

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

STUDY MATERIALS



a complete app for ktu students

Get it on Google Play

www.ktuassist.in

Reg No.: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
EIGHTH SEMESTER B.TECH DEGREE EXAMINATION, MAY 2019

Course Code: CS404
Course Name: Embedded Systems

Max. Marks: 100

Duration: 3 Hours

PART A

Answer all questions, each carries 4 marks.

Marks

- | | | |
|----|--|-----|
| 1 | What is an embedded computing system? Write two functionalities of an embedded system. | (4) |
| 2 | Explain the problems of hardware software co-design in an embedded system. | (4) |
| 3 | Draw a concurrent program model for Seat Belt Warning System of an automobile. | (4) |
| 4 | Explain the library file in assembly language context. What is the benefit of 'library file'. | (4) |
| 5 | Briefly describe out of circuit programming in Embedded System. | (4) |
| 6 | Differentiate generic IDEs with IDEs used in embedded firmware development with suitable examples. | (4) |
| 7 | Explain hard real-time considerations and soft real-time considerations | (4) |
| 8 | Differentiate monolithic kernel with microkernel | (4) |
| 9 | Explain System on Chip technique (SOC) | (4) |
| 10 | Write any 4 bottlenecks available in the embedded industry. | (4) |

PART B

Answer any two full questions, each carries 9 marks.

- | | | |
|----|--|-----|
| 11 | a) With a suitable example, explain the specification phase of an embedded system. | (5) |
| | b) Show the UML representation of an object and a class with a suitable example. | (4) |
| 12 | a) Design a coin operated public telephone unit based on FSM model for the following requirements. | (9) |
| | 1. The calling process is initiated by lifting the receiver (off-hook) of the telephone unit. | |
| | 2. After lifting the phone the user needs to insert a 1 rupee coin to make the call. | |

3. If the line is busy, the coin is returned on placing the receiver back on the hook (on-hook).
4. If the line is through, the user is allowed to talk till 60 seconds and at the end of 45th second, prompt for inserting another one rupee coin for continuing the call is initiated.
5. If the user doesn't insert another 1 rupee coin, the call is terminated on completing the 60 seconds time slot.
6. The system is ready to accept new call request when the receiver is placed back on the hook (on-hook).
7. The system goes to the "Out of Order" state when there is a line fault.

(No need to take care of the scenarios like user doesn't insert a coin within the specified time after lifting the receiver, user inserts coins other than a one rupee etc.)

- 13 a) List and explain the non functional requirements in an embedded system. (4)
- b) Draw a class diagram for a basic microwave oven, cooking time should be adjusted from 1 min to 60 min. Include classes for door, front panel and heating elements. (5)

PART C

Answer any two full questions, each carries 9 marks.

- 14 a) With a neat diagram explain the steps in converting assembly language to machine language (9)
- 15 a) Explain the Debuggers used in Embedded System Development Environment (5)
- b) Briefly describe (i) decompiler (4)
- (ii) disassemblers
- 16 a) Is it possible to embed the firmware into the target processor/controller memory at the time of chip fabrication? Justify your answer. (3)
- b) Explain the merits and demerits of assembly language based embedded firmware development . (6)

PART D

Answer any two full questions, each carries 12 marks.

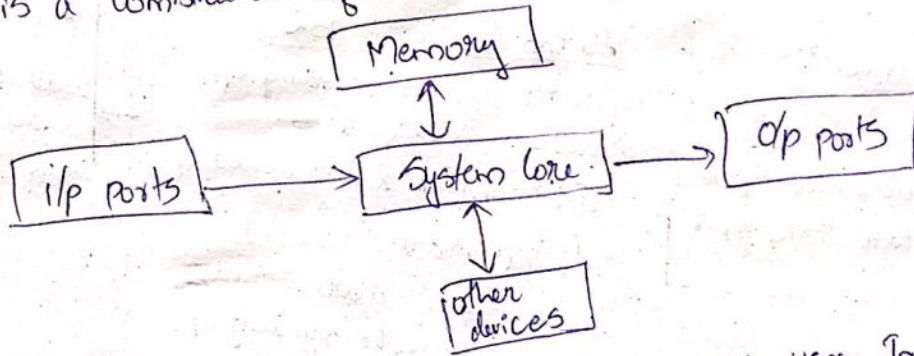
- 17 Explain the different types of Inter Task Communication mechanisms supported by MicroC/OS-II kernel. (12)
- 18 (a) Explain the various steps involved in the development of an embedded system (5)

using Waterfall model.

- (b) Explain the need for product Re-engineering in embedded product development. (4)
- (c) What are the factors that lead to the disposal of an embedded product. (3)
- 19 a) Consider a mobile phone device and look at the main menu. Explain how the events of touching the screen at different points on the screen are handled by an RTOS using two-level ISR handling. (6)
- (b) Explain various types of testing performed in Embedded product development. (6)

1) What is embedded Computing System? Write two functionalities of an embedded system.

A:) An embedded system can be thought of as a computer h/w s/w having s/w embedded in it. An embedded s/w can be independent s/w or it can be a part of a large s/w. An embedded s/w can be independent ^{is a MC or MP} based s/w which is designed to perform a specific task. Eg. Fire alarm. It is a combination of h/w, s/w and Mechanical components (PDS.)



Functionalities include Complex algorithms and User Interface.

1) Complex algorithms: The operations performed by the MP may be very sophisticated. For ex: the MP that controls an automobile engine must perform complicated filtering jobs to optimize the performance of the car while minimizing pollution.

2) User Interface: MP are frequently used to control complex user interfaces that may include multiple menus & many options. The moving maps in GPS navigation are ex. of this.

2) Describe the problems of h/w-s/w co-design in ES.

1) Model Selection: A model captures & describes the s/w characteristics.

A model is a formal s/w consisting of objects and composition rules:

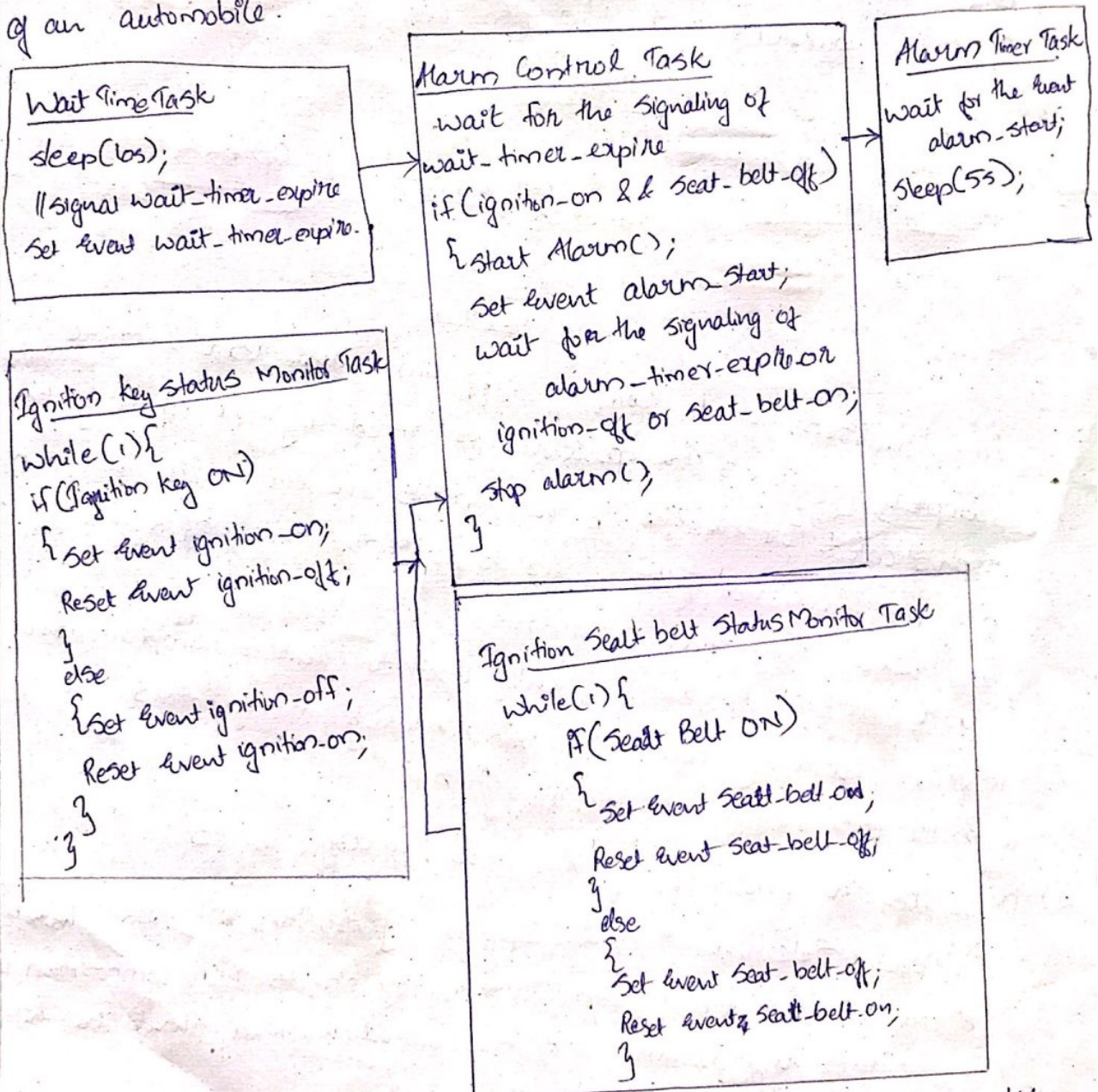
- It is hard to make a decision on which model should be followed in a particular s/w design.
- Most often designers switch b/w a variety of models from the requirements specifications to the implementation aspect of s/w design.

2) Architecture Selection: The architecture specifies how a s/w is going to implement in terms of the no. & types of different components & the interconnection among them.

3) Language Selection: A programming lang captures a 'Computational Model' of s/w.

it into architecture. A model can be captured using multiple programming lang. like C, C++ etc for s/w implementations & lang like VHDL, SystemC etc for h/w implementations. The only pre-requisite in selecting a programming lang for capturing a model is that the lang should capture the model easily.

3) Draw a concurrent program model for Seat Belt Warning s/w of an automobile.



→ Explain library file in assembly lang. Context. What is the benefit of 'library file'?

Libraries are specially formatted, ordered program collection of object modules that may be used ~~for~~ by the linker at a later time. When a linker process a library, only those ~~subset~~ object modules in the library that

are necessary to create the program are used. Library files are generated with extension '.lib'. Library file is some kind of source code hiding technique. If you don't want to reveal the source code hiding technique behind the various fns you have written in your prgm and at the same time you want to be distributed to application developers for making use of them in their applications, you can supply them as library files and give them the details of the public fns available from the library. For using library files in a project, add library to the project.
eg: LIB51 from keil s/w.

- 5) Briefly describe ~~about~~ ^{out of} circuit programming in ES.
- Out-of circuit programming is performed outside the target board. The processor or mpy chip into which the firmware needs to be embedded is taken out of the target board and it is programmed with the help of programming device. The programming device is dedicated unit which contains the necessary h/w ckt to generate the programming signals. The programmer contains a ZIF socket with locking pin to hold the device to be programmed. The programming device will be under the control of the utility prgm running on a PC. The commands to control the programmer are sent from the utility program to the programmer through the interface.
- Drawbacks: High development time and not suitable for batch production.

- 6) Differentiate generic IDEs with IDEs used in embedded firmware development with suitable examples.
- In ES, IDE stands for an integrated environment for developing and debugging the target processor specific embedded firmware. An IDE is also known as integrated design environment or integrated debugging environment. IDE is a s/w package which bundles a "Text Editor", "Cross Compiler", "Linker" and a "Debugger". IDE is a s/w application that provides facilities to computer programmers for s/w development. IDEs can either be command line based or GUI based. IDE consists of
- 1) Text editor / Source Code Editor
 - 2) A Compiler & an interpreter
 - 3) Build automation tools
 - 4) Debugger
 - 5) Emulators & logic analyzer.
 - 6) Simulators.

An ex of IDE is Turbo C/C++ which provides platform on windows for development of application prgms with Command Line interface. ^{Generic} IDE used for high level lang based development for desktop applications are different from the IDEs used in Embedded firmware. In ES, IDE ~~are~~ is either applied to supplied by the target processor/ Controller manufacturer or by third party vendors or as Open source.

7) Explain hard - real-time considerations & soft real-time considerations.

~~Soft~~ A hard - real time system is a slm in which a single failure to meet the deadline may lead to a complete slm failure while a soft - real time slm is a slm in which one or more failures to meet the deadline is not considered as complete slm failure, but its performance is considered degraded.

For soft real-time systems, missing a deadline may not be critical and can be tolerated to a certain stage.

For hard-real-time slm's - missing a program/task execution time deadline can have catastrophic consequences.

10-a) With suitable examples, explain the specification phase of an ES

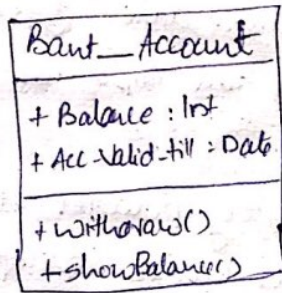
The specification is more precise - it serves as the contract b/w the customer and the architects. The specification must be carefully written so that it accurately reflects the customer's requirements and does so in a way that can be clearly followed during design. The specification should be understandable enough so that someone can verify that it meets slm requirements & overall expectations of the customer.

It should also be unambiguous enough that designers know what they need to build. If the global characteristics of the specification are wrong or incomplete, the overall slm architecture derived from the specification may be inadequate to meet the needs of implementation.

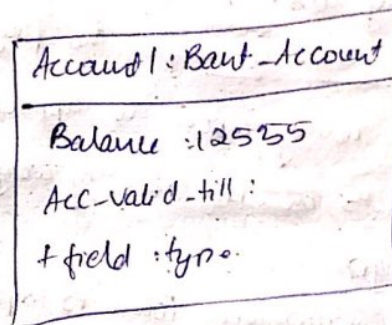
Specification of the GPS slm would include several components:

1) Data received from the GPS satellite constellation.
2) Map data (3) User Interface (4) Operations that must be performed to classify customer requests

- 11-b) Show the UML representation of an object and class with a suitable eg: (5)
- Object is an instance of a particular moment. A state UML diagram is an instance of a class diagram, it shows a snapshot of the detailed state of a sm at a point in time, thus an object diagram encompasses objects and their relationship at a point in time.
- An object diagram is similar to a class diagram except it shows the instances of classes in the sm. We depict actual classes and their relationships making use of class diagrams. On the other hand an object diagram represents specific instances of classes & relationship b/w them at a point of time.

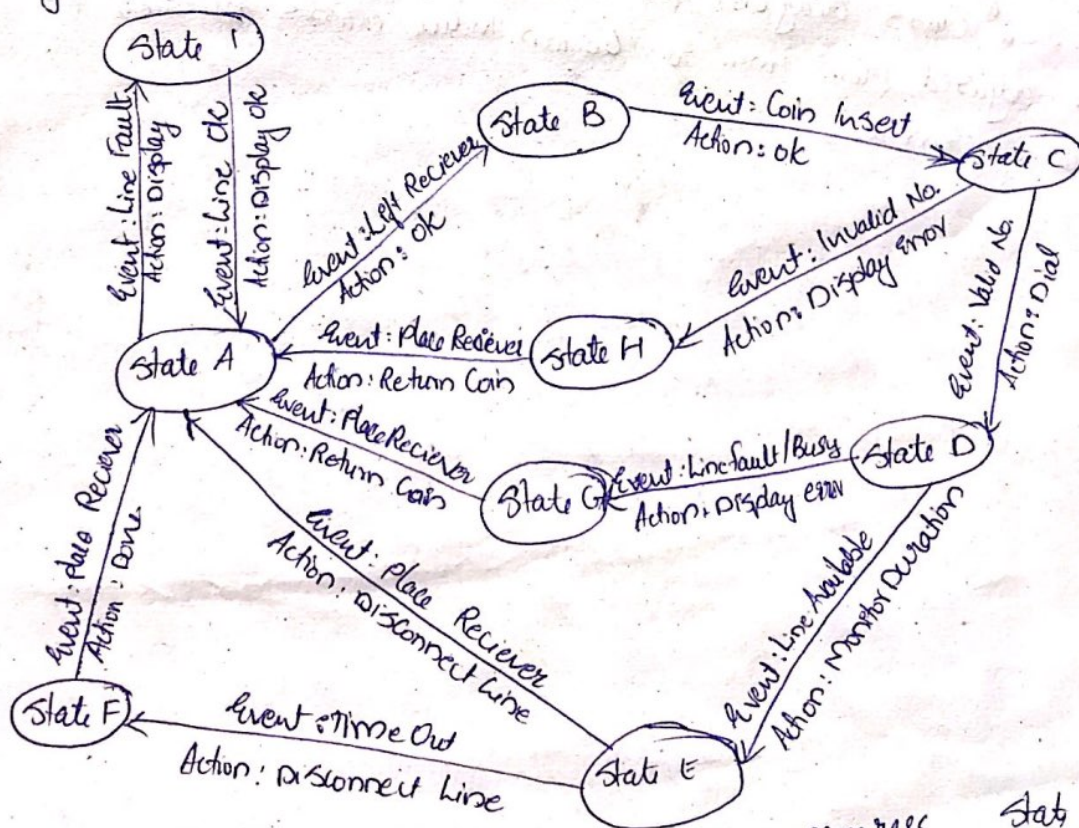


A class



An object

- 12-a) Design a coin operated public telephone unit based on FSM model.



State A : Ready
 State B : Wait for coins
 State C : Wait for no.
 State D : Dialing

State E : Call in progress
 State F : Call terminated
 State G : Unable to make call
 State H : Invalid no. ip

State I : Out of order

13-a) List and explain non-functional requirements in a FS. (6)

1) Performance: The speed of the S/W is often a major consideration both for the usability of the S/W and for its ultimate cost. As to Performance may be combination of soft performance metrics such as appraised time to perform a user-level task & hard deadlines by which a particular operation must be completed.

2) Cost: The target of purchase price for the S/W is almost always a consideration. Cost typically has manufacturing cost and nonrecurring engineering cost.

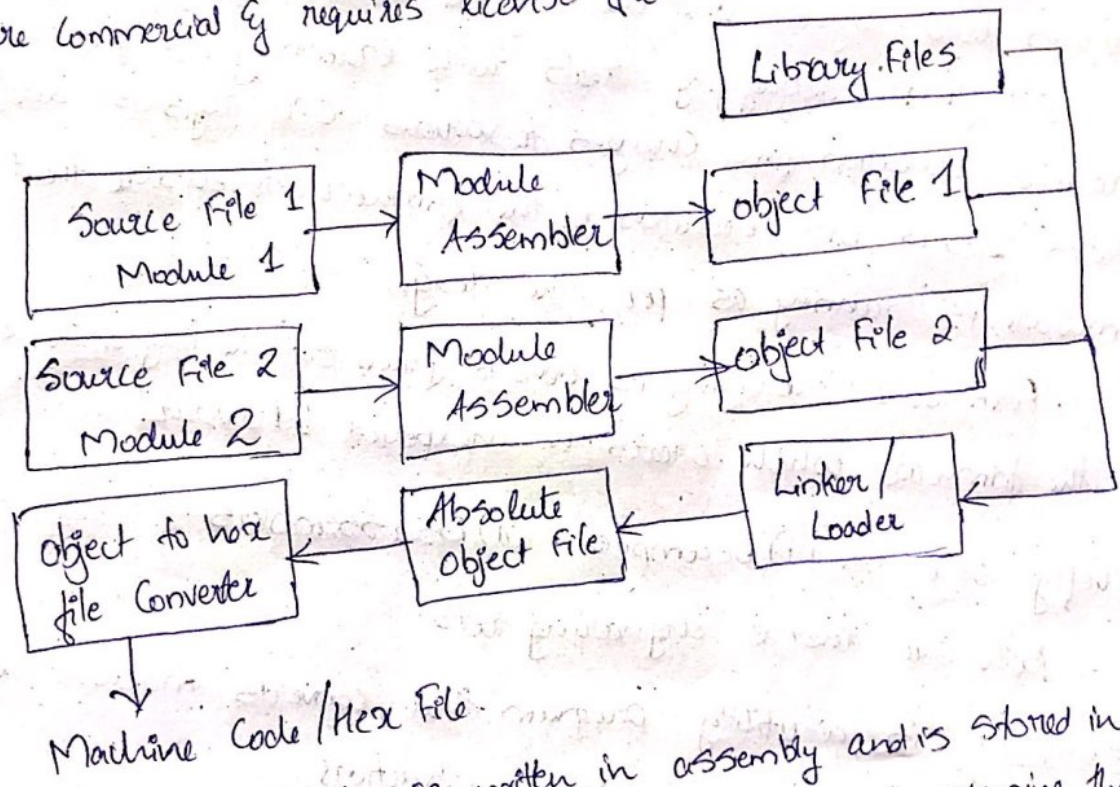
3) Physical size & weight: The physical aspects of the final S/W can vary greatly depending upon the application. A handheld device typically has tight requirements on both size & weight ~~upon~~ that can ripple through the entire design.

4) Power Consumption: Power is important in battery-powered S/W & is often important in other applications as well. Power can be specified in the requirements stage in terms of battery life.

13-b) Draw a class diagram for a basic microwave oven, cooking time should be adjusted from 1 min to 60 min. Include classes for door, front panel & heating elements.

14-a) With a neat diagram explain the steps in converting assembly lang to machine lang.

The translation of assembly code to machine code is performed by assembler. The assemblers for different target machines are different and it is common that assemblers from multiple vendors are available in the market for the same target machines. Some assemblers are supplied by single vendor only are some are freely available. Some are commercial & requires license from vendors.



Each source module is written in assembly and is stored in .src or .asm file. Each file can be assembled separately to examine the syntax errors and incorrect assembly instructions. On assembling of each .src/.asm file a corresponding object file is created with extension .obj. The object file does not contain the absolute address of where the generated code need to be placed on the program memory. Hence it is called relocatable segment. It can be placed at any code memory location and it is responsibility of the linker/loader to assign absolute address for this module. Each module can share variables and subtractive among them. Exporting a variable from a module is done by declaring that variable as PUBLIC in source module.

15-a) Explain the debuggers used in Embedded System Development Environment. (8)

Debugging in ES embedded application is the process of diagnosing the firmware execution, monitoring the target processor's registers and memory while the firmware is running and checking the signals from various buses of Embedded h/w.

classified as 1) Hardware debugging: Deals with monitoring of various bus signals and checking the status lines of target h/w.
2) Software debugging: deals with examining the firmware execution, execution flow, changes to various CPU registers and status registers on execution of the firmware to ensure that the firmware is running as per the design.

Firmware debugging is done to figure out the bug or the error in the firmware which creates the unexpected behaviour.

15-b) Briefly describe (i) Decompiler (ii) Disassemblers.

- Both are reverse engineering tools.

i) Decompiler: is a utility program that converts machine lang instruction to high level lang instructions.

- Performs reverse operation of compiler or cross compiler.

ii) Disassembler: utility program that convert machine code into assembly Code.

- It is Complementary to assembly or cross assembly.

16-a) Is it possible to embed the firmware into target processor/controllers only at the time of chip fabrication? Justify.

It is possible. Such chips are known as 'Factory Programmed chips'. The OS based ES are programmed using

(9)

the In Str programming technique. OS ES contain a special piece of code called 'Boot loader' program which takes control of the OS and application firmware embedding and copying of the OS image to the RAM of the Str for execution.

16-b) Explain the merits & demerits of assembly lang based embedded firmware development.

Ans) Advantages:

* Efficient Code Memory and data memory usage (Memory Optimization)

Since the developer is well versed with the target processor architecture and memory organization, optimized code can be written for performing operations. This lead to the less utilization of code mly and efficient utilization of data mly.

* High Performance.

Optimized Code not only improve the code mly usage but also improve the total Str performance. Though effective assembly coding optimum performance can be achieved for target applications.

Drawbacks:

* High development time.

Assembly lang programs are much harder to prgm than high level lang. Learning the inner details of the processor & its assembly instructions are high time consuming and it create delay impact in product development.

Solution is to use a readily available developer who is well versed in action which target processor architecture assembly instructions.

* Developer Dependency.

In assembly lang programming, ~~also~~ developers have the freedom to choose the different only locations and registers.

If the approach done by developer is not documented properly at the development stage, it may not be able to recollect at later stage or when a new developer is instructed to analyze the code, he may not be able to understand what is done and why it is done.

Hence upgrading/modifying on later stage is more difficult.

Solution is well documentation.

* Non portable.

Target applications written in assembly instructions are valid only for that particular family of processors. Cannot be reused for another target processors. If the target processor changes, a

Complete rewriting of the application using assembly instructions for the new target processor is required.

try it now

A KTU
STUDENTS
PLATFORM

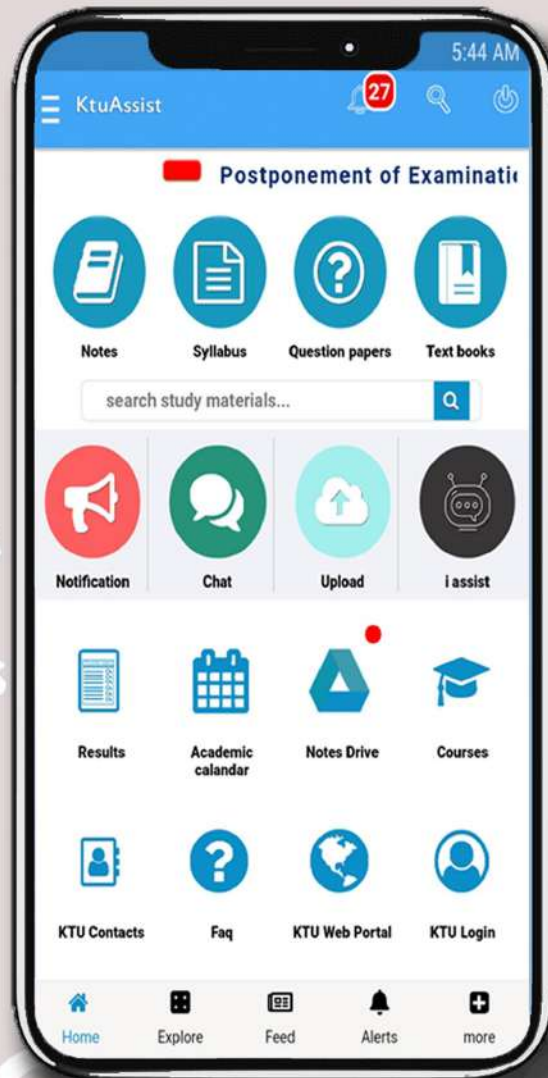
SYLLABUS

NOTES

TEXT BOOKS

QUESTION PAPERS

DOWNLOAD
IT
FROM
GOOGLE PLAY



CHAT
A
LOGIN
FAQ
E
N
D
A

MUCH MORE

DOWNLOAD APP



ktuassist.in

instagram.com/ktu_assist

facebook.com/ktuassist