Software Application development at device level

Module : Introduction to Microprocessors & Microcontrollers

Basic Concepts on Microprocessors & Microcontrollers

Microprocessor

- Microprocessor is a multipurpose, clock driven, register based, digital integrated circuit that accepts binary data as input, processes it according to instructions stored in its memory, and provides results as output
- heart of the PC based computer system
- ALU and Registers are main part of microprocessor
- 32 bit, 16 bit, 8 bit variants
- ex. 8085, 8086, ARM

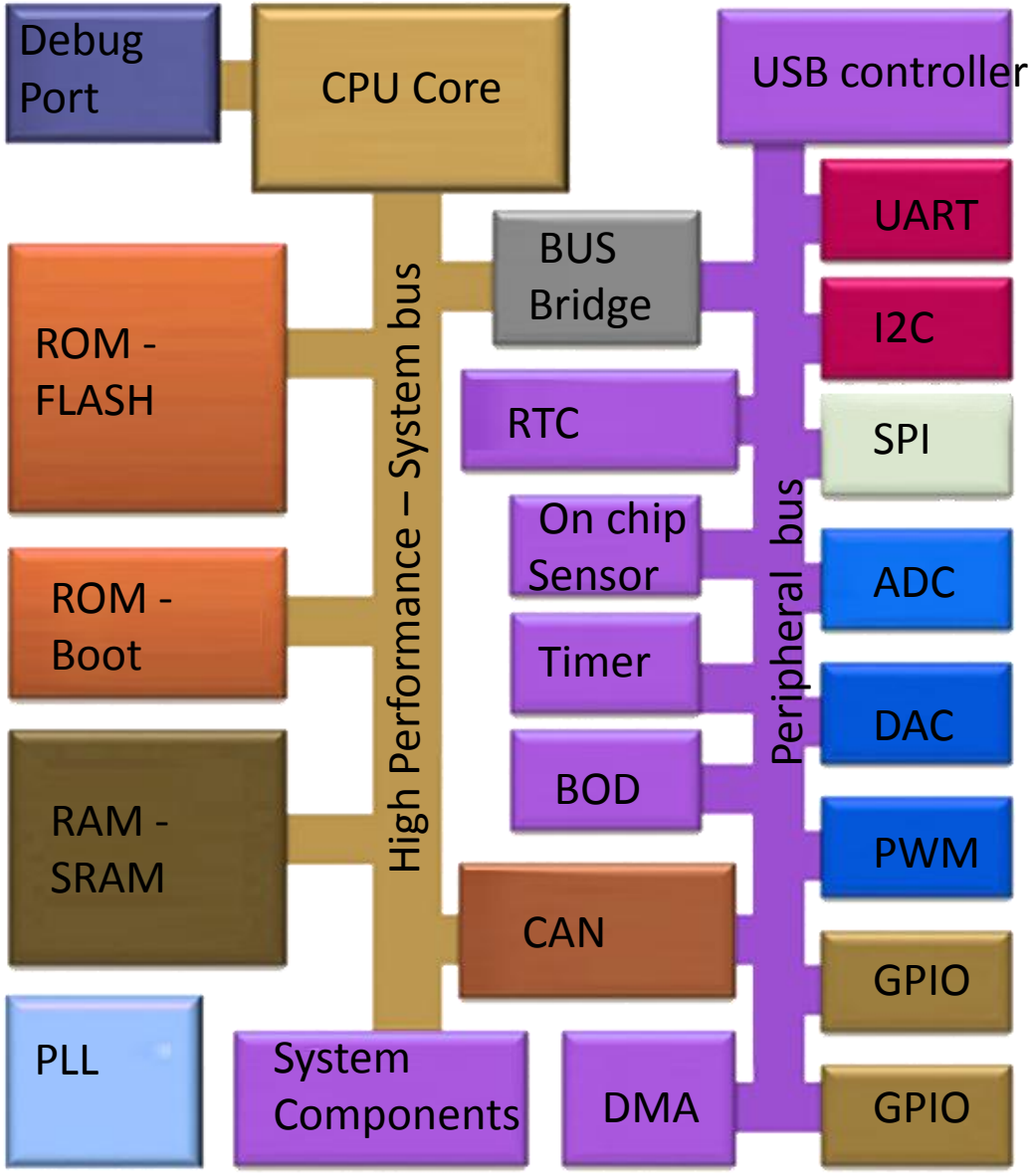
Ref : https://en.wikipedia.org/wiki/Microprocessor

Microcontroller

- 10 % of silicon is occupied by processor
- 90 % of silicon is occupied by peripherals
- Microcontroller has microprocessor plus various peripherals interconnected via buses
- peripherals – ram, rom, communication interfaces, timers, adc, dac, dma, power management
- single chip computer or SOC
- eg. avr, st microelectronics, freescale

Processor and Controller

- Processor core interacts with core/system peripherals over high performance System Bus
- Peripherals are interconnected with processor core via Peripheral Bus



Microprocessor & Microcontroller

Microprocessor	Microcontroller
Heart of PC based computer systems	Heart of embedded/IoT systems
Contains ALU, GP registers and Clock timing circuit	It has processor & various peripherals included in single chip
Over all circuit is large	Over all circuit is small
General purpose computing	Special purpose computing
Can`t be used in Embedded systems	Designed to be used in embedded systems
High system cost	Lesser cost
Power consumption is high	Power consumption is low

Microprocessor & Microcontroller

Microprocessor	Microcontroller
No/less efficient power saving features	Efficient power saving features
External memory interactions are slow	On chip memory access so faster
Less number of GP registers	More number of registers
ex : 8085, 8086, 80386	example : 8051, AVR, ARM

Von-Newman & Harvard Architecture

Von-Newman Architecture	Harvard architecture
Same physical memory Buses for instruction and data	separate physical memory Buses for instruction and data
Single memory storing both instructions & data	Separate memory blocks for storing data & instructions with separate buses
simpler control unit design	control unit for 2 buses is complicated
Instruction fetch & accessing data memory can’t be done simultaneously	can be performed at the same time

Von-Newman & Harvard Architecture

Von-Newman Architecture	Harvard architecture
No parallelism	Parallelism
low performance as compared to Harvard architecture	easier to pipeline, so high performance can be achieved
comparatively cheaper	comparatively high cost
e.g. 8085, 68K series, ARM7	ARM Cortex M, ARM9, PIC and DSP processors

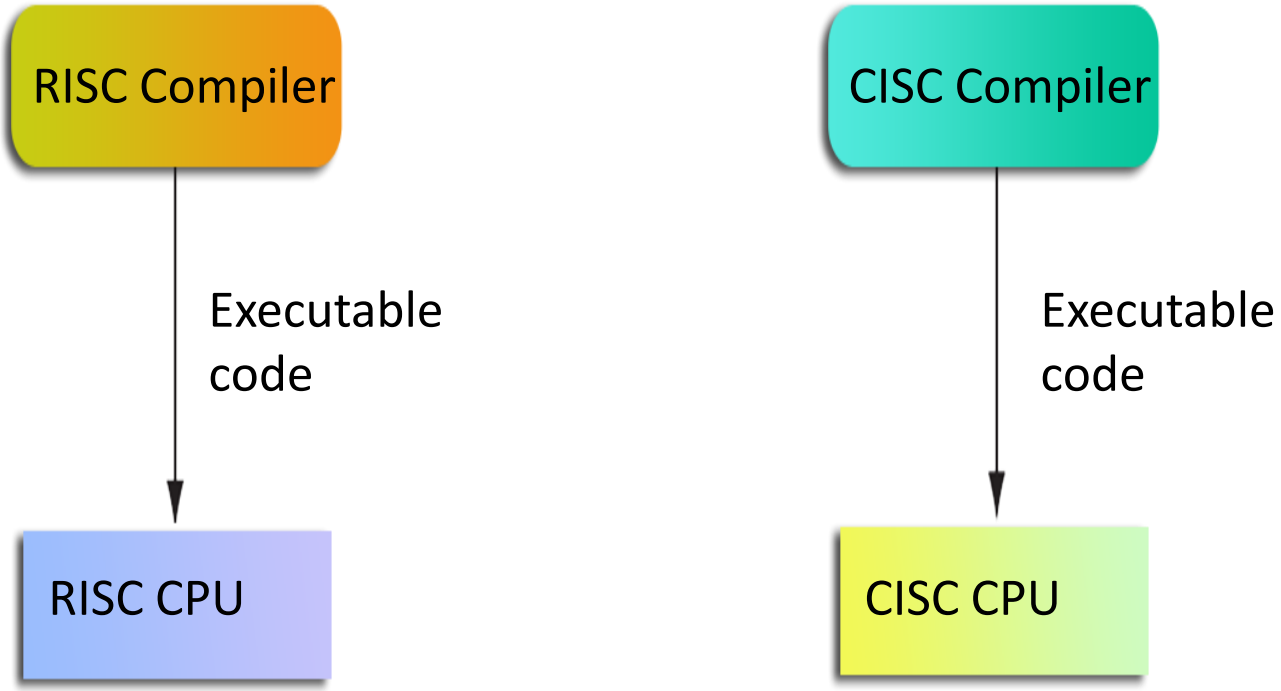
RISC vs. CISC

RISC	CISC
Reduced instruction set computing – no. of instructions & its complexity	Complex instruction set computing – no. of instructions & its complexity
Designed to make hardware simple	Designed to make hardware complex
ALU instructions are single-cycle, and reduced instruction	Includes multi-clock, complex instructions
Instruction width is fixed	Variable size instructions
Follows LOAD/STORE architecture	Data Processing & memory access can be combined in single instructions

RISC vs. CISC

RISC	CISC
Large code sizes	Small code sizes
e.g. ARM	e.g. 8085,8086,80386
Requires less transistors for implementation, so low power consumption	Requires more transistors for implementation, so higher power consumption
pipelined	Pipelining is difficult

RISC & CISC Architectures



CISC – emphasis on hardware complexity

RISC – emphasis on software complexity

Little Endian and Big Endian

Little endian	Big endian
lsb is stored at lower memory address	lsb is stored at higher memory address
Intel x86 and x86_64	Motorola 68000 series

Polling Vs. Interrupt Technique

Polling	Interrupt
CPU polls/checks peripherals periodically for event	Peripherals triggers signals to CPU for events
Events are handled through software	Events are handled by hardware
Latency can’t be deterministic	Deterministic latency
Not efficient	Efficient

IO Mapped IO Technique

- IO devices are directly interfaced to the I/O space of the processor.
- Separate instruction set available for accessing the I/o (Like IN & OUT)
- This does not use memory related instructions.

Memory Mapped IO Technique

- IO devices are interfaced in the memory space of the processor.
- The memory space is the RAM space of the processor.
- The memory space could be internal RAM or External RAM.
- Internal RAM memory mapping of I/o devices is defined by the processor.
- External RAM memory mapping of I/o devices has to designed by the hardware designer.

References

- ARM System Developers Guide Designing and Optimizing System Software by Andrew N Sloss
- The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors Third Edition