

BECE403E-EMBEDDED SYSTEM DESIGN

MODULE-2

EMBEDDED HARDWARE DESIGN

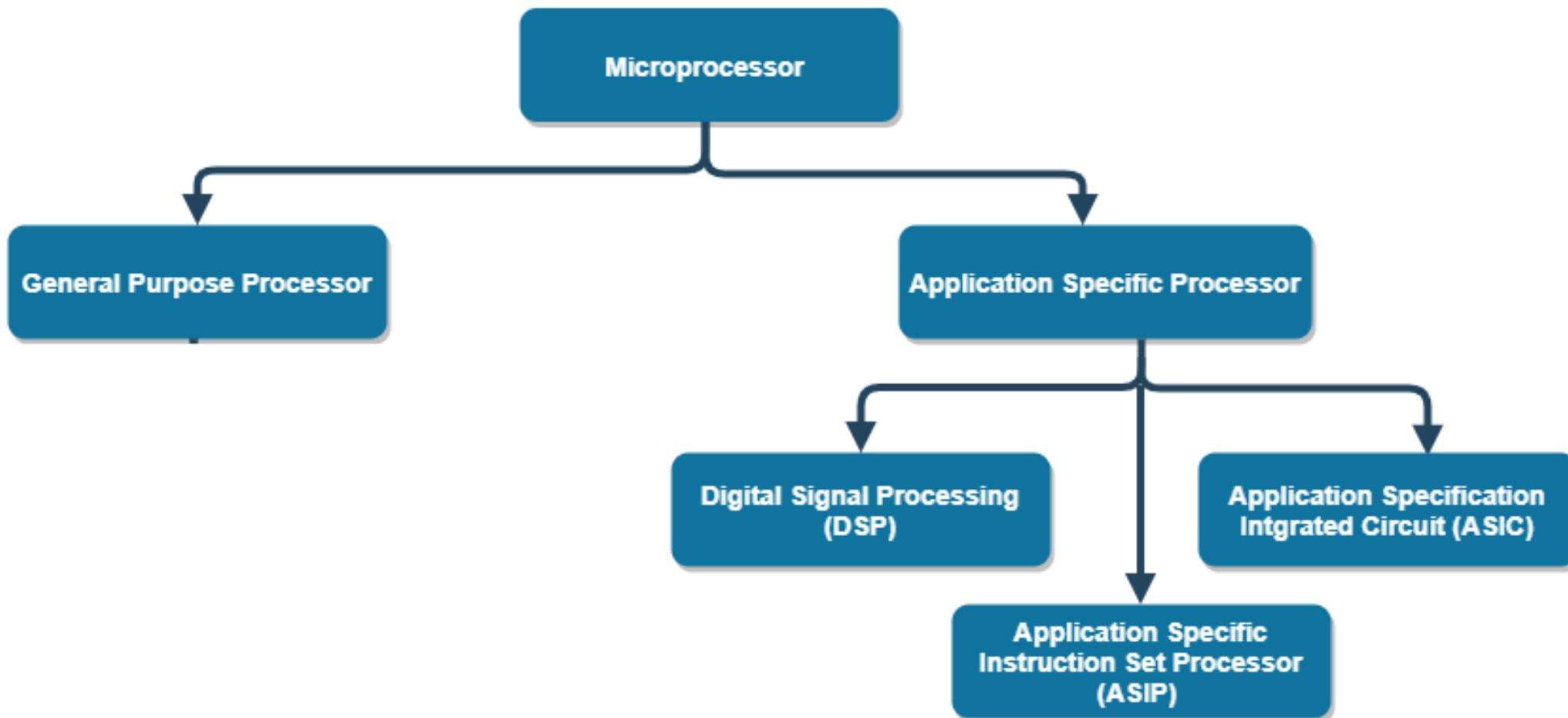
MODULE-2

Embedded Hardware Design

Processor classification - general purpose, customized, application specific processors, Microcontroller architectures (RISC, CISC), Embedded Memory, Strategic selection of processor and memory, Power Supply Design Considerations for Embedded Systems.

PROCESSOR CLASSIFICATION

PROCESSOR CLASSIFICATION



CLASSIFICATION OF EMBEDDED SYSTEM

GENERAL PURPOSE PROCESSOR (GPP)

- ❑ GPP are designed to **perform multiple tasks** and used in a variety of applications
- ❑ The system designer **only needs to program the processor's memory** to carry out the required functionality
- ❑ Biggest advantage of such system is the **flexibility but lack in performance** in certain task
- ❑ Advantages:
 - ✓ Easy to design and use
 - ✓ Design time & cost is low
 - ✓ Reprogrammability
- ❑ Disadvantages:
 - ✗ Performance is not very good
 - ✗ Large in size
 - ✗ They consume much power
- ❑ Example: **intel “i” series processors**

CLASSIFICATION OF EMBEDDED SYSTEM

APPLICATION SPECIFIC PROCESSORS (ASP)

- ❑ ASPs emerged as a solution for **high performance and cost effective processors**.
- ❑ Designed to **execute exactly one program** and contains only the components needed to execute a single program
- ❑ Designer creates a single-purpose processor by designing a **custom digital circuit**.
- ❑ **Advantages:**
 - ✓ Performance is very good
 - ✓ Small size
 - ✓ Consume less power
- ❑ **Disadvantages:**
 - ✗ Difficult to design hence design time is high
 - ✗ Design cost is higher
 - ✗ reprogramming is difficult and limited flexibility
- ❑ **Types of ASPs:** Digital Signal Processor (DSP), Application Specific Instruction Set Processors (ASIP), Application Specific Integrated Circuit (ASIC).

CLASSIFICATION OF EMBEDDED SYSTEM

APPLICATION SPECIFIC PROCESSORS (ASP)

- **Digital Signal Processor (DSP):** Programmable microprocessor for extensive real-time mathematical computations.
- **Application Specific Instruction Set Processors (ASIP):** Programmable microprocessor where hardware and instruction set are designed together for one special application.
- **Application Specific Integrated Circuit (ASIC):** Algorithm completely implemented in hardware.

CLASSIFICATION OF EMBEDDED SYSTEM

DIGITAL SIGNAL PROCESSORS (DSP)

- ❑ DSPs are specialized Mp optimized for the need of performing digital signal processing.
- ❑ DSP gained their importance with the increased demand on data intensive applications such as video and internet browsing on mobile devices.
- ❑ DSP satisfy the need for powerful processor while maintaining low cost and low power consumption.
- ❑ DSP Architecture Features:
 - Memory architecture designed for streaming data, using DMA extensively
 - Deliberate exclusion of a Memory Management Unit (MMU)
 - Bit-reversed addressing, a special addressing mode useful for calculating FFTs
 - Special arithmetic operations, such as fast Multiply–Accumulates (MACs)
 - Separate program and data memories (Harvard architecture)

CLASSIFICATION OF EMBEDDED SYSTEM

APPLICATION-SPECIFIC INSTRUCTION SET PROCESSORS (ASIP)

- ❑ ASIP is typically a programmable architecture that is designed in a **specific way** to perform certain tasks more efficiently
- ❑ As the name suggests, the Instruction set seems to be the **core characteristic** of any ASIP based platform; but this is entirely not true.
- ❑ Considering a whole platform, other very important attributes like interfaces and micro-architecture **do contribute a lot** to the overall system performance.
- ❑ The term “Application” in ASIP is not necessarily related to **software applications**, it actually describe the class of tasks the ASIP platform was designed to efficiently accomplish.
- ❑ This extra efficiency is not exclusively associated with **faster performance**.

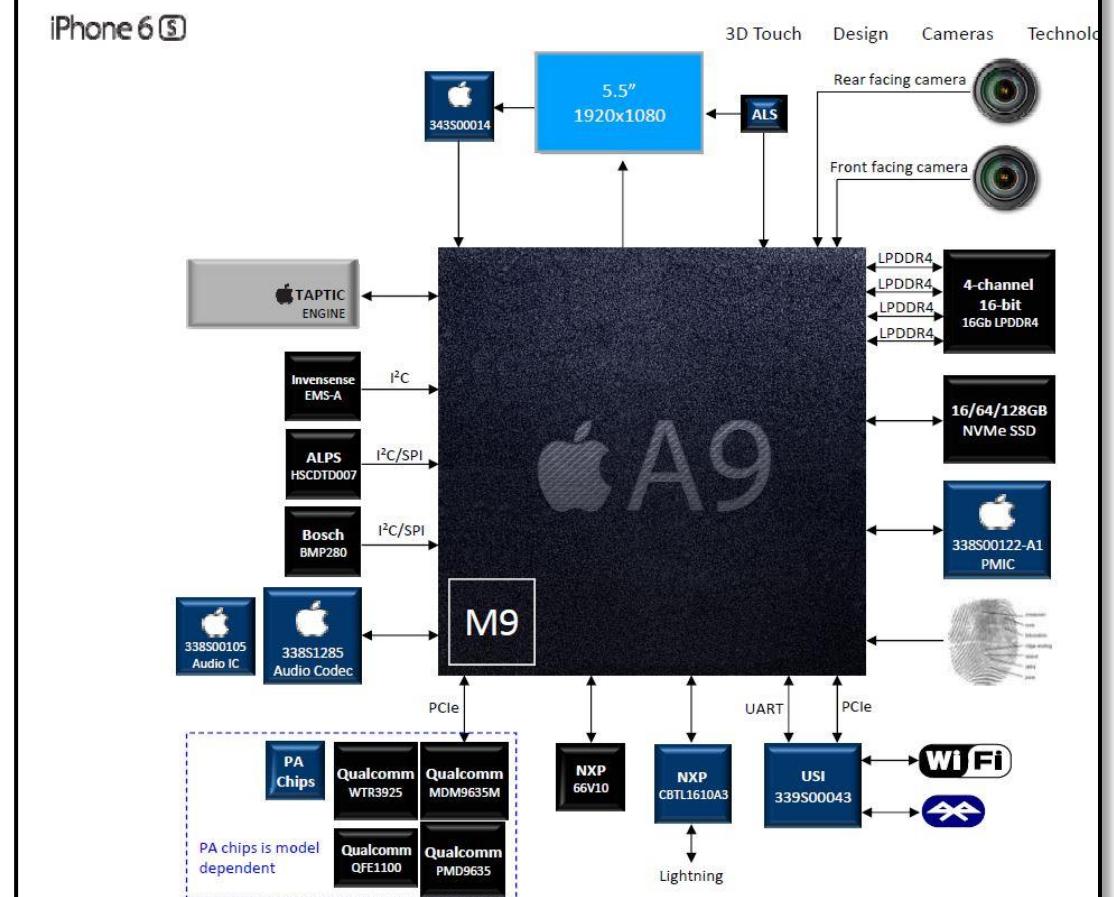
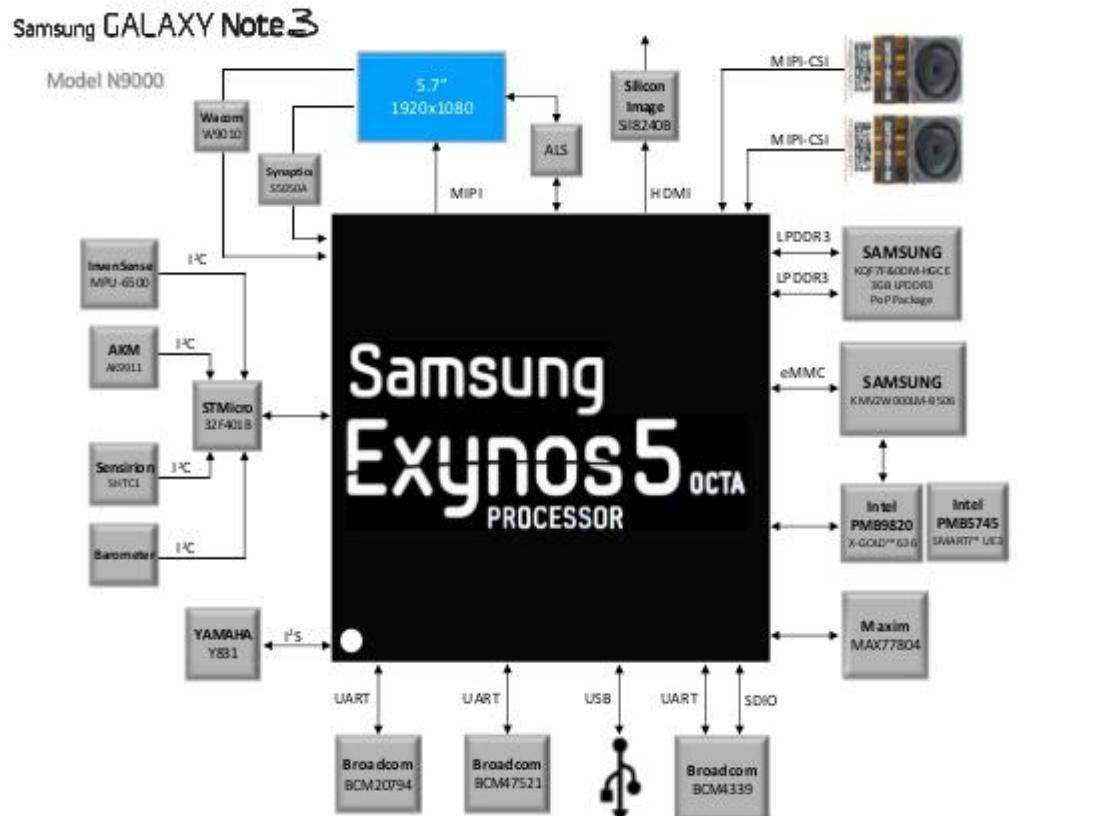
CLASSIFICATION OF EMBEDDED SYSTEM

APPLICATION-SPECIFIC INTEGRATED CIRCUITS (ASIC)

- ❑ ASIC is basically an IC designed and used by a single company in a specific system.
- ❑ Example: An IC designed for a specific line of cellular phones of a company, whereby no other company can use it.
- ❑ They are incredibly expensive, time-consuming, and resource-intensive to develop but extremely high performance coupled with low power consumption.
- ❑ Types of ASIC ICs:
 - ❖ Full-custom: entirely tailor-fitted to a particular application from the very start
 - ❖ Semi-custom: designed to allow a certain degree of modification during the manufacturing process.
 - ❖ Structured: built from a group of 'platform slices', with a 'platform slice' being defined as a pre-manufactured device, system, or logic for that platform.
 - ❖ Gate-array: ASIC are transistors which are predefined on the silicon wafer.

CLASSIFICATION OF EMBEDDED SYSTEM

APPLICATION-SPECIFIC INTEGRATED CIRCUITS (ASIC) - EXAMPLES



CLASSIFICATION OF EMBEDDED SYSTEM

GPP vs ASIP vs ASIC

| Key factors | GPP | ASIP | ASIC |
|-------------|--------------|------------------|------------------|
| Performance | Low | High | Very High |
| Flexibility | Excellent | Good | Poor |
| HW Design | None | Large | Very Large |
| SW Design | Small | Large | None |
| Power | Large | Medium | Small |
| Reuse | Excellent | Good | Poor |
| Market | Very Large | Relatively Large | Small |
| Cost | Mainly on SW | SOC | Volume sensitive |

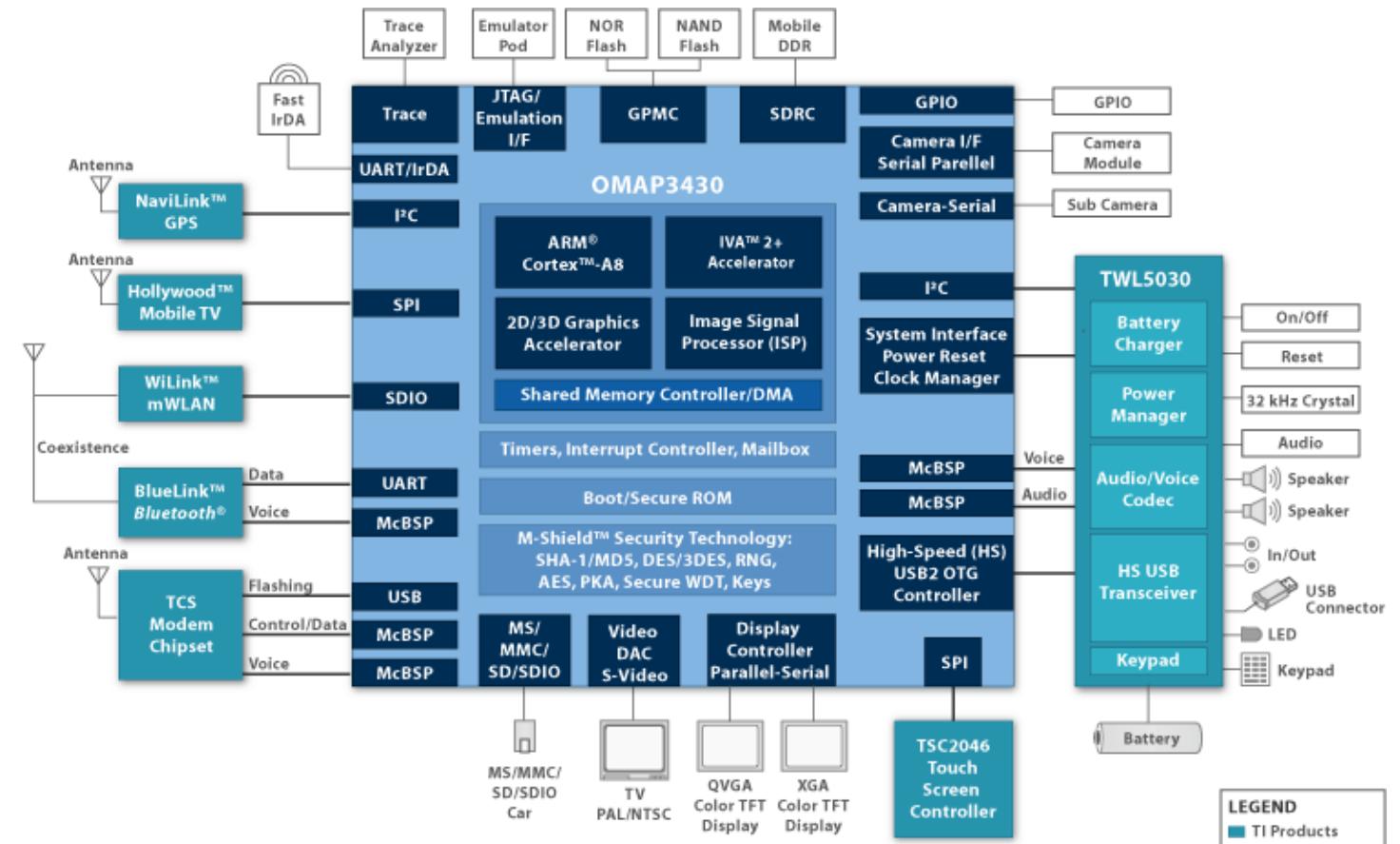
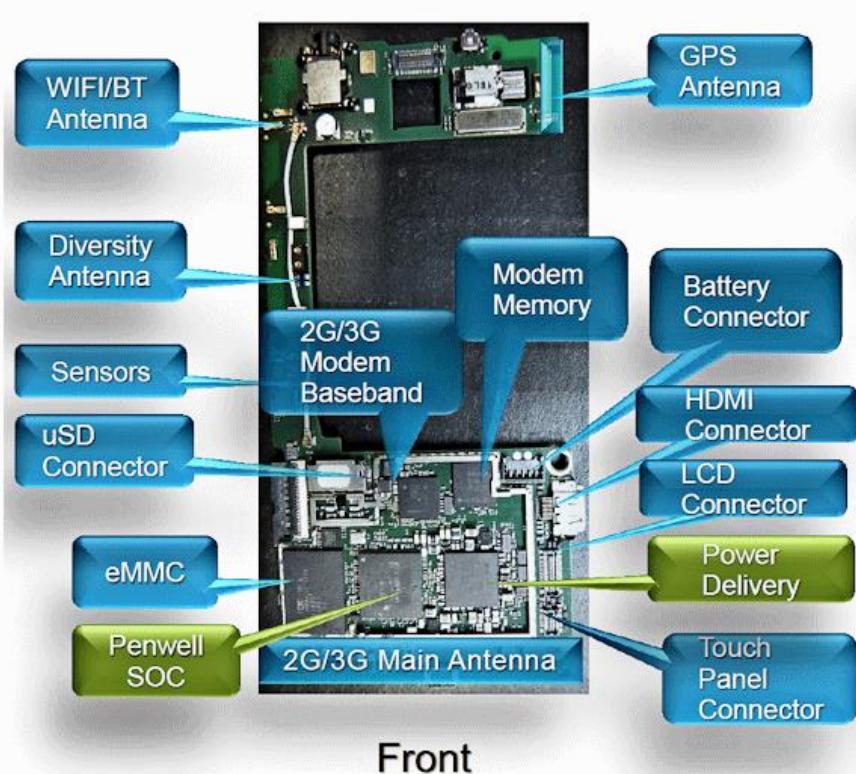
CLASSIFICATION OF EMBEDDED SYSTEM

SYSTEM ON CHIP (SoC)

- ❑ SoC is an integrated circuit (IC) that takes a single platform and **integrates an entire electronic system onto it.**
- ❑ It is, exactly as its name suggests, an **entire system on a single chip.**
- ❑ SoC contains **one or more processor cores** — microprocessors (MPs) and/or microcontrollers (MCs) and/or digital signal processors (DSPs) — **along with on-chip memory, hardware accelerator functions, peripheral functions etc.,**
- ❑ SoC can **perform a variety of functions** including signal processing, wireless communication, artificial intelligence etc.

CLASSIFICATION OF EMBEDDED SYSTEM

SYSTEM ON CHIP (SoC) - EXAMPLE



CLASSIFICATION OF EMBEDDED SYSTEM

SYSTEM ON CHIP (SoC) – EXAMPLE REDMI NOTE 4 (CPU-Z APP)

The figure consists of four screenshots of the CPU-Z application interface on an Android device, illustrating the System-on-Chip (SoC) components and system details for a Redmi Note 4.

SOC Tab:

- Model:** Qualcomm Snapdragon 625
- Brand:** Qualcomm
- Board:** msm8953
- Cores:** 8
- Architecture:** 8x ARM Cortex-A53 @ 2.02 GHz
- Revision:** r0p4
- Process:** 14 nm
- Clock Speed:** 652 MHz – 2.02 GHz
- CPU 0:** 2016 MHz
- CPU 1:** 2016 MHz
- CPU 2:** 2016 MHz
- CPU 3:** 2016 MHz
- CPU 4:** 2016 MHz
- CPU 5:** 2016 MHz
- CPU 6:** 2016 MHz
- CPU 7:** 2016 MHz
- CPU Load:** 57 %
- GPU Vendor:** Qualcomm
- GPU Renderer:** Adreno (TM) 506
- GPU Load:** 0 %

DEVICE Tab:

- Model:** Xiaomi Redmi Note 4
- Brand:** xiaomi
- Board:** msm8953
- Hardware:** qcom
- Screen Size:** 5.50 inches
- Screen Resolution:** 1080 x 1920 pixels
- Screen Density:** 400 dpi
- Dimensions:** 151.0 x 76.0 x 8.4 mm
- Weight:** 175 g
- Total RAM:** 3598 MB
- Available RAM:** 582 MB (16%)
- Internal Storage:** 52.16 GB
- Available Storage:** 1.37 GB (2%)
- Release Date:** 2016-08-01

SYSTEM Tab:

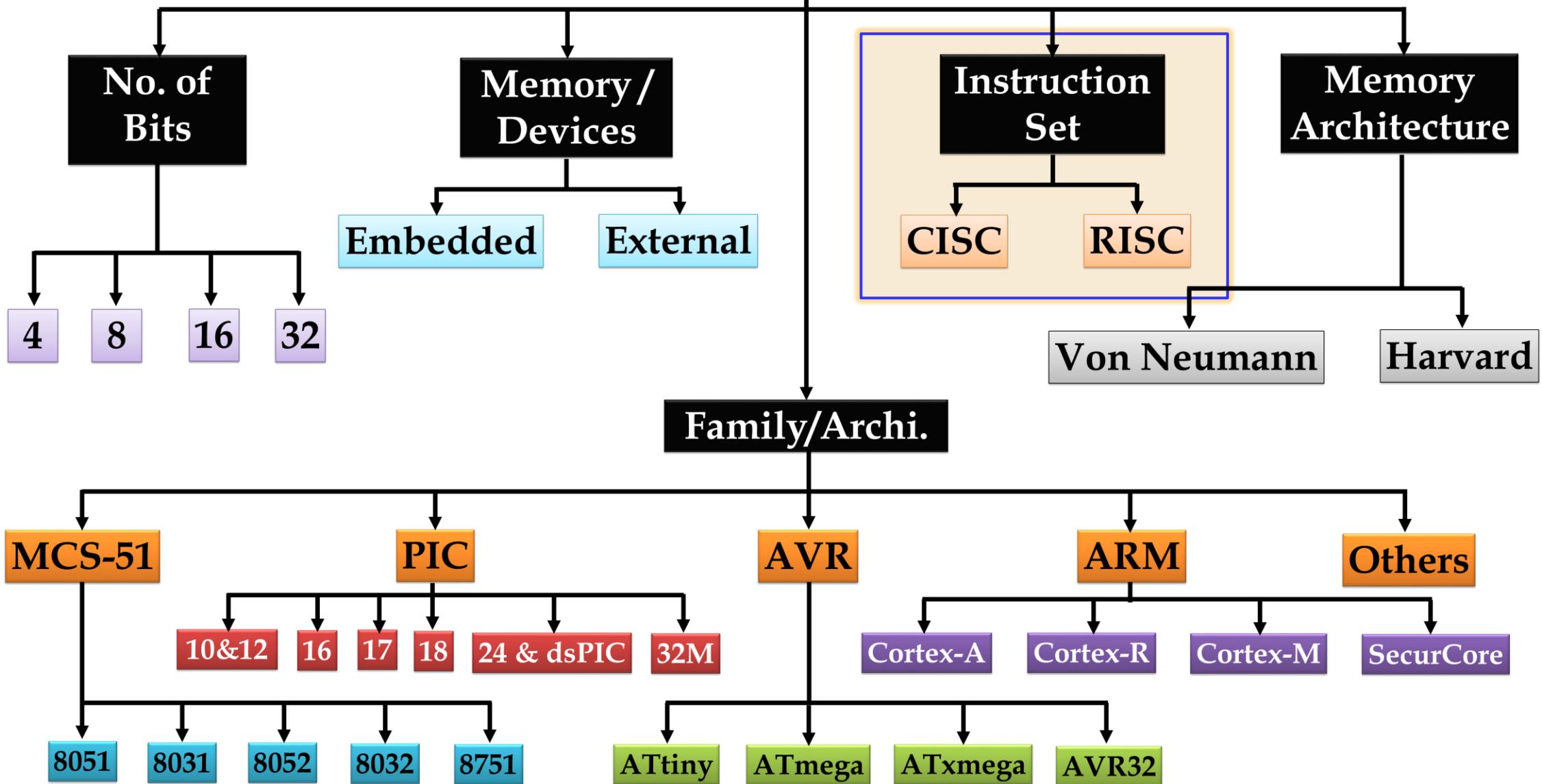
- Android Version:** 7.0
- API Level:** 24
- Security Patch Level:** 2018-05-01
- Bootloader:** unknown
- Build ID:** ALI1512-mido-build-20180614204110
- Java VM:** ART 2.1.0
- OpenGL ES:** 3.2
- Kernel Architecture:** armv8l
- Kernel Version:** 3.18.51-perf-g9a864f9 (V9.6.1.0.NCFMIFD)
- Root Access:** No
- System Uptime:** 3 days, 22:56:34

BATTERY Tab:

- LSM6DS3 Accelerometer:** X= 1.1 m/s² Y= 6.4 m/s² Z= 7.5 m/s²
- YAS537 Magnetometer:** 36.0 µT
- YAS537 Magnetometer Uncalibrated:** 153.8 µT
- LSM6DS3 Gyroscope:** X= 0.1 rad/s Y= 0.0 rad/s Z= 0.0 rad/s
- LSM6DS3 Gyroscope Uncalibrated:** X= 0.1 rad/s Y= 0.0 rad/s Z= 0.0 rad/s
- LTR579 ALSPS:** LTR579 ALSPS 79.0 lux
- LSM6DS3 Accelerometer – Wakeup Secondary:** X= 1.0 m/s² Y= 6.4 m/s² Z= 7.5 m/s²
- YAS537 Magnetometer – Wakeup Secondary:** 36.0 µT
- YAS537 Magnetometer Uncalibrated – Wakeup Secondary:** 153.8 µT
- LSM6DS3 Gyroscope – Wakeup Secondary:** X= 0.1 rad/s Y= 0.0 rad/s Z= 0.0 rad/s
- LSM6DS3 Gyroscope Uncalibrated – Wakeup Secondary:** X= 0.1 rad/s Y= 0.0 rad/s Z= 0.0 rad/s
- LTR579 ALSPS – Non Wakeup Secondary:** LTR579 ALSPS – Non Wakeup Secondary

MICROCONTROLLER ARCHITECTURES (RISC, CISC)

Types of Microcontrollers



MICROCONTROLLER ARCHITECTURES

INSTRUCTION SET – CISC & RISC

- **CISC - Complex Instruction Set Computer:** Allow single (complex) instructions to perform numerous low-level (simple) operations like a load from memory, arithmetic operation, store into memory with multiple clock cycle. Ex: Motorola 68K, 8051, x86 processors

MUL A, B : Get the value of A and B from registers, compute multiplication by repeated addition and store results back to registers

- **RISC - Reduced Instruction set Computer:** Reduce the instruction execution complexity by having several simple instructions which achieve low-level operation within a single clock cycle. Ex: AVR, PIC, ARM

LDR for loading, ADD with loop count for multiplication then STR for storing operations

MICROCONTROLLER ARCHITECTURES

INSTRUCTION SET – CISC & RISC

□ Example:

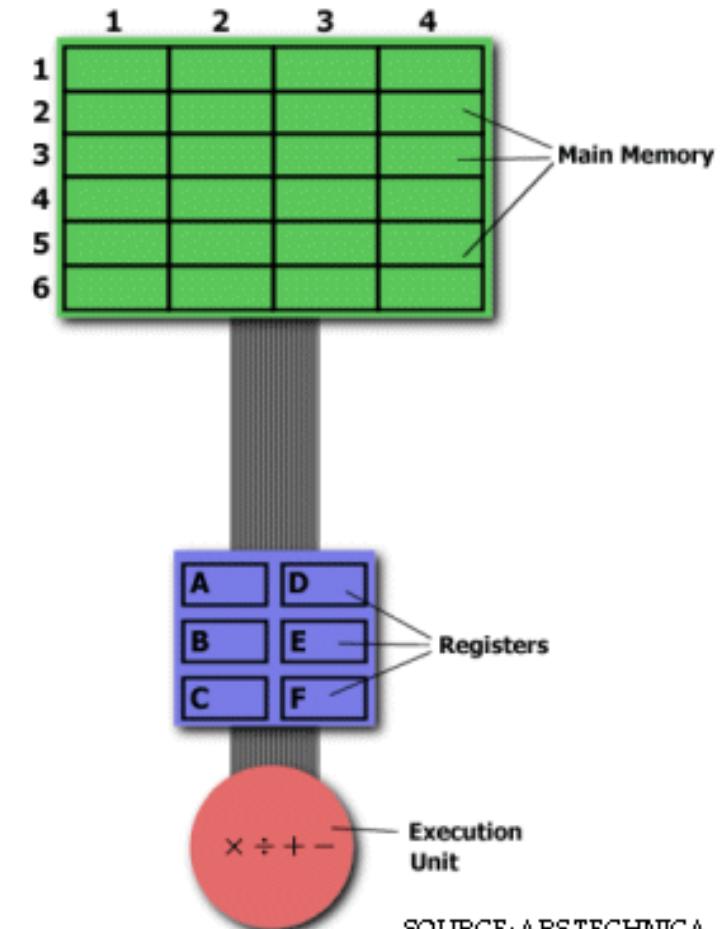
- Let's say we want to find the product of two numbers - one stored in location 2:3 and another stored in location 5:2 - and then store the product back in the location 2:3.

CISC Approach:

MULT 2:3, 5:2

RISC Approach:

**LOAD A, 2:3
LOAD B, 5:2
PROD A, B
STORE 2:3, A**



SOURCE: ARSTECHNICA

MICROCONTROLLER ARCHITECTURES

INSTRUCTION SET – CISC APPROACH

- ❑ The primary goal of **CISC** is to complete a task in as few lines of assembly as possible. This is achieved by building processor hardware that is capable of understanding and executing a series of operations.
- ❑ When **MULT 2:3, 5:2** executed, this instruction loads the two values into registers, multiplies the operands in the execution unit, and stores the product in the appropriate register.
- ❑ Thus, entire task of **multiplying two numbers can be completed with one instruction.**
- ❑ **MULT** is what is known as a "complex instruction". It operates directly on the memory and does not require the programmer to call any loading or storing functions.
- ❑ The main advantage of this system is that the **compiler has to do very little work to translate a high-level language statement into assembly**. Because the length of the code is relatively short, very little RAM is required to store instructions.

MICROCONTROLLER ARCHITECTURES

INSTRUCTION SET – RISC APPROACH

- ❑ RISC only use simple instructions that can be executed within one clock cycle. Thus, the "MULT" command could be divided into three separate commands:
 - "LOAD," which moves data from the memory bank to a register,
 - "PROD," which finds the product of two operands located within the registers,
 - "STORE," which moves data from a register to the memory banks.
- ❑ In order to perform the steps described in the CISC, 4 lines of code required in RISC. At first, this may seem like a less efficient way of completing the operation because there are more lines of code, more RAM is needed to store instructions.
- ❑ However, each instruction in RISC requires only one clock cycle, the entire program will execute in approximately the same amount of time as the multi-cycle "MULT".
- ❑ RISC require less transistors of hardware space, leaving more room for general purpose registers. Because all of the instructions execute in a uniform amount of time (i.e. one clock), pipelining is possible.

MICROCONTROLLER ARCHITECTURES

CISC vs RISC

| Key parameters | CISC | RISC |
|--------------------------|--|---|
| Program unit | microprogramming unit | hard-wired unit of programming |
| Performance optimization | Hardware based | Software based |
| Number of instructions | Large | Small |
| Type of instruction | Complex | Simple |
| Instruction cycle | One or More than one | One |
| Instruction length | Variable | Fixed |
| Instruction decoding | Complex | Simple |
| Instruction execution | Less Pipelined | Highly Pipelined |
| Execution time | More | Less |
| Applications | low-end applications such as security systems, home automation | high-end applications such as video processing, telecommunication |

EMBEDDED MEMORY

MEMORY ORGANISATION

Memory (both RAM and ROM) divided into a set of storage locations, each of which can hold 1 byte(8 bits) of data.

The storage locations are numbered, and the number of a storage location called its *address*, is used to tell the memory system which location the processor refers.

Important characteristics of a computer system is the width of the addresses it uses, which limits the amount of memory that the processor can address.

Most current computers use either 32-bit or 64-bit addresses, allowing them to access either 2^{32} or 2^{64} bytes of memory.

ADDRESS

| Address |
|----------------|
| 00000000 |
| 00000001 |
| : |
| . |
| 11111100 |
| 11111101 |
| 11111110 |
| 11111111 |

| Contents |
|-----------------|
| 11100011 |
| 10101001 |
| : |
| . |
| 00000000 |
| 11111111 |
| 10101010 |
| 00110011 |

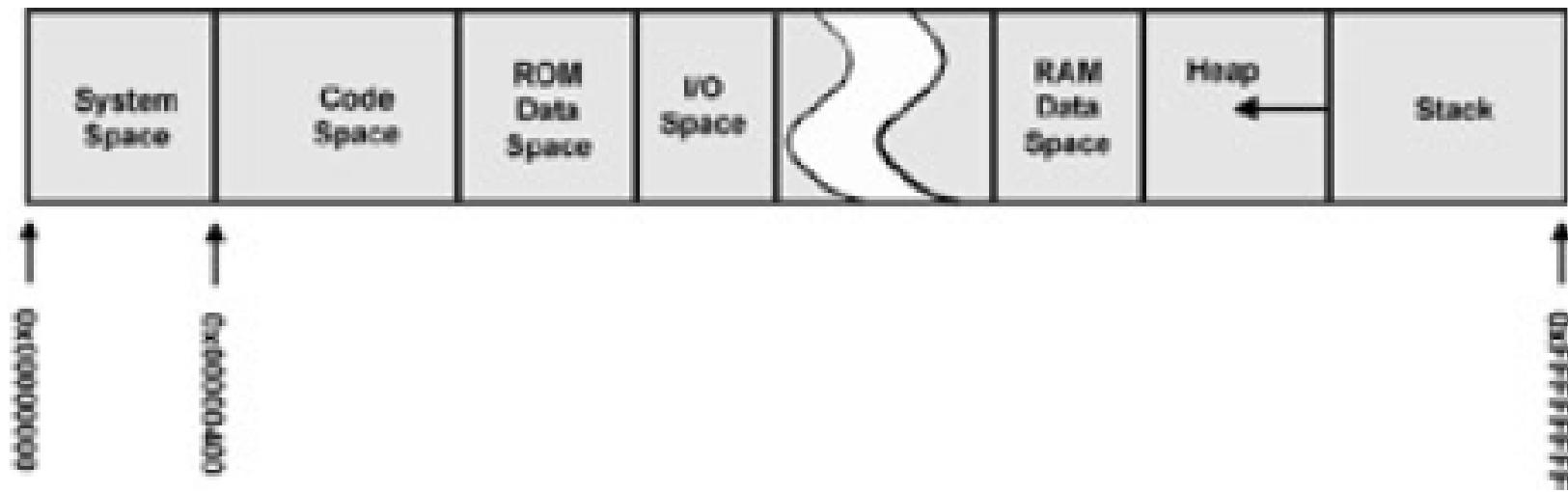
STORE AND LOAD INSTRUCTION

Most high performance organisations allows more than 1- byte of memory to be loaded or stored at a time.

Load – Store operation operates on a quantity of data equal to system bus width.

EMBEDDED MEMORY

MEMORY ORGANISATION



**Figure 4.1: Memory map of processor.
Memory model for a 68K family processor.**

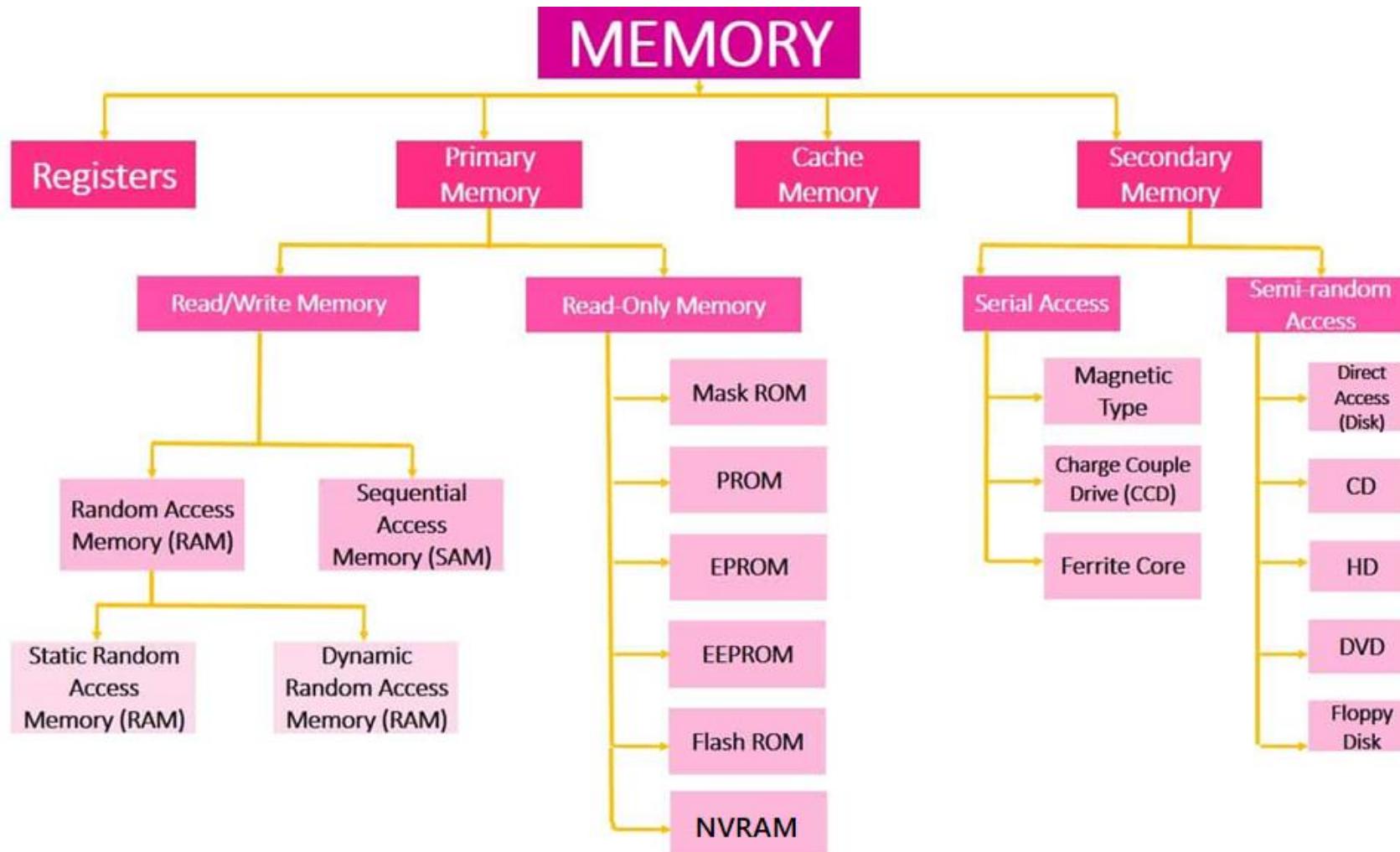
EMBEDDED MEMORY

MEMORY ORGANISATION

- **System Space – Exception Vectors**
- **Code Space – Stores the Instruction**
- **ROM Data Space – Stores the constants e.g. error messages**
- **Stack – Context Switching, grows downwards**
- **Free Memory – All Statically allocated variables**
- **Heap – All dynamically allocated variables**
- **I/O Space – Memory mapped I/O devices**

EMBEDDED MEMORY

EMBEDDED MEMORY DEVICES - CLASSIFICATION



EMBEDDED MEMORY

MEMORY CLASSIFICATION - RAM

□ **Static RAM (SRAM):**

- Retains its contents as long as electrical power is applied
- Uses bi-stable latching circuitry to store each bit
- Offer extremely fast access time
- Complicated storage process
- High production cost
- Limited Storage capacity
- Used in applications where access speed is extremely important
- Used as instruction and Data caches in a processor system

EMBEDDED MEMORY

MEMORY CLASSIFICATION - RAM

Dynamic RAM (DRAM):

- Stores each bit in a storage cell consisting of capacitor and transistors
- Extremely short data lifetime (typically few ms)
- Each bit in the DRAM refreshed periodically to maintain its contents
- Specialized controller required to take care of refreshing
- Lower cost-per-byte compared to SRAM due to higher density
- Less access speed
- High power consumption
- Used in applications whenever large amounts of RAM are required

EMBEDDED MEMORY

MEMORY CLASSIFICATION - ROM

□ MASKED ROM:

- Data stored in ROM remains unchanged even power is turned off
- Contains a pre-programmed set of data/inst. and it cannot be modified
- Allow only read and cannot perform write operation
- Produced by arranging transistors before manufacturing actually begin
- Cost is low when large quantities of the same ROM are produced
- IC area per bit for masked ROMs is generally lower
- Common application is the BIOS in the computer

EMBEDDED MEMORY

MEMORY CLASSIFICATION - ROM

□ PROM (Programmable ROM):

- Also known as one-time programmable (OTP) devices
- Generally comes in an unprogrammed state
- Once programmed, contents can never be changed
- Programming require special equipment called device programmer
- Writes data to the device one word at a time
- Apply an electrical charge to input pins of the chip for writing data
- PROMs are inexpensive
- Used for prototyping the data for a ROM before costly ROM fabrication

EMBEDDED MEMORY

MEMORY CLASSIFICATION - ROM

□ **EPROM (Erasable and Programmable ROM):**

- A single field effect transistor constitutes a storage location
- Programmed in exactly the same manner as a PROM
- Supports erasing and reprogramming multiple times
- To erase expose the device to a strong source of ultraviolet light
- More expensive than PROMs
- Not possible to erase a particular byte of data and take more time
- Static power consumption is quite high.
- Used in software development and testing process

EMBEDDED MEMORY

MEMORY CLASSIFICATION - ROM

□ **EEPROM(Electrically Erasable and Programmable ROM):**

- Memory cell will comprise two FET - storage & access transistor
- Uses the same basic principle used by EPROM memory technology
- Erase operation is performed electrically instead of ultraviolet light
- Since no equipment required, chip need not to remove for reprogram
- Individual bytes of data can be erased and reprogrammed but slow
- Number of times it can be reprogrammed is limited (10-1000 cycles)
- Used in computers and other electronic devices to store small amount of data that must be saved when the power supply is removed

EMBEDDED MEMORY

MEMORY CLASSIFICATION - ROM

□ FLASH:

- Similar to EEPROM except flash erases one sector at a time
- Typical sector sizes are in the range 256 bytes to 16KB
- High density hence smaller in size
- Operates with very low power consumption
- Low cost and fast (to read, but not to write)
- Designed for about 10,000 – 100,000 write cycles
- Susceptible to bit disturbance, so it require error detection algorithm
- Mostly used in µCs and other electronics device to store the firmware

EMBEDDED MEMORY

CHARACTERISTICS OF VARIOUS EMBEDDED MEMORY DEVICES

| Type | Volatile? | Writeable? | Erase Size | Max Erase Cycles | Cost | Speed |
|------------|-----------|------------|-------------|------------------|-------------|-----------------------------------|
| SRAM | Yes | Yes | Byte | Unlimited | Expensive | Fast |
| DRAM | Yes | Yes | Byte | Unlimited | Moderate | Moderate |
| Masked ROM | No | No | n/a | n/a | Inexpensive | Fast |
| PROM | No | Once | n/a | n/a | Moderate | Fast |
| EPROM | No | Yes | Entire Chip | Limited | Moderate | Fast |
| EEPROM | No | Yes | Byte | Limited | Expensive | Fast to read, slow to erase/write |
| Flash | No | Yes | Sector | Limited | Moderate | Fast to read, slow to erase/write |
| NVRAM | No | Yes | Byte | Unlimited | Expensive | Fast |

EMBEDDED MEMORY

APPLICATIONS

- Embedded microcontrollers usually have both SRAMs (a few kB for critical data path) and DRAMs (in MB for everything else)
- Masked ROM serve the function of storing the bootloaders in microcontrollers and to store microcode on microprocessors.
- PROM used to store firmware and constants in the source code of applications like TV, washing machine and microwave ovens
- Similar to PROM, EPROM are also used to store firmware and constants in the source code especially in development phase.

EMBEDDED MEMORY

APPLICATIONS

- ❑ EEPROM used for storing updatable firmware and runtime constants after production also for storing current date & time, port status
- ❑ Microcontrollers uses flash memory for storing firmware of large size, constant data and large lookup tables as needed by application
- ❑ Flash also used for storing user data like picture in a digital camera, voice data in a voice recorder, messages and contacts in mobile
- ❑ NVRAM is used in RTOS applications where start-up time is extremely important, and we cannot afford to lose even μ s of time

STRATEGIC SELECTION OF PROCESSOR & MEMORY

PROCESSOR SELECTION CRITERIA

- ❑ To design an efficient embedded system, selection of right processor is very important and challenging task
- ❑ Types of processors: μP, μC, Digital signal processor (DSP)
- ❑ μP are offered in 4 to 64-bit size with distinct features like cost, speed, no. of CPU core, address & data line are used in simple toys to network router
- ❑ μC plays an important role in embedded system design and majorly used in low-end to high-end control applications
- ❑ DSP are majorly used for high computation intensive applications such as image processing, communication devices, voice to text converter etc.,

PROCESSOR SELECTION CRITERIA

- Sequence of analysis to be made selecting an appropriate processor for embedded system applications as follows,
 - Application requirement analysis: understand the purpose of application and arrive specific requirement
 - Processor Architecture analysis: MCS51, ARM, PIC, PowerPC, MIPS etc.,
 - Peripheral set analysis: Includes on-chip (RAM, ROM ,I/O Ports, ADC) and specialized processing units (FPU, MMU, DMA)
 - Technical analysis: Execution speed, operating voltage, power consumption, and data & address bus size etc.,
 - Non-technical analysis: Cost, software tools, package type, vendor reputation, support etc.,

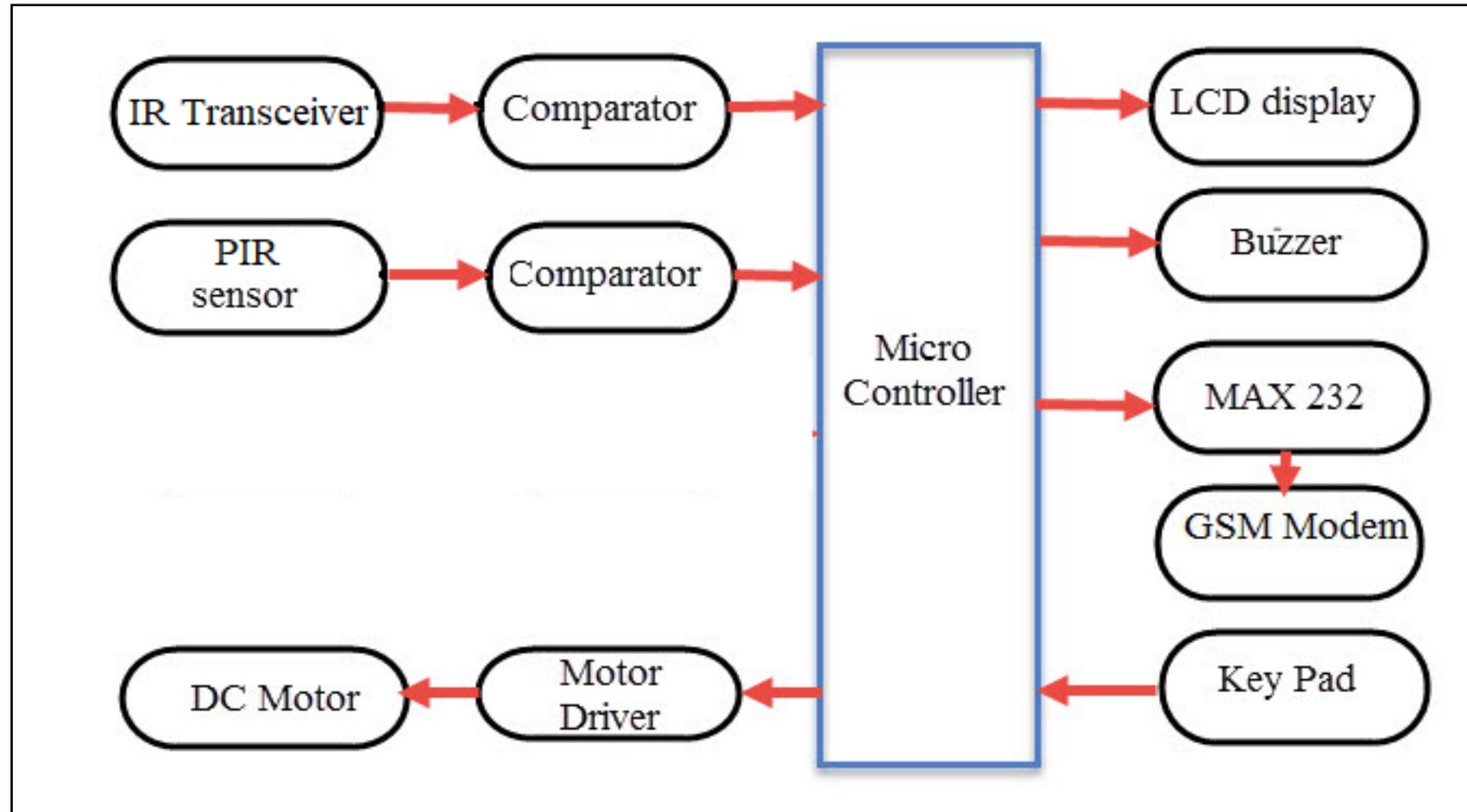
PROCESSOR SELECTION CRITERIA

Case study-1: Home security system

This application consists of three main modules (1) Intruder detection (2) password based door lock system (3) Send SMS to owner using GSM module. The intruder detection system consists of IR and PIR sensor interfaced with processing unit to detect and alert under human presence condition. In password based door lock system numeric keypad to accept the password from user and LCD to display the message whether permission is granted or not. Upon receiving correct password signal, enable motor to open the door. In case of password mismatch or intruder detection condition activate the buzzer and send a SMS to owners' mobile to alert the house owner. All these actions are expected to carried with execution speed of approximately $1\mu\text{s}$ per instruction. Since no complex time-constraint action need to be performed a simple 8-bit microcontroller of CISC architecture is sufficient. Also, special peripherals or processing units like ADC, FPU, DMA are not required since on-chip peripheral and memory is sufficient for implementation. Hence, low-range 8-bit microcontrollers are suitable choice.

PROCESSOR SELECTION CRITERIA

Case study-1: Home security system



PROCESSOR SELECTION CRITERIA

Case study-2: Smartwatch

Smartwatch require moderate processing power to manage complex algorithms and perform sensor fusion to provide better information for the user. This require a pipelined super scalar RISC architecture based processing unit. Also, the processor consume low power with 'always-on, always-aware' feature for continuous monitoring of sensors. Necessary ADC modules need to process analog signals from various sensors. But floating-point unit is not required since complex mathematical computations are not involved. Running a simple RTOS at an operating frequency of between 20MHz and 150MHz, the design can provide months of battery life. In addition, necessary memory unit to store user information and peripherals to support small LCD/OLED display. To interact with user necessary communication module such as Bluetooth, NFC and GPS module the processor should have necessary capability. Any of the ARM Cortex (M0, M0+, M3, M4) ultra-low-power processor cores can be used for 'always-on' sensor fusion processing and other modules can be interface external to this processor.

PROCESSOR SELECTION CRITERIA

Case study-2: Smartwatch

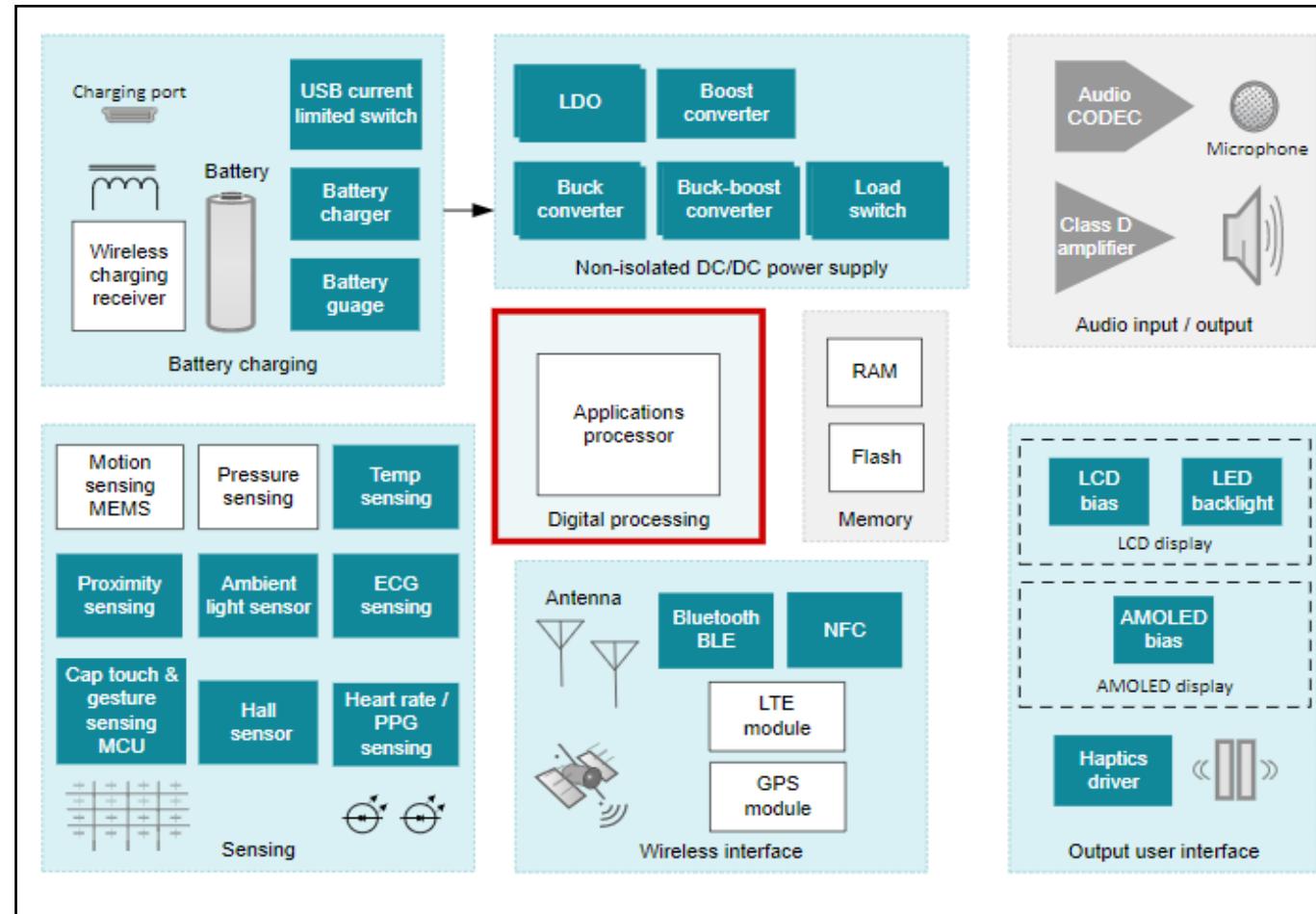


Image source: ti.com Ref. URL: <https://www.ti.com/solution/smartwatch?variantid=34352&subsystemid=27272#technicaldocuments>

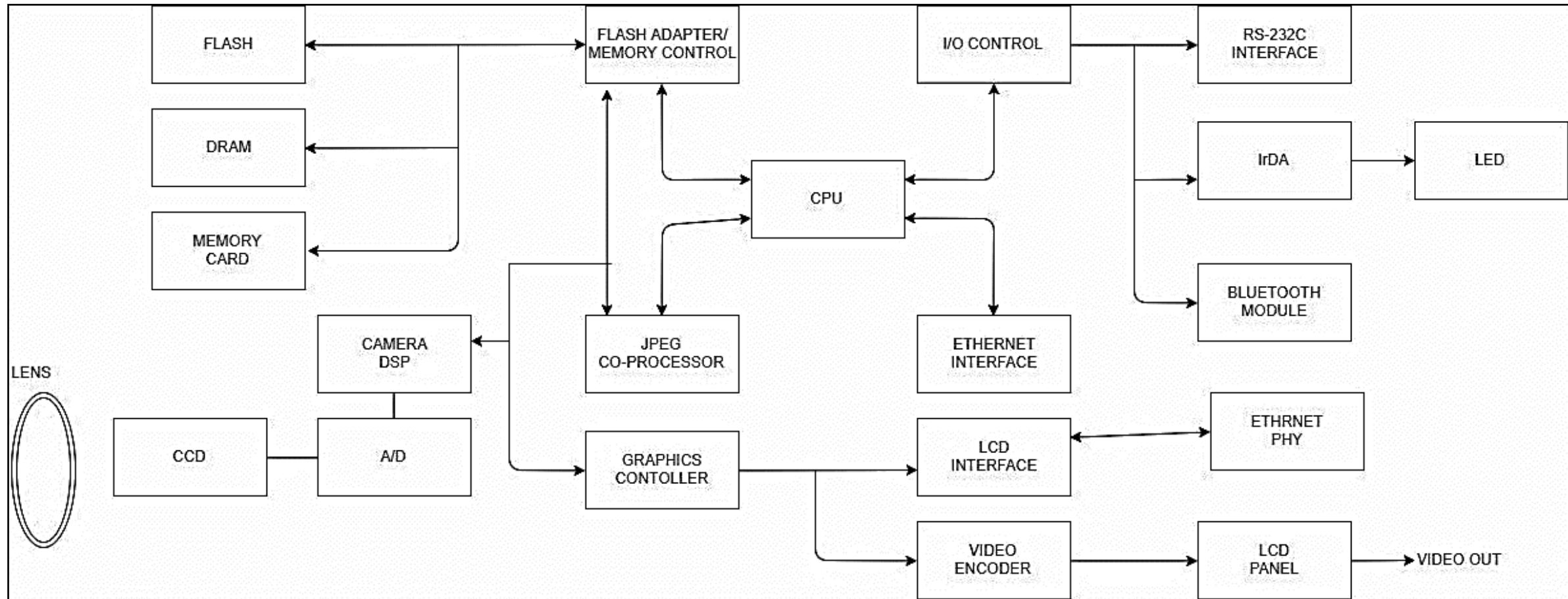
PROCESSOR SELECTION CRITERIA

Case study-3: Digital Camera

In digital camera, high computation capability based processor required for real-time image/video processing. Also the host processor should control various complicated operations such as image rotation, shadow correction, adjusting brightness, contrast, colour and hue, image stabilization, image sharpening, filter function, lens adjustment etc., **Battery recharging after 400 pictures. Shooting a 4M pixels still picture in 0.5s with 25 pictures per minutes.** Allow to save image/video in standard format (.JPEG) on memory card. Allow to transfer files to PC or other device through **USB port or Bluetooth connectivity.** Display the picture on the screen after capturing along with details of the picture such as date, time, size and serial number. A multiprocessor based processing unit of ARM processor with DSP deliver a best performance for this application. **The camera DSP processes the images taken by CCD camera after it is converted to digital form.** ARM processor perform various control operation including displaying the images and videos on the LCD panel through the LCD controller interface. The JPEG co-processor is mainly meant to compress and decompose image into JPEG format.

PROCESSOR SELECTION CRITERIA

Case study-3: Digital Camera



PROCESSOR SELECTION CRITERIA

| | REQUIRED FEATURES | HOME SECURITY SYSTEM | SMARTWATCH | DIGITAL CAMERA |
|------------------------|------------------------|----------------------|----------------|------------------------------|
| Processor | Processor required | Microcontroller | Microprocessor | Multiprocessor(μ P+DSP) |
| On-chip peripherals | Processor architecture | CISC | RISC | RISC |
| Pipelined execution | No | | Yes | Yes |
| Performance level | Low | | Moderate | Very high |
| On-chip ROM | Sufficient | | Not sufficient | Not sufficient |
| On-chip RAM | Sufficient | | Not sufficient | Not sufficient |
| GPIO pins | 20 or more | | 40 or more | 40 or more |
| ADC | No | | External | Yes |
| Timers | 2 | | 5 | 5 |
| Interrupts | 2 | | 10-15 | 16-32 |
| Real-time clock | No | | Yes | Yes |
| Communication protocol | UART | | BT, USB, UART | UART, BT, USB |

PROCESSOR SELECTION CRITERIA

| | REQUIRED FEATURES | HOME SECURITY SYSTEM | SMARTWATCH | DIGITAL CAMERA |
|-----------------------|---------------------------|--------------------------------|-----------------------------------|---|
| Specialized processor | MMU | No | No | Yes |
| | FPU | No | No | Yes |
| | DMA | No | No | Optional |
| | External ROM | No | Yes | Yes |
| | External RAM | No | Yes | Yes |
| Technical | Data bus width | 8 | 32 | 32 or 64 |
| | Address bus width | 16 | 32 | 32 or 64 |
| | Clock frequency | 10 -50MHz | 20-150MHz | 66-40 0MHz |
| | Execution speed (in mips) | 1 μ s | < 1 μ s | 1 to 5 ns |
| | Operating voltage | 3.3 – 5V | 1.8 – 3.3V | 1.8 – 5V |
| | Low power mode support | No | Yes | No |
| | RTOS support | No | Yes | Yes |
| | Suitable processor family | 8051, PIC16F Series, ATmega | ARM Cortex-M (M0,M3,M4) family | ARM Cortex-A with TMS series DSP, Power PC |

MEMORY SELECTION CRITERIA

- ❑ Other than processor, **memory device play a major role** in deciding the performance of the embedded device
- ❑ Systems memory requirement **depend primarily on the nature of the application** that is planned to run on the system
- ❑ Memory performance and capacity **requirement for low cost systems are small** hence memory within the microcontroller meet the requirements
- ❑ While larger/complex system demand external memory and **small access time to achieve high performance level**
- ❑ In addition to many technical factor, few **non-technical factors also plays important role** in memory selection process

MEMORY SELECTION CRITERIA

□ Technical factors

- Data capacity : KB, MB, GB
- Access time : Read/write operation
- Data & address bus width : 8/16, 16/32, 32/32, 32/64
- Data storage size : Byte, Page, Block, Sector
- Erase/write cycles : 1 to 1,00,000 cycles
- Power consumption : 1.8 to 5.5V

□ Non-technical factors

- Cost : High (volatile), low (non-volatile)
- Battery life : For DRAM, NVRAM
- Tools for rewrite operation : Programmer device
- Data retention period : 0 – 10 years

MEMORY SELECTION CRITERIA

Case study-1: Home security system

In this application, the system is not performing any high computational operation like image or video processing. Therefore small amount of **on-chip RAM memory (128 – 512 Bytes)** is sufficient. However, to store the status of sensors, GSM module related information an EEPROM/FLASH is needed. Since this information consume few bytes of memory an **on-chip EEPROM of size 1 to 2KB** more appropriate. In addition, **4 – 8 KB of on-chip ROM** memory required to store program. Therefore, no external memory devices are required since all memory requirements are fulfilled by on-chip memory of the microcontroller unit.

MEMORY SELECTION CRITERIA

Case study-2: Smartwatch

Smartwatch require moderate processing power to manage complex algorithms and perform sensor fusion to provide better information for the user. This require a microprocessor based control unit hence all necessary memory devices needs to be connected externally. To process information from sensor, power unit, communication module and user interface external **RAM of 128-512 MB** required. To handle multi-functional operation complex algorithm and embedded software storage require **ROM of 8-64 MB**. Additional **flash memory of 1-4 GB** need to recording information from all modules.

MEMORY SELECTION CRITERIA

Case study-3: Digital Camera

In digital camera, high computation capability based processor required for real-time image/video processing. Also the host ARM processor should control various complicated operations such as image rotation, shadow correction, adjusting brightness, contrast, colour and hue, image stabilization, image sharpening, filter function, lens adjustment etc., To manage all these operation sufficient amount of **RAM size 128 – 512MB** for storing temporary variables and stack. And, **ROM size of 64 – 512 MB** for application codes and RTOS codes for scheduling the tasks. To store pictures a memory stick size of **16 – 64 GB** of **flash** based memory stick is required. The camera DSP processes the images taken by CCD camera after it is converted to digital form. To carry out this operation **on-chip RAM of 256-4096 KB** and **ROM of 512-4096 KB** required.

MEMORY SELECTION CRITERIA

| | MEMORY TYPE | HOME SECURITY SYSTEM | SMARTWATCH | DIGITAL CAMERA |
|----------|------------------------|----------------------|----------------|-------------------------------|
| INTERNAL | Processor used | Microcontroller | Microprocessor | Multiprocessor (μ P+DSP) |
| | RAM | 128 – 512 bytes | - | 256 – 4096 KB |
| | ROM | 4 – 8 KB | - | 512 – 4096 KB |
| | EEPROM or FLASH | 1 – 2 KB | - | |
| EXTERNAL | RAM | - | 128 - 512 MB | 128 - 512 MB |
| | ROM | - | 8 – 64 MB | 64 – 512 MB |
| | EEPROM or FLASH | - | 1 – 4 GB | 16 – 64 GB |

POWER SUPPLY DESIGN CONSIDERATIONS FOR EMBEDDED SYSTEMS

POWER SUPPLY DESIGN CONSIDERATIONS

- ❑ The power supply **provides the necessary electrical energy** to the embedded components, ensuring their proper functioning.
- ❑ Designing the power supply for embedded systems is a critical aspect of the overall system design, also it **influence some of the product's capabilities and functions**.
- ❑ Some key considerations when designing power supplies for embedded systems are:
 1. **Power Requirements Analysis:** Understand the power requirements of each component in the embedded system. Consider both active (during operation) and standby (during idle or sleep mode) power requirements.
 2. **Power Source Selection:** Select an appropriate power source based on the specific application. Common sources include batteries, AC mains, or a combination of both.

POWER SUPPLY DESIGN CONSIDERATIONS

3. **Voltage Regulation:** Use voltage regulators to ensure a stable and reliable power supply. This avoid damage to sensitive components due to voltage fluctuations.
4. **Efficiency:** Choose power supply components with high efficiency to minimize energy waste and maximize battery life in portable embedded systems.
5. **Battery Management:** If using batteries, implement effective battery management techniques, such as charging control, over-discharge protection to extend battery life.
6. **Temperature Considerations:** Ensure that the power supply components can operate within the specified temperature limits.
7. **Fault Tolerance:** Design the power supply with built-in fault tolerance mechanisms to handle issues such as overvoltage, undervoltage, overcurrent, and short circuits.

POWER SUPPLY DESIGN CONSIDERATIONS

8. **EMI/RFI Mitigation:** Implement filtering techniques to reduce electromagnetic interference (EMI) and radio-frequency interference (RFI) generated by the power supply, ensuring compliance with regulatory standards.
 9. **Size and Form Factor:** Choose power supply components that meet the size and form factor constraints of the embedded system, especially in space-constrained applications.
 10. **Low Power Modes:** Implement low-power modes for components during idle periods to conserve energy, especially in battery-powered embedded systems.
- By carefully addressing these considerations, you can design a robust and efficient power supply for your embedded system, **ensuring reliable and optimal performance.**

THANK YOU

NOU NHATH

