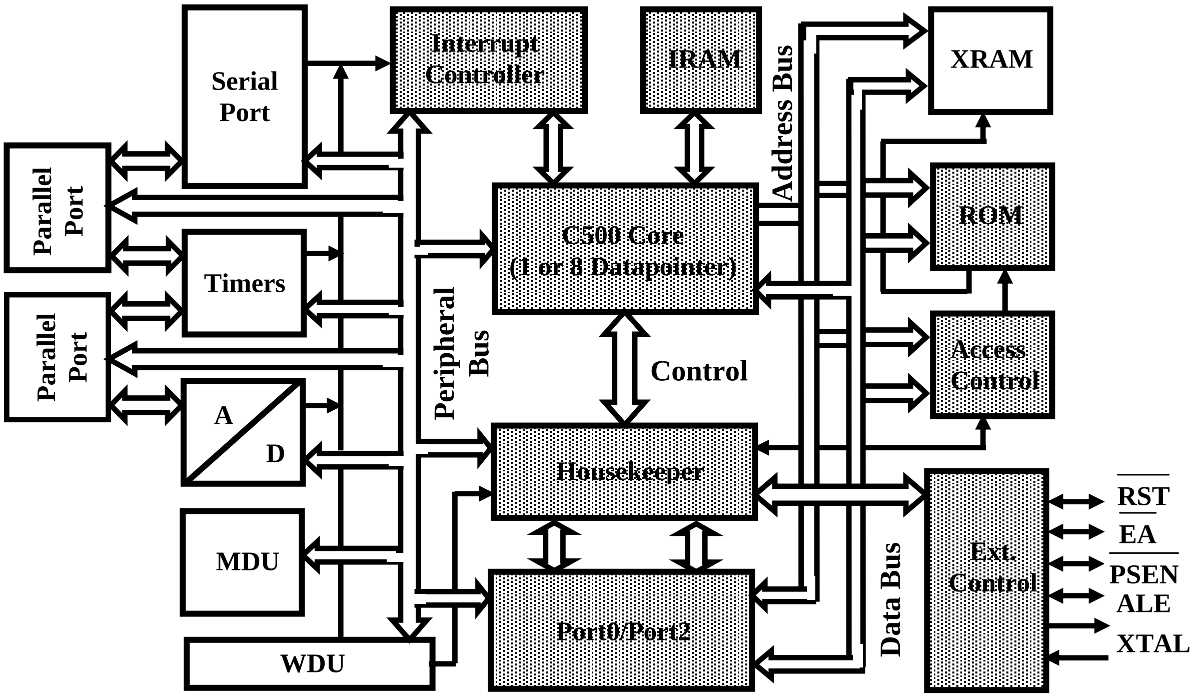


Fig. 4.2 The architecture of a typical microcontroller named as C500 from Infineon Technology, Germany

**The double-lined blocks are core to the processor. Other blocks are on-chip*

The various units of the processors (Fig. 4.2) are as follows:

The C500 Core contains the CPU which consists of the Instruction Decoder, Arithmetic Logic Unit (ALU) and Program Control section

The housekeeper unit generates internal signals for controlling the functions of the individual internal units within the microcontroller.

Port 0 and Port 2 are required for accessing external code and data memory and for emulation purposes.

The external control block handles the external control signals and the clock generation.
The access control unit is responsible for the selection of the on-chip memory resources.
The IRAM provides the internal RAM which includes the general purpose registers.
The XRAM is another additional internal RAM sometimes provided
The interrupt requests from the peripheral units are handled by an *Interrupt Controller Unit.*
Serial interfaces, timers, capture/compare units, *A/D* converters, watchdog units (*WDU*), or a multiply/divide unit (*MDU*) are typical examples for on-chip peripheral units. The external signals of these peripheral units are available at multifunctional *parallel I/O ports* or at dedicated pins.

Digital Signal Processor (DSP)

These processors have been designed based on the modified Harvard Architecture to handle real time signals. The features of these processors are suitable for implementing signal processing algorithms. One of the common operations required in such applications is array multiplication. For example convolution and correlation require array multiplication. This is accomplished by multiplication followed by accumulation and addition. This is generally carried out by Multiplier and Accumulator (MAC) units. Some times it is known as MACD, where D stands for Data move. Generally all the instructions are executed in single cycle.

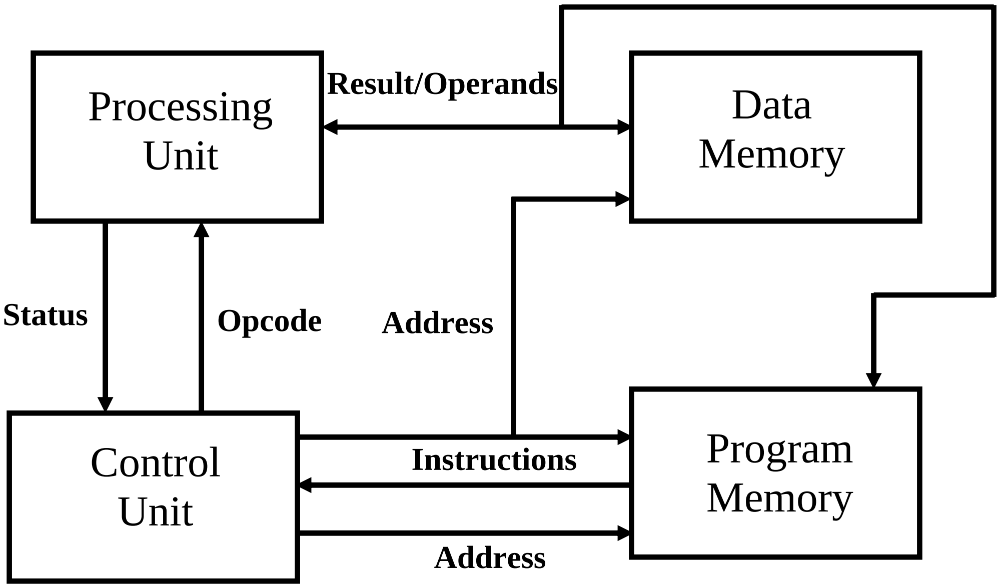


Fig. 4.3 The modified Harvard architecture

The MACD type of instructions can be executed faster by parallel implementation. This is possible by separately accessing the program and data memory in parallel. This can be accomplished by the modified architecture shown in Fig. 4.3. These DSP units generally use Multiple Access and Multi Ported Memory units. Multiple access memory allows more than one access in one clock period. The Multi-ported Memory allows multiple addresses as well Data ports. This also increases the number of access per unit clock cycle.

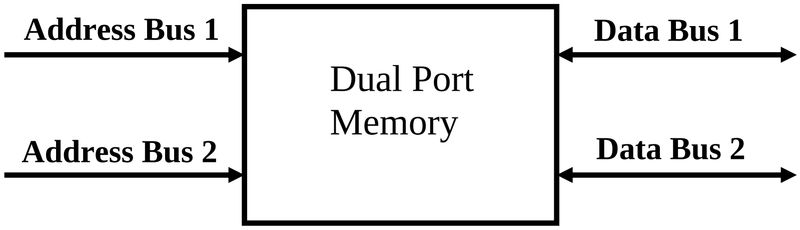


Fig. 4.4 Dual Ported Memory

The Very Long Instruction Word (VLIW) architecture is also suitable for Signal Processing applications. This has got a number of functional units and data paths as seen in Fig. 4.5. The long instruction words are fetched from the memory. The operands and the operation to be performed by the various units are specified in the instruction itself. The multiple functional units share a common multi-ported register file for fetching the operands and storing the results. Parallel random access to the register file is possible through the read/write cross bar. Execution in the functional units is carried out concurrently with the load/store operation of data between RAM and the register file.

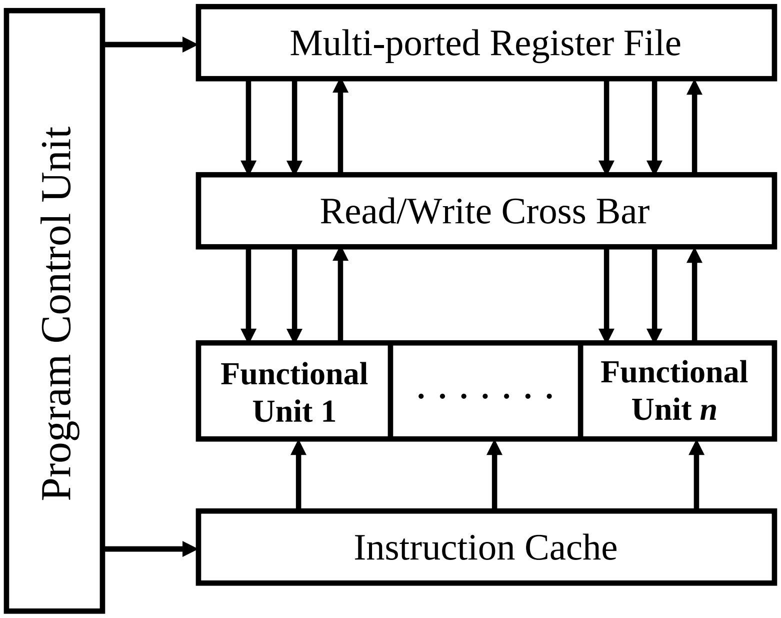


Fig. 4.5 Block Diagram of *VLIW* architecture

Microprocessors vs Microcontrollers

A **microprocessor** is a **general-purpose** digital computer's central processing unit. To make a complete microcomputer, you add memory (ROM and RAM) memory decoders, an oscillator, and a number of I/O devices. The prime use of a microprocessor is to read data, perform extensive calculations on that data, and store the results in a mass storage device or display the results. These processors have complex architectures with multiple stages of pipelining and parallel processing. The memory is divided into stages such as multi-level cache and RAM. The development time of General Purpose Microprocessors is high because of a very complex VLSI design.

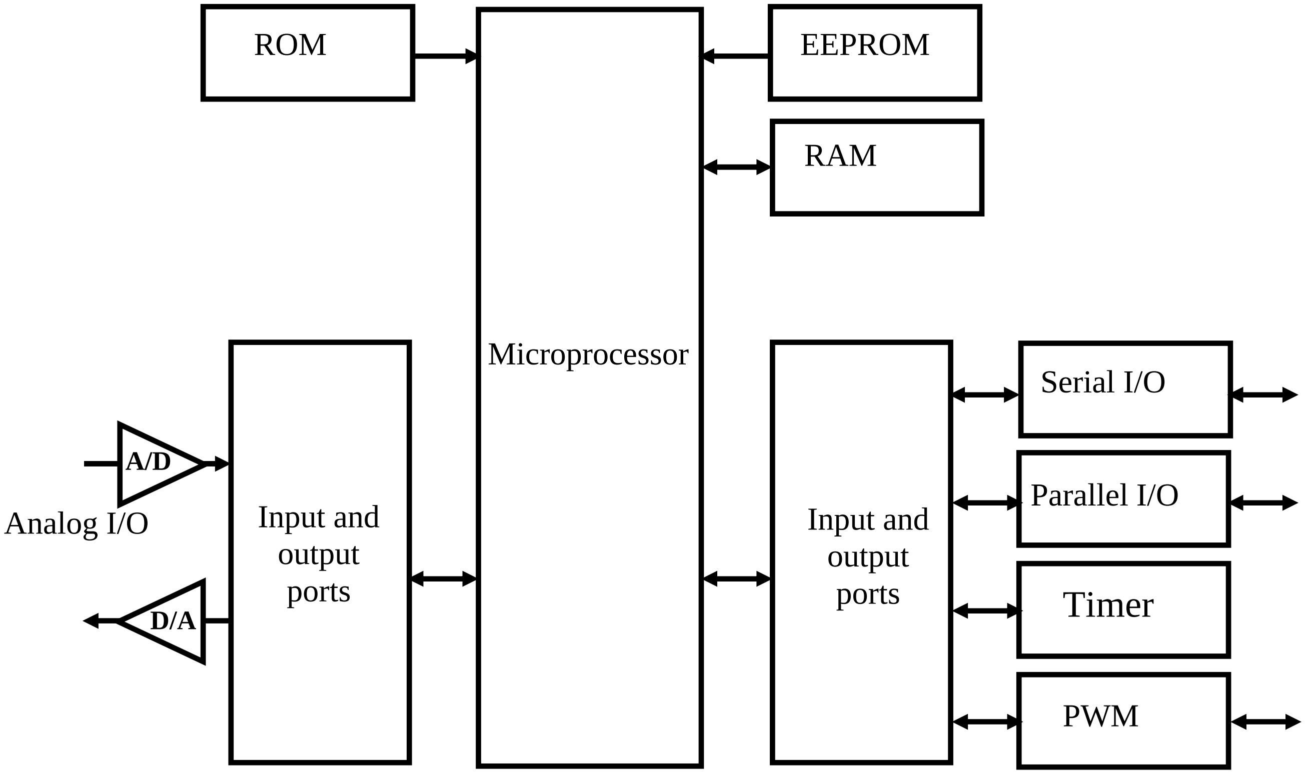


Fig. 4.6 A Microprocessor based System

The design of the **microcontroller** is driven by the desire to make it as expandable and flexible as possible. Microcontrollers usually have on chip RAM and ROM (or EPROM) in addition to on chip i/o hardware to minimize chip count in single chip solutions. As a result of using on chip hardware for I/O and RAM and ROM they usually have pretty low performance CPU. Microcontrollers also often have timers that generate interrupts and can thus be used with the CPU and on chip A/D D/A or parallel ports to get regularly timed I/O. The prime use of a microcontroller is to control the operations of a machine using a fixed program that is stored in ROM and does not change over the lifetime of the system. The microcontroller is concerned with getting data from and to its own pins; the architecture and instruction set are optimized to handle data in bit and byte size.

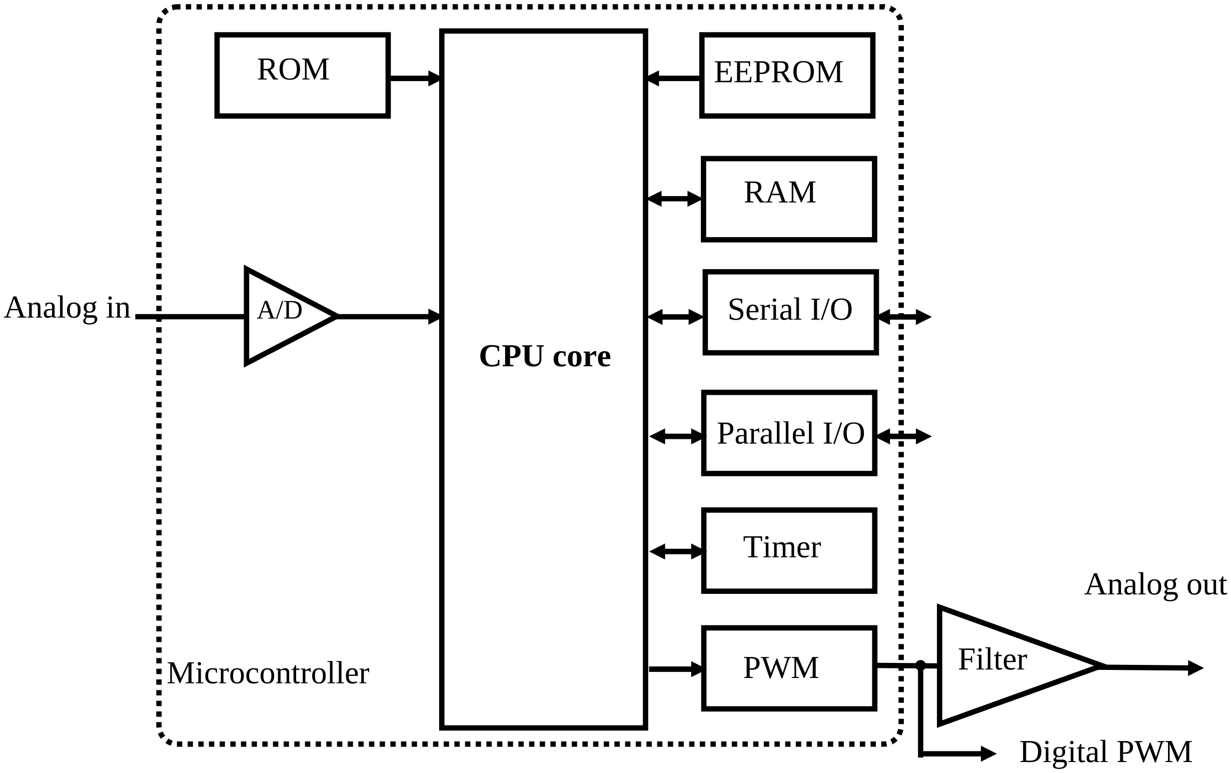


Fig. 4.7 A Microcontroller

The contrast between a microcontroller and a microprocessor is best exemplified by the fact that most microprocessors have many operation codes (opcodes) for moving data from external memory to the CPU; microcontrollers may have one or two. Microprocessors may have one or two types of bit-handling instructions; microcontrollers will have many.

A basic Microprocessors vs a basic DSP

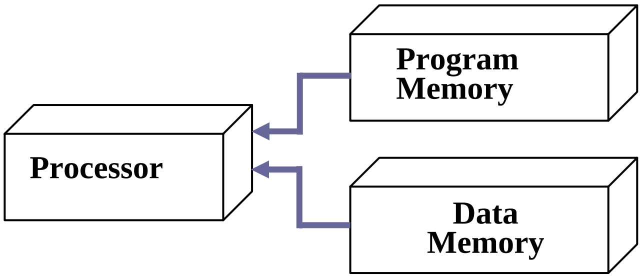


Fig. 4.8 The memory organization in a DSP

DSP Characterization

1. Microprocessors specialized for signal processing applications
2. Harvard architecture
3. Two to Four memory accesses per cycle
4. Dedicated hardware performs all key arithmetic operations in 1 cycle

5. Very limited SIMD(*Single Instruction Multiple Data*) features and Specialized, complex instructions
6. Multiple operations per instruction
7. Dedicated address generation units
8. Specialized addressing [• Auto-increment • Modulo (circular) Bit-reversed]
9. Hardware looping.
10. Interrupts disabled during certain operations
11. Limited or no register Shadowing
12. Rarely have dynamic features
13. Relatively narrow range of DSP oriented on-chip peripherals and I/O interfaces
14. synchronous serial port

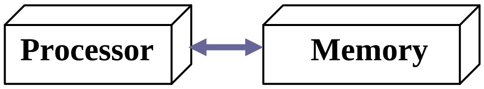


Fig. 4.9 Memory Organization in General Purpose Processor

Characterization of General Purpose Processor

1. CPUs for PCs and workstations E.g., Intel Pentium IV
2. Von Neumann architecture
3. Typically 1 access per cycle
4. Most operations take more than 1 cycle
5. General-purpose instructions Typically only one operation per instruction
6. Often, no separate address generation units
7. General-purpose addressing modes
8. Software loops only
9. Interrupts rarely disabled
10. Register shadowing common
11. Dynamic caches are common
12. Wide range of on-chip and off-chip peripherals and I/O interfaces
13. Asynchronous serial port...

Memory

Memory serves processor short and long-term information storage requirements while registers serve the processor's short-term storage requirements. Both the program and the data are stored in the memory. This is known as *Princeton Architecture* where the data and program occupy the same memory. In *Harvard Architecture* the program and the data occupy separate memory blocks. The former leads to simpler architecture. The later needs two separate connections and hence the data and program can be made parallel leading to parallel processing. The general purpose processors have the Princeton Architecture.

The memory may be Read-Only-Memory or Random Access Memory (RAM). It may exist on the same chip with the processor itself or may exist outside the chip. The on-chip memory is faster than the off-chip memory. To reduce the access (read-write) time a local copy of a portion of memory can be kept in a small but fast memory called the *cache* memory. The memory also can be categorized as Dynamic or Static. Dynamic memory dissipate less power and hence can be compact and cheaper. But the access time of these memories are slower than their Static counter parts. In Dynamic RAMs (or DRAM) the data is retained by periodic refreshing operation. While in the Static Memory (SRAM) the data is retained continuously. SRAMs are much faster than DRAMs but consume more power. The intermediate cache memory is an SRAM.

In a typical processor when the CPU needs data, it first looks in its own data registers. If the data isn't there, the CPU looks to see if it's in the nearby Level 1 cache. If that fails, it's off to the Level 2 cache. If it's nowhere in cache, the CPU looks in main memory. Not there? The CPU gets it from disk. All the while, the clock is ticking, and the CPU is sitting there waiting.

Input/Output Devices and Interface Chips

Typical RTES interact with the environment and users through some inbuilt hardware. Occasionally external circuits are required for communicating with user, other computers or a network.

In the mobile handset discussed earlier the input output devices are, keyboard, the display screen, the antenna, the microphone, speaker, LED indicators etc. The signal to these units may be analog or digital in nature. To generate an analog signal from the microprocessor we need an Digital to Analog Converter(DAC) and to accept analog signal we need and Analog to Digital Converter (ADC). These DAC and ADC again have certain control modes. They may also operate at different speed than the microprocessor. To synchronize and control these interface chips we may need another interface chip. Similarly we may have interface chips for keyboard, screen and antenna. These chips serve as relaying units to transfer data between the processor and input/output devices. The input/output devices are generally slower than the processor. Therefore, the processor may have to wait till they respond to any request for data transfer. Number of idle clock cycles may be wasted for doing so. However, the input-output interface chips carry out this task without making the processor to wait or idle.

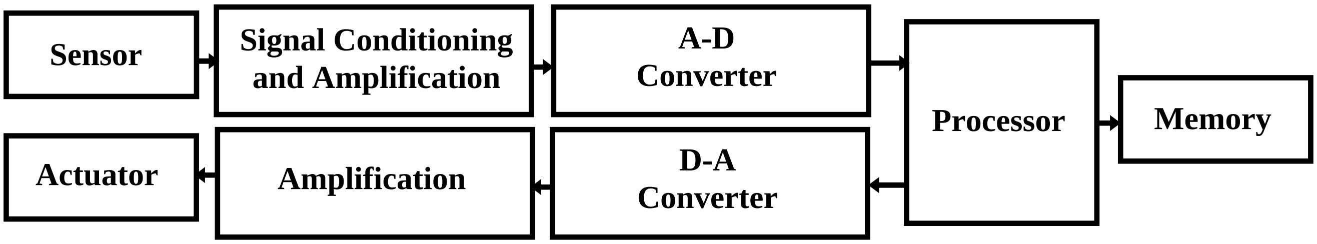


Fig. 4.10 The typical input/output interface blocks

Conclusion

Besides the above units some real time embedded systems may have specific circuits included on the same chip or circuit board. They are known as Application Specific Integrated Circuit (ASIC). Some examples are

1. MODEMs (modulator, demodulator units)

It is used to modulate a digital signal into high-frequency analog signal for wire-less transmission. There are various methods to convert a digital signal into analog form.

Amplitude Shift Keying (ASK)
Frequency Shift Keying (FSK)
Phase Shift Keying (PSK)
Quadrature Phase Shift Keying (QPSK)

The same unit is also used to demodulate the analog signal into digital forms.

2. CODECs (Compress and Decompress Units)

It is generally used to process digital video and/or audio files. A CODEC reduces the amount of data to be transmitted by discarding redundant data on the transmitting end and reconstituting the signal on the receiving end.

3. Filters

Filters are used to condition the incoming signal by eliminating the out-band noise and other unnecessary signals. A specific class of filters called Anti-aliasing filters, are used before the A-D converters to prevent aliasing while acquiring a broad-band signal (signal with a very wide frequency spectrum)

4. Controllers

These are specific circuits for controlling, motors, actuators and light-intensities etc.

Questions-Answers

Q1. Enumerate the similarities and differences between the Microcontroller and Digital Signal Processor

Ans:

Microcontrollers usually have on chip RAM and ROM (or EPROM) in addition to on chip i/o hardware to minimize chip count in single chip solutions. As a result of using on chip hardware for I/O and RAM and ROM they usually have pretty low performance CPU. Microcontrollers also often have timers that generate interrupts and can thus be used with the CPU and on chip A/D D/A or parallel ports to get regularly timed I/O. The prime use of a microcontroller is to control the operations of a machine using a fixed program that is stored in ROM and does not change over the lifetime of the system. The microcontroller is concerned with getting data from and to its own pins; the architecture and instruction set are optimized to handle data in bit and byte size.

Digital Signal Processors have been designed based on the modified Harvard Architecture to handle real time signals. The features of these processors are suitable for implementing signal processing algorithms. One of the common operations required in such applications is array multiplication. For example convolution and correlation require array multiplication. This is accomplished by multiplication followed by accumulation and addition. This is generally carried out by Multiplier and Accumulator (MAC) units. Some times it is known as MACD, where D stands for Data move. Generally all the instructions are executed in single cycle. These DSP units generally use Multiple Access and Multi Ported Memory units. Multiple access memory allows more than one access in one clock period. The Multi-ported Memory allows multiple addresses as well Data ports. This also increases the number of access per unit clock cycle.

Q2. Name few chips in each of the family of processors such as: Microcontroller, Digital Signal Processor, General Purpose Processor

Ans:

Microcontroller: Intel 8051, Intel 80196, Motorola 68705
Digital Signal Processors: TI 3206711, TI 3205000
General Purpose Processor: Intel Pentium IV, Power PC

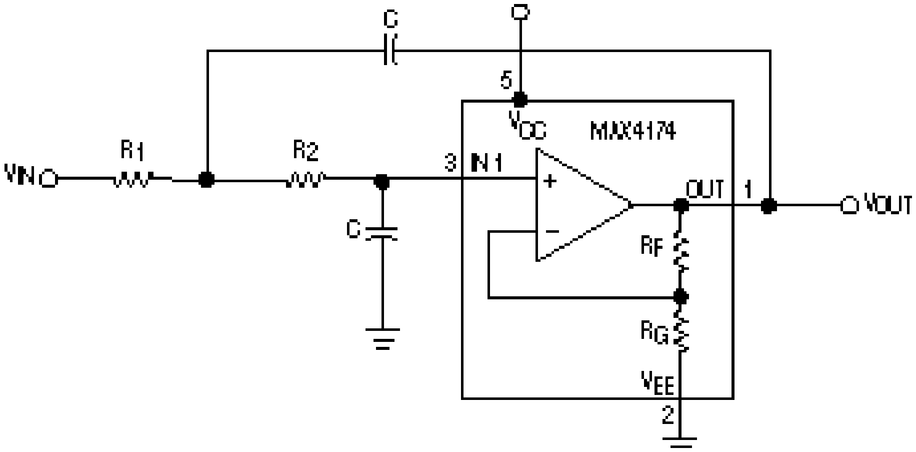
Q3. Enlist the following in the increasing order of their access speed
Flash Memory, Dynamic Memory, Cache Memory, CDROM, Hard Disk, Magnetic Tape, Processor Memory

Ans:

Magnetic Tape, CDROM, Hard Disk, Dynamic Memory, Flash Memory, Cache Memory, Processor Memory

Q4. Draw the circuit of an anti-aliasing Filter using Operational amplifiers

Ans:



Low Pass Sallen Key Butterworth Filter

Q5. Is it possible to implement an anti-aliasing filter in the digital form?

Ans:

No it is not possible to implement an anti-aliasing filter in digital form. Because aliasing is an error introduced at the sampling phase of analog to digital converter. If the sampling frequency is less than twice of the highest frequency present the higher signal frequencies fold back to lower frequency band and hence can be distinguished in the digital/discrete domain.

Q6. Download any free emulator of some simple microcontrollers such as 8051, 68705 etc and learn about it.

Home work

Q7. Draw the internal architecture of 8051 and explain the functions of various units.
See http://www.atmel.com/products/8051/

Q8. State with justification if the following statements are right (or wrong)
Cache memory can be a static RAM
Dynamic RAMs occupy more space per word storage
The full-form of SDRAM is static-dynamic RAM
BIOS in your PC is not a Random Access Memory (RAM)

Ans:
Cache memory can be a static RAM **right**
The cache memory need to have very fast access time which is possible with static RAM.

Dynamic RAMs occupy more space per word storage **wrong**
DRAMs are basically simple MOS based capacitors. Therefore occupy much lower space as compared to static RAMs.

The full-form of SDRAM is static-dynamic RAM w**rong**
SDRAM is Synchronous Dynamic RAM. *Covered in later chapters*

BIOS in your PC is not a Random Access Memory (RAM) **Wrong**
The BIOS is a CMOS based memory which can be accessed uniformly.

Q9. Explain the function of the following units in a general purpose processor

Instruction Register
Program Counter
Instruction Queue
Control Unit

Ans:

Instruction Register: A register inside the CPU which holds the instruction code temporarily before sending it to the decoding unit.

Program Counter: It is a register inside the CPU which holds the address of the next instruction code in a program. It gets updated automatically by the address generation unit.

Instruction Queue: A set of memory locations inside the CPU to hold the instructions in a pipeline before rending them to the next instruction decoding unit.

Control Unit: This is responsible in generating timing and control signals for various operations inside the CPU. It is very closely associated with the instruction decoding unit.