

- To familiarise the student with microcomputer hardware and interface devices.
- To develop the programming techniques in the construction of an example real-time system.
- To develop an understanding of the problems in real-time software design.
- To familiarise the student with the facilities provided by modern real-time programming languages.

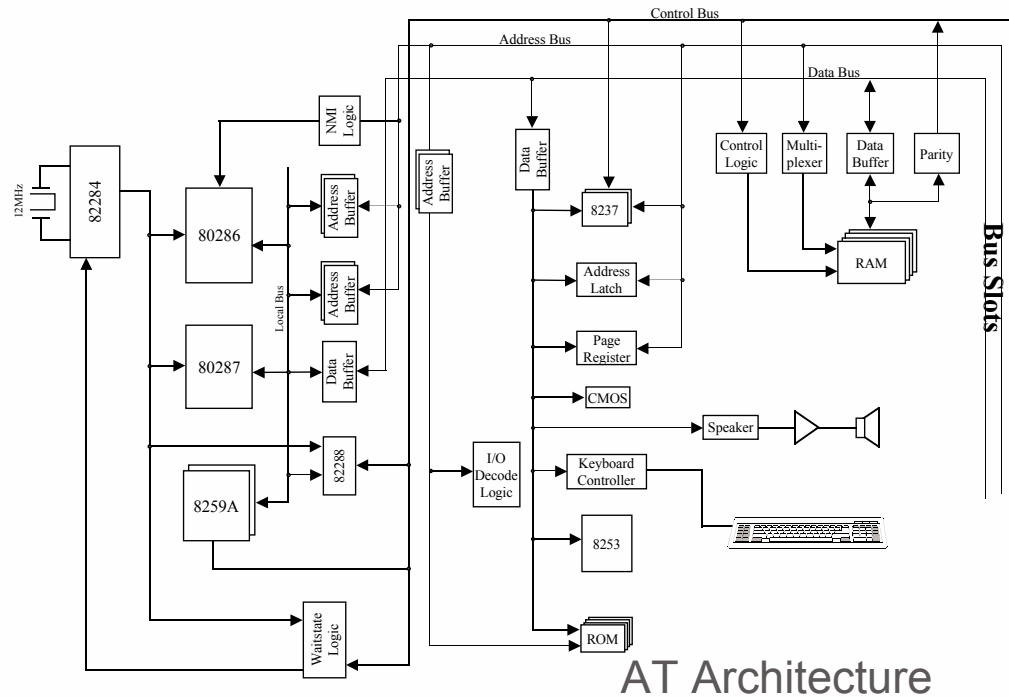
- Understand microcomputer hardware in relation to interface devices.
- Implement machine to machine communication protocol in a suitable system programming language.
- Design and implement a simple real-time multiprocess control system using suitable hardware and real-time executives.
- Produce an appraisal of their experience and the techniques they have used.

**UNIX and C**  
By P. Cornes  
Van Norstrand, 1989

**Introduction to Real-Time Software Design**  
By S.T. Allworth and R. N. Zobel  
MacMillan, 1987

**Microprocessor Hardware Interface & Applications**  
By Barry B. Brey  
Charles E. Merrill Publishing Co., 1984

AT  
Architecture



# Real Time Systems



## Real-Time Systems Topics

- Definition of Real-Time Systems
- Classes of Real-Time Systems
- Characteristics of Real-Time Systems



## Real-Time System

What is a **Real-Time System**?

Any system in which the *time* at which an *output* is produced is *significant*.

or

Any information processing activities or systems which have to *respond* to externally generated stimuli *within a finite and specified period of time*.

## Classes Of Real-Time Systems

---

- **Hard Real-Time Systems**

Systems whereby the response time has to occur within a specified deadline.

- **Soft Real-Time Systems**

Response time is important but the system will still function correctly if the deadline is occasionally missed.

## Real-Time Facilities

---

- Access to a clock
- Delaying a process
- Programming Time-outs
- Deadline Specification and scheduling

## Characteristics of Real-Time Systems

---

- Size and Complexity
- Manipulation of real numbers
- Extreme Reliability and Safety
- Concurrent control of separate system components
- Real-Time facilities (and control)
- Interaction with Hardware Interfaces
- Efficient Implementation

Reliability &  
Fault  
Tolerance

- Definition of Reliability
- Sources & Types of Faults
- Fault Handling
- Safety

- Inadequate (or unclear) specifications
- Design errors in software components
- Faults due to processor failure
- Faults caused by transient or permanent interference

A measure of success with which the system conforms to some authoritative specifications of its behaviour.

- **Transient Faults**

Faults which appears, remains for a short duration and then disappears.

- **Intermittent Faults**

Faults which occur from time to time.

- **Permanent Faults**

A kind of fault which appears and remains throughout the duration of the use of the system.

- **Fault Prevention**

Attempt to eliminate any possibility of faults creeping into the system before it goes operational.

- **Fault Tolerance**

Enable the system to continue functioning even in the event that faults are present.

Freedom from those conditions that can cause death, injury, occupational illness, damage to (or loss of) equipment (or property), or environment harm.

- **Full Fault Tolerance**

system continues to operate in the presence of faults, for a limited period, with no significant loss of functionality or performance.

- **Graceful Degradation**

system continues to operate in the presence of errors, accepting a partial degradation of functionality or performance during recovery or repair.

- **Fail-safe**

system maintains its integrity while accepting a temporary halt in its operations.