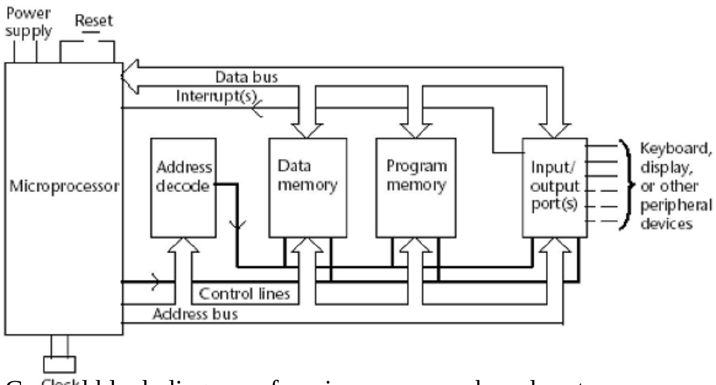
Embedded Systems Hardware Core

Embedded systems are domain and application specific and are built around a central core. The core of the embedded system mainly consists of a processor. For an embedded system designer, knowledge of microprocessors and microcontrollers is a prerequisite.

General purpose Processors

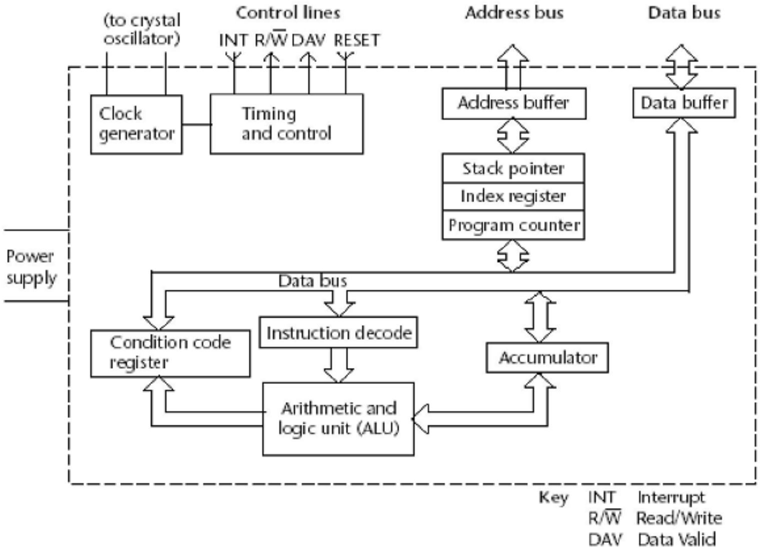
Almost 80% of the embedded systems are processor/controller based. The processor may be microprocessor or a microcontroller or digital signal processor, depending on the domain and application. A microprocessor is a single VLSI silicon chip that has a CPU and may also have some other units (caches, floating point processing arithmetic unit, pipelining, and super-scaling units) that are additionally present and that result in faster processing of instructions. A microprocessor is a dependent unit and it requires the combination of other hardware like memory, timer unit, and interrupt-controller, etc. for proper functioning. A microprocessor has two essential units: Program Flow Control Unit (CU) and Execution Unit (EU). The CU includes a fetch unit for fetching instructions from the memory. The EU has circuits that implement the instructions pertaining to data transfer operations and data conversion from one form to another. The EU includes the Arithmetic and Logical Unit (ALU) which may be capable of processing 8, 16, 32 or 64 bit words at an instant. It also includes circuits that execute instructions for a program control task, say, halt, interrupt, or jump to another set of instructions. It can also execute instructions for a call or branch to another program and for a call to a function.

As can be seen from the figure, it consists of the microprocessor a section of memory to store the program, another section to store temporary data some contact with the outside world (through the input/output port), a means of interconnecting these elements (i.e. data and address bus, together with some control lines).



General block diagram of a microprocessor based system.

The block diagram of a general purpose microprocessor is shown in the below figure. The computing function takes place in the Arithmetic Logic Unit (ALU), where arithmetic and logical operations take place. Part of the ALU is the accumulator. This is the register where the operand, the number on which the operation is being performed, is held. The size of the accumulator, in number of bits, determines the size of number that the processor can operate on. It is reflected across the whole microcomputer system, for example in the size of the data bus and memory locations. The ALU, together with the control section around it, is known as the Central Processing Unit (CPU).



General block diagram of a microprocessor.

The action of the microprocessor is synchronized to the clock generator, often based on a quartz crystal oscillator. Any microprocessor can only operate within a certain range of clock frequencies, whose limits are set by the fabrication technology of the device and specified by the manufacturer. Each has a maximum (for microcontrollers usually in the range 4MHz to around. 30MHz). Those based on dynamic logic have a minimum as well. Those which can operate down to DC are known as fully static. The clock oscillator frequency is divided down within the microprocessor (generally by a factor between 4 and 12, depending on the microprocessor), giving a lower internal operating frequency. One period of this internal frequency is sometimes called a machine cycle, or an instruction cycle. All instruction execution is made up of integer numbers of machine or instruction cycles. In normal system operation the processor works down the list of instructions which make up the program. It fetches each one from program memory, decodes it with its Instruction Decode circuit, and then executes it. The instruction is in many cases accompanied by further pieces of code, also stored in program memory, which are treated as operand data, or addresses where the operand data may be found. The microprocessor uses the Program Counter to always hold the address of the next instruction to be executed. In order to fetch the next instruction, the processor places the value held in the Program Counter on the address bus, and signals through the control lines that it wishes to read data. Memory corresponding to that address will, upon receiving the address and control signals, place the instruction word on the data bus, which the processor can then read. As each word is read from program memory, the Program Counter is incremented.

Microprocessor Design Options

1. Harvard architecture vs Von Neumann architecture

Harvard	Von Neumann (Princeton)
Separate buses for instruction as well as data fetching.	It shares single common bus for instruction and data fetching.
Easier to pipeline, so high performance can be achieve.	Low performance as compared to Harvard architecture due to separate data and instruction fetches.
Comparatively high cost due to more silicon complexity for additional hardware logic.	It is cheaper.
Since data memory and program memory are stored physically in different locations, no chances exist for accidental corruption of program memory.	Accidental corruption of program memory may occur if data memory and program memory are stored physically in the same chip.

2. Reduced Instruction Set Computing (RISC) vs Complex Instruction Set Computing (CISC)

RISC	CISC
Lesser number of instructions.	Greater number of instructions
Instruction pipelining and increased execution speed.s	Instruction pipelining feature does not exist.
Orthogonal instruction set (allows instructions to operate on any register and use any addressing mode)	Non-orthogonal instruction set (instructions are not allowed to operate on any register or use any addressing mode)
Operations are performed on registers only;The only memory operations	Operations are performed on registers or memory, depending onthe

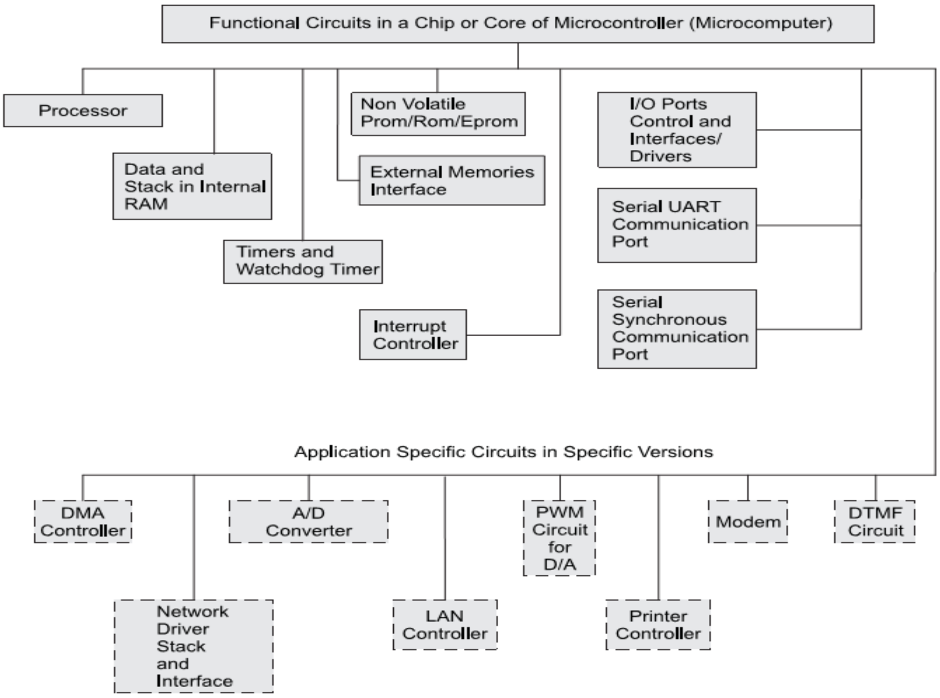
include load and store	instruction
A larger number of registers are available	The number of general purpose registers is very limited.
More instructions required for achieving the task as the instructions are very simplified.	A programmer can achieve the desired functionality with a single instruction which in turn provides the effect of using more simpler instructions in RISC
Fixed instruction length.	Variable instruction length.
Less silicon usage and pin count	More silicon: Additional decoder logic is required to implement the complex instruction decoding
With Harvard Architecture	Can be Harvard or Von-Neumann.

Microcontrollers

A microcontroller differs from a microprocessor in several important ways. The early name for a microcontroller was microcomputer. The big difference between a microprocessor and a microcomputer/microcontroller is the completeness of the machine each represents. A microprocessor is simply the heart of a computer. To put a microprocessor into use, a designer required memory, peripheral chips, and serial and parallel ports to make a completely functional computer. A microcontroller is a single chip, self-contained computer which incorporates all the basic components of a personal computer on a much smaller scale.

A highly integrated chip that contains a CPU, scratch pad RAM, special and general purpose register arrays, on chip ROM/FLASH memory for program storage, timer and interrupt control units and dedicated I/O ports.

Some embedded system applications require only 8 bit controllers whereas some requiring superior performance and computational needs demand 16/32 bit controllers. The instruction set of a microcontroller can be RISC or CISC. Microcontrollers are designed for either general purpose application requirement or domain specific application requirement.



General block diagram of a microcontroller.

The microcontroller typically includes: CPU (Central Processing Unit), RAM (Random Access Memory), EPROM/PROM/ROM (Erasable Programmable Read Only Memory), I/O (input/output) – serial and parallel, timers, interrupt controller. For example, Intel 8051 is 8-bit microcontroller and Intel8096 is 16-bit microcontroller. By only including the features specific to the task (control), cost is relatively low. A typical microcontroller has bit manipulation instructions, easy and direct access to I/O (input/output), and quick and efficient interrupt processing. It is important to note that the processor core of a microcontroller can be thought of having similar characteristics to that of a general purpose processor discussed earlier with Harvard & Von-Neumann, RICS & CISC and Pipelined & Non-pipelined design options.

All components of a microcontroller are connected via an internal bus and are all integrated on one chip. The modules are connected to the outside world via I/O pins. A microcontroller can be 8bit, 16bit, 32bit depending upon the bus width and whether the ALU performs arithmetic and logical operations on a byte (8bits), on a word (16bits) or a double word (32bits) respectively. Examples of 8-bit microcontrollers are Intel 8051 family and Motorola MC68HC11 family, 16-bit microcontrollers are Intel 8096 family and Motorola MC68HC12 and MC68332 families. Examples of 32-bit microcontrollers are Intel80960 family and Motorola M683xx and Intel/Atmel 251 family.

Moreover, a microcontroller can be *embedded* or *external*. When an embedded system has a microcontroller unit that has all the functional blocks (including program as well as data memory) available on a chip is called an embedded microcontroller. For example, 8051 having Program & Data Memory, I/O Ports, Serial Communication, Counters and Timers and Interrupt Control logic on the chip is an embedded microcontroller. On the other hand, when an embedded system has a microcontroller unit that has not all the functional blocks available on a chip is called an external memory microcontroller. In external memory microcontroller, all or part of the memory units are externally interfaced using an interfacing circuit called the glue circuit. For example, 8031 has no program memory on the chip is an external memory microcontroller.

Microprocessor vs Microcontroller

Microprocessor	Microcontroller
A silicon chip representing a central processing unit capable of performing arithmetic and logical operations	Highly integrated chip with a CPU, RAM, general purpose register array, on chip ROM/FLASH for program storage, timer, interrupt controlunits and dedicated I/O ports
Dependent unit: requires the combination of other chips like timers, program /data memory chips, interrupt controllers etc for functioning	Self-contained
General purpose in design and operations	Application-oriented or domain-specific
Requires external Peripheral Programmable Interface Chips (PPIC) like 8255 for I/O functionality	Most microcontrollers contain multiple built in I/O ports which can be operated as 8, 16, 32 bit ports or individual port pins
Limited power saving and heat dissipation options	Various power saving a heat dissipation options are available