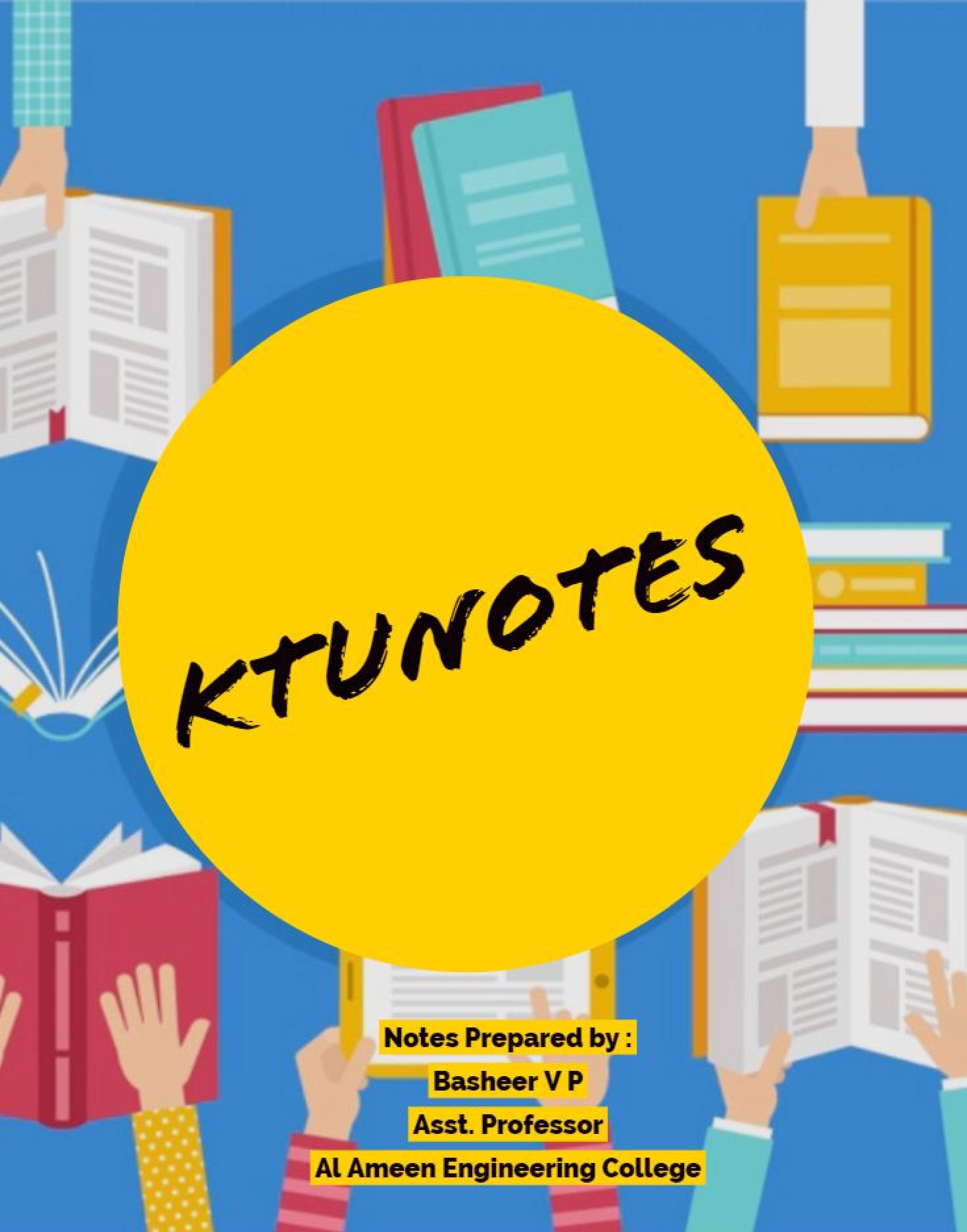


KTUNOTES

Notes Prepared by :
Basheer V P
Asst. Professor
Al Ameen Engineering College

EC 308 EMBEDDED SYSTEMS

MODULE-1

SYLLABUS

Introduction to Embedded Systems– Components of embedded system hardware– Software embedded into the system – Embedded Processors - CPU architecture of ARM processor (ARM9) – CPU Bus Organization and Protocol.

Design and Development life cycle model - Embedded system design process – Challenges in Embedded system design

Basheer V P
Asst. Professor
Al Ameen Engineering College

1.1 Introduction to Embedded Systems

- An embedded system is an electronic system, which includes a single chip microcomputers(Microcontrollers) like the ARM or Cortex.
- It is configured to perform a specific dedicated application.
- Software is programmed into the on chip ROM of the single chip computer. This software is not accessible to the user , and software solves only a limited range of problems.
- Here the microcomputer is embedded or hidden inside the system. Every embedded microcomputer system , accepts inputs, performs computations, and generates outputs and runs in “real time.”

For Example, a typical automobile now a days contains an average of ten microcontrollers. In fact, modern houses may contain as many as 150 microcontrollers and on average a consumer now interacts with microcontrollers up to 300 times a day. General areas that employ embedded systems covers every branch of day to day science and technology, namely Communications, automotive, military, medical, consumer, machine control etc...

Ex: Cell phone , Digital camera , Microwave Oven,MP3 player, Portable digital assistant & automobile antilock brake system etc.

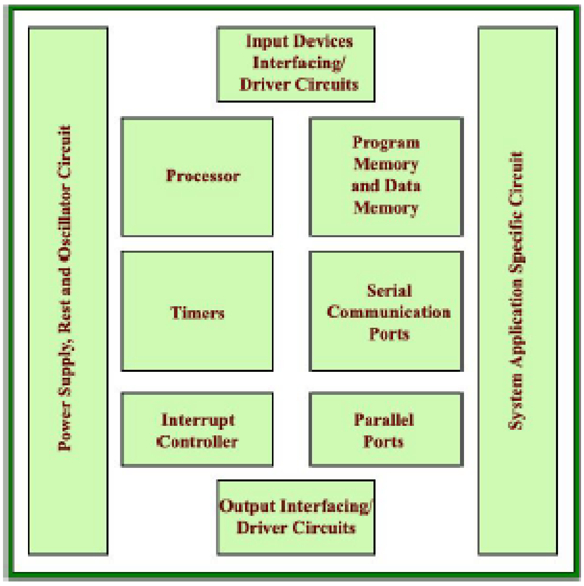
1.2 Components of embedded system:

An embedded system has three components:

1. Hardware.
2. Application software.
   This may perform concurrently the series of tasks or multiple tasks.
3. Real Time Operating system (RTOS) that supervises the application software and provide mechanism to let the processor run a process as per scheduling by following a plan to control the latencies. RTOS defines the way the system works. It sets the rules during the execution of application program. A small scale embedded system may not have RTOS.

Basheer V P
Asst. Professor
Al Ameen Engineering College

Hardware:



Embedded System hardware

Processor

- A Processor is the heart of the Embedded System.
- The main criteria for the processor are the processing power needed to perform the tasks within the system.

Processors can be of the following categories:

- General Purpose Processor (GPP)
- Microprocessor
- Microcontroller
- Embedded Processor
- Digital Signal Processor
- Media Processor
- Application Specific System Processor (ASSP)
- Application Specific Instruction Processors (ASIPs)

Power Source

Three possible methods of providing power to an embedded system are

- System own supply with separate supply rails for IOs, clock, basic processor and memory.

- Supply from a system to which the embedded system interfaces, for example in a network card.
- Charge pump concept used in a system of little power needs, for examples, in the mouse or contact-less smart card.

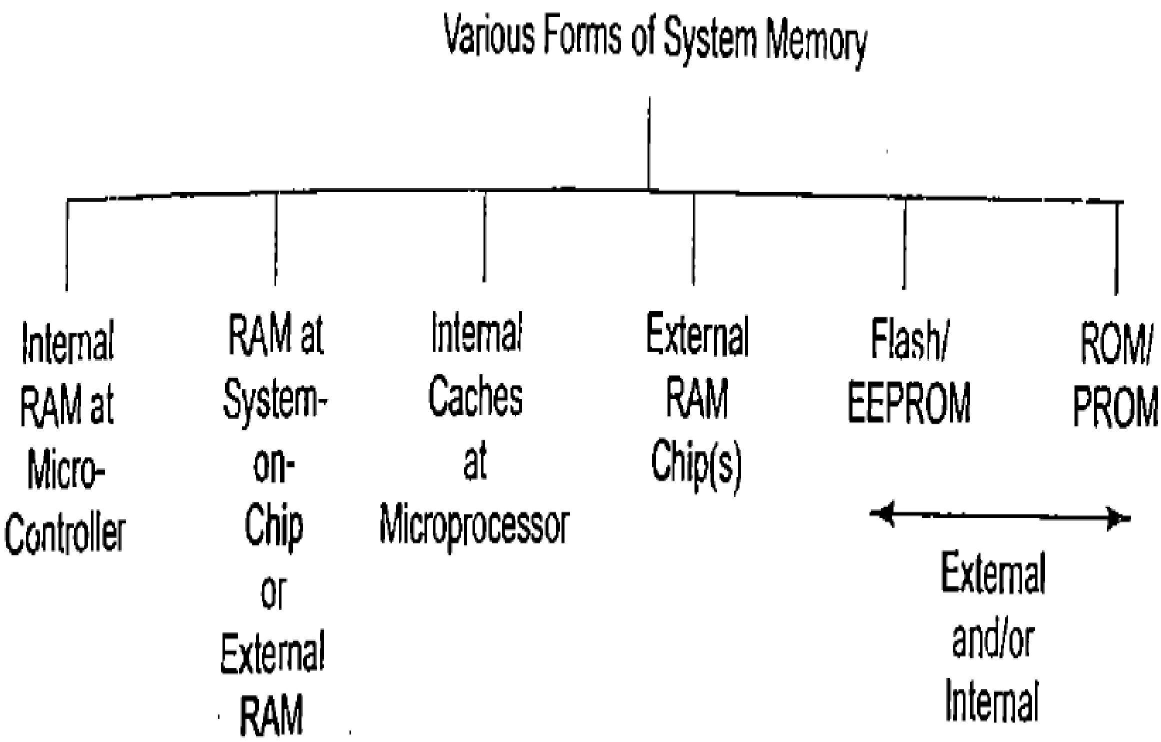
Clocking Circuits

- The clock controls the time for executing an instruction. The clock controls the various clocking requirements of CPU, Timer etc
- For this, clocking circuit provide highly stable clock pulses.

Memory

- An embedded system uses different types of memory modules for a wide range of tasks such as storage of software code and instructions for hardware.
- There are different varieties of memories in embedded system, each having their own particular mode of operation.
- An efficient memory increases the performance of embedded systems.

Various forms system memory:



a. Functions Assigned to the ROM or EPROM or Flash

1. Storing 'Application' program from where the processor fetches the instruction codes
2. Storing codes for system booting, initializing, Initial input data and Strings.
3. Storing Codes for RTOS.
4. Storing Pointers (addresses) of various service routines.

b. Functions Assigned to the Internal, External and Buffer RAM

1. Storing the variables during program run,
2. Storing the stacks,
3. Storing input or output buffers for example, for speech or image .

c. Functions Assigned to the EEPROM or Flash

Storing non-volatile results of processing.

d. Functions Assigned to the Caches

1. Storing copies of the instructions, data and branch-transfer instructions in advance from external memories and
2. Storing temporarily the results in write back caches during fast processing

Timer

- Embedded systems often require mechanisms for counting the occurrence of events and for performing tasks at regular intervals.
- Embedded processors are often equipped with hardware support for this functionality.
- Timer is a device, which counts the input at regular interval using clock pulses at its input.
- The count increment on each pulse and store in a register, called count register
- Timer is used for generating delay and for generating waveforms with specific delay.

Serial Port

- A **serial port** is a serial communication interface through which information transfers in or out one bit at a time.
- Serial data transmission is much more common in new communication protocols due to a reduction in the I/O pin count, hence a reduction in cost.
- Common serial protocols include UART, SPI, SCI and I^2C etc

- In most of the embedded systems at least two serial ports are provided.

Parallel Port

- A **parallel port** is a type of interface found on computers or embedded systems for connecting peripherals.
- The name refers to the way the data is sent; parallel ports send multiple bits of data at once.
- Parallel ports require multiple data lines in their cables and port connectors, and tend to be larger than contemporary serial ports.

Interrupt Controller

- An *interrupt* is a signal to the processor emitted by hardware or software indicating an event that needs immediate attention.
- Interrupts allow an embedded system to respond to multiple real world events in rapid time.
- By managing the interaction with external systems through effective use of interrupts can dramatically improve system efficiency and the use of processing resources.
- In an embedded system there are usually multiple interrupt sources. These interrupt sources share a single pin. The sharing is controlled by a piece of hardware called an **interrupt controller** that allows individual interrupts to be either enabled or disabled.

System Application Specific Circuit

- These are the dedicated circuits for the implementation of the application of particular system.
- This may varies from one system to other.

1.3 ARM Processor

- Advanced RISC machine (ARM) is the first reduced instruction set computer (RISC) processor for commercial use, which is currently being developed by ARM Holdings.

RISC

- A Reduced Instruction Set Computer (RISC) is a microprocessor that has been designed to perform a small set of instructions, with the aim of increasing the overall speed of the processor.
- The RISC concept first originated in the early 1970's when an IBM research team proved that 20% of instruction did 80% of the work.
- The RISC architecture follows the philosophy that one instruction should be performed every clock cycle.

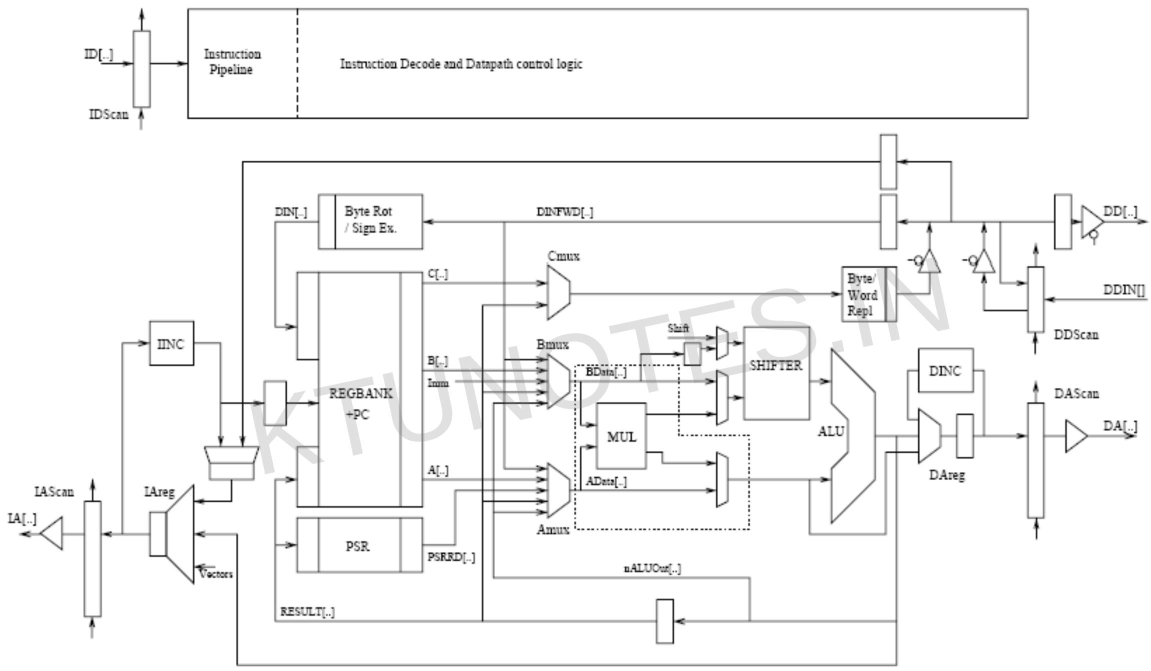
Comparison between CISC and RISC

CISC	RISC
Emphasis on hardware	Emphasis on software
Multiple instruction sizes and formats	Instructions of same set with few formats
Less registers	Uses more registers
More addressing modes	Fewer addressing modes
Extensive use of microprogramming	Complexity in compiler
Instructions take a varying amount of cycle time	Instructions take one cycle time
Pipelining is difficult	Pipelining is easy

1.4 ARM Architecture (ARM9)

- The ARM9TDMI is a member of the ARM family of general-purpose microprocessors.
- The ARM9TDMI is targeted at embedded control applications where high performance, low die size and low power are all important.
- The ARM9TDMI supports both the 32-bit ARM and 16-bit Thumb instruction sets, allowing the user to trade off between high performance and high code density.

- The ARM9TDMI supports the ARM debug architecture and includes logic to assist in both hardware and software debug.
- The ARM9TDMI supports both bidirectional and unidirectional connection to external memory systems.
- The ARM9TDMI also includes support for coprocessors.
- The ARM9TDMI processor core is implemented using a five-stage pipeline consisting of fetch, decode, execute, memory and write stages.
- The device has a Harvard architecture, and the simple bus interface eases connection to either a cached or SRAM-based memory system.



ARM 9 Architecture

The main parts of the ARM processor are:

1. Register file: The processor has a total of 37 registers made up of 31 general 32 bit registers and 6 status registers
2. Booth Multiplier
3. Barrel shifter
4. Arithmetic Logic Unit (ALU)
5. Control Unit.

Registers

- The processor has a total of 37 registers made up of 31 general 32 bit registers and 6 status registers.
- At any one time 16 general registers (R0 to R15) and one or two status registers are visible to the programmer.
- The visible registers depend on the processor mode
- In all modes 16 registers, R0 to R15, are directly accessible. All registers except R15 are general purpose and may be used to hold data or address values. Register R15 holds the Program Counter (PC).
- A seventeenth register (the CPSR - Current Program Status Register) is also accessible. It contains condition code flags and the current mode bits and may be thought of as an extension to the PC.
- R14 is used as the subroutine link register and receives a copy of R15 when a Branch and Link instruction is executed.
- Program status register: It contains the processor flags (Z, S, V and C).
- The modes bits also exist in the program status register in addition to the interrupt and fast interrupt disable bits
- Some special registers: Some registers are used like the instruction register, memory data read and write register and memory address registe

Multiplexers:

- Many multiplexers are used to control the operation of the processor buses

Arithmetic Logic Unit (ALU)

- The ALU has two 32-bits inputs.
- The first comes from the register file while the other comes from the shifter.
- ALU outputs modify the status register flags.

Booth multiplier

- The multiplier has three 32-bit inputs.
- All the inputs come from the register file.
- The multiplier output is only the 32 least significant bits of the product.

Barrel shifter

- The barrel shifter has a 32-bit input to be shifted.
- This input is coming from the register file or it could be immediate data.

Basheer V P
Asst. Professor
Al Ameen Engineering College

- The shifter has other control inputs coming from instruction register.
- Shift field in the instruction controls the operation of the barrel shifter.
- This field indicates the type of shift to be performed (logical left or right, arithmetic right or rotate right).
- The amount by which the register should be shifted is contained in an immediate field in the instruction or it could be the lower 6 bits of a register in the register file.

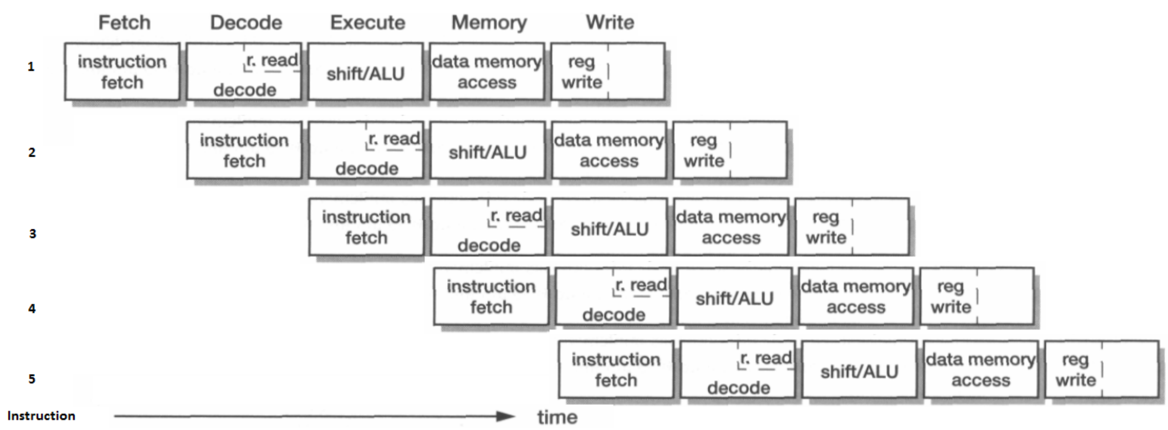
Control Unit

- For any microprocessor, control unit is the heart of the system.
- It is responsible for the system operation and so the control unit design is the most important part in the hole design.
- Control unit is usually a pure combinational circuit.
- The processor timing is also included in the control unit.

Pipeline implementation

The ARM9TDMI implementation uses a five-stage pipeline design. These five stages are:

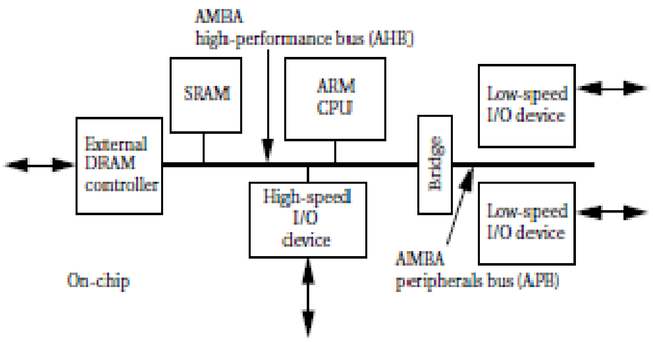
• Instruction fetch (F)
• Instruction decode (D)
• Execute (E)
• Data memory access (M)
• Register write (W).



Five stage pipeline operation

1.5 ARM Bus Organization

- ARM has created a separate bus specification for single-chip systems. The AMBA bus [ARM99A] supports CPUs, memories, and peripherals integrated in a system-on-silicon.
- As shown in Figure , the AMBA specification includes two buses.
- The AMBA high-performance bus (AHB) is optimized for high-speed transfers and is directly connected to the CPU. It supports several high-performance features: pipelining, burst transfers, split transactions, and multiple bus masters.
- A bridge can be used to connect the AHB to an AMBA peripherals bus (APB). This bus is designed to be simple and easy to implement; it also consumes relatively little power.
- The AHB assumes that all peripherals act as slaves, simplifying the logic required in both the peripherals and the bus controller.
- It also does not perform pipelined operations, which simplifies the bus logic.



ARM Bus Structure

1.6 Embedded Product Development Life Cycle (EDLC)

- EDLC is Embedded Product Development Life Cycle
- It is an Analysis – Design – Implementation based problem solving approach for embedded systems development.

Different Phases of EDLC:

The following figure depicts the different phases in EDLC:

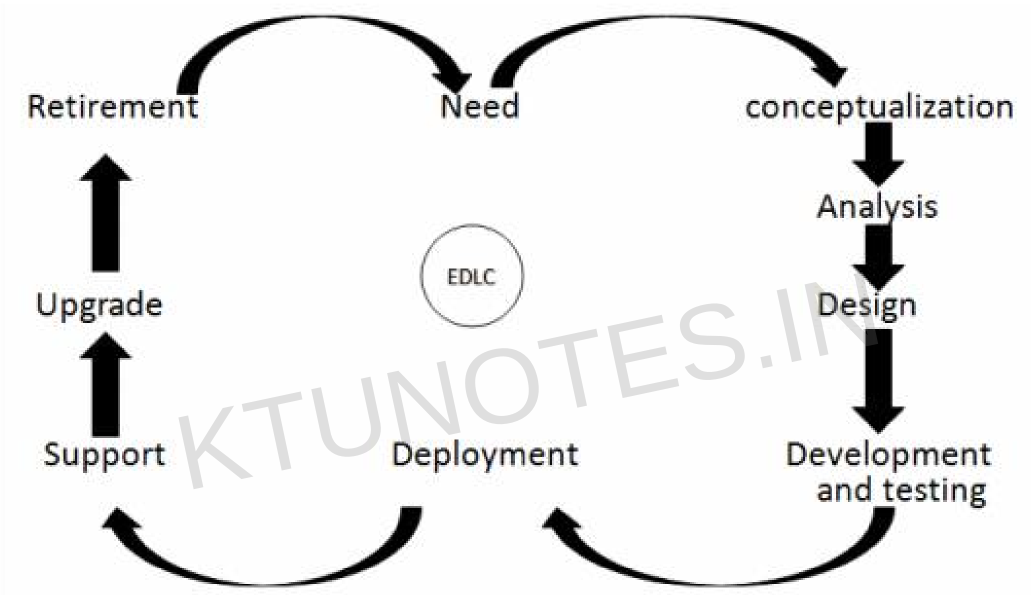


Figure : Phases of EDLC

Need

- The need may come from an individual or from the public or from a company.

Conceptualization

- Defines the scope of concept, performs cost benefit analysis and feasibility study and prepare project management and risk management plans.

Analysis

- The product is defined in detail with respect to the inputs, processes, outputs, and interfaces at a functional level.

Design

- The design phase identifies application environment and creates an overall architecture for the product.

Development and Testing

- Development phase transforms the design into a realizable product.

Deployment

- Deployment is the process of launching the first fully functional model of the product in the market.

Support

- The support phase deals with the operational and maintenance of the product in the production environment.

Upgrades

- Deals with the development of upgrades (new versions) for the product which is already present in the market.

Retirement/Disposal

- The retirement/disposal of the product is a gradual process.
- This phase is the final phase in a product development life cycle where the product is declared as discontinued from the market.

1.7 Waterfall Model

- Linear or waterfall model is one of the EDLC models which adopted in most of the olden systems.

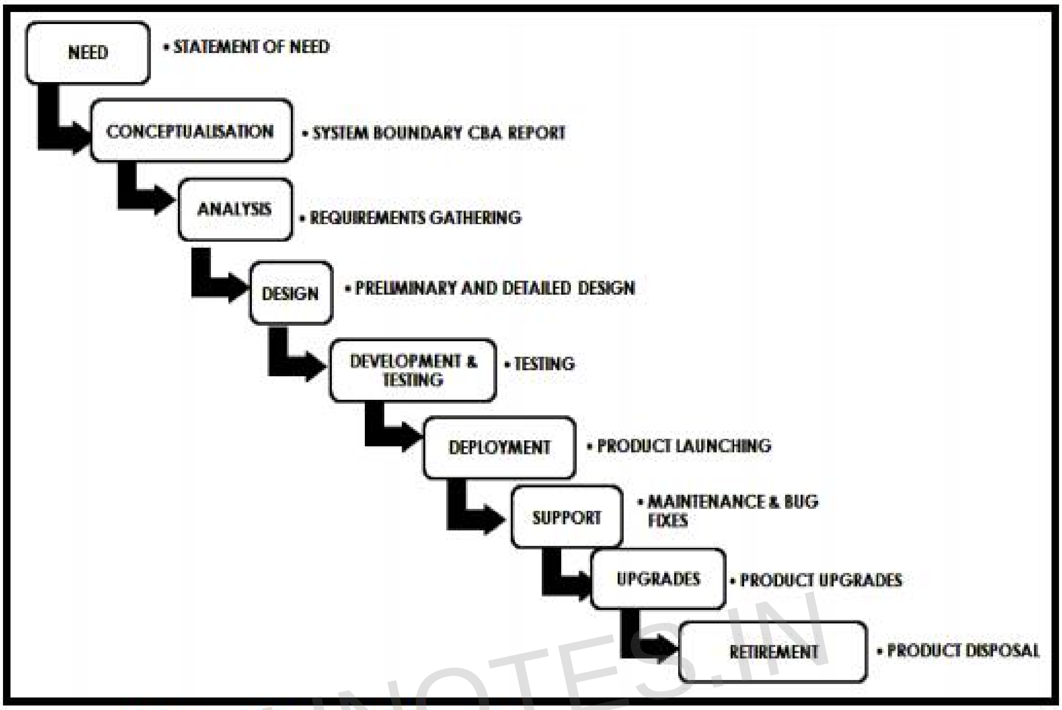


Figure: Waterfall Model

- In this approach each phase of EDLC (Embedded Development Product Lifecycle) is **executed in sequence.**
- It establishes analysis and design with **highly structured development phases**.
- The execution flow is **unidirectional.**
- The **output of one phase serves as the input of the next phase**
- All activities involved in each phase are well planned so that what should be done in the next phase and how it can be done.
- The feedback of each phase is available only after they are executed.
- It implements extensive review systems to ensure the process flow is going in the right direction.
- One significant feature of this model is that even **if you identify bugs in the current design the development process proceeds with the design.**
- The fixes for the bug are postponed till the support phase.

Advantages

Product development is rich in terms of:

- Documentation
- Easy project management
- Good control over cost & Schedule

Drawbacks

- It assumes all the analysis can be done without doing any design or implementation
- The risk analysis is performed only once.
- The working product is available only at the end of the development phase
- Bug fixes and correction are performed only at the maintenance/support phase of the life cycle.

1.8 Challenges in Embedded Systems

1. Amount and type of hardware needed.
    - Optimizing various hardware elements for a particular design.
2. Taking into account the design metrics
    - Design metrics examples –power dissipation, physical size, number of gates and engineering, prototype development and manufacturing costs.
3. Optimizing the Power Dissipation.
    - Clock Rate Reduction and Operating Voltage Reduction
4. Disable use of certain structural units of the processor to reduce power dissipation the processor to reduce power dissipation.
    - Control of power requirement, for example, by screen auto-brightness control
5. Process Deadlines
    - Meeting the deadline of all processes in the system while keeping the memory, power dissipation, processor clock rate and cost at minimum is a challenge.
6. Flexibility and Upgradeability

- Ability to offer the different versions of a product for marketing and offering the product in advanced versions later on.

7. Reliability
    - Designing reliable product by appropriate design and thorough testing, verification and validation is a challenge.
8. Testing, Verification and Validation
    - Testing – to find errors and to validate that the implemented software is as per the specifications and requirements to get reliable product.