Embedded System - combination of computer HW & SW either fixed in capability or "programmable", that is specifically designed for a particular kind of application device.

=> Incorporated microprocessors within their implementations

=> SW is read only

=> To modify SW, must reprogram/Flash ROM containing the application

Embedded system: non-general purpose, Fridge, Car things, R2D2, smart-phones redefining ESys.

Multi-Tasking -> separate application into different tasks

-> programming - requires switching back and forth among multiple tasks

-> Task **appear to be executing concurrently**

Multi-Threading -> gives appearance that simultaneous execution of threads is occurring, but in reality only one thread is executing at a time.

**Concurrency -> the appearance of simultaneous execution of multiple tasks /threads**

Real time system -> System that processes events. System input event causes other events to happen as system outputs

Hard RT: System MUST meet a specific deadline, else system may fail to operate properly **Brake System, Pacemaker, Army, Stock Market, Nuclear Reactor Shutdown**Soft RT: System designed to computer the response as fast as possible but has no explicit deadline

**MCUs** Microcontroller units -> Consists of a CPU which is the "CPU" of the "microcontroller" chip. However peripherals/ components within the unit may be designed by other manufacturers which have a license for that specified processor, so they can put it in their silicon designs

|  |  |
| --- | --- |
| CortexM3 Core <-> Debug  ^ ^ | |
| BUS | |
| v Peripherals | v  Memory |
| Clk/Rst | I/O |

ARM categories - **A** High performance applications  
**R** High end embedded systems for RT systems  
**M** embedded uC type systems (low power)

RISC - Good for embedded systems.