Embedded Systems - Introduction

4. SOFTWARE FOR EMBEDDING IN A SYSTEM

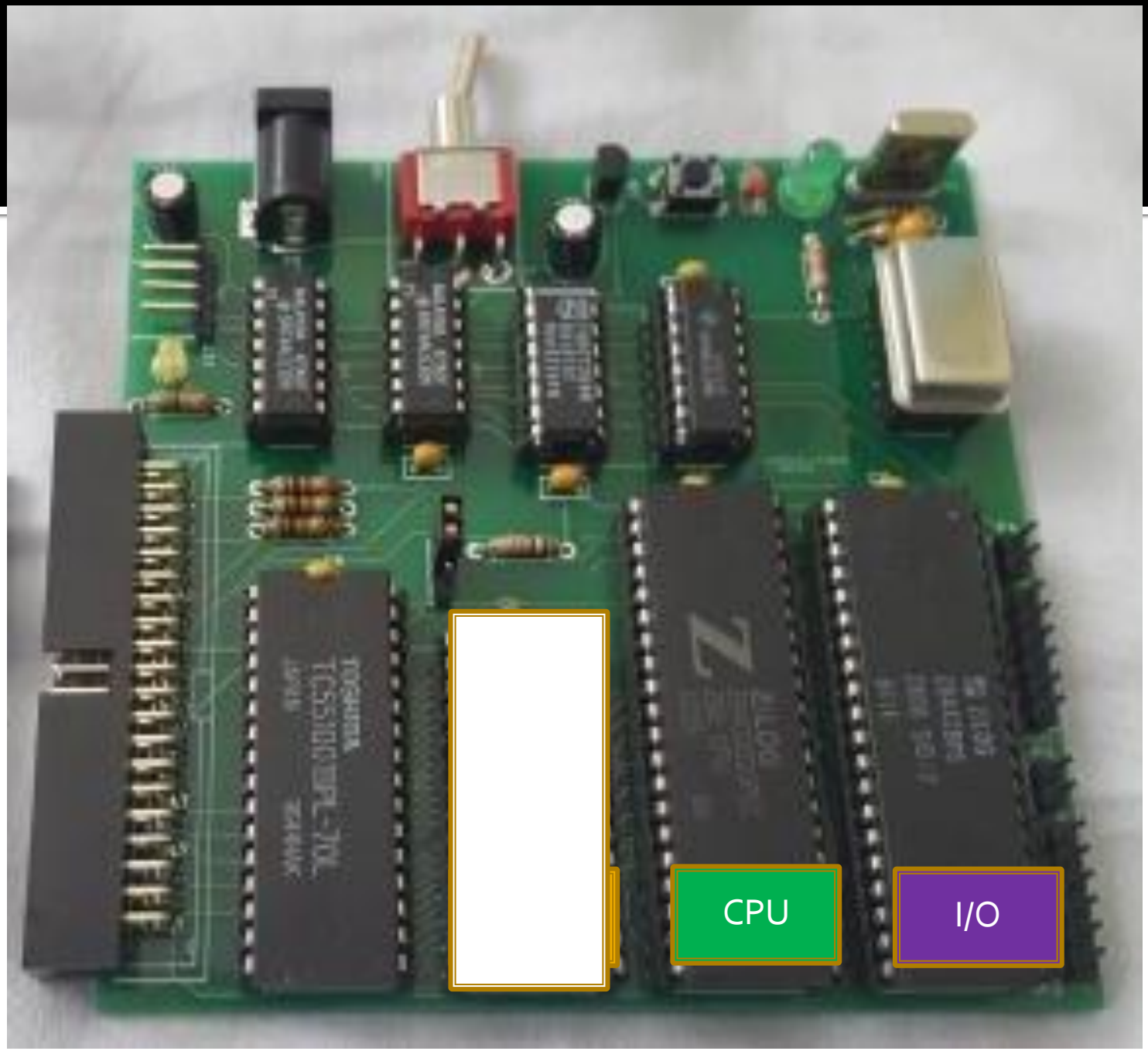
Part I

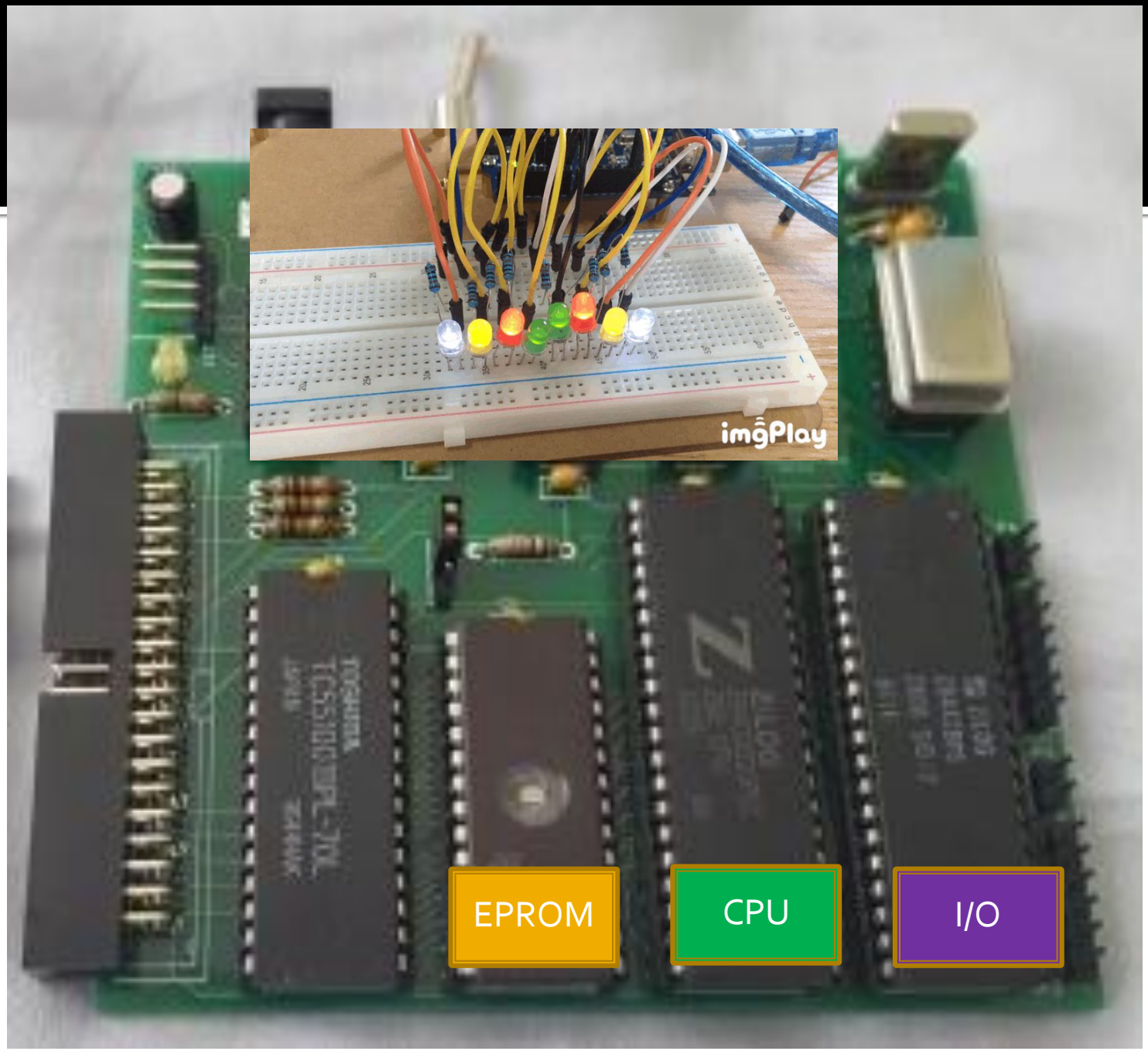
- ROM image
- Programming Languages
- Program models

4.1. ROM Image

- Final stage software also called ROM image*

* (Just as an image is a unique sequence and arrangement of pixels, embedded software is also a unique placement and arrangement at each ROM address of bytes for instructions and data.)


CPU
I/O

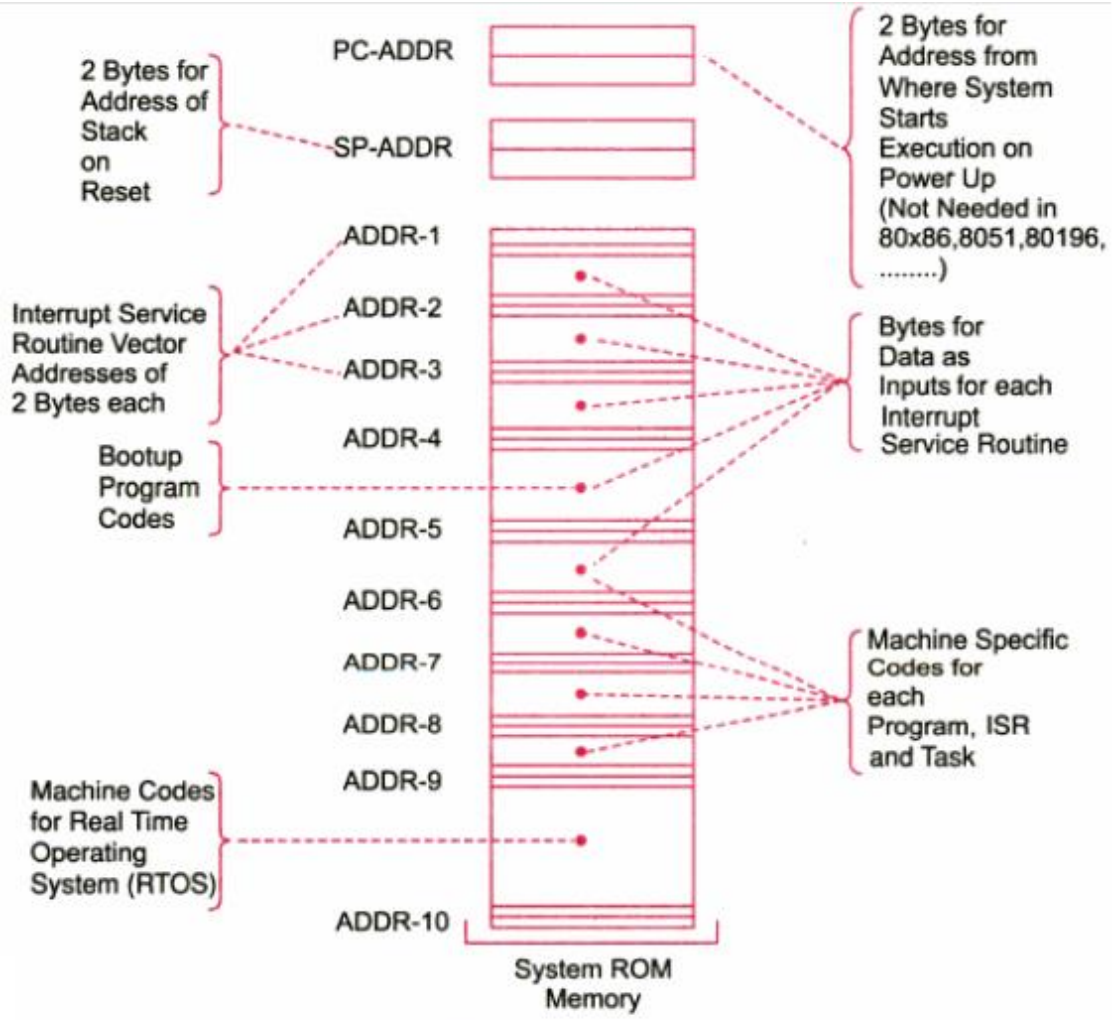

imgPlay
EPROM
CPU
I/O

4.1. ROM Image

ซอฟต์แวร์ขั้นตอนสุดท้ายเรียกอีกอย่างว่า ROM image

โดยในภาพต่อไปนี้ คือการจัดเรียงของซอฟต์แวร์ ที่อยู่หน่วยความจำถาวร(ROM,PROM) ตำแหน่งต่างๆ

4.1. ROM Image



System ROM memory embedding the software, RTOS, data and vector addresses

4.1.1 Final machine software

- Bytes at each address จะถูกใส่ไว้ ROM image
- และถ้ามีการเปลี่ยนแปลงหรืออัพเกรด image นี้เกิดขึ้นถึงแม้จะเป็น hardware platform เดียวกัน ผลการทำงานที่ได้ก็จะแตกต่างกันโดยสิ้นเชิง

4.1.2 Distinct ROM image in a distinct Embedded System

- Hardware elements ที่มีบนระบบที่แตกต่างกันนั้น สามารถที่จะทำงานเหมือนกันได้ แต่ตัวของ software ที่ถูกสร้างขึ้นมานั้นจะต้องสร้างให้เข้ากับระบบแต่ละแบบซึ่งระบบแต่ละแบบนั้นก็จะมีความแตกต่างกัน

4.1.3 Compressed Codes and Data

- ROM image อาจจะถูกเก็บไว้ในรูปของ compressed software ได้ (เช่นในรูปของ zip format) และตัว data (เช่นภาพในรูปแบบของ jpg หรือ gif format) โดยใช้ software ที่ต้องมีส่วนของ decompression algorithm

4.2. Programming Languages

4.2.1 ภาษาเครื่อง (Machine Language)

สามารถเข้าถึงตำแหน่งของหน่วยความจำได้ในระดับบิตและไบต์

โดยสามารถควบคุมระบบฮาร์ดแวร์ได้โดยตรง อีกทั้งยังเข้าถึงระบบการเข้ารหัสภาษาเครื่อง

4.2. Programming Languages

2. Assembly Language Coding

- Needed for Invoking Processor Specific Instructions
- Requires understanding of the processor and instruction set.
- A program or a small specific part coded in the assembly language using an Assembler (software used for developing codes in assembly).

4.2. Programming Languages

2. ภาษาแอสเซมบลี(Assembly Language)

ต้องการความเข้าใจเกี่ยวกับโปรเซสเซอร์และชุดคำสั่งโปรแกรมหรือชิ้นส่วนเฉพาะขนาดเล็กที่เข้ารหัสในภาษาแอส-เซมบลีโดยใช้แอสเซมเบลอร์ (ซอฟต์แวร์ที่ใช้สำหรับการพัฒนาโค้ดในแอสเซมบลี)

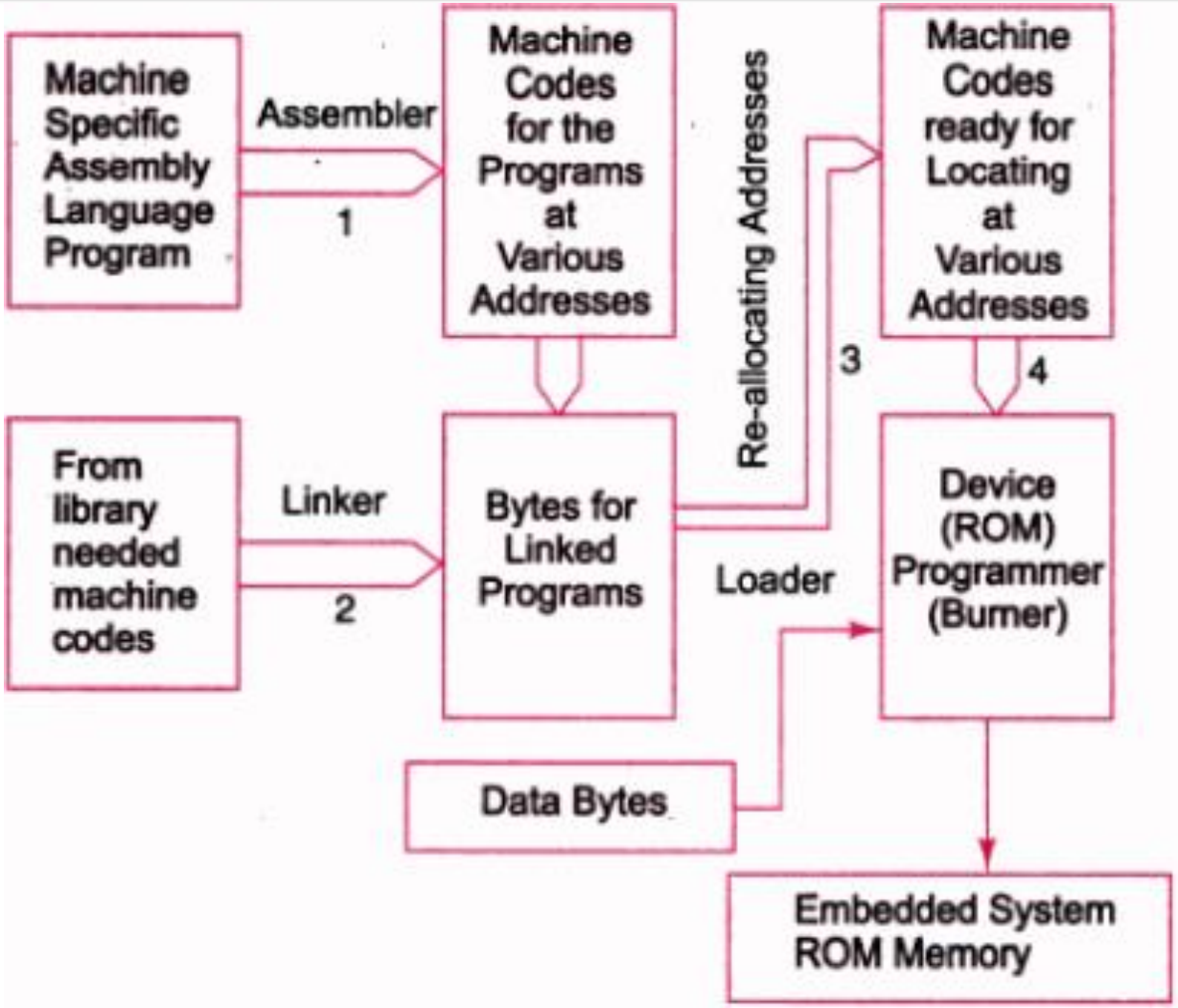
4.2. Programming Languages

Three steps when using assembly language

- 'Assembler',
- 'Linker' and
- 'Locator'

before finally burned at the ROM

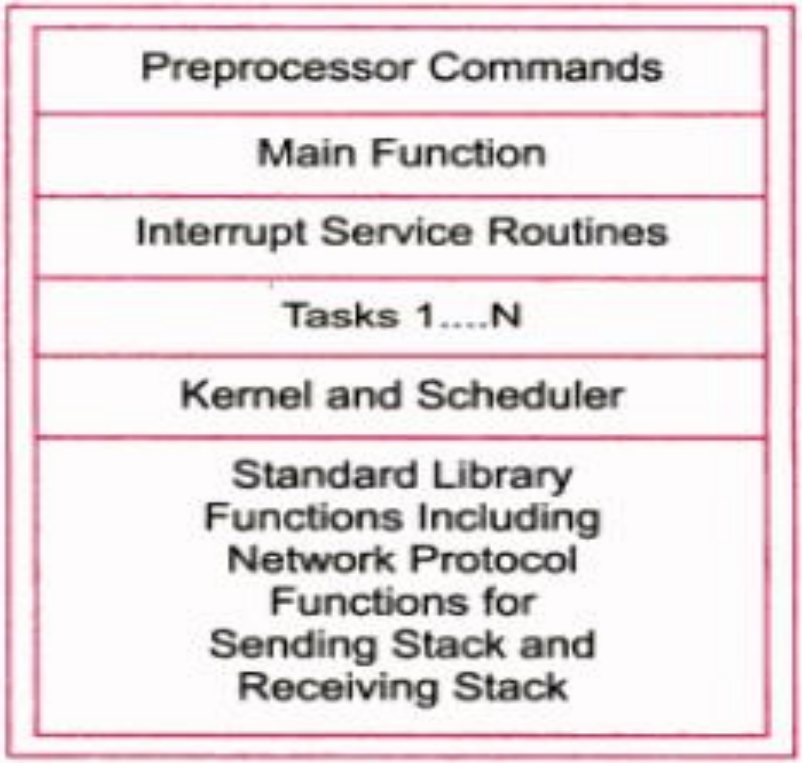
4.2. Programming Languages



Assembly - Conversion Process Assembly to ROM image

4.2. Programming Languages

3) Programming language C or C++ or Visual C++ or Java .



Application Software - Different Program Layers

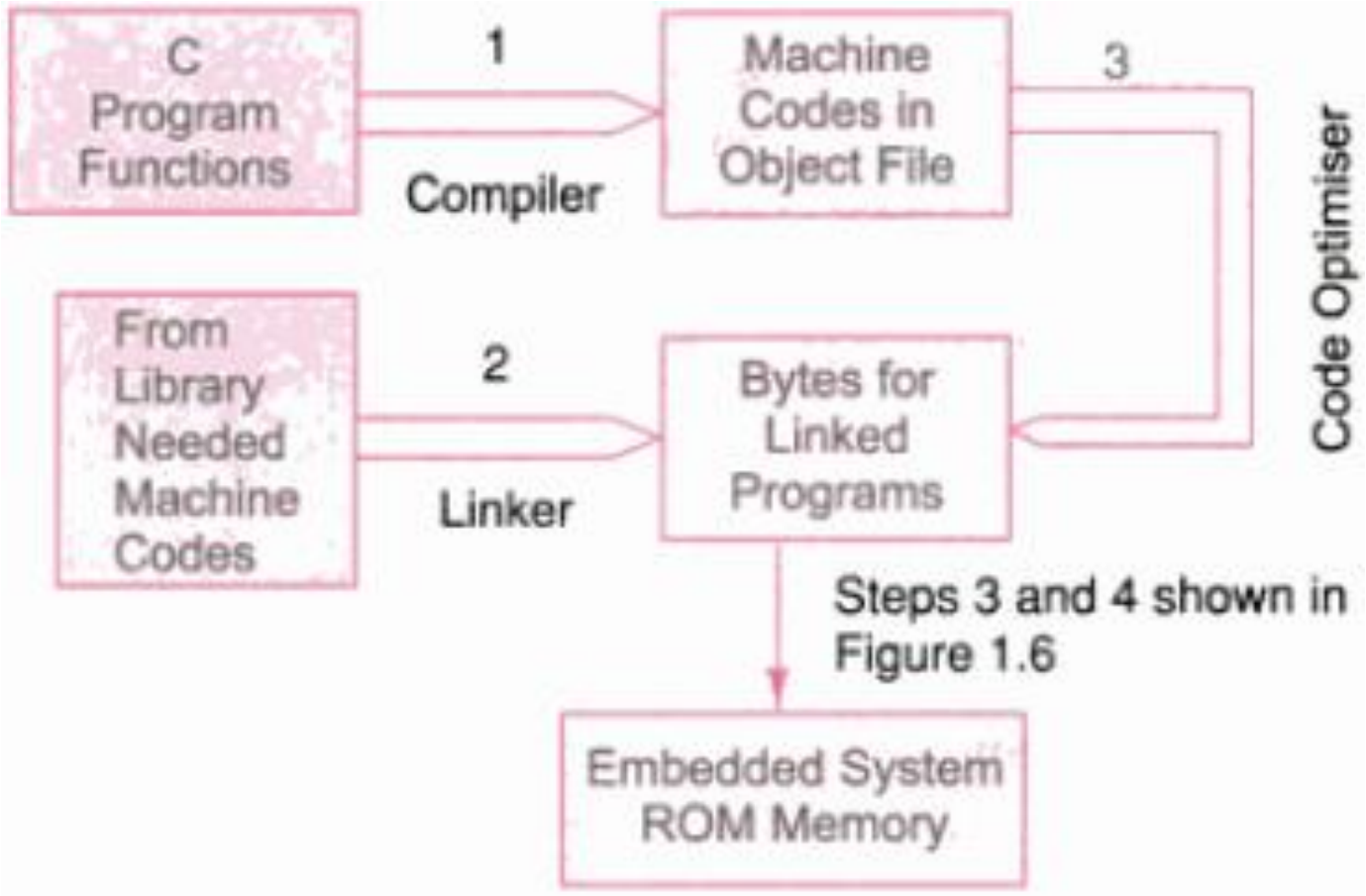
4.2. Programming Languages

Compiler

- Generates an object file. Using linker and locator, the file for ROM image is created for the targeted hardware. C++ and Java are other languages used for software coding.

4.2. Programming Languages

Converting a C program into ROM image



4.3. Program models

- Sequential Programming Model
- Object Oriented Programming Model
- Control and Data flow graphs or Synchronous Data Flow (SDF) Graph or Multi Thread Graph (MTG) Model
- Finite State Machine for data path
- Multithreaded Model
- Concurrent Processing of processes or thread or tasks

4. SOFTWARE FOR EMBEDDING IN A SYSTEM

Part II

Device drivers, Device manager, OS,

RTOS and Software tools

4.4. Device drivers

- In an embedded system, there are number of *physical devices*.
- Physical devices – keypad, LCD display or touch screen, memory stick (flash memory), wireless networking device, parallel port and network card…
- In an embedded system, there are number of *virtual devices*.
- Virtual devices – pipe, file, RAM disk, socket, …

4.4. Device drivers

- A *device driver* is software for controlling (configuring), receiving and sending a byte or a stream of bytes from or to a device.
- A set of generic functions, such as create ( ),open ( ), connect ( ), listen ( ), accept ( ),read ( ), write ( ), close ( ), delete ( ) for use by high level programmers
- Each generic function calls a specific software (interrupt service routine), which controls a device function or device input or output

4.4. Device drivers

Device controls and functions by :

1. Calling an ISR (also called Interrupt Handler Routine) on hardware or software interrupt
2. Placing appropriate bits at the control register or word.
3. Setting status flag(s) in the status register for interrupting, therefore running (driving) the ISR, Resetting the status flag after interrupt service.

4.5. Device manager

- Device Management software (usually a part of the OS) provide codes for detecting the presence of devices, for initializing (configuring) these and for testing the devices that are present.
- Also includes software for allocating and registering port(s) or device codes and data at memory addresses for the various devices at distinctly different addresses, including codes for detecting any collision between the allocated addresses , if any

4.6. Multitasking using an operating system (OS) and Real time operating system (RTOS)

- Concurrent Processes, tasks or threads
- A System is composed of two or more concurrent processes that execute
- Operating System
- Multitasking (multiprocessing or multithreaded) software
- Scheduling multiple tasks,
- Processes, memory, device, ports, network, file system, timers, event functions, inter-processor communication, shared memory, security, GUIs, ... management

4.6. Multitasking using an operating system (OS) and Real time operating system (RTOS)

- Real Time Operating System (RTOS)
  - Embedded software is most often designed for deterministic performance and task and ISR latencies in addition to the OS functions
  - Performing multiple actions and controlling multiple devices and their ISRs with defined real time constraints and with deadlines for these
  - Task and ISRs priority allocations, their preemptive scheduling, ..

4.6. Multitasking using an operating system (OS) and Real time operating system (RTOS)

- Task and ISRs priority allocations, their preemptive scheduling, ..
- RTOS and concurrent processes
- OS for providing deterministic performance during concurrent processing and execution with hard (stringent) or soft timing requirements with priority allocation and preemption

4.6. Multitasking using an operating system (OS) and Real time operating system (RTOS)

"RTOS is needed when the tasks for the system have real time constraints and deadlines for finishing the tasks"

4.7. Software tools

- Development Tools

1. Editor,
2. Interpreter,
3. Compiler,
4. Assembler and Cross Assembler,
5. IDE,
6. Prototyper

4.7. Software tools

- Application Software Development Tools
- Source Code Engineering Tools
- Stethoscope (tracks the switching from one task to another as a function of time, stores beats)
- Trace Scope (traces changes in a parameter(s) as a function of time)

4.7. Software tools

- Simulator - A Simulator... To simulate the target processor and hardware elements on a host PC and to run and test the executable module
- Project Manager - To manage the files that associates with a design stage project and keep several versions of the source file(s) in an orderly fashion.

Q n A