****

**Contents:**

* Embedded Systems
* Domains od Embedded Systems
* Classifications of Embedded System
* Microcontroller Vs Microprocessor
* Embedded System model
* Processor Characteristics
* Memory Architecture (Harvard vs Von Neumann)
* Multibyte Data Representation (Endianness)
* Instruction Set Architecture (RISC vs CISC)
* Processor throughput (Pipelining)
* Automotive Industry
* Functional Areas
* Functional Requirements
* Domains and Potential roles offered

**Embedded systems**

It is an application specific computing system, as it is performing the specific task that has been assigned to it.

* An embedded system is a combination of computer hardware and software—and perhaps additional parts, either mechanical or electronic—designed to perform a dedicated function. A good example is the microwave oven. Almost every household has one, and tens of millions of them are used every day, but very few people realize that a computer processor and software are involved in the preparation of their lunch or dinner.
* The design of an embedded system to perform a dedicated function is in direct contrast to that of the personal computer. It too is comprised of computer hardware and software and mechanical components (disk drives, for example). However, a personal computer is not designed to perform a specific function. Rather, it is able to do many different things. Many people use the term general-purpose computer to make this distinction clear. As shipped, a general-purpose computer is a blank slate; the manufacturer does not know what the customer will do with it. One customer may use it for a network file server, another may use it exclusively for playing games, and a third may use it to write the next great American novel.
* Frequently, an embedded system is a component within some larger system. For example, modern cars and trucks contain many embedded systems. One embedded system controls the antilock brakes, another monitors and controls the vehicle’s emissions, and a third displays information on the dashboard. Some luxury car manufacturers have even touted the number of processors (often more than 60, including one in each headlight) in advertisements. In most cases, automotive embedded systemsare connected by a communications network.
* It is important to point out that a general-purpose computer interfaces to numerous embedded systems. For example, a typical computer has a keyboard and mouse, each of which is an embedded system. These peripherals each contain a processor and software and is designed to perform a specific function. Another example is a modem, which is designed to send and receive digital data over an analog telephone line; that’s all it does. And the specific function of other peripherals can each be summarized in a single sentence as well.
* The existence of the processor and software in an embedded system may be unnoticed by a user of the device. Such is the case for a microwave oven, MP3 player, or alarm clock. In some cases, it would even be possible to build a functionally equivalent device that does not contain the processor and software. This could be done by replacing the processor-software combination with a custom integrated circuit (IC) that performs the same functions in hardware. However, the processor and software combination typically offers more flexibility than a hardwired design. It is generally much easier, cheaper, and less power intensive to use a processor and software in an embedded system.

Todd Morton – author of Embedded Microcontrollers.

Embedded Systems are electronic systems that contain a microprocessor or a microcontroller, but we do not think of them as computers- the computer is hidden or embedded in the system.

Difficulty to choose between the hardware’s what we want to work as there is an advancement in the technology and there is dramatic decrease in cost of Hardware and software**.**

**Embedded System Domains**

The domains where the embedded systems are most widely used is

* Consumer Electronics and Durables
* Telecommunications
* Automotive Electronics
* IT Hardware and Office Automation
* Industrial Electronics
* Test and Measuring Instruments
* Industrial Automation
* Defense Electronics
* Medical Electronics
* Internet of Things

The core of the embedded system is nothing but it need to have a controller or a Processor inside it. For an Embedded Engineer or Embedded System Designer, knowledge of the Microcontroller and Microprocessor is must.

**Classification of Embedded Systems**

Embedded systems are classified into four categories based on their performance and functional requirements:

* Stand alone embedded systems
* Real time embedded systems
* Networked embedded systems
* Mobile embedded systems

Embedded Systems are classified into three types based on the performance of the microcontroller such as

1. Small scale embedded systems
2. Medium scale embedded systems
3. Sophisticated embedded systems

Stand Alone Embedded Systems

Stand alone embedded systems do not require a host system like a computer, it works by itself. It takes the input from the input ports either analog or digital and processes, calculates and converts the data and gives the resulting data through the connected device-Which either controls, drives and displays the connected devices. Examples for the stand alone embedded systems are mp3 players, digital cameras, video game consoles, microwave ovens and temperature measurement systems.

**Real Time Embedded Systems**

A real time embedded system is defined as, a system which gives a required o/p in a particular time.These types of embedded systems follow the time deadlines for completion of a task. Real time embedded systems are classified into two types such as soft and hard real time systems.

**Networked Embedded Systems**

These types of embedded systems are related to a network to access the resources. The connected network can be LAN, WAN or the internet. The connection can be any wired or wireless. This type of embedded system is the fastest growing area in embedded system applications. The embedded web server is a type of system wherein all embedded devices are connected to a web server and accessed and controlled by a web browser. Example for the LAN networked embedded system is a home security system wherein all sensors are connected and run on the protocol TCP/IP

**Mobile Embedded Systems**

Mobile embedded systems are used in portable embedded devices like cell phones, mobiles, digital cameras, mp3 players and personal digital assistants, etc. The basic limitation of these devices is the other resources and limitation of memory.

**Small Scale Embedded Systems**

These types of embedded systems are designed with a single 8 or 16-bit microcontroller, that may even be activated by a battery. For developing embedded software for small scale embedded systems, the main programming tools are an editor, assembler, cross assembler and integrated development environment (IDE).

**Medium Scale Embedded Systems**

These types of embedded systems design with a single or 16 or 32bit microcontroller, RISCs or DSPs. These types of embedded systems have both hardware and software complexities. For developing embedded software for medium scale embedded systems, the main programming tools are C, C++, JAVA, Visual C++, RTOS, debugger, source code engineering tool, simulator and IDE.

**Sophisticated Embedded Systems**

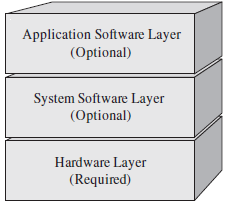
These types of embedded systems have enormous hardware and software complexities, that may need ASIPs, IPs, PLAs, scalable or configurable processors. They are used for cutting-edge applications that need hardware and software Co-design and components which have to assemble in the final system.

|  |  |
| --- | --- |
| **MICROPROCESSOR** | **MICROCONTROLLER** |
| The functional blocks are ALU, registers, timing & control units | It includes functional blocks of microprocessors & in addition has timer, parallel i/o, RAM,EPROM, ADC&DAC *(on chip peripherals);* |
| Higher capability and Higher MIPS | Lower capability and lower MIPS |
| For complex systems, Wide memory range. | They are used for designing application specific dedicated systems, and lesser memory |

**Embedded System Model**

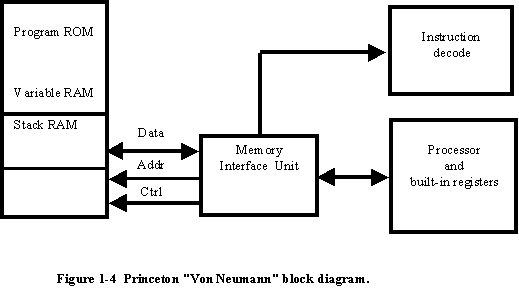
What the embedded systems architecture model shows is that all embedded systems proportion one similarity at the very best degree; that is, all of them have a minimum one layer (hardware) or all layers (hardware, device software and alertness software) into which all additives fall. The hardware layer consists of all the important physical components placed on an embedded board, whereas the system and alertness software layers incorporate all of the software program placed on and being processed by the embedded gadget. This reference model is basically a layered (modular) illustration of an embedded systems structure from which a modular architectural structure can be derived. Whilst the concept of layering isn’t unique to embedded gadget design (architectures are relevant to all PC structures, and an embedded system is a form of PC system), it is a useful tool in visualizing the possible mixtures of masses, if not hundreds, of hardware and software additives that can be utilized in designing an embedded device.

System software and application software are computer programs. The system software is also installed during the installation of the operating system. However, the application software utilizes the capabilities of the computer on which it is installed.



### HarvardVSPrinceton(Von-Neumann)

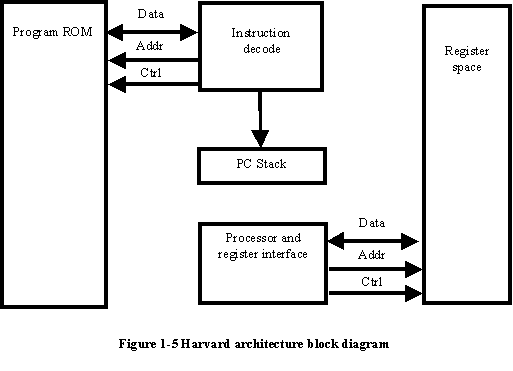
Many years ago, the United States government asked Harvard and Princeton Universities to come up with a computer architecture to be used in computing tables of Naval artillery shell distances for varying elevations and environmental conditions. Princeton's response was a computer that had common memory for storing the control program as well as variables and other data structures. It was best known by the chief scientist's name "Von Neumann".



The *memory interface unit* is responsible for arbitrating access to the memory space between reading instructions (based upon the current program counter) and passing data back and forth with the processor and its internal registers.

It might at first seem that the memory interface unit is a bottleneck between the processor and the variable/RAM space (especially with the requirement for fetching instructions at the same time); however, in many Princeton architected processors, this is not the case because the time required to execute a given instruction can be used to fetch the next instruction (this is known as *pre-fetching* and is a feature on many Princeton architected processors.

In contrast, Harvard's response was a design that used separate memory banks for program store, the processor stack, and variable RAM.



The Princeton architecture won the competition because it was better suited for the technology of the time. Using one memory was preferable because of the unreliability of current electronics (this was before transistors were in widespread use). A single memory interface would have fewer things that could go wrong.

The Harvard architecture was largely ignored until the late 1970s when microcontroller manufacturers realized that the architecture had advantages for the devices they were currently designing.

What are the advantages of the two architectures?

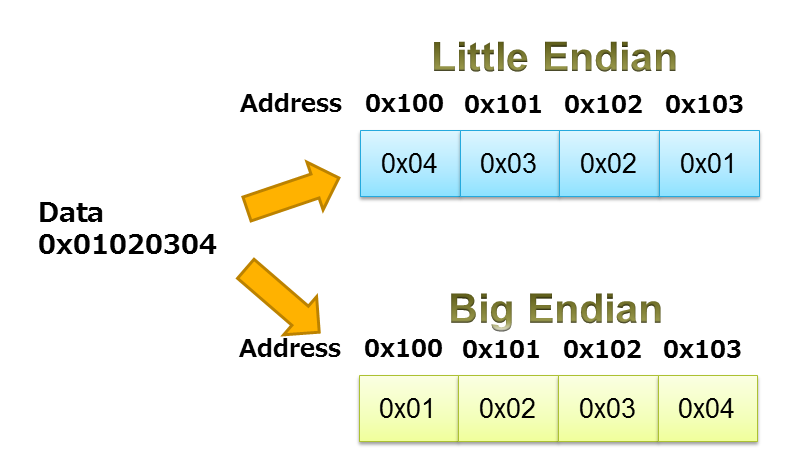
The Von Neumann architecture's largest advantage is that it simplifies the microcontroller chip design because only one memory is accessed. For microcontrollers, its biggest asset is that the contents of RAM (random-access memory) can be used for both variable (data) storage as well as program instruction storage. An advantage for some applications is the program counter stack contents that are available for access by the program. This allows greater flexibility in developing software, primarily in the areas of real-time operating systems.

The Harvard architecture executes instructions in fewer instruction cycles that the Von Neumann architecture. This is because a much greater amount of instruction *parallelism* is possible in the Harvard architecture. Parallelism means that fetches for the next instruction can take place during the execution of the current instruction, without having to either wait for a "dead" cycle of the instruction's execution or stop the processor's operation while the next instruction is being fetched.

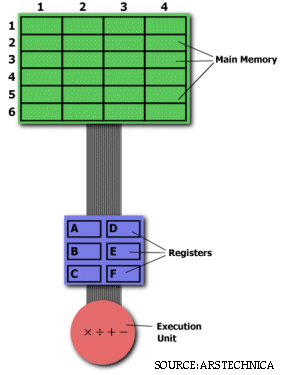
Endianness can be a major migration problem, causing migrated software to not work properly. This article shows techniques which can be useful when migrating from a big-endian to a little-endian CPU.

**Endianness**

Endianness and endian are terms that describe the order in which a sequence of bytes is stored in memory. Big-endian is an order in which the most significant value in the sequence is stored first. In little-endian systems the least significant value in the sequence is stored first. For example, in a big-endian CPU, the four bytes of data 0x01020304 would be stored 0x01(address+0), 0x02(address+1), 0x03(address+2), 0x04(address+3). In a little-endian CPU, the same bytes would be stored 0x04(address+0), 0x03(address+1), 0x02(address+2), 0x01(address+3).



If your program uses a simple data structure such as "int" and "short", there is little trouble. But if the data structure is similar to the following example, you might run into problems. In this case, the union variable can store "unsigned int" and "unsigned char [4]" which both have 4 bytes of memory space. This kind of union is commonly used in application programs. When input and output data is transferred, 4 bytes access is used. When the data is accessed in the program, one byte access is used, and sometimes also bitfield access.

**RISC vs CISC**

The simplest way to examine the advantages and disadvantages of RISC architecture is by contrasting it with it's predecessor: CISC (Complex Instruction Set Computers) architecture.

**Multiplying Two Numbers in Memory**  
On the right is a diagram representing the storage scheme for a generic computer. The main memory is divided into locations numbered from (row) 1: (column) 1 to (row) 6: (column) 4. The execution unit is responsible for carrying out all computations. However, the execution unit can only operate on data that has been loaded into one of the six registers (A, B, C, D, E, or F). 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.

**The CISC Approach**  
The primary goal of CISC architecture 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. For this particular task, a CISC processor would come prepared with a specific instruction (we'll call it "MULT"). When executed, this instruction loads the two values into separate registers, multiplies the operands in the execution unit, and then stores the product in the appropriate register. Thus, the entire task of multiplying two numbers can be completed with one instruction:

MULT 2:3, 5:2

MULT is what is known as a "complex instruction." It operates directly on the computer's memory banks and does not require the programmer to explicitly call any loading or storing functions. It closely resembles a command in a higher level language. For instance, if we let "a" represent the value of 2:3 and "b" represent the value of 5:2, then this command is identical to the C statement "a = a \* b."

One of the primary advantages 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. The emphasis is put on building complex instructions directly into the hardware.

**The RISC Approach**   
RISC processors only use simple instructions that can be executed within one clock cycle. Thus, the "MULT" command described above 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, and "STORE," which moves data from a register to the memory banks. In order to perform the exact series of steps described in the CISC approach, a programmer would need to code four lines of assembly:

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

At first, this may seem like a much less efficient way of completing the operation. Because there are more lines of code, more RAM is needed to store the assembly level instructions. The compiler must also perform more work to convert a high-level language statement into code of this form.

|  |  |
| --- | --- |
| **CISC** | **RISC** |
| Emphasis on hardware | Emphasis on software |
| Includes multi-clock | Single-clock, |
| complex instructions | reduced instruction only |
| Memory-to-memory: | Register to register: |
| "LOAD" and "STORE" | "LOAD" and "STORE" |
| incorporated in instructions | are independent instructions |
| Small code sizes, | Low cycles per second, |
| high cycles per second | large code sizes |
| Transistors used for storing | Spends more transistors |
| complex instructions | on memory registers |

However, the RISC strategy also brings some very important advantages. Because each instruction requires only one clock cycle to execute, the entire program will execute in approximately the same amount of time as the multi-cycle "MULT" command. These RISC "reduced instructions" require less transistors of hardware space than the complex instructions, 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.

Separating the "LOAD" and "STORE" instructions actually reduces the amount of work that the computer must perform. After a CISC-style "MULT" command is executed, the processor automatically erases the registers. If one of the operands needs to be used for another computation, the processor must re-load the data from the memory bank into a register. In RISC, the operand will remain in the register until another value is loaded in its place.

**The Performance Equation**  
The following equation is commonly used for expressing a computer's performance ability:

https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/risccisc/options/performanceeq.gif

The CISC approach attempts to minimize the number of instructions per program, sacrificing the number of cycles per instruction. RISC does the opposite, reducing the cycles per instruction at the cost of the number of instructions per program.

**RISC Roadblocks**  
Despite the advantages of RISC based processing, RISC chips took over a decade to gain a foothold in the commercial world. This was largely due to a lack of software support.

Although Apple's Power Macintosh line featured RISC-based chips and Windows NT was RISC compatible, Windows 3.1 and Windows 95 were designed with CISC processors in mind. Many companies were unwilling to take a chance with the emerging RISC technology. Without commercial interest, processor developers were unable to manufacture RISC chips in large enough volumes to make their price competitive.

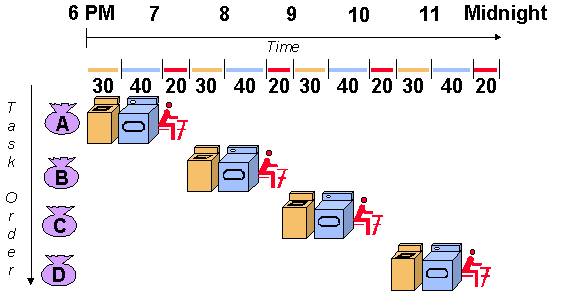
Another major setback was the presence of Intel. Although their CISC chips were becoming increasingly unwieldy and difficult to develop, Intel had the resources to plow through development and produce powerful processors. Although RISC chips might surpass Intel's efforts in specific areas, the differences were not great enough to persuade buyers to change technologies.

**The Overall RISC Advantage**  
Today, the Intel x86 is arguable the only chip which retains CISC architecture. This is primarily due to advancements in other areas of computer technology. The price of RAM has decreased dramatically. In 1977, 1MB of DRAM cost about $5,000. By 1994, the same amount of memory cost only $6 (when adjusted for inflation). Compiler technology has also become more sophisticated, so that the RISC use of RAM and emphasis on software has become ideal.

**How Pipelining Works**

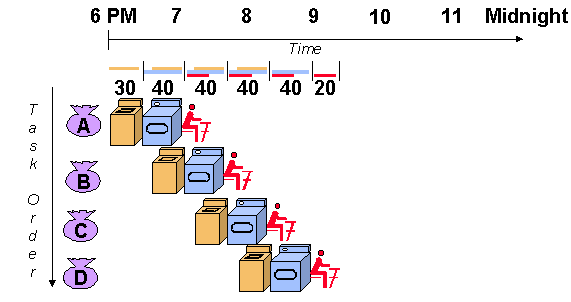
Pipelining, a standard feature in RISC processors, is much like an assembly line. Because the processor works on different steps of the instruction at the same time, more instructions can be executed in a shorter period of time.

A useful method of demonstrating this is the laundry analogy. Let's say that there are four loads of dirty laundry that need to be washed, dried, and folded. We could put the the first load in the washer for 30 minutes, dry it for 40 minutes, and then take 20 minutes to fold the clothes. Then pick up the second load and wash, dry, and fold, and repeat for the third and fourth loads. Supposing we started at 6 PM and worked as efficiently as possible, we would still be doing laundry until midnight.



Source: <http://www.ece.arizona.edu/~ece462/Lec03-pipe/>

However, a smarter approach to the problem would be to put the second load of dirty laundry into the washer after the first was already clean and whirling happily in the dryer. Then, while the first load was being folded, the second load would dry, and a third load could be added to the pipeline of laundry. Using this method, the laundry would be finished by 9:30.



Source <http://www.ece.arizona.edu/~ece462/Lec03-pipe/>

**RISC Pipelines**  
A RISC processor pipeline operates in much the same way, although the stages in the pipeline are different. While different processors have different numbers of steps, they are basically variations of these five, used in the MIPS R3000 processor:

1. fetch instructions from memory
2. read registers and decode the instruction
3. execute the instruction or calculate an address
4. access an operand in data memory
5. write the result into a register

If you glance back at the diagram of the laundry pipeline, you'll notice that although the washer finishes in half an hour, the dryer takes an extra ten minutes, and thus the wet clothes must wait ten minutes for the dryer to free up. Thus, the length of the pipeline is dependent on the length of the longest step. Because RISC instructions are simpler than those used in pre-RISC processors (now called CISC, or Complex Instruction Set Computer), they are more conducive to pipelining. While CISC instructions varied in length, RISC instructions are all the same length and can be fetched in a single operation. Ideally, each of the stages in a RISC processor pipeline should take 1 clock cycle so that the processor finishes an instruction each clock cycle and averages one cycle per instruction (CPI).

**Pipeline Problems**  
In practice, however, RISC processors operate at more than one cycle per instruction. The processor might occasionally stall aa result of data dependencies and branch instructions.

A data dependency occurs when an instruction depends on the results of a previous instruction. A particular instruction might need data in a register which has not yet been stored since that is the job of a preceeding instruction which has not yet reached that step in the pipeline.

For example:

add $r3, $r2, $r1  
add $r5, $r4, $r3  
*more instructions that are independent of the first two*

In this example, the first instruction tells the processor to add the contents of registers r1 and r2 and store the result in register r3. The second instructs it to add r3 and r4 and store the sum in r5. We place this set of instructions in a pipeline. When the second instruction is in the second stage, the processor will be attempting to read r3 and r4 from the registers. Remember, though, that the first instruction is just one step ahead of the second, so the contents of r1 and r2 are being added, but the result has not yet been written into register r3. The second instruction therefore cannot read from the register r3 because it hasn't been written yet and must wait until the data it needs is stored. Consequently, the pipeline is stalled and a number of empty instructions (known as *bubbles* go into the pipeline. Data dependency affects long pipelines more than shorter ones since it takes a longer period of time for an instruction to reach the final register-writing stage of a long pipeline.

MIPS' solution to this problem is code reordering. If, as in the example above, the following instructions have nothing to do with the first two, the code could be rearranged so that those instructions are executed in between the two dependent instructions and the pipeline could flow efficiently. The task of code reordering is generally left to the compiler, which recognizes data dependencies and attempts to minimize performance stalls.

Branch instructions are those that tell the processor to make a decision about what the next instruction to be executed should be based on the results of another instruction. Branch instructions can be troublesome in a pipeline if a branch is conditional on the results of an instruction which has not yet finished its path through the pipeline.

**For example:**

|  |  |
| --- | --- |
| Loop : | add $r3, $r2, $r1 sub $r6, $r5, $r4 beq $r3, $r6, Loop |

The example above instructs the processor to add r1 and r2 and put the result in r3, then subtract r4 from r5, storing the difference in r6. In the third instruction, beq stands for branch if equal. If the contents of r3 and r6 are equal, the processor should execute the instruction labeled "Loop." Otherwise, it should continue to the next instruction. In this example, the processor cannot make a decision about which branch to take because neither the value of r3 or r6 have been written into the registers yet.

The processor could stall, but a more sophisticated method of dealing with branch instructions is branch prediction. The processor makes a guess about which path to take - if the guess is wrong, anything written into the registers must be cleared, and the pipeline must be started again with the correct instruction. Some methods of branch prediction depend on stereotypical behavior. Branches pointing backward are taken about 90% of the time since backward-pointing branches are often found at the bottom of loops. On the other hand, branches pointing forward, are only taken approximately 50% of the time. Thus, it would be logical for processors to always follow the branch when it points backward, but not when it points forward. Other methods of branch prediction are less static: processors that use dynamic prediction keep a history for each branch and uses it to predict future branches. These processors are correct in their predictions 90% of the time.

Still other processors forgo the entire branch prediction ordeal. The RISC System/6000 fetches and starts decoding instructions from both sides of the branch. When it determines which branch should be followed, it then sends the correct instructions down the pipeline to be executed.

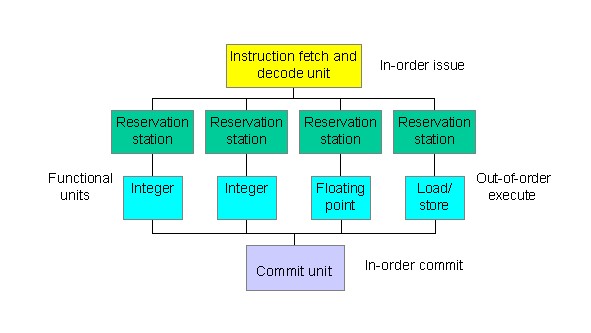
**Pipelining Developments**  
In order to make processors even faster, various methods of optimizing pipelines have been devised.

Superpipelining refers to dividing the pipeline into more steps. The more pipe stages there are, the faster the pipeline is because each stage is then shorter. Ideally, a pipeline with five stages should be five times faster than a non-pipelined processor (or rather, a pipeline with one stage). The instructions are executed at the speed at which each stage is completed, and each stage takes one fifth of the amount of time that the non-pipelined instruction takes. Thus, a processor with an 8-step pipeline (the MIPS R4000) will be even faster than its 5-step counterpart. The MIPS R4000 chops its pipeline into more pieces by dividing some steps into two. Instruction fetching, for example, is now done in two stages rather than one. The stages are as shown:

1. Instruction Fetch (First Half)
2. Instruction Fetch (Second Half)
3. Register Fetch
4. Instruction Execute
5. Data Cache Access (First Half)
6. Data Cache Access (Second Half)
7. Tag Check
8. Write Back

Superscalar pipelining involves multiple pipelines in parallel. Internal components of the processor are replicated so it can launch multiple instructions in some or all of its pipeline stages. The RISC System/6000 has a forked pipeline with different paths for floating-point and integer instructions. If there is a mixture of both types in a program, the processor can keep both forks running simultaneously. Both types of instructions share two initial stages (Instruction Fetch and Instruction Dispatch) before they fork. Often, however, superscalar pipelining refers to multiple copies of all pipeline stages (In terms of laundry, this would mean four washers, four dryers, and four people who fold clothes). Many of today's machines attempt to find two to six instructions that it can execute in every pipeline stage. If some of the instructions are dependent, however, only the first instruction or instructions are issued.

Dynamic pipelines have the capability to schedule around stalls. A dynamic pipeline is divided into three units: the instruction fetch and decode unit, five to ten execute or functional units, and a commit unit. Each execute unit has reservation stations, which act as buffers and hold the operands and operations.



While the functional units have the freedom to execute out of order, the instruction fetch/decode and commit units must operate in-order to maintain simple pipeline behaviour. When the instruction is executed and the result is calculated, the commit unit decides when it is safe to store the result. If a stall occurs, the processor can schedule other instructions to be executed until the stall is resolved. This, coupled with the efficiency of multiple units executing instructions simultaneously, makes a dynamic pipeline an attractive alternative.

The Automotive Industry consists of a broad range of organizations and companies with a critical objective of designing, developing, marketing, manufacturing, and selling of motor vehicles. The automotive industry makes a vital part of the world's economic sectors by revenue Automobiles, however, are not entirely included in the industry. The industry also does not include companies or organizations dedicated to the maintenance of automobiles such as fuel filling stations and automobile service and repair shops.

The industry consists of producers, however not limited to original equipment manufacturers (OEMs). The original equipment manufacturers consist of light trucks, cars, heavy equipment, heavy trucks, and motorbike manufacturers. Another category in the original equipment manufacturers includes; wholesalers and automotive suppliers, distributors, dealers, and importers.

Companies in the automotive industry fall into two categories that are car manufacturers and car parts manufacturers. Vehicles in the modern world are becoming more complex and involve more electronic parts than in the past years. It, therefore, increases the number of components manufactured by suppliers rather than the manufacturers.

The modern automotive industry is in a continual state of flux. The success of any automobile industry relies on the salesroom as well as the expertise of many different professionals. The sector offers numerous employment opportunities in several positions such as mechanical, sales, assembly, financial, creative, scientific, technical, and business position. Other employment opportunities in the manufacturing plants of the automotive industry include quality control workers, safety engineers, managers, supervisors, designers, and executives.

Since car manufacturing is expensive, there are a few numbers of manufacturers in the automotive industry. Globally the world leaders in the automotive sector include Toyota, Honda, Volkswagen, Nissan Motors, and Hyundai. The automotive industry is also an example of a mixed oligopoly. It is because it has only a few producers who produce differentiated products.

The industry is entering a period of intense change, and the automotive industry would change to mobility industry. The trends in the industry are as a result of the combination of business models, digital sciences, and new technologies in the material.

The transformation of the industry is more about people's connectivity to automobiles. The change in the automotive sector will entail connectivity, autonomous vehicles, redefined mobility, and electrification. The integration of mobile information to the industry would also set a pathway for predictive maintenance that monitors and warns consumers about the operational performance of the automobile.

**What are the Key Segments in Automotive Industry?**

**Light vehicles**  
  
These are motorcycles and cars used for business transportation involving goods and people. Examples include SUVs, passenger cars, minivans, and pick-up trucks.

**Trucks and buses**

They are mainly commercial vehicles used for transportation of goods and people. Examples include coaches, trailers, and light and heavy buses.

**Construction and agriculture.**

Industrial machinery used for mining, agricultural and construction transportation tasks. Examples include earth movers and commercial vehicles.

**Electric cars and Plugin Hybrids**

Include automobiles propelled by a single or several electric motors. They use electrical energy stored in rechargeable batteries. The electric motors provide electric cars with instant torque, thus creating a stable and smooth acceleration.

**Autonomous Vehicle**

These are self-driving, driverless or robotic cars that can sense their environment and navigate without human input. They use global positioning system, radar, and computer vision to detect their surroundings.

**Automotive Industry Value Chain?**

The value chain covers all the activities starting from conception of the product to the final delivery in the hands of a customer. In cases where the company involved deals with the production of goods, their value chain commences from the acquisition of raw materials. The processes in the value chain add value to the product until its ready for sale. Analysis of the value chain helps in analyzing and identifying where problems are to make improvements and increase the efficiency of the operations. A compelling value chain analysis would also assist in generating competitive advantage. The value chain of the automotive industry starts from:

**Inbound logistics**

It is the initial step in the line of production. The action involves receiving of raw materials from suppliers who are in different locations all over the world. After acquiring the raw materials, they industry distributes them among the manufacturing units based on the requirement.

**Design and Manufacture:**

It is the initial step in the line of production. The action involves receiving of raw materials from suppliers who are in different locations all over the world. After acquiring the raw materials, they industry distributes them among the manufacturing units based on the requirement.

**Engineering**

In this step, it primarily deals with the flexibility of the engineers to manufacture vehicles. It also involves minimizing of engineering times by researching and developing solutions. Engineers also have to ensure the production line is more flexible by planning and designing new production processes.

**Quality and warranty**:

It is an important part that touches all parts of the automotive industry. The agreement among suppliers and industry practitioners in the automotive sector on what should entail any quality and warranty program are; managing of corrective actions to improve the quality of products, tracking and managing aspects of warranty operations, and analysing contributing factors and failures to warranty cost. With this, OEMs and suppliers will be in good terms.

**Connected vehicles services**

These involve services that enable a car to share internet access with other devices. The facilities also consist of specialized technology that taps into the internet to offer additional benefits to the driver.

**Marketing and sales**

It is a critical part of the value chain to vehicle producing companies. This part includes distribution, management of sales force, advertising, promotions, and management of customer relationship. The objective of this step is ensuring that the product reaches to the targeted consumer segment as well as making the target market aware of the products' advantages and features. The focus of marketing and sales is driving up profitability and sales for the automobile company. These automobile companies use online and traditional channels to advertise and market their brands to remain competitive in the industry.

**Service.**

It is the final activity in the automotive industry value chain and which adds value to the product. It comprises of customer support after the sale of the product by providing continued support relevant to the maintenance of their vehicles. These support activities also ensure that attaining customer retention. By providing better customer support, the brand would have a better image and have a high number of retained customers.

**Key Terms in Automotive Industry.**

**Advanced driver assistance systems (ADAS):**

Conventionally, ADAS technology enables the car to detect objects, perform necessary clarification, and alert the driver of any hazardous road conditions. In other cases, the ADAS technology can slow down or stop the vehicle. It also brings applications such as blind spot monitoring, forward collision warnings, and lane change assistance.

**Autonomous vehicles:**

They are self-driving vehicles. The technologies used in these vehicles offer significant benefits that reduce crashes, reduce fuel consumption, congestion and increase mobility.

**Internet of Things (IoT):**

Technologies in the IoT perform various tasks such as linking to smartphones, registering real-time alerts, offer emergency roadside assistance among others. Vehicles also will have the ability to send and receive data.

**Telematics:**

These are methods of monitoring a vehicle through the combination of a GPS system with on-board diagnostics. Through this combination, one can map and record exactly where a car is and how fast it is traveling as well as how the car is behaving internally.

**Electric vehicle (EV):**

It is a vehicle operating with a single or several electric motors using energy stored in rechargeable batteries. There are three types of EVs based on the degree of electricity they use. They are;

**Hybrid-electric vehicles (HEVs):**

They generate their power from both electricity and petrol. The car's braking system makes its electric energy to recharge the battery.

**Plug-in hybrid-electric vehicles (PHEVs):**

They generate their power from both petrol and electricity. They use plugging-in to an external charging outlet and regenerative braking to recharge their battery.

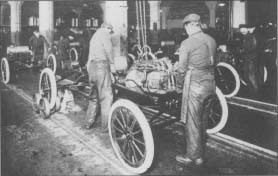
**Battery electric vehicle:**

They are fully electric vehicles and use an external electrical charging outlet to charge the battery.

**Functional Areas of Automotive**

**Chassis**

The typical car or truck is constructed from the ground up (and out). The frame forms the base on which the body rests and from which all subsequent assembly components follow. The frame is placed on the assembly line and clamped to the conveyer to prevent shifting as it moves down the line. From here the automobile frame moves to component assembly areas where complete front and rear suspensions, gas tanks, rear axles and drive shafts, gear boxes, steering box components, wheel drums, and braking systems are sequentially installed.

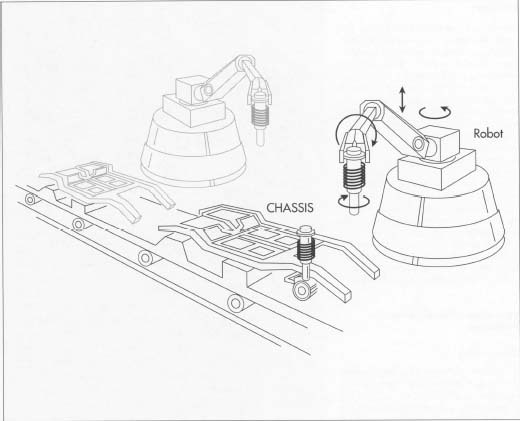


*Workers install engines on Model Ts at a Ford Motor Company plant. The photo is from about 1917.*

The automobile, for decades the quintessential American industrial product, did not have its origins in the United States. In 1860, Etienne Lenoir, a Belgian mechanic, introduced an internal combustion engine that proved useful as a source of stationary power. In 1878, Nicholas Otto, a German manufacturer, developed his four-stroke "explosion" engine. By 1885, one of his engineers, Gottlieb Daimler, was building the first of four experimental vehicles powered by a modified Otto internal combustion engine. Also in 1885, another German manufacturer, Carl Benz, introduced a three-wheeled, self-propelled vehicle. In 1887, the Benz became the first automobile offered for sale to the public. By 1895, automotive technology was dominated by the French, led by Emile Lavassor. Lavassor developed the basic mechanical arrangement of the car, placing the engine in the front of the chassis, with the crankshaft perpendicular to the axles.

In 1896, the Duryea Motor Wagon became the first production motor vehicle in the United States. In that same year, Henry Ford demonstrated his first experimental vehicle, the Quadricycle. By 1908, when the Ford Motor Company introduced the Model T, the United States had dozens of automobile manufacturers. The Model T quickly became the standard by which other cars were measured; ten years later, half of all cars on the road were Model Ts. It had a simple four-cylinder, twenty-horsepower engine and a planetary transmission giving two gears forward and one backward. It was sturdy, had high road clearance to negotiate the rutted roads of the day, and was easy to operate and maintain.

*William S. Pretzer*

An off-line operation at this stage of production mates the vehicle's engine with its transmission. Workers use robotic arms to install these heavy components inside the engine compartment of the frame. After the engine and transmission

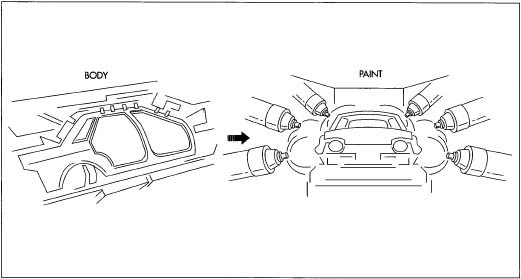
On automobile assembly lines, much of the work is now done by robots rather than humans. In the first stages of automobile manufacture, robots weld the floor pan pieces together and assist workers in placing components such as the suspension onto the chassis.

worker attaches the radiator, and another bolts it into place. Because of the nature of these heavy component parts, articulating robots perform all of the lift and carry operations while assemblers using pneumatic wrenches bolt component pieces in place. Careful ergonomic studies of every assembly task have provided assembly workers with the safest and most efficient tools available.

### Body Design.

Generally, the floor plan is the largest body component to which a multitude of panels and braces will subsequently be either welded or bolted. As it moves down the assembly line, held in place by clamping fixtures, the shell of the vehicle is built. First, the left and right quarter panels are robotically disengaged from pre-staged shipping containers and placed onto the floor pan, where they are stabilized with positioning fixtures and welded.

The front and rear door pillars, roof, and body side panels are assembled in the same fashion. The shell of the automobile assembled in this section of the process lends itself to the use of robots because articulating arms can easily introduce various component braces and panels to the floor pan and perform a high number of weld operations in a time frame and with a degree of accuracy no human workers could ever approach. Robots can pick and load 200-pound (90.8 kilograms) roof panels and place them precisely in the proper weld position with tolerance variations held to within .001 of an inch. Moreover, robots can also tolerate the



The body is built up on a separate assembly line from the chassis. Robots once again perform most of the welding on the various panels, but human workers are necessary to bolt the parts together. During welding, component pieces are held securely in a jig while welding operations are performed. Once the body shell is complete, it is attached to an overhead conveyor for the painting process. The multi-step painting process entails inspection, cleaning, undercoat (electrostatically applied) dipping, drying, topcoat spraying, and baking. smoke, weld flashes, and gases created during this phase of production.

As the body moves from the isolated weld area of the assembly line, subsequent body components including fully assembled doors, deck lids, hood panel, fenders, trunk lid, and bumper reinforcements are installed. Although robots help workers place these components onto the body shell, the workers provide the proper fit for most of the bolt-on functional parts using pneumatically assisted tools.

**Drive-train**

The drivetrain of a motor vehicle is the group of components that deliver power to the wheels. This excludes the engine or motor that generates the power. In contrast, the [powertrain](https://en.wikipedia.org/wiki/Powertrain) is considered to include both the engine or motor and the drive train.

**Function of Drive-train**

The function of the drivetrain is to couple the engine that produces the power to the driving wheels that use this mechanical power to rotate the axle. This connection involves physically linking the two components, which may be at opposite ends of the vehicle and so requiring a long [propeller shaft](https://en.wikipedia.org/wiki/Propeller_shaft) or [drive shaft](https://en.wikipedia.org/wiki/Drive_shaft). The operating speed of the engine and wheels are also different and must be matched by the correct [gear ratio](https://en.wikipedia.org/wiki/Gear_ratio). As the vehicle speed changes, the ideal engine speed must remain approximately constant for efficient operation and so this gearbox ratio must also be changed, either manually, automatically or by an automatic [continuous variation](https://en.wikipedia.org/wiki/Continuously_variable_transmission).

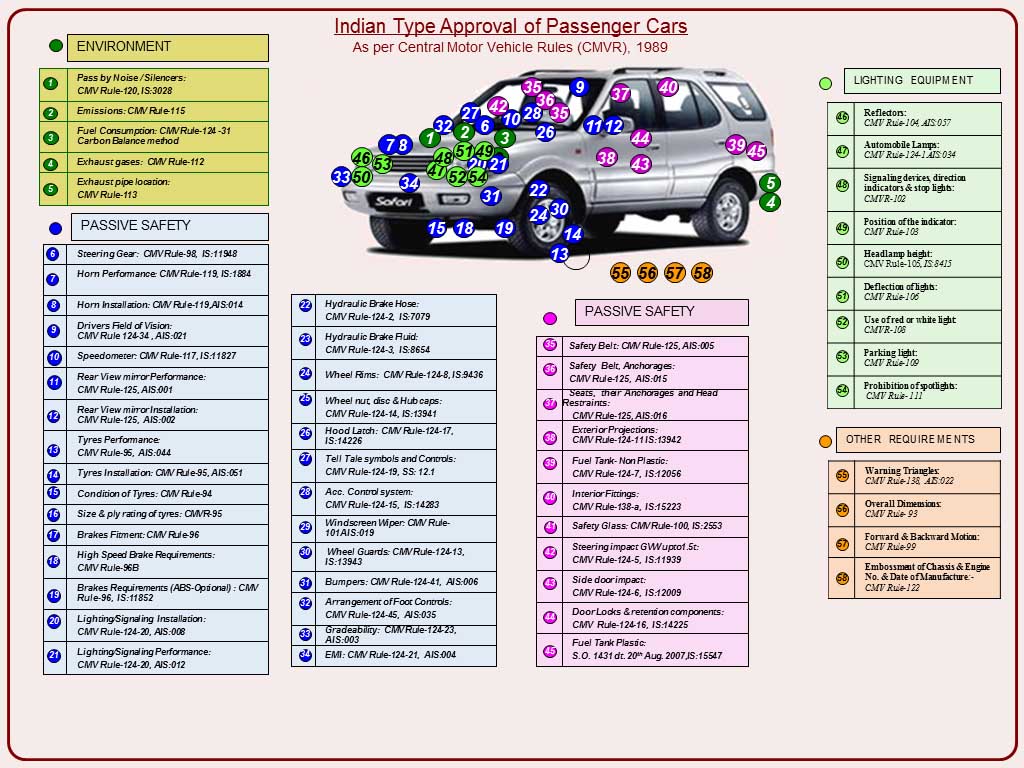
**Telematics**

Telematics is an interdisciplinary field that encompasses [telecommunications](https://en.wikipedia.org/wiki/Telecommunication), vehicular technologies, for instance, [road transportation](https://en.wikipedia.org/wiki/Road_transport), [road safety](https://en.wikipedia.org/wiki/Road_safety), electrical engineering (sensors, instrumentation, [wireless communications](https://en.wikipedia.org/wiki/Wireless_communication), etc.), and [computer science](https://en.wikipedia.org/wiki/Computer_science) (multimedia, [Internet](https://en.wikipedia.org/wiki/Internet), etc.). Telematics can involve any of the following:

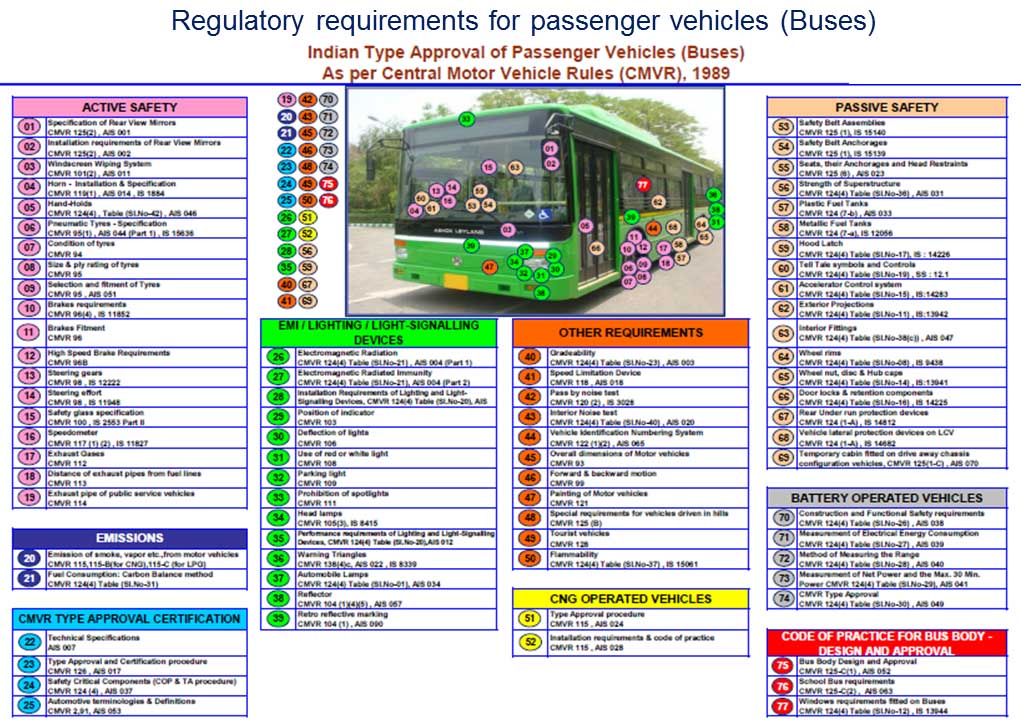
* the technology of sending, receiving and storing information using telecommunication devices to control remote objects
* the integrated use of [telecommunications](https://en.wikipedia.org/wiki/Telecommunication) and [informatics](https://en.wikipedia.org/wiki/Informatics) for application in vehicles and to control vehicles on the move
* [global navigation satellite system](https://en.wikipedia.org/wiki/Satellite_navigation) technology integrated with computers and [mobile communications](https://en.wikipedia.org/wiki/Mobile_communications) technology in [automotive navigation systems](https://en.wikipedia.org/wiki/Automotive_navigation_system)
* (most narrowly) the use of such systems within [road vehicles](https://en.wikipedia.org/wiki/Road_vehicle), also called vehicle telematics

**Functional Requirements**

#### **Vehicular Safety Standards & Regulations**

  
Environmental imperatives and safety requirements are two critical issues facing the automotive industry worldwide. Indian Automobile Industry in the last decade has made significant progress on the environmental front by adopting stringent emission standards, and is progressing towards technical alignment with international safety standards.

Vehicles manufactured in the country have to comply with relevant Indian Standards (IS) and Automotive Industry standards (AIS). Indian Standards are being issued since the late 1960s and these standards for Automotive Components were based on EEC/ISO/DIN/BSAU/FMVSS, etc. at that time.



India signed the UN WP 29 1998 Agreement in Feburary 2006. It continues to actively participate in the Global Technicla Regulation (GTR) formulation by contributing data and subject matter expertise. SIAM members chair the different expert groups formed to formulate India stance on the various safety regulations. This has helped in developement of GTRs taking into consideration the traffic and driving conditions in the developing countries.

India has curently more than 70% safety regulations which are either partially or fully technically aligned with GTRs and UN Regulations while retaing Indian specific driving and environmental conditions.

Regulations are reviewed periodically by AISC and amendments are recommended to the Technical standing Committee on CMVR for adoption and subsequent notification by MoRT&H under the CMVR.

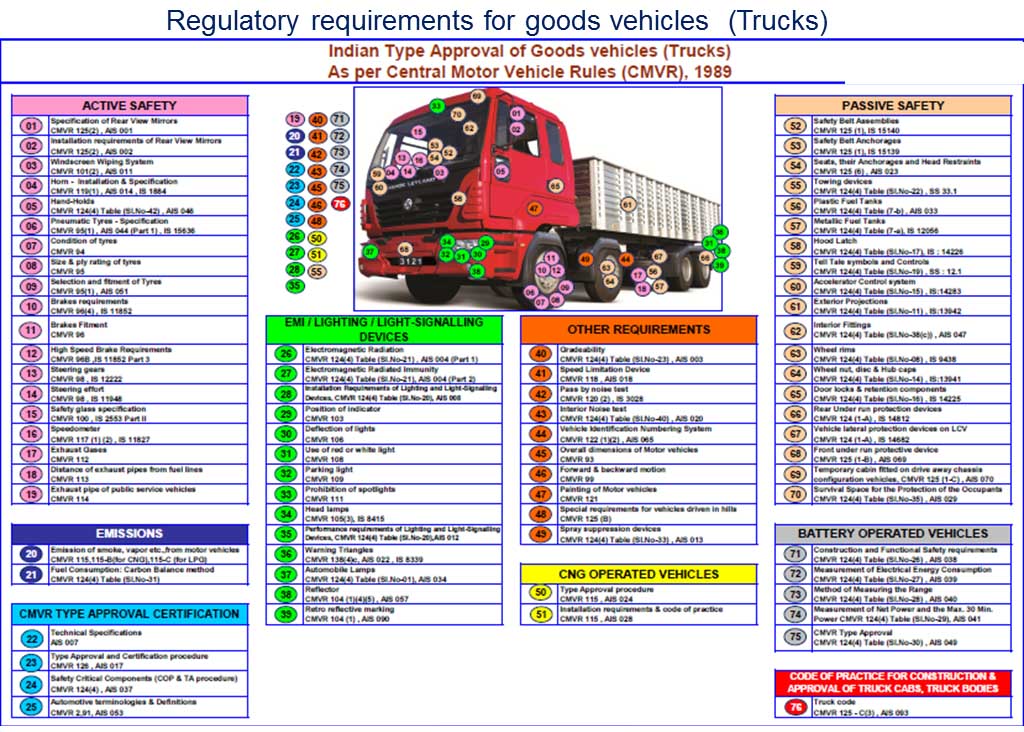
States also have their State Motor Vehicle Rules. These rules are mostly related to seating arrangment for transport vehicles, etc.

Since 2000, ECE Regulations have been referred to as basis for formulating Indian regulations and since 2003, increased efforts are being made to technically align with GTR / ECE. Variance from GTR / ECE exists on formatting, phraseology and administration related issues.

Alignment of Indian regulations (AIS / BIS) with GTRs / ECE is being attempted as per the broad roadmap drafted by SIAM.

In order to have a planned approach for the introduction of advanced safety features, SIAM drew a roadmap for Automobile Safety Standards. The roadmap was prepared by the CMVR, Safety & Regulations Committee.

The current traffic conditions, driving habits, traffic density and road-user behaviour necessitate that maximum safety be built into the vehicles. Progressive tightening of safety standards taking into account unique India requirements has been addressed in the roadmap with a view to reduce the impact of accidents, thereby improving safety of the vehicle occupants and vulnerable road users.



The roadmap was presented to the Government in January 2002 which received an in-principle approval of the Ministry of Road Transport & Highways. Based upon discussions with all stakeholders, a roadmap has been finalized by the Ministry and work has commenced on drafting standards and notifications for the various stages. The roadmap is reviewed from time to time to align with the changing environment and new regulations being formulated in UN WP 29.

Safety critical components are covered under AIS-037 to bring about better control at the OEM and after market. The regulation also brings these components under Conformity of Production.

## **List of Components Covered Under AIS-037**

* Tyres
* Rear View Mirrors
* Speed Limiting Devices
* Safety Belts
* Warning Triangle
* Lighting & Light Signalling Devices
* Retro Reflectors
* Bulbs
* Safety Glass
* Brake Hose
* Wheel Rims
* Horns
* CNG / LPG Regulators, Vaporizers
* CNG / LPG Kit Components

**Crash Requirements will be introduced in the country mandatorily 2017 onwards after the test facilities are commissioned in 2016.**

**Hard real-time constraints**: it is imperative that all safety related systems comply to real-time constraints. Systems like ABS should be able to react within mere milliseconds to ensure a timely intervention. This not only puts a constraint on the processors-speed, but also on the speed of the communication system used throughout the car, because these systems often rely on input signals from other systems and need to send outputs to different systems as well (e.g. [ESC](https://en.wikipedia.org/wiki/Electronic_stability_control)-system must communicate with engine management).

**Fault-detection:** every sub-system of a safety-critical function should be capable of auto-diagnosing of its function and should be able to switch to a safe shutdown condition in case of failure. Obviously also the communication system should be able to perform data-transfer checks. Legislation dictates that if a fault is detected, this should also be shown to the driver ([MIL Malfunction Indicator Lamp](https://en.wikipedia.org/wiki/Malfunction_Indicator_Lamp)) and be stored in the fault-memory of the controller.

Redundancy: critical sensors, like the throttle in drive-by-wire systems, should have redundancy to minimize malfunction probability.

CostAs in all industries manufacturing cost is also very important in automotive.

**Emissions / fuel-economy**

The main driver behind emissions-constraints has always been legislation, even though more recently fuel-economy has become an increasingly more important sales-argument. A very important aspect of this legislation is the On-board diagnostics ([OBD](https://en.wikipedia.org/wiki/On-board_diagnostics)), which monitors the engine-system continuously to ensure compliance with emission-laws in everyday use.

## **Introduction**

The automobile industry has to address the following issues at all stages of vehicle manufacturing:

* Environmental Imperatives
* Safety Requirements
* Competitive Pressures
* Customer Expectations

There is a strong interlinkage amongst all these forces of change influencing the automobile industry. These have to be addressed consistently and strategically to ensure competitiveness.

Since pollution is caused by various sources, it requires an integrated and multidisciplinary approach. The different sources of pollution have to be addressed in an integerated approach to acheive the objective of cleaner environment and meet National Air Quality standards.

## The parameters determining emission from vehicles are:

* Vehicular Technology
* Fuel Quality
* Inspection & Maintenance of In-Use Vehicles
* Road and Traffic Management

While each one of the four factors mentioned above have direct environmental implications, the vehicle and fuel systems have to be addressed as a whole as requiste fuel quality is required to meet the emission standards.

## **Vehicular Technology**

In India, vehicle technolgy has evolved to meet the emission and safety regulations notified as per the Auto Fuel Policy specifing the emission road map and safety regulations as per the Safety Road map adopted by the CMVR-TSC, respectively. Today the vehicle technolgy in India is at par with the international bench marks as Indian safety standards are being alligned with Global Technical Regulations (GTR) and UN Regulations. India is a signatory to UN WP 29 1998 agreement which develops GTRs. India actively particiates in the UN WP 29 body and contributes significantly so that the GTR reflect the driving conditions and requirements of the developing countries.

## **History of Emission Norms in India**

Vehicles are one of the contributors to air pollution and there is need to reduce vehicular emissions on a continous basis. Indian Automotive Industry recognises this fact and is continuously working towards controlling emissions as per the roadmap suggested by the Auto Fuel Policy and proactively developing environment-friendly technologies. India today has some of the most fuel efficient vehicles in the world.

The first stage of mass emission norms came into force for petrol vehicles in 1991 and in 1992 for diesel vehicles.

From April 1995, mandatory fitment of catalytic converters in new petrol passenger cars sold in the four metros, Delhi, Calcutta, Mumbai and Chennai along with supply of Unleaded Petrol (ULP) was affected. Availability of ULP was further extended to 42 major cities; and it is now available throughout the country.

In the year 2000, passenger cars and commercial vehicles met  Euro I equivalent India 2000 norms, while two wheelers were meeting one of the tightest emission norms in the world.

Euro II equivalent Bharat Stage II norms were in force 2001 onwards in Delhi, Mumbai, Chennai and Kolkata.

The first Auto Fuel Policy was announced in August 2002 which layed down the Emission and Fuel Roadmap upto 2010. As was given in the roadmap, four-wheeled  vehicles moved to Bharat Stage III emission norms in 13 metro cities from April 2005 and rest of the country moved to Bharat Stage II norms.

Bharat Stage IV for 13 Metro cities was implemented April 2010 onwards and the rest of the country moved to Bharat Stage III. Bharat stage IV norms were extended to additonal 20 cities October 2014 onwards.

The Auto Fuel Policy 2025 was submitted to the Minstry of Petroleum & Natural Gas (MoP&NG) which had constituted an expert committe for the formulation of the same in December 2013. The document is currently hosted at the  MoP&NG's website. This policy document laid down the emission and fuel road map upto 2025.

The proposed road map envisaged implementation of BS IV norms across the country by April 2017 in a phased manner and BS V emission norms in 2020/2021 and BS VI from 2024.

However, the Delhi, NCR region of North India became notorious for its drastic rise in air pollution levels. This attracted attention and subsquently led to the government making a conscious decision of leapfrogging Bharat Stage V emission norms that were subject to implementation in 2020, as well as advancing introduction of Bharat Stage VI emission norms from 2024 to 2020.

Since India embarked on a formal emission control regime only in 1991, a gap in implementaion of these norms in comparison to Europe can be noticed. However, this gap has helped in the technologies to mature which in turn faciltated the Indian Auto sector in meeting the regulations at an affordable cost for the Indian consumers.

## **Fuel Technology**

In India we are yet to address the vehicle and fuel system as a whole. It was in 1996 that the Ministry of Environment and Forests formally notified fuel specifications. Maximum limit for critical ingredients such as benzene level in petrol has been reduced continuously, from time to time, and was specified as 5% m/m and 3% m/m pa India and metroes, respectively. This limit now stands at 1%, which in line with international practices.

To address the high pollution in metro cities, 0.05% sulphur for petrol and diesel has been introduced since 2000-2001. The same has been reduced to 0.005% in April 2010 in 13 metro cities for both petrol and diesel. 350 and 150 ppm for diesel and petrol, respectively, in rest of the country, the limit on sulphur content for petrol and diesel is 150ppm and 350ppm, respectively. This content would be reduced further to 10 ppm in BS V and BS VI fuels in line with Auto Fuel Policy 2025. There is a need completely align the fuel properties with Europeon fuel quality so that vehicles can meet BS VI emission norms and also the durability requirement.

## **Inspection & Maintenance (I&M) of In-use Vehicles**

It has been estimated that at any point of time, new vehicles comprise only 8% of the total vehicle population. In India, currently only transport vehicles, that is, vehicles used for hire or reward are required to undergo periodic fitness certification. The large population of personalised vehicles are not yet covered by any such mandatory requirement.

In most countries that have been able to control vehicular pollution to a substantial extent, Inspection & Maintenance (I&M) of all categories of vehicles has been one of the chief tools used. Developing countries in the South-East Asian region, which till a few years back had severe air pollution problems, have introduced an I&M system and an effective traffic management plan.

## **Road & Traffic Management**

Inadequate and poor quality of road surface leads to increased vehicle operation costs, thereby increased pollution. It has been estimated that improvements in roads will result in savings of about 15% of vehicle operation costs.

**Conclusion**

The need for an integrated holistic approach for controlling vehicular emissions cannot be over-emphasised. More importantly, the auto and oil industries need to come together for evolving fuel quality standards and vehicular technology to meet the air quality targets.

**Frequently Asked Questions**

1. What is the difference between Harvard and Von-Neumann architecture?
2. Differentiate CISC and RISC
3. What is embedded system? explain with any one example
4. How many classification of embedded systems are present?
5. Give any 4 major difference between microprocessor and micro controller.
6. Mention what are buses used for communication in embedded system?
7. What is pipelining?
8. What is Embedded Automotive?
9. What are the functional areas of Embedded Automotive?
10. What are the Key Segments in Automotive Industry?

**Domain & Potential Roles offered**

* Engineering
* Automotive
* IT Services
* Electronics
* Telecom & Wireless
* Semi Conductor
* Aerospace
* Testing
* Desktop GUI Web technology
* Gaming & Enterprise
* Cloud based technology
* Big data technologies

**Potential Roles**

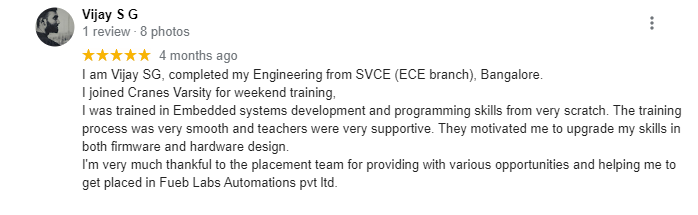
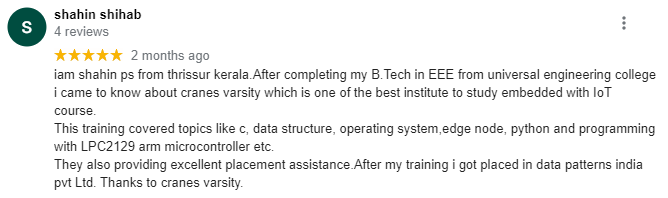
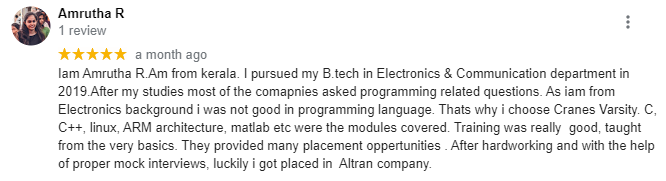
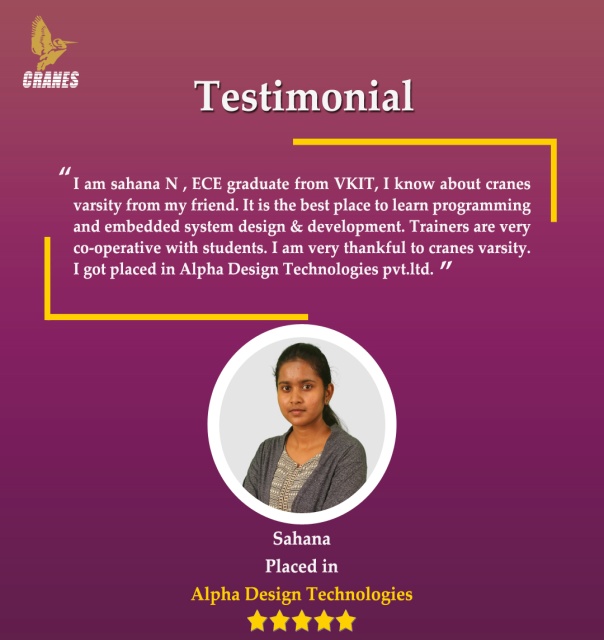
* Embedded Developer
* Software developer
* Technical Analyst
* Junior software Engineer
* Network Engineer
* Junior embedded IOT engineer
* AI Platform Engineer

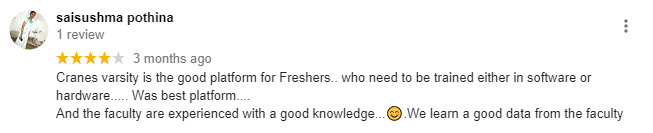
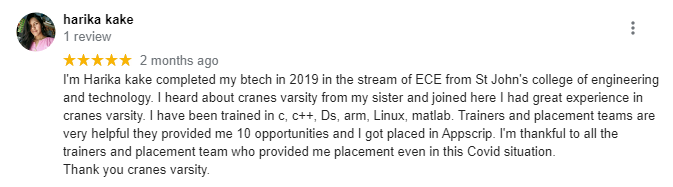
**Packages Offered**

* 5-8 LPA (Tier I )
* 2-4 LPA (Tier II )

**Testimonials:**

****

**  
**



**Placements:**



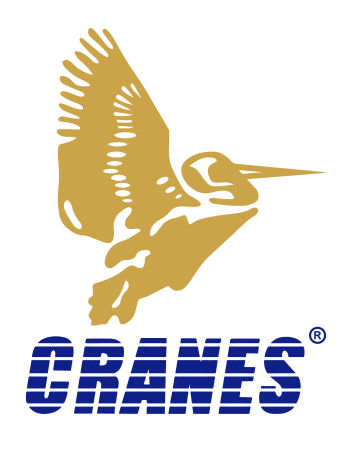


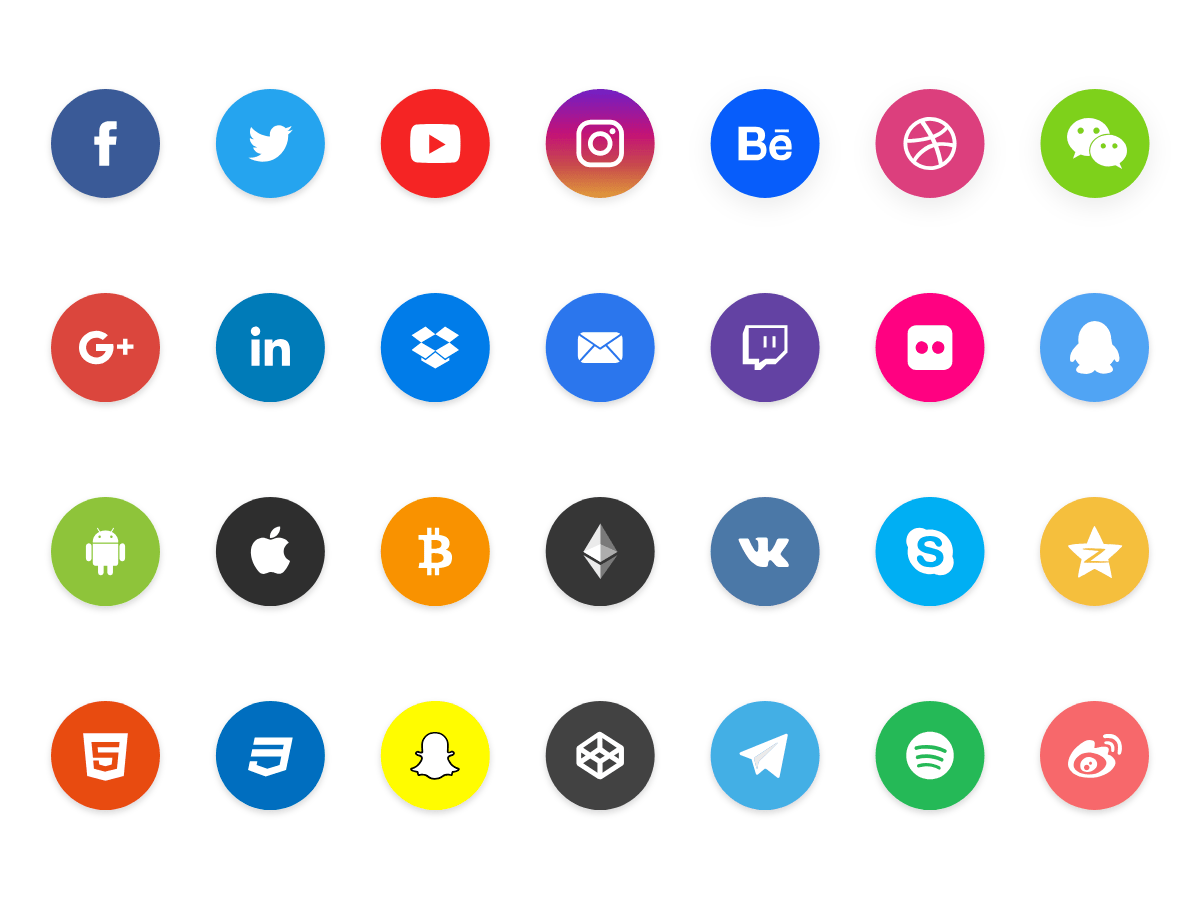
**Cranes Varsity always welcomes You !**

**Job Assistance Courses:**

Training & Placement Program for **ENGINEERS:**

Embedded system design &AutomotiveInternet of Things & Data scienceVLSI Design & VerificationFull stack java development   
Data science using python

  
[**www.cranesvarsity.com**](file:///C:\Users\jeyar\Downloads\www.cranesvarsity.com)

[](https://www.instagram.com/cranesvarsity/)

# 82, Presidency Building, 3rd & 4th Floor, St Marks Rd,   
Bangalore, Karnataka 56000.