Module

8

Industrial Embedded and Communication Systems

Lesson
36

Introduction to Real Time Embedded Systems

Instructional Objectives

After going through this lesson the student would be able to

- Define a Real Time Embedded System
- Describe major hardware components of an Embedded system.
- Describe typical architectures for such systems

Introduction

Definition

A Real Time Embedded System (RTES) an RTES is essentially a special-purpose computational system built into another machine with the sole purpose of controlling the embedder. Below we explain the key terms used in the above definition.

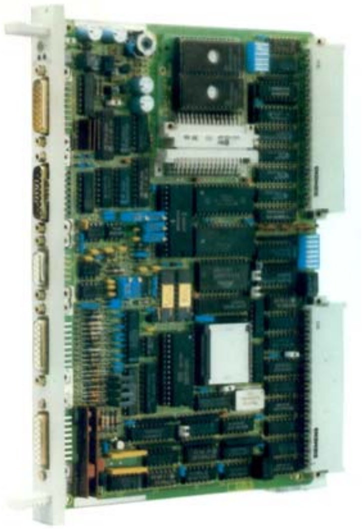
Real Time

“Real’-time usually means time as prescribed by external sources”. For example the time struck by clock (however fast or late it might be). The timings are generated by the requirements of the over all system. This external timing requirements imposed by the user is the real-time for the embedded system.

Embedded

Embedded, as per the Collins Cobuild dictionary, means, “firmly and deeply fixed within a surrounding mass”. Thus, an embedded system is deep within a surrounding system so that, from an external appearance such a system is not visible. For example, a CNC system controller is not apparent in the overall machine. On the other hand, a standalone system such as the PC does not have any such surrounding and is visible by itself. The system is firmly fixed, in the sense that it is an integral part of the overall system, both functionally and structurally.

Thus “A Real Time Embedded System” (RTES), in our context, is a computational subsystem within an overall system built to discharge a specific industrial task (such as precision machining), that is to be carried out in a definite manner with respect to time. Henceforth we refer to them as RTES.



Technical Features

- Shaft encoder input
- Onboard dedicated high speed CPU
- Motor drive set point output
- Digital IO
- Programmer port
- Set point from data bus

Fig. 36.1 Embedded Single Board Position Controller Module

RTES as a generic term may mean a wide variety of systems in the real world. However we will be concerned about those, which use programmable devices such as microprocessors or microcontrollers and have specific functions. We shall also restrict ourselves mainly to industrial RTES applications although not explicitly so. Much of the discussion is also applicable to other application domains of RTES, and is therefore presented in a general setting.

Typical Characteristics of an RTES

Single-Functioned

Here “single-functioned” means specific functions. The RTES is usually meant for very specific functions. Generally a special purpose microprocessor executes a program over and over again for a specific purpose. The same program is executed repeatedly throughout almost the whole life of the systems, unless reprogrammed for upgrading.

These operations are monitored and controlled by an operating system called as Real Time Operating System (RTOS) which is simpler, than a general purpose OS like Windows because these are built for a specific set of functionality, use a specific set of computing resources and try to achieve specific performance metrics.

Tightly Constrained

The constraints on the design of RTES are more severe than their non-real-time non-embedded counter parts. Time-domain constraints are the first thing that is taken care while developing such a system. Size, weight, power consumption and cost are the other major factors. However, RTES often turned out to be quite complex to achieve performance in the face of resource and timing constraints. The major burden of complexity often goes to the software.

Reactive and Real Time

Many embedded systems must continually react to events and processes in the system’s environment and must compute results in real time. For example, an industrial controller continually monitors and reacts to process the signals generated by analog and digital sensors. It must compute control inputs and must generate outputs repeatedly within a limited time to actuators and indicators; a delayed computation could result in a failure to maintain control of the industrial process. In contrast, a desktop computer system typically focuses on computations, with relatively infrequent (from the computer’s perspective) reactions to input devices. In addition, a delay in those computations, while perhaps inconvenient to the user, typically does not result in a system failure.

Point to Ponder: 1

A. *Give one example of an embedded system and of one that is not. Justify your examples.*

B. *Name an industrial embedded system. Explain whether the above characteristics hold for it.*

C. *Name an Industrial Real Time systems, which may not be called embedded. Explain your answer.*

Common Architecture

Unlike general-purpose computers, a generic architecture cannot be rigidly defined for Real Time Embedded System. There are many types of architecture proposed by manufacturers of a spectacular variety of embedded processors. However for the sake of our understanding we can discuss some common form of systems at the block diagram level. Any system can hierarchically divided into subsystems. Each sub-system may be further segregated into smaller systems. Each of these smaller systems, in turn, consists of discrete parts. Together, this is referred to as the hardware configuration.

Some of these parts may be programmable and therefore must have memory to store the programs. In an RTES the on-chip or on-board non-volatile memory is used to store these programs. These programs constitute both a Real Time Operating System (RTOS) as well as the application programs. The overall software system continually runs, as long as the RTES receives power. Particularly, for industrial applications, embedded systems often runs for days and even months on end.

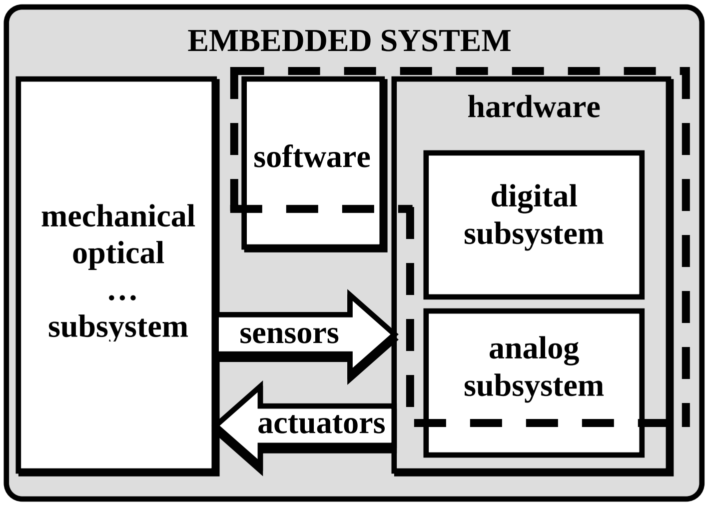


Fig. 36.2 Typical Embedded System Architecture

Components of an Embedded System

Architecture of a typical embedded-system is shown in Fig. 36.2. The hardware unit consists of the above units along with a digital as well as an analog subsystem. The software in the form of a RTOS resides in the memory. A typical embedded system consists, by and large, of the following units housed on a single board or chip.

1. Processor
2. Memory
3. Input/Output devices and interface chips
4. Real Time System Operating System Software
5. Application Software

1. Microprocessor

The microprocessors used in RTES are different from the general-purpose microprocessors like Pentium etc. They are often designed to meet specific application requirements. For example, a special function module for a PLC card may contain a special processor for high speed, real time, multi-channel I/O.

2. Memory

The microprocessor and memory must co-exit on the same PCB or same chip. Compactness, speed and low power consumption are the characteristics required for the memory to be used in an RTES. For housing the operating system, Read Only Memory (ROM) is used. Sometimes it is required to load-reload programs more often than never. The program or data loaded might exist for considerable duration. In these cases the memory should be capable of retaining the information even after the power is removed. In other words the memory should be non-volatile and should be easily programmable too. It is achieved by using Flash memories.

3. Input Output Devices and Interfaces

Input/Output interfaces are necessary to make the RTES interact with the external world. This is one of the most important subsystems for industrial RTES. The RTES needs to interact with a wide variety of sensors and actuators. These RTES should also have standard hardware interfaces to other devices such as Desktop Computers, Local Area Networks (LAN) and other RTES. These input/output devices along with standard software protocols in the RTOS provide the necessary interface to these standards.

4. Software

The software in an RTES can be classified as System and Application Software. A Real Time Operating System sitting on the non-volatile memory of the RTES, The application software are essentially cyclic or interrupt driven programs that execute read-compute-write cycles. Multiple such programs execute under a priority driven scheduling policy managed by the RTOS. These issues are discussed in the next lesson.

Besides the above, an RTES may have various other components and Application Specific Integrated Circuits (ASIC) for specialized functions such as motor control or counting. In the sequel, in this lesson, we shall discuss the main hardware components of an RTES.

Point to Ponder: 2

A. *State one characteristics each of industrial embedded systems in respect of Processors, Memory and I/O.*

B. *Describe typical i/o organization of industrial embedded systems.*

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

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 as a sequential machine. FPGA’s 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 inexpensive because these are manufactured in large quantities. The NRE (Non-recurring Engineering Cost) is, therefore, 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. 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. 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 and a control unit tightly linked with the memory. The *Data Path* consists of 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 connects 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 (e.g. 8085 has an 8-bit data path).

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 size of the address space is dictated by the capacity of 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 a number of clock cycles.

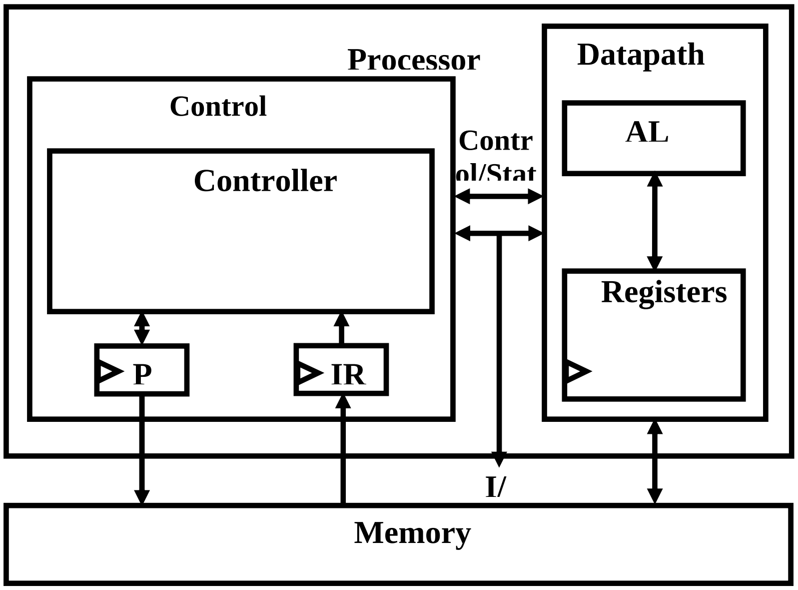


Fig. 36.3 The architecture of a General Purpose Processor

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.36.4.

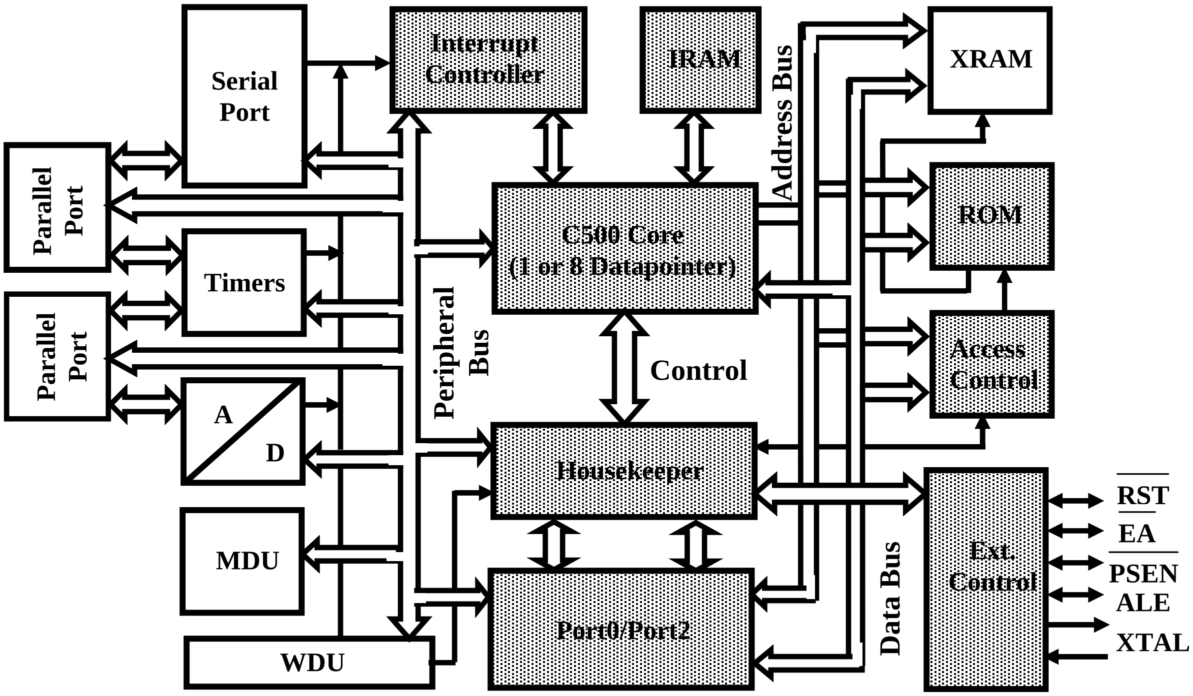


Fig. 36.4 The architecture of a typical microcontroller named as C500 from Infineon Technology, Germany

The various units of the processors are as follows:

The C500 Core contains the CPU which consists of the Instruction Decoder, 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 provided sometimes

Interrupt requests from 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. Sometimes it is known as MACD, where D stands for Data move. Generally all the instructions are executed in single cycle.

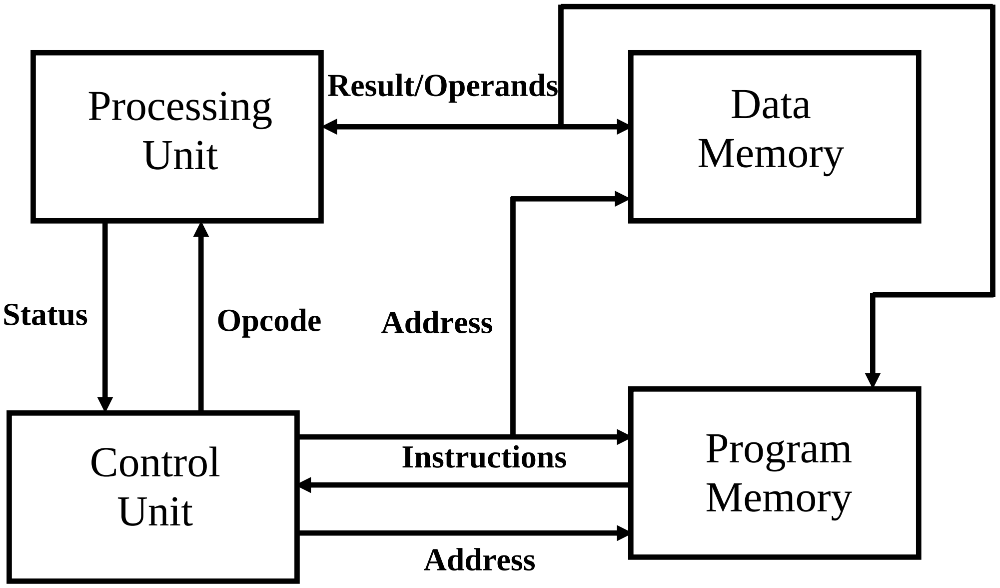


Fig. 36.5 The modified Harvard architecture

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.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.

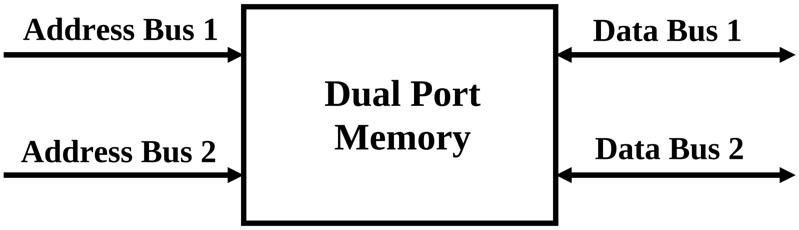


Fig. 36.6 Dual Ported Memory

The Very Long Instruction Word (VLIW) architecture is also suitable for Signal Processing applications. This has a number of functional units and data paths as seen in Fig.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

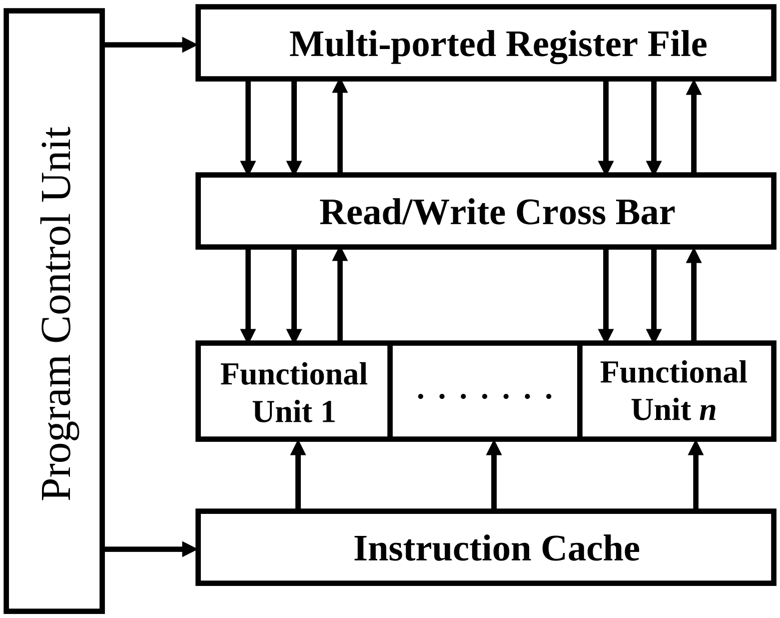


Fig. 36.7 Block Diagram of *VLIW* architecture

Microprocessors vs Microcontrollers

A **microprocessor** is a **general-purpose** digital computer's central processing unit. To make a complete microcomputer, memory (ROM and RAM) memory decoders, an oscillator, and a number of I/O devices are added. 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.

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.

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.

Microprocessors vs. DSP

DSPs are microprocessors specialized for signal processing applications. They are designed following a Harvard architecture, contrasted with the Von Neuman architecture, shown below.

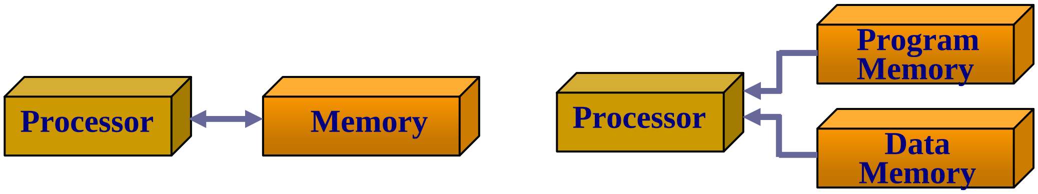


Fig. 36.8 The Von Neumann and the Harvard architechtures

DSPs, make concurrent, multiple memory accesses per cycle. GPPs use a multi-level hierarchy of memory, with an elaborate virtual memory system. Dedicated DSP hardware performs all key arithmetic operations in 1 cycle. This involves features such as, complex addressing of multiple data streams for each instruction, hardware loop control unit, typically unavailable for GPPs. DSPs support a narrower range of on-chip peripherals.

Point to Ponder: 3

A. *Give one example each of an embedded system that can use a GPP, a microcontroller and a DSP. Justify your examples.*

B. *Name one each of commercial versions of a GPP, a microcontroller and a DSP that is used in industrial embedded applications.*

Memory

Memory serves processor's 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. In the *Princeton Architecture* the data and program occupy the same physical 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. General-purpose processors typically have a 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.

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. If not there, the CPU gets it from disk. All the while, the clock is ticking, and the CPU is sitting there waiting.

Point to Ponder: 4

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

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

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.

To generate an analog signal from the microprocessor we need a 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.

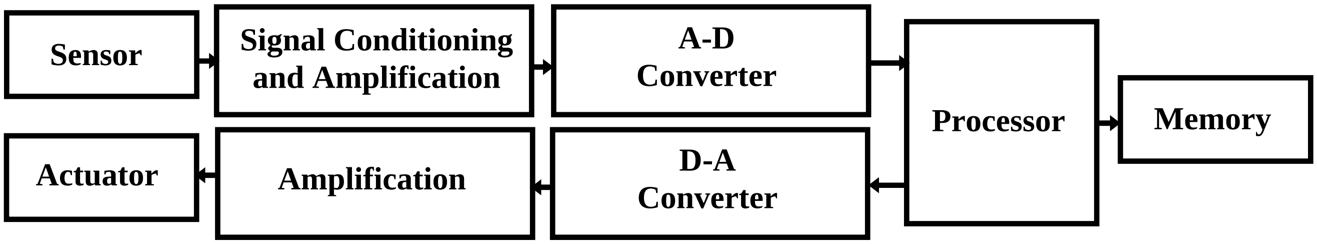


Fig. 36.9 The typical input/output interface blocks

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. **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.
2. **Controllers:** These are specific circuits for controlling, motors, actuators and light-intensities etc.

Lesson Summary

The following topics were covered.

a. Definition and characteristics of RTES
b. Architecture and Components of RTES
c. Processors
d. Memory
e. I/O and interface modules

Answers, Remarks and Hints to Points to Ponder

Point to Ponder: 1

A. *Give one example of an embedded system and of one that is not. Justify your examples.*

Ans: The controller for a microwave oven is an embedded system. Its existence is deep within the oven casing and not immediately visible at all. However, it controls each functionality of the microwave oven. On the other hand a pocket calculator is not an embedded system although it has very similar structural characteristics and handles similar kind of i/o, like LCD display and keypad. The calculator is a standalone computing device, which does not have a embedding environment.

B. *Name an industrial embedded system. Explain whether the above characteristics hold for it.*

Ans: Consider a CNC milling machine controller. One such machine is shown. Note that the controller is not seen on the machine, although a display and a keyboard are. The controller is custom designed to control the CNC milling machine. It cannot be used, as a PC for example. Thus it is single-functioned. It is clearly reactive and real-time, since it interacts with sensors every sampling cycle. Its design is likely to be constrained by cost. However, design of industrial systems is less constrained compared to other categories, in respect of power, size, weight and consequently, memory, processor speed etc.

Point to Ponder: 2

A. *State one characteristics each of industrial embedded systems in respect of Processors, Memory and I/O.*

Ans: Processors for industrial embedded systems are generally characterized by a sophisticated interrupt handling system. This is necessary to support an asynchronous i/o subsystem, that is typical of industrial i/o as well as to facilitate a preemptive priority scheduler that is an integral part of an RTOS. Memory configurations for industrial embedded systems are characterized by the fact that there is generally enough available memory to preclude a virtual memory system. This is because virtual memory systems can introduce significant variations in task latencies and are therefore not good reliable for real-time applications. One important characteristics of industrial embedded i/o is that it involves both processor driven cyclic i/o as well device interrupt driven asynchronous i/o.

B. *Describe typical i/o organization of industrial embedded systems.*

Ans: I/O in industrial embedded systems can be divided into field i/o and device i/o. Field i/o typically means signals from sensors and signals to actuators and indicators. These are generally sensed cyclically, as for every scan of an RLL. Other kinds of i/o involve data exchange with various system devices such as programmers, MMI devices or communication processors, keyboards and displays. Finally the last kind of i/o involves network i/o following standard protocols such as the Ethernet or the fieldbus.

Point to Ponder: 3

A. *Give one example each of an embedded system that can use a GPP, a microcontroller and a DSP. Justify your examples.*

Ans: A large industrial controller like a rack-based PLC can use a GPP. Motor drive controllers often use DSPs. A single-loop temperature controller can use a microcontroller. These are chosen depending on the number of computing tasks, special kind of floating point computational requirements and the need for a simple system with limited off-chip resource requirement.

B. *Name one each of commercial versions of a GPP, a microcontroller and a DSP that is used in industrial embedded applications.*

Ans: 80186 processors have been used in the past to build embedded controller cards. More recently processors like the Power-PC may be used for industrial applications. Similarly, 8031 processors have been used for building small industrial controllers. More recently one can use ARM processors to build industrial embedded devices like intelligent sensors. Texas instruments has a special category of DSPs that have been designed for motor control (TMS 658030)

Point to Ponder: 4

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

Ans: a. Yes, because it is very fast and limited in density; b. No, they have the highest densities among memory technologies; c. No. it is synchronous dynamic RAM; d. Yes, BIOS cannot be on volatile memory like RAM.

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

Ans: CDROM, Magnetic tape, Hard disk, Flash memory, Dynamic memory, Cache memory, Processor memory

Source:http://www.nptel.ac.in/courses/Webcourse-contents/IIT%20Kharagpur/Industrial%20Automation%20control/pdf/L-36(SM)%20(IA&C)%20((EE)NPTEL).pdf