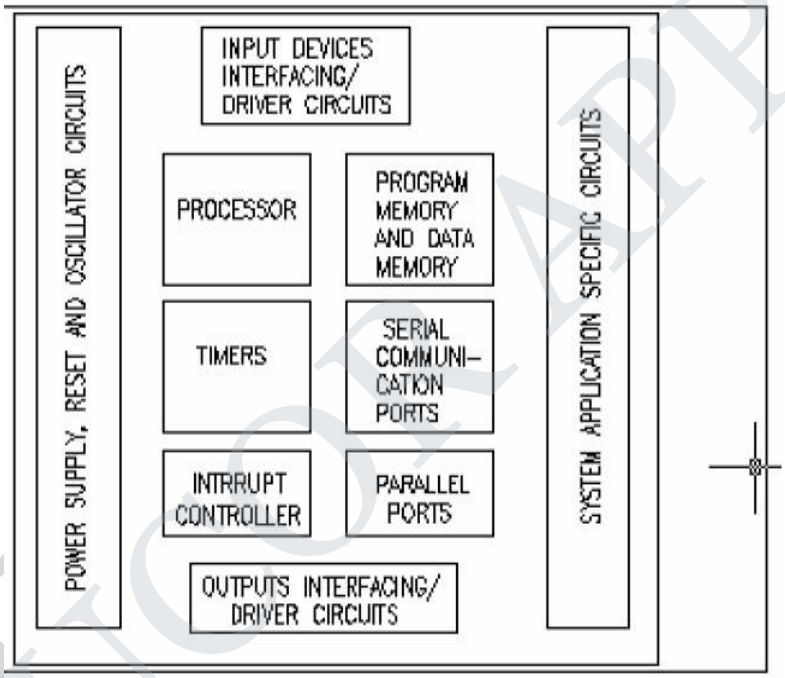
Subject Code:EE8691 **Year/Semester: III /06**
Subject Name: EMBEDDED SYSTEMS **Subject Handler: Vinnarasi Ponniury.P**

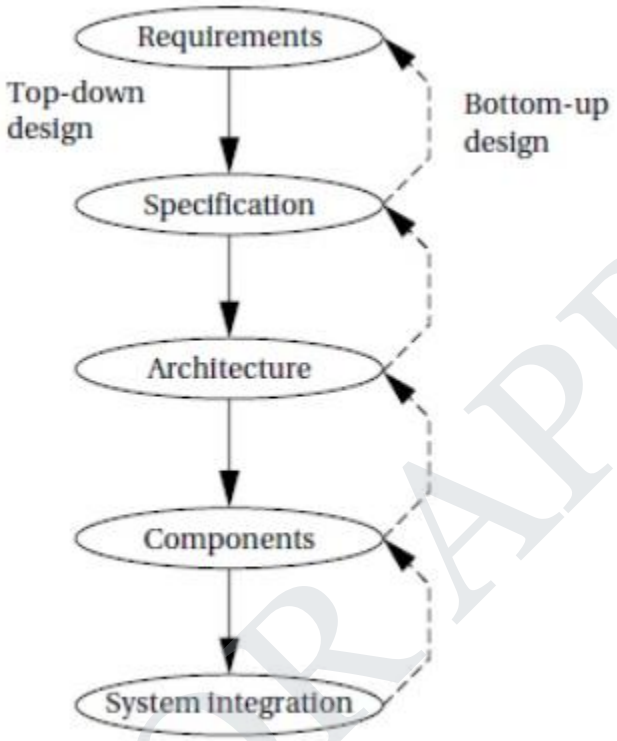
UNIT I INTRODUCTION TO EMBEDDED SYSTEMS	
Introduction to Embedded Systems – The build process for embedded systems-Structural units in Embedded Processor , selection of processor & memory devices - DMA – Memory management methods-Timer and Counting devices, Watchdog Timer, Real Time Clock, In circuit emulator, Target Hardware Debugging.	
Part*A	
Q.No	**Question**
1.	**List out the challenges in building embedded system.(May/June 2016, Nov/Dec 2016)**BTL1 • Clock rate reduction • Voltage reduction • Wait • Stop & cache
2.	**What is the need of Watch Dog timer? .(May June 2016)**BTL1 A Watch dog timer is an additional timer that does a monitoring job and resets the system, if necessary. Most embedded systems are expected to self- reliant, when the software is detected to be malfunctioning the best way is to reset and start again.
3.	**What are the steps involved in build processor? (Nov/Dec 2016)(April/May 2017)** BTL1 • Processor in an embedded system • Microprocessor • Micro controller • Special purpose processor
4.	**Whatisanembeddedsystem?**BTL1 Anembeddedsystememploysacombinationofhardware&software(a"computationalengine")toperformaspecificfunction;ispartofalargersystemthatmaynotbea"computer"worksinareactiveandtime-constrainedenvironment.
.5.	**What arethetypicalcharacteristics ofanembedded system? (April/May 2017)** BTL1 Typicalcharacteristics: Performasingleortightlyknitsetoffunctions. Increasinglyhigh-performance&real-timeconstrained,Power,costandreliabilityareoftenimportantattributesThatinfluencedesign. Application specificprocessor design can be a significant component of someembeddedsystems. Othercharacteristics: • Applicationspecific • DigitalsignalprocessinginECS • Reactive • Real-time • Distributed

6.	**What arethe advantages of embedded system?** BTL1 • User friendly environment. • Customizationyieldslowerarea • Power • Cost
7.	**What arethe disadvantages of embedded system**? BTL1 Disadvantages: HigherHW/softwaredevelopmentoverheaddesign,compilers,debuggers,etc.,mayresultin delayedtimetomarket.
8.	**Whatarethecomplicatingfactorsinembeddeddesign?**BTL1 • Complicatingfactorsinthedesignofembeddedsystems • Manyofthesubtasksindesignareintertwined. • Allocationdependsonthepartitioning,andschedulingpresumesacertainallocation. • Predictingthetimeforimplementingthemodulesinhardwareorsoftwareisnotvery easy,Particularlyfortasksthathavenotbeenperformedbefore.
9.	**Whataretheapplicationsofanembeddedsystem?** BTL1 **EmbeddedSystems:Applications:** • Consumerelectronics,e.g.,cameras,camcorders. • Consumerproducts,e.g.,washers,microwaveovens. • Automobiles(anti-lockbraking,enginecontrol. • Industrialprocesscontrollers&avionics/defenseapplications. • Computer/Communicationproducts,e.g.,printers,FAXmachines. • Emergingmultimediaapplications&consumerelectronics.
10.	**Whatarethereal-timerequirementsof anembeddedsystem?**BTL1 Hard-realtimesystems:wherethereisahighpenaltyformissingadeadlinee.g.,control systemsforaircraft/space probes/nuclearreactors;refresh ratesforvideo,orDRAM. Softreal time systems:wherethereisasteadilyincreasingpenaltyifadeadlineismissed. e.g.,laserprinter:ratedbypages-per-minute,butcantakedifferingtimestoprintapage (dependingonthe\"complexity\"ofthepage)withoutharmingthemachineorthecustomer.
11.	**Whatarethefunctionalrequirementsofembeddedsystem?**BTL1 • DataCollection • Sensorrequirements • Signalconditioning • Alarmmonitoring
12.	**Whatarethemaincomponentsofanembeddedsystem?**BTL1 Threemaincomponentsofembeddedsystems: • Hardware • ApplicationSoftware • RTOS
13.	**Defineembedded microcontroller.**BTL2 Anembeddedmicrocontrollerisparticularlysuitedforembeddedapplicationstoperform

	dedicatedtaskoroperation. Example:68HC11xx,8051,PIC,16F877.
14.	**Explaindigitalsignalprocessinginembeddedsystemcontinueddigitizationofsignals increasingtheroleofDSPinES.**BTL2 • Signalsarerepresenteddigitallyassequenceof"samples" • ADC'saremovingclosertosignals
15.	**Whatarethevariousclassificationsofembeddedsystems?**BTL1 • Smallscaleembeddedsystems • Mediumscaleembeddedsystems • Sophisticatedembeddedsystems
16.	**Whatarethetwoessentialunitsofaprocessoronanembeddedsystem?**BTL1 • Programflowcontrolunit(CU) • Executionunit(EU)
17.	**Giveexamplesforgeneralpurposeprocessor.**BTL1 • Microprocessor • Microcontroller • Embeddedprocessor • DigitalSignalProcessor • MediaProcessor
18.	**Definemicroprocessor.** BTL2 Amicroprocessorfetchesandprocessesthesetofgeneral-purposeinstructionssuchasdata transfer,ALUoperations,stackoperations,I/Ooperationsandotherprogramcontroloperations.
19.	**WhenisApplicationSpecificSystemprocessors(ASSPs)usedinanembeddedsystem?**BTL2 AnASSPisdedicatedtoreal-timevideoprocessingapplicationssuchasvideoconferencing,videocompressionanddecompressionsystems.Itisusedasanadditionalprocessingunitfor runningapplicationspecifictasksintheplaceofprocessingusingembedded software.
20.	**Whatistheneedfor LCDandLEDdisplays?** BTL1 UsesofLCDand LEDdisplay: • Itisusedfordisplayingandmessaging. • Example:Trafficlightstatusindicator,remotecontrols, signals,etc., Thesystemmustprovidenecessarycircuitandsoftwarefortheoutputto LCD controller
21.	**Explain distributedsystems.**BTL2 • Consistofcomponentsthatmaynecessarilybephysicallydistributed. • Consistofcommunicatingprocessesonmultipleprocessorsand/ordedicatedhardware **Motivation:** • Economical multipleprocessorstohandlemultipletimecriticaltasks physicallydistributed • Devicesundercontrolmaybephysicallydistributed.
22.	**What arethe temporal requirements?**BTL1 • Tasksmayhavedeadlines • Minimallatencyjitter • Minimalerrordetectionlatency • Timingrequirementsduetotightsoftwarecontrolloops

	• Humaninterfacerequirements
Part*B	
Q.No	**Question**
1.	**Explain in detail about the build process for embedded systems. (May/June 2016, Nov/Dec 2016, April/May 2017).** (13M) BTL1 **Answer: Page 1.3 –Dr.G. Ramprabu** • **Diagram (5M)** • **Explanation (8M)** • Heart of the embedded system – important unit  • Two units – Program Flow unit and Execution unit • Program Flow unit – Fetch unit • Execution unit – implementing instruction – ALU Types of Embedded system Processor • General purpose processor – Microprocessor and embedded processor • Application Specific instruction processor – Micro controller , DSP , Network processor • Single purpose processor as additional processor – coprocessor, Accelerator , controllers • Programable logic device • Application Specific system processor • Multicore processor or multiprocessor
2.	**Explain the Design processor for Embedded system. (13M)** BTL1 **Answer: Page 1.14 –Dr.G. Ramprabu**

- **Diagram (6M)**
- **Explanation (7M)**



Requirements :
- General description – idea od product – list of ideas

Types
- functional requirement - input and out put devices
- non-functional requirement – cost, performance, physical size and weight, power conception

Specifications
- Behavior of product
- Global characteristics

Architecture design
- Plan for overall product
- Design the components first phase
- Refining the hardware

Design software and hardware design
- Architectural design tells the components
- Components include both hardware and software
- Components – ready made

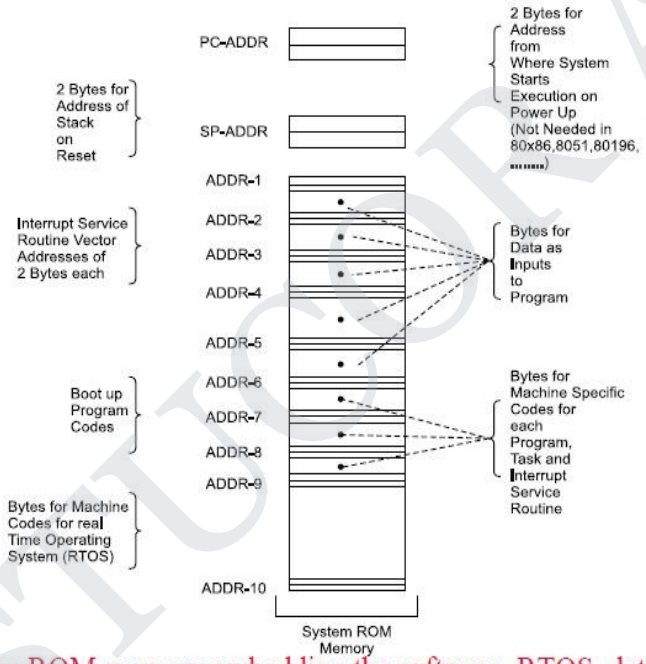
System Integration
- Plugging everything together – software and hardware

	• Debug modules at that time • Good planning will help the system integration • Make – user friendly
3.	**Discuss in detail about the timer and counter devices in Embedded System . (13M)** BTL4 **Answer: Page 1.30 – Dr.G. Ramprabu** • **Timer (7)** • **Counter (6)** TIMINGAND COUNTINGDEVICES TIMER • Timer - which counts the input at regular interval (δT)using clock pulses- inputs • Thecounts increment on each pulse and storeina register,calledcount register • EvaluationofTime- counts multiplied bythe interval δTgive thetime. • The(present counts −initial counts) ×δT intervalgives the timeinterval between two instances when present count bits arereadand initial counts were read or set. • Has an input pin (ora control bit in control register)for resettingit forall count bits = 0s. • Has an output pin (orastatus bit in status register) foroutput whenall count bits = 0s • after reachingthe maximum value, which also means aftertimeout oroverflow Counter • A device, which countsthe input due to the events at irregularor regular intervals. • Thecountsgives the numberof input events or pulses sinceit waslast read. • Has a register to enableread of present counts • Functions - timerwhen counting regularinterval clock pulses • Has an input pin (ora control bit in control register)for resettingit forall count bits = 0s. • Has an output pin (orastatusbit in status register) foroutput whenall count bits = 0s .
4.	**Explain the following Embedded Hardware Units** **(i) Watch Dog Timer (5M)** **(ii) Memory (4M)** (iii) **Input/ Output-Port (4M). (April/May 2017)** BTL2 **Answer: Page 1.32, 1.29–Dr.G. Ramprabu** (i) Watchdog timer. • Baud orBit Rate Control forserial communication on alineor network. • Timer timeout interrupts definethe timeof each baud • Input pulsecountingwhen using a timer, which is ticked by givingnon periodicinputs instead of theclock inputs. • Thetimeracts asa counterif, in placeof clock inputs, the inputs -given to the timerfor each instanceto be counted. • Schedulingof various tasks. • A chain of software-timers interrupt and RTOS uses these interrupts to schedule the tasks. • Timeslicingof various tasks.

- A multitasking- multi-programmed operatingsystem presents theillusion that multiple tasks
- programs are running simultaneouslyby switchingbetween programs veryrapidly, forexample, after every16.6 ms.

(ii) Memory

- ROM - read only memory- isnon-volatile memory used for program information - permanent data.
- Themicrocontroller usesROM memory spaceto store program instructionsit will execute when it is started or reset.
- Program instructions must be saved in non-volatile memory so that they are not affected by loss of power.
- The microcontroller usually cannot write data to program memory space.
- RAM- random access memory - used to write and read data values as a programruns.
- RAM - volatile: if you remove the power supply its contents are lost.
- Any variablesused ina program are allocated fromRAM.



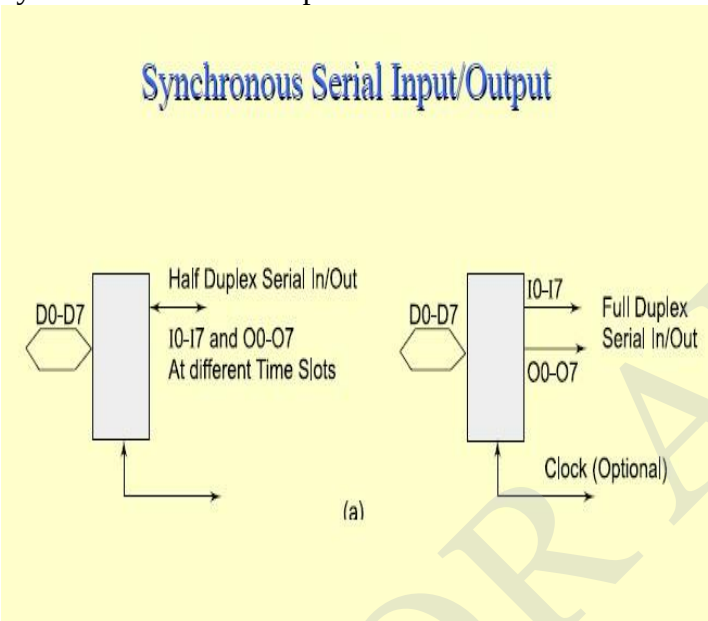
System ROM memory embedding the software, RTOS, data, and vector addresses

- Arrangement at each ROM address of bytes for instructions and data.
- Bytes at eachaddress defined for creating the ROM image.
- thesamehardware plat formworkdifferently
- Canbeused for entirelydifferent applications or fornew upgradesofthe same system.
- Hardwareelementsbetweenthe distinctsystemscanbeidentical.
- Softwarethatmakes asystem uniqueand distinct from theother..
- CompressedCodesandDataROMimagemayalternativelybecompressedsoftware.

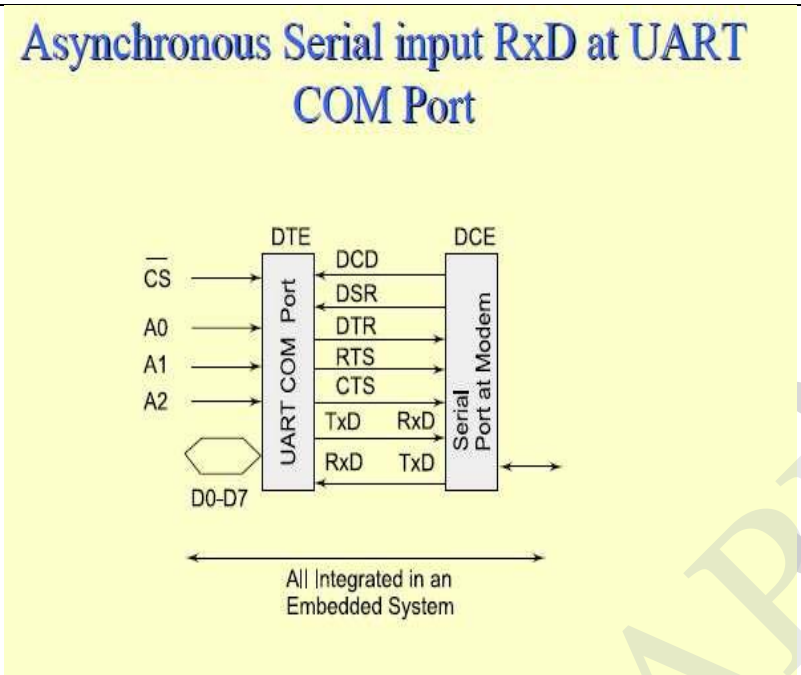
(iii) IO Port Types

Types of Serial ports

- Synchronous SerialInput
- Synchronous Serial Output



- Each bit in each byteis in synchronization at input
- Each bit in each byteis in synchronization atoutput with themasterclock output.
- Asynchronous Serial UART input
- Asynchronous Serial UART output(both as inputand as output, forexample ,modem.)



- Does not receive theclock pulses or clockinformation alongwith thebits.
- Each bit is received in each byte at fixed intervalsbut each received byte-not in synchronization.
- Bytes separate bythevariableintervals or phasedifferences.
- Asynchronous serial input also called UART inputif serial input - accordingto UART protocol

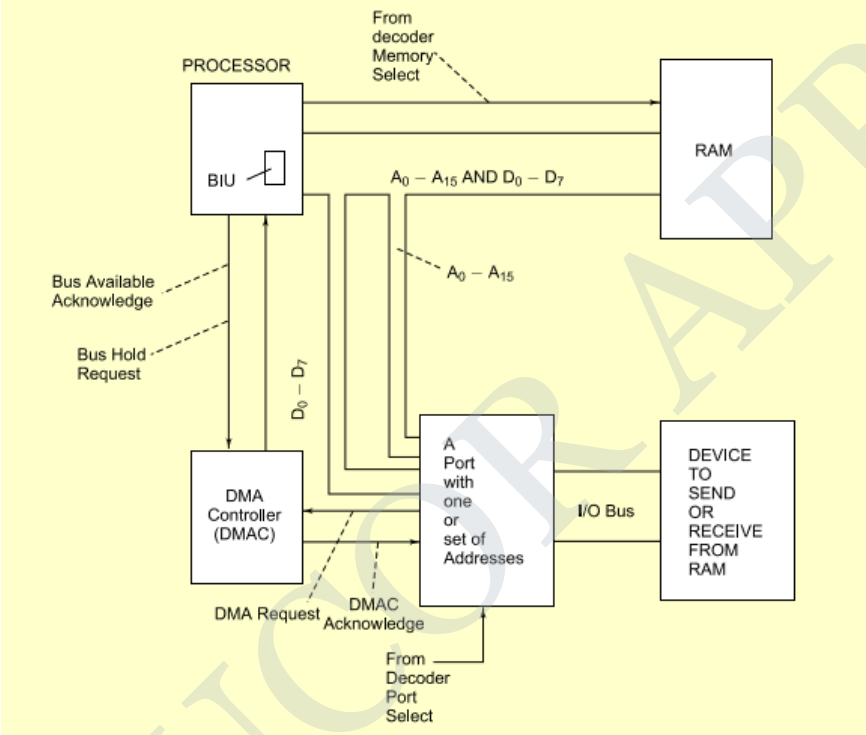
5. **Briefly explain about Real Time Clock .(13M)** BTL1
Answer: Page 1.34–Dr.G. Ramprabu

- which is based on the interrupts at preset intervals.
- interrupt service routine executes on each timeout.
- once started never resets or never reloaded with another value.
- Used in a system to save the time and date.(6M)
- each tick (interrupt) a service routine runs and updates at a memory location.
- Within 256 days there will be 232 ticks
- A battery is used to protect the memory for long period.
- Interrupt service routine can generate a port bit output after every time it runs.
- RTC disabled or enabled by the I bit in 8/

Part*C

Q.No	Question
1.	**With a neat diagram explain the working of Direct Memory Access (DMA) with architecture and timing diagram. (May/June 2016, Nov/Dec 2016, April/May 2017)(15M)**BTL2 **Answer: Page 1.23 –Dr.G. Ramprabu** • **Diagram (6M)**

- **Explanation (9M)**
- required when a multi-byte data set or a burst of data.
- a block of data is to be transferred between the external device and system or two systems.
- A device facilitates DMA transfer with a processing element
- Repeatedly interrupting the processor for transfer of every byte during bulk transfer of data.
- System performance improves by separate processing of the transfers from and to the peripherals.



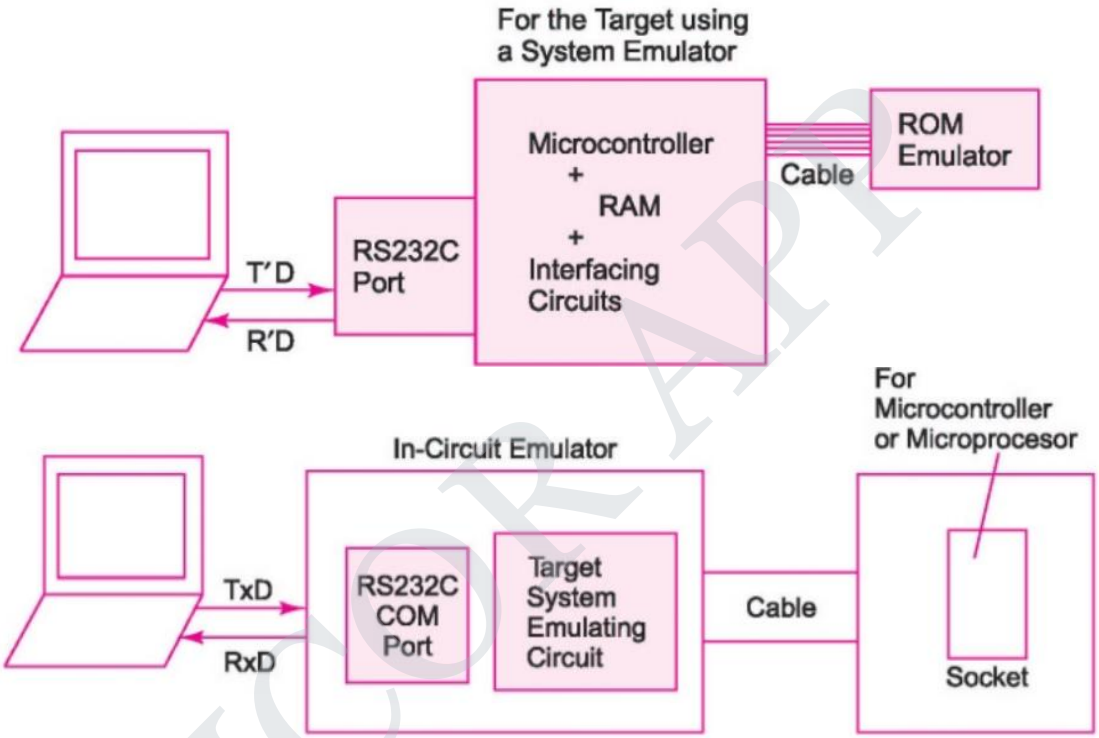
- DMAC hold request - After an ISR initiates - DMAC sends a hold request to the CPU
- CPU acknowledges that if the system memory buses are free to use
- Three modes - Single transfer at a time-Burst transfer at a time- Bulk transfer and then release of the hold
- DMA proceeds without the CPU intervening.
- An ISR is called only at the beginning of the transfer to program the controller (DMAC)
- Another ISR is called only at the end of the transfer.
- Separate set of registers for programming each channel.
- Separate interrupt signals in the case of a multi-channel DMAC.

system memory with the least processor intervention using DMAC.

2. **Explain (i) In circuit Emulator (8M)**
(ii)Target Hardware Debugging. (7M)BTL2
Answer: Page 1.35, 1.37 –Dr.G. Ramprabu

In-Circuit Emulator

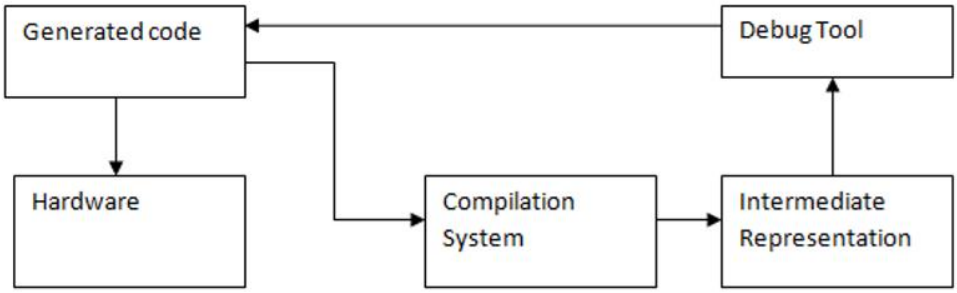
- Circuit for emulating target system remains independent of a particular targeted system and processor
- emulating target system remains independent of a particular targeted system and processor
- ICE provides great flexibility
- ease for developing various applications on a single system in place of testing that multiple targeted systems



- Emulates controlled outputs for the peripheral interfaces/systems.

Target Hardware Debugging

- A Debugger or debugging tool - a computer program that is used to test and debug other programs.
- should be running on an instruction set simulator to identify the fault.
- The debugger can be used to identify if the program - running correctly
- a source-level debugger, the debugger can show the actual position in the original code
- the run time errors as in general software development
- Real-time analysis - following code flow in real time with real-time trace analysis
- Memory substitution - replacing ROM-based memory with RAM for rapid and easy code download, debugand repair cycles

	
3.	**Discuss about the structural units in embedded processor and how a processor is selected for an embedded system application. (May/June 2016, Nov/Dec 2016)(15M)**BTL4 **Answer: Page 1.11, 1.19–Dr.G. Ramprabu** Selection of the processor (7M) • Can operate at higher speed • More instruction per second • High computing performance • Context switching with multi tasking system • Burst mode accesses external memories fast, reads fast and write fast • Special programming skills Selection of memory devices (8M) • Actual memory requirement – known after the design • ROM and RAM allocation for various segments • Prior estimation of the memory • Available memory are 1 KB , 4KB ,16 KB, 32KB etc • 100KB needed 128 KB is chosen

UNIT II EMBEDDED NETWORKING	
Embedded Networking: Introduction, I/O Device Ports & Buses– Serial Bus communication protocols - RS232 standard – RS422 – RS485 -CAN Bus -Serial Peripheral Interface (SPI) – Inter Integrated Circuits (I2C) –need for device drivers.	
Part*A	
Q.No	**Question**
1.	**How SPI is differed from other serial interfaces ?(May/June 2016)** BTL1 • The master sends a bit on the MOSI line which the slave reads from the same line. • The slave sends a bit on MISO line and the master reads it from that same line.

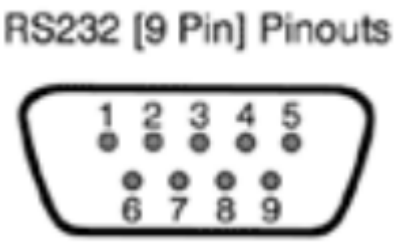
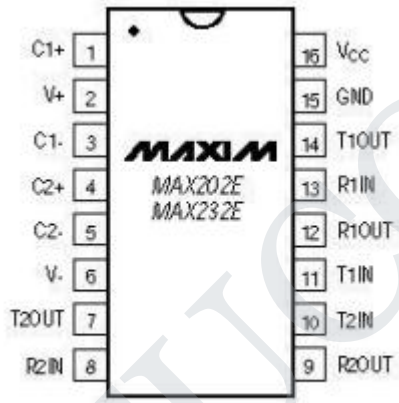
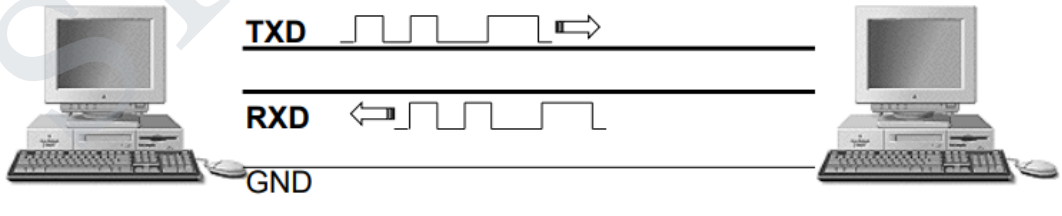
2.	**What is the need of Device drivers?(May/June 2016, Nov/Dec 2016)** BTL1 • It is a software interface to hardware device that handle request from the kernel regarding the use of the particular I/O devices • There is a well defined interface for the kernel to make these request, because of this adding new device is easy.
3.	**Mention the features of CAN.(Nov/Dec 2016, April/May 2017)** BTL1 • When common mode signals, usually noise appear on the bus they are subtracted OOF and this makes the CAN bus resistance to noise. • In CAN and most other modern serial protocols differential signaling with NRZ coding ids used to reduce the effect of noise.
4.	**Mention some serial bus communication protocols.(April/May 2017)** BTL1 • I^2C Bus • CAN Bus • Universal serial Bus
5.	**Definebus.** BTL1 Buses:Theexchangeofinformation. Informationistransferred betweenunits of themicrocomputer by collections of conductorscalledbuses.Therewillbeone conductorforeachbitofinformationtobepassed,e.g.,161inesfora16bitaddressbus.Therewillbeaddress,control,anddatabuses.
6.	**Whataretheclassificationsof110devices?** BTL1 • Synchronousserialinputandoutput • AsynchronousserialUARTinputandoutput • Parallelonebitinputand output • .Parallelportinputandoutput
7.	**Givethestepsforaccomplishinginput outputdatatransfer .** BTL1 -Accomplishing input/outputdatatransfer Therearethreemainmethodsusedtoperform/controlinput/outputdatatransfers.Theyare, • Softwareprogramming(scanningorpolling) • interruptcontrolled • Directmemoryaccess(DMA)
8.	**Givethelimitations ofpollingtechnique.** BTL1 Thepollingtechnique,however,haslimitations. Itiswastefuloftheprocessorstime,asitneedlesslychecksthestatusofalldevicesallthetime.Itisinherentlyslow,asitchecksthestatusofallI/Odevicesbeforeitcomesbacktocheck anygivenoneagain.Whenfastdevicesareconnectedtoasystem,pollingmaysimplynotbefastenoughto satisfytheminimumservicerequirements.Priorityofthedeviceisdetermined.
9.	**Whatdo youmeantbybusarbitration?** BTL1 BusArbitration Mostprocessorsusespecialcontrollinesforbusarbitration,ie,controllingtheuseofthe addressanddatabus, • AninputwhichtheDMACusestorequestthebus • Anoutput(s)indicatingthebusstatus • AnoutputindicatingacceptanceoftheDMAC\'sbusrequest
10.	**Whatarethetwocharacteristicsofsynchronouscommunication?** BTL1

	Bytes/framesmaintainconstantphasedifferenceandshouldnotbesentatrandom timeintervals.Nohandshakingsignalsareprovidedduringthecommunication.Clockpulseisrequiredtotransmitabyteorframeserially.Clockrateinformationis transmittedbythetransmitter.
11.	**Whatdoyoumeanbyasynchronouscommunication?** BTL1 Themostbasicwayofsharingdataisbycopyingthedatainquestiontoeachserver.Thiswill onlyworkifthedataischangedinfrequently andalways by someone with administrativeaccesstoalltheserversinthecluster.
12.	**Whatarethecharacteristicsofasynchronouscommunication?** BTL1 • Variablebitrate-neednotmaintainconstantphasedifference • Handshakingmethodisused • Transmitterneednottransmitclockinformationalongwithdatabitstream
13.	**Whatarethethreewaysofcommunicationforadevice?** BTL1 • Separateclockpulsealongwithdatabits • Databitsmodulatedwithclockinformation • Embeddedclockinformationwithdatabitsbeforetransmitting
14.	**WhatarethefeaturesofSPI?(Nov/Dec 2016)** BTL1 • SPIhasprogrammableclockrates • Full-duplexmode • Crystalclockfrequencyis8MHz • Opendrainortotem poleoutputfrommastertoslave
15.	**Definesoftwaretimer.** BTL1 Asoftwaretimeris softwarethatexecutes theincrease/decreasecountvalueonaninterruptfrom timerorRTC.Softwaretimerisusedasvirtualtimingdevice.
16.	**Whataretheformsoftimer?** BTL1 • Hardwareinterrupttimer • Softwaretimer • Usersoftwarecontrolledhardwaretimer • RTOScontrolledhardwaretimer • UP/DOWNcountactiontimer • One-shottimer(Noreloadafteroverflowandfinishedstates)
17.	**DefineRTC.** BTL1 RTCStandsforRealTimeSystems.Once thesystemstarts,donotstop/resetandthe countvaluecannotbereloaded.
18.	**WhatarethefeaturesoftheUSBprotocol?** BTL1 Adevicecanbeattached,configuredandused,reset,reconfiguredandused,detachedand reattached,sharethebandwidthwithotherdevices.
19.	**WhatarethefourtypesofdatatransferusedinUSB?** BTL1 • Controlledtransfer • Bulktransfer • Interruptdrivendatatransfer • Iso-synchronoustransfer
20.	**What are the function of PCIandPCI/Xbuses?** BTL1 • UsedformostPCbasedinterfacing. • ProvidessuperiorthroughputthanEISA

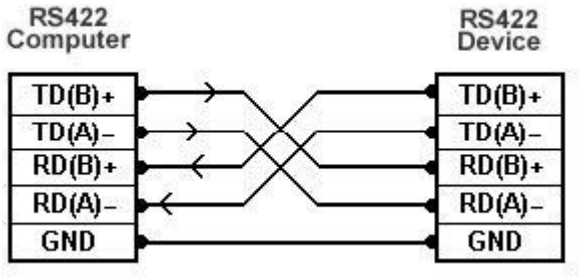
	• Platform-independent • Clockrateisnearesttosub-multiplesofsystemclock
21.	**Mentionany two advantage ofbusstandardprotocols.** BTL2 • GMII(GigabitEthernetMACInterchangeInterface) • XGMI(10GigabitEthernetMACInterchangeInterface)
22.	**Whatdoyoumeantbyhighspeeddeviceinterfaces?** BTL1 Fail-overclusteringwouldnotbepracticalwithoutsomewayfortheredundantserversto accessremotestoragedeviceswithouttakingalargeperformancehit,aswouldoccuriftheseedevices weresimplylivingonthelocalnetwork.Twocommonsolutionstothisproblemaredouble-ended SCSIandfibre-channel.
23.	**MentionsomeI/Ostandardinterfaces.** BTL2 HSTL-HighSpeedTransceiverLogic(Usedinhighspeedoperations) SSTL-StubSeriesTerminatedLogic(Usedwhenthebusesareneededtoisolatefromthe largeno.ofstubs)
24.	**GivesomeexamplesforserialinputI/Odevices.** BTL1 Audioinput,videoinput,dialtone,transceiverinput,scanner,serialIObusinput,etc.,

Part*B

Q.No	Question
1.	**Describe one type of serial communication bus with its communication protocol.(13M)**BTL2 **Answer: Page 2.7 –Dr.G. Ramprabu** • **Diagram (6M)** • **Explanation (7M)** SERIAL BUSCOMMUNICATION PROTOCOLS– USB Three types • Simplex connection – data flow in only one direction – computer to printer • Half duplex connection – data flow in one direction or other but not in same time – walkie talkie • Full duplex connection – data travel – both direction simultaneously – mobile phone • Transmission rate - unit – bits per second bps • Synchronous bus – clocked -very fast – interface logic small • Asynchronous bus – not clocked – lengthened USBHost Applications Connecting • flash memorycards, • pen-likememorydevices, • digital camera, • printer, • mouse-device, • Pocket PC, • video games, Universal Serial Bus (USB) • Serial transmission and reception between host and serial devices

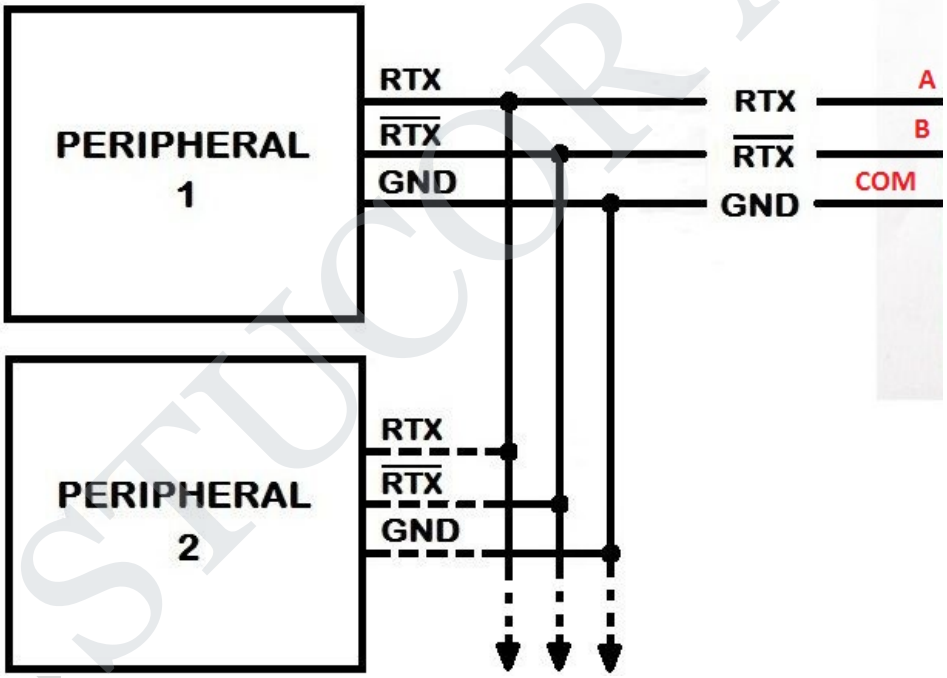
	• Thedata transfer - four types:(a)Controlleddata transfer, (b) Bulk data transfer, (c)Interrupt driven data transfer,(d) Iso-synchronous transfer • A bus between thehost system and interconnected numberof peripheral devices
2.	**Explain the RS 232, RS 422 and RS 485 protocol. (May/June 2016, April/May 2017)** (13M) BTL1 **Answer: Page 2.8–Dr.G. Ramprabu** • **Diagram (6M)** • **Explanation (7M)**  • Signals between +25V and -25V; some say ±15V usually +12V to -12V   • 1 start bit - 8 data bits -1 stop bit (optional 1 parity bit) • serial connection historically found on IBM-compatible PCs • connecting a mouse, printer, or modem, as well as industrial instrumentation • 1Driver-1 Receiver

RS-422



- serial connection historically used on Apple Macintosh computers
- uses a differential electrical signal, as opposed to unbalanced signals referenced to ground with the RS-232
- uses two lines each for transmit and receive signals which results in greater noise immunity
- better fit for industrial applications.
- 1Driver-10 Receivers

RS-485



- devices from 10 to 32 and defines the electrical characteristics necessary to ensure adequate signal voltages under maximum load
- enhanced multi-drop capability
- 32Drivers*- 32 Receivers

3. **Explain I/O Device ports and its characteristics.(13M)**BTL2
Answer: Page 2.2–Dr.G. Ramprabu

- **Diagram (6M)**
- **Explanation (7M)**

I/O Device ports

The largest, most complex subsystem – OS

Block transfer

- Disk, tape, CD, DVD
- Network
- Clocks
- Internal, external
- Graphics
- GUI, games

Multimedia

- Audio, video

Other

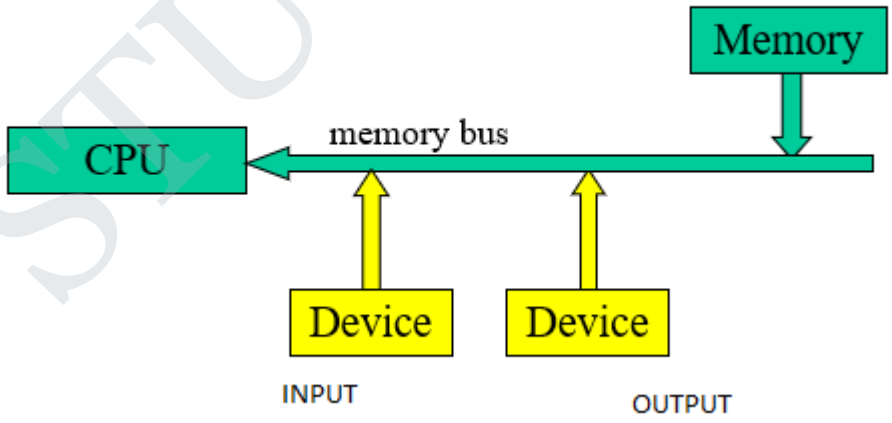
- Sensors, controllers

Special I/O instructions

- Opcode to stop, start, query, etc.
- Separate I/O address space
- Kernel mode only

Memory-mapped I/O control registers

- Each register has a physical memory address
- Writing to data register is output
- Reading from data register is input
- Writing to control register causes action
- Can be mapped to user-level virtual memory



4.	**What are the device drivers in embedded systems and explain its types.(13M)**BTL1

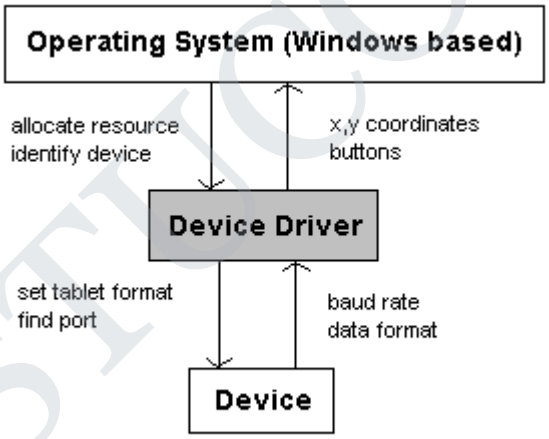
Answer: Page 2.33–Dr.G. Ramprabu

- **Diagram (6M)**
- **Explanation (7M)**

- Software interface to hardware device
- Well defined interface – kernel
- Adding new device – easy

Classification

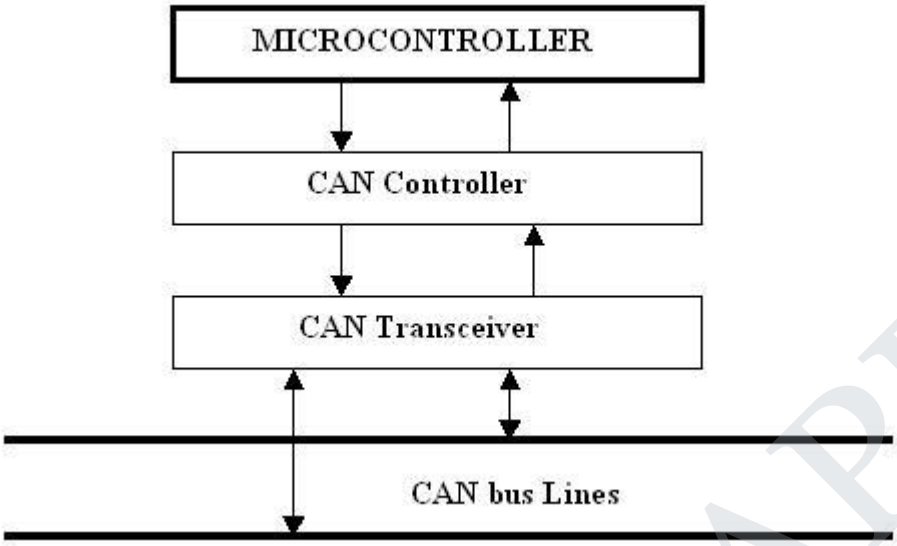
- Block device drive – transfer data as blocks – disk drive – block sized buffer
- Character device drive – data as character – does not need buffer – line printer
- Network device drive – data transmission and reception – wifi devices
- A device driver has a set of routines (functions) - high-level language programmer.
- Does the interaction with the device hardware - sends control commands to the device - communicates data to the device.
- ISR relates to a device driver command.
- Programmer uses generic commands for device driver for using a device.
- Device drivers - Different in different operating system.
- Same device may have different code for the driver when system - using different operating system.
- Same device may have different code for the driver when system - using different operating system.
- Device driver can be considered software layer between an application program and the device



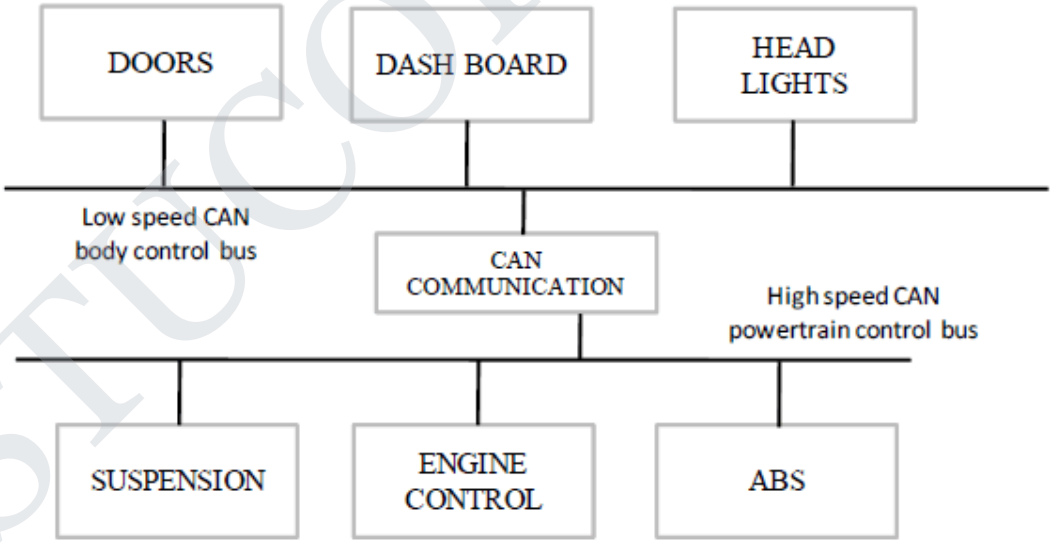
Part*C

Q.No	Question
1.	**Summarizeshort notes on CAN BUS. (May/June 2016)**(15M) BTL2 **Answer: Page 2.17 –Dr.G. Ramprabu** • **Diagram (7M)**

- **Explanation (8M)**



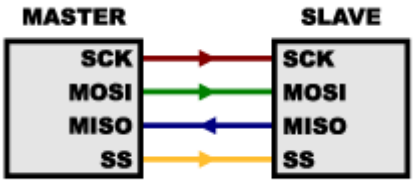
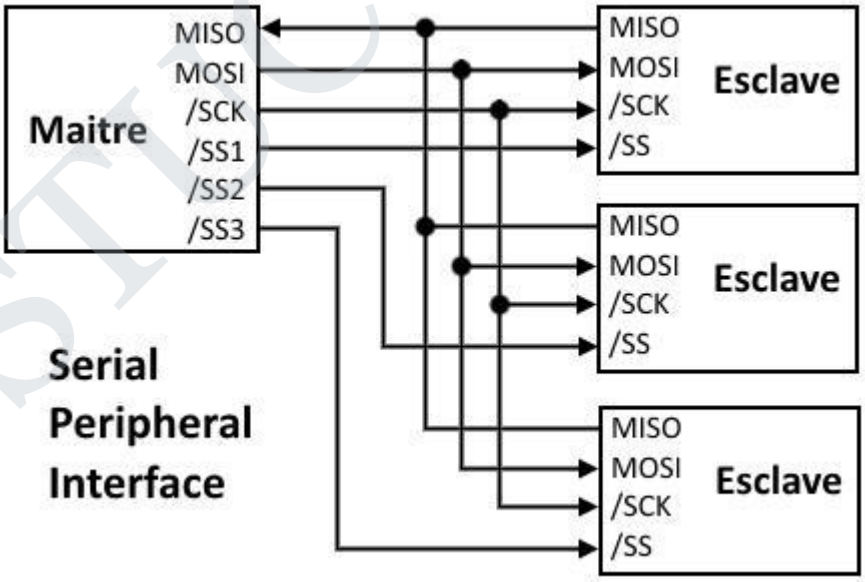
- Control AreaNetwork example -anetwork of embedded systems in automobile
- Developed by Bosch company – reduce the wiring inside the vehicle – 1984
- 1990 – very popular
- gives the input andgets output between the physical and data link layersat thehost node

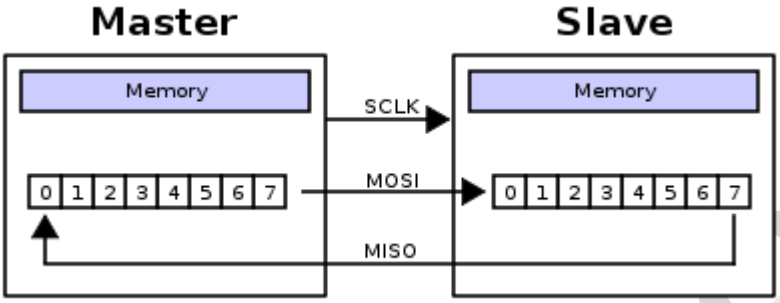
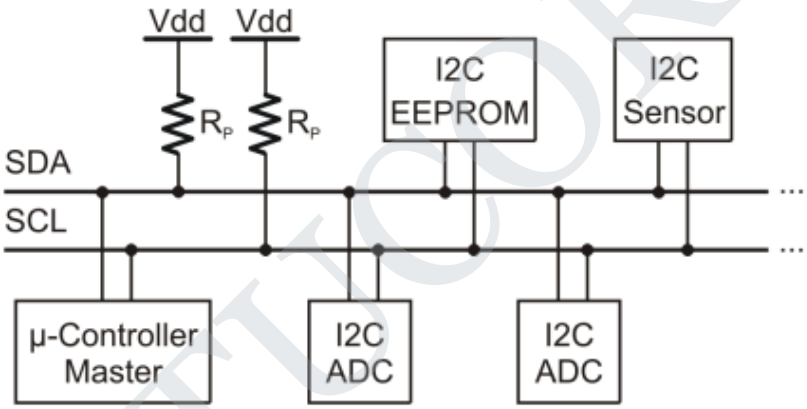


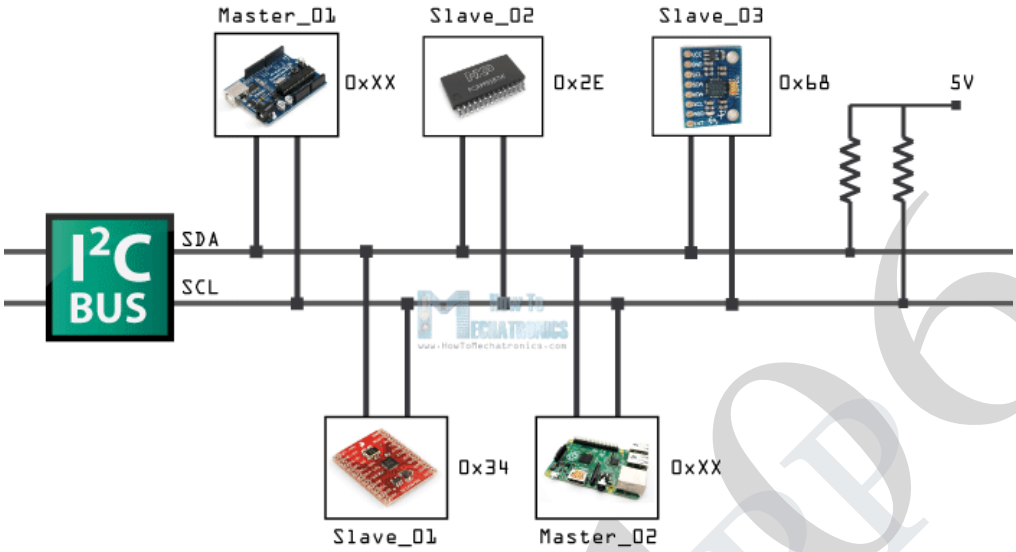
- bus interfaceunit consistingof buffer and driver
- Interconnecting networks

Threestandards:

- low speed CAN – 125 Kbps -11 bit identifier – door lock, seat control, climate control,
- standard CAN 2.0A – 1 Mbps - 11 bit identifier
- Extended CAN 2.0B – 1 Mbps- 29 bit identifier

	• Interconnecting buses is called bridges • Reliable, low cost and efficient • Message based protocol • Four message format – data, remote, error and overland frames
2.	**Explain the Serial Peripheral Interfacing SPIbus.(May/June 2016, Nov/Dec 2016)(15M)**BTL2 **Answer: Page 2.24–Dr.G. Ramprabu** • **Diagram (6M)** • **Explanation (9M)**  • Developed by Motorola • Synchronous and full duplex • Single master multi slave system • Only one slave – enabled at a time • Signals – MOSI , MISO,SCLK and SS • MOSI – master out slave in • MISO – master in slave out  • Large shift registers – shared between master and slave

	• Clock – shared by two devices • Ring buffer – master sends a byte to slave and slave send a byte to master • MOSI – slave reads the message. • MISO – master reads the message.  • Higher speed – no acknowledgement signal • Works best for single slave system
3.	**Explain in briefly about Inter Integrated Circuit (I2C)protocol.(May/June 2016, Nov/Dec 2016, April/May 2017)(15M)** BTL2 **Answer: Page 2.29–Dr.G. Ramprabu** • **Diagram (6M)** • **Explanation (9M)**  • Developed by Philips – 1980 – TV application • I2C – I^2C • Widely used in embedded system • Synchronous, half duplex, serial protocol and byte oriented • Acknowledgement – to slave



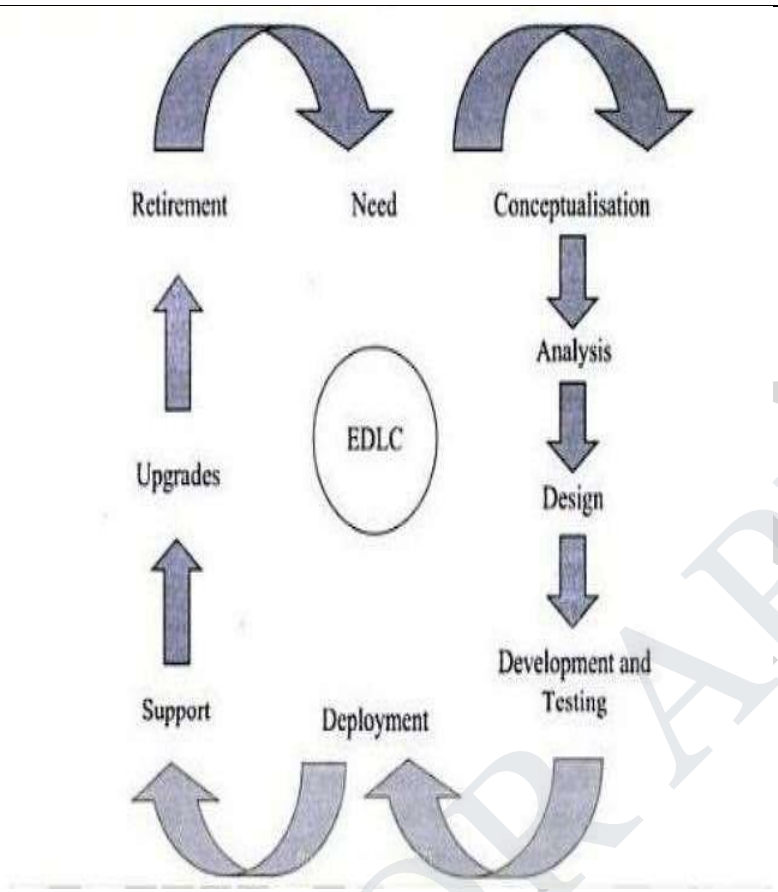
- ROM read only – LCD
- Mater – MCU – micro controller Unit
- Slave – input – output devices
- SDA – serial data – each devices carry its unique address
- Slow – under 100 Kbps
- Fast – 400 Kbps
- High speed – 3.4 Mbps
- Very simple – not fast
- Include EPROM, thermal sensor and real time clock
- Easily – implemented protocol

UNIT III EMBEDDED FIRMWARE DEVELOPMENT ENVIRONMENT	
Embedded Product Development Life Cycle-objectives, different phases of EDLC, Modelling of EDLC; issues in Hardware-software Co-design, Data Flow Graph, state machine model, Sequential Program Model, concurrent Model, object oriented Model.	
Part*A	
SL No.	**Questions**
1.	**Mention different models used for the development of an embedded system.(May/June 2016)** BTL1 • Linear or Waterfall Model • Iterative/Incremental or Fountain Model • Prototyping/Evolutionary Model • Spiral Model
2.	**What are the processes involved in Co design ? (May/June 2016, Nov/Dec 2016)** BTL1 • Selecting the model

	• Selecting the architecture • Selecting the process • Partitioning system requirements into hardware and software.
3.	**What is state machine model?(Nov/Dec 2016, April/May 2017)** BTL1 The state machine model describes the system behavior with States, Events, Actions and Transitions. The state machine model is used for modeling reactive or event-driven embedded systems whose processing behavior are depend on the transitions.
4.	**List the different phases of EDLC.(April/May 2017)** BTL1 • Need • Conceptualization • Analysis • Design • Development and testing • Deployment • Support • Upgrades • Retirement/Disposal
5.	**WhataretheadvantagesofAssemblylanguage?** BTL1 • Itgives theprecise control of theprocessor internal devices • Fulluseof processorspecificfeaturesinitsinstructionsetsandaddressingmodes. • Themachinecodesarecompact,whichrequiresonlysmallmemory. • Devicedriversneedonlyfewassemblyinstructions.
6.	**Whatareadvantages ofhighlevellanguages?**BTL1 • Datatypedeclaration • Typechecking • Controlstructures • Probability ofnon-processorspecificcodes
7.	**DefineIn-lineassembly**BTL1 Insertinganassemblycodeinbetweenthe processes of a system issaidtobein-lineassembly.
8.	**MentiontheelementsofCprogram.**BTL1 • Headerfiles • Sourcefiles • Configurationfiles • Preprocessor directives • Functions: • Macrofunction • Mainfunction • Interruptserviceroutinesordevicedrivers • Others: • Datatypes • Datastructures • Modifiers • Statements

	• Loopsandpointers
9.	**What istheuseofMACROfunction?** BTL1 Amacrofunctionexecutesanamedsmallcollectionofcodes,withthevaluespassed bythecallingfunctionthroughitsarguments.Ithasconstantsavingandretrievingoverheads.
10.	**Whatistheuseofinterruptservice routinesordevicedrivers?** BTL1 Itisusedforthedeclarationoffunctionsanddatatypes,typedefandexecutesnamed setofcodes.ISRmustbesmall(short),reentrantormusthavesolutionforshareddataproblem.
11.	**WhatarethedatatypesavailableinClanguage?** BTL1 Char-8bit;byte-8bit;short-16bit;unsignedshort-16bit;unsignedint-32bit;int-32bit;longdouble-64bit;float-32bit;double-64
12.	**Mention thedatastructuresavailableinClanguage.** BTL1 • Queue • Stack • Array(1-dimentionalandmulti-dimensional) • List • Tree • Binary-tree
13.	**WritethesyntaxfordeclarationofpointerandNull-pointer.** BTL2 Syntaxforpointer: void*portAdata SyntaxforNull-pointer: #defineNULL(void*)0x0000
14.	**Explainpassbyvalues.** BTL2 • Thevaluesarecopiedintotheargumentsofthefunction. • Calledprogramsdoesnotchangethevaluesofthevariables
15.	**Definequeue.** BTL1 • Astructurewithaseriesofelements. • UsesFIFOmode. • Itisusedwhenanelementisnotdirectlyaccessedusingpointerandindexbut onlythroughFIFO. • Twopointersareusedforinsertionanddeletion.
16.	**Definestack.** BTL1 • AstructurewithaseriesofelementswhichusesLIFOmode. • Anelementcanbepushedonlyatthetopandonlyonepointerisusedfor POP. • Usedwhenanelementisnotaccessiblethroughpointerandindex,butonly throughLIFO.
17.	**DefineList.** BTL1 • Eachelementhasapointertoitsnextelement. • Onlythefirstelementisidentifiableanditisdoneusinglist-toppointer(header). • Other elementhasnodirectaccessandisaccessedthroughthefirstelement.
18.	**What is Objectoriented programming?** BTL1 Anobject-orientedprogramminglanguageisusedwhenthereisaneedforre-usabilityof definedobjectsorasetofobjectsthatarecommonformanyapplications.
19.	**WhataretheadvantagesofOOPs?** BTL1 • Dataencapsulation. • Data Abstraction.

	• Reusablesoftwarecomponents. • Inheritance • Polymorphism
20.	**WhatarethecharacteristicsofOOPs?** BTL1 • Anidentity-referencetoamemoryblock • Astate-data,fieldandattributes • Abehavior-methodstomanipulatethestateoftheobject
21.	**DefineClass.** BTL1 Aclassdeclarationdefinesanewtypethatlinkscodeanddata.Itisthenusedtodeclare objectsofthatclass.Thusaclassisanlogicalabstractionbutanobjecthasphysical existence.
22.	**What is MultipleInheritance?** BTL1 Inheritanceistheprocessbywhichobjectsofoneclassacquirethepropertiesofobjectsof anotherclass.InOOP,theconceptofinheritanceprovidestheideaofreusability.
23.	**WhatisaPreprocessorDirective?** BTL1 Apreprocessordirectivestartswith'#'sign. Thefollowingarethetypesofpreprocessor directives: Preprocessorglobalvariables Preprocessorconstants
24.	**Mentiontheflagsavailable forqueue.** BTL1 • QerrrorFlag • HeaderFlag • TrailingFlag • cirQuFlag • PolyQuFlag
Part *B	
SL NO.	**QUESTIONS**
1.	**Illustrate with functional description about the different phases of Embedded Design Life Cycle Method. (May/June 2016, Nov/Dec 2016, April/May 2017) (13M)** BTL4 **Answer: Page 3.7 –Dr.G. Ramprabu** • **Diagram (6M)** • **Explanation (7M)**



- Typicalsimpleproductcontainsfiveminimalphasesnamely: 'requirementanalysis'. 'Design','developmentandtest','deployment'and'maintenance'.

Need

- articulatedtoinitiatetheProductDevelopmentLife.
- based ontheneedfortheproduct,a'StatementofNeed'or'ConceptProposal'isprepared.
- 'ConceptProposal'mustbereviewedbytheseniormanagementandfundingagency

Conceptualization

- the'ProductConceptDevelopmentPhase -beginsimmediately afteraConceptProposal-formallyapproved
- twotypesofactivities,namely;'Planning Activity'and'AnalysisandStudyActivity'.
- Cost Benefit Analysis – similar to loss and gain.
- Marketchoicebased benefitmeasurement
- Targetendusers - ProductScope

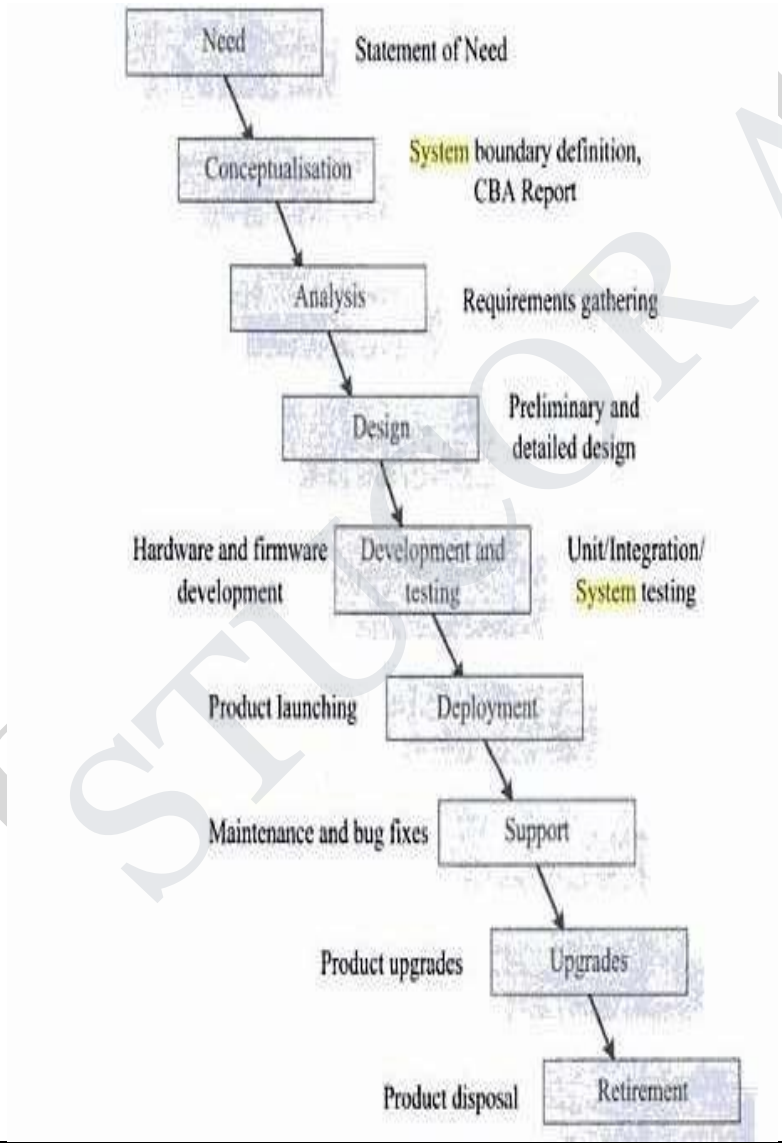
Analysis

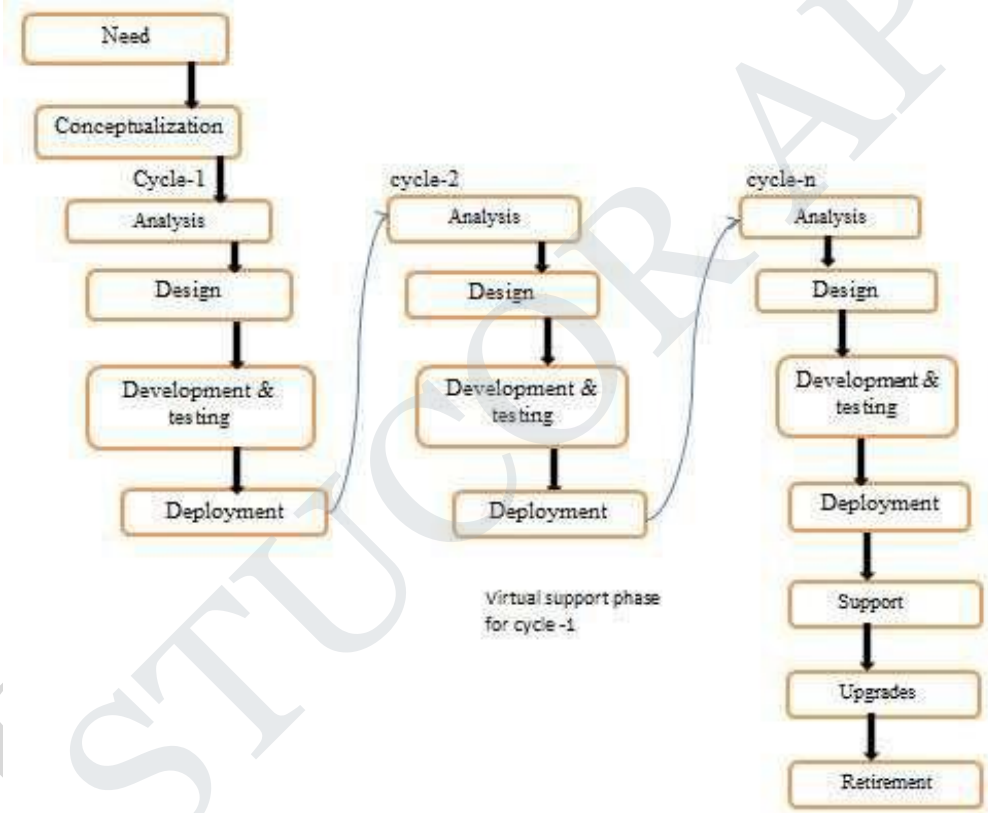
- startsimmediatelyafterthedocumentssubmitteddurin the'Conceptualization'
- thedevelopmentofdetaileduserrequirements.
- variousactivities performedduring'Requirement analysis'.

Design

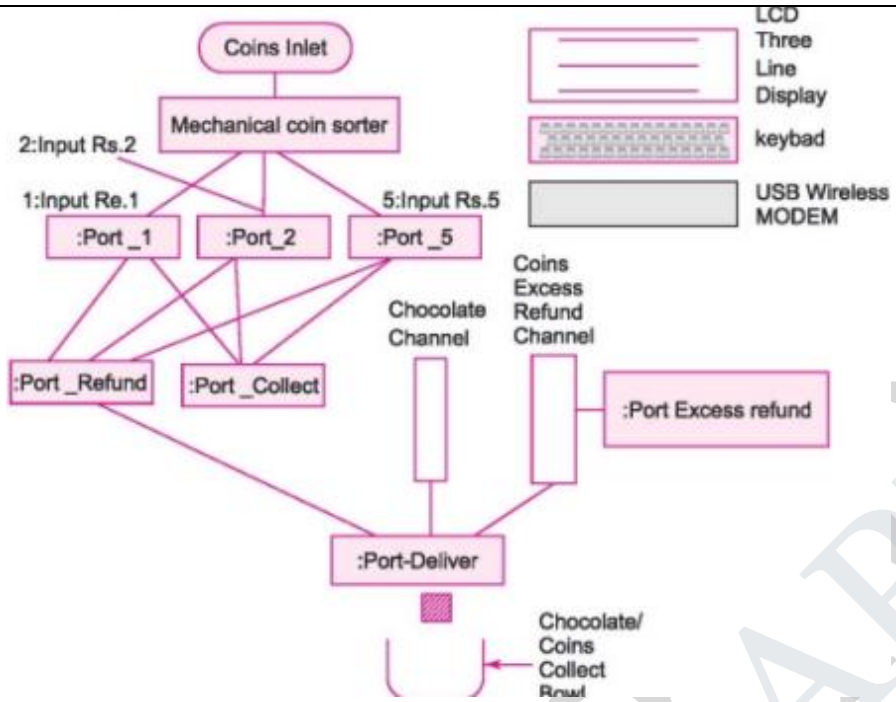
- 'Designphase'dealswiththe entiredesignoftheproduct.
- designphaseidentifiestheapplicationenvironment

	• createsan overall architectureforthe product. • the'PreliminaryDesignDocument(PDD)issentforreviewtothe end- user/client **DevelopmentandTesting** • 'DevelopmentPhasetransformsthedesignintoarealizableproduct. • theinstallationandsettingupofvariousdevelopment tools–performed. • partitionedintoembeddedhardwaredevelopment, embeddedfirmwaredevelopmentandproductenclosuredevelopment. **Deployment** • theprocessoflaunchingthefirstfullyfunctionalmodel. • handingoverthefullyfunctionalinitial modeltoanenduser/client. • ProductInstallation. **Support** • theoperationsandmaintenanceoftheproductina productionenvironment. • meetstherequirementsput forwardbytheenduser/client - productmal-functioning orunexpectedbehaviororany operationalerror **Upgrades** • upgradephaseofproductdevelopmentdealswith thedevelopmentofnew versions. • subjecttodesignmodificationtofixthemajorbugsreported • embeddedproducts,theupgradesmaybefortheproductresidentfirmware. • featureenhancementscanalsobeperformedeasily. **Retirement/Disposal** • technologyyoufeelasthemostadvanced andbest todaymaynotbethesametomorrow. • productcannotsustaininthemarket foralongtime. • theproduct manufacturerrealizesthatthereisanotherpowerfultechnology – old technology – disposal
2.	**Mention the essential and objectives of EDLC. Discuss in detail about the different phases of EDLC.(13M)** BTL1 **Answer: Page 3.2–Dr.G. Ramprabu** • **Diagram (6M)** • **Explanation (7M)** • Aimofanyembeddedproductinacommercialproductionsetup - ReturnonInvestment (R0I). • Aproductissaidtobeprofitableonly-turnoverfrom theselling-product-morethanthatoftheoverallinvestmentexpenditure • Ensurethathighqualityproductsaredelivered toenduser. • Risk minimizationand defect prevention in product development through project management. • The budgetallocationmighthavedoneafterstudyingthemarkettrends. • project management-essentialinproductdevelopment. • 'Projectmanagementisessentialforpredictability,co-ordinationandrisk minimization'. • Estimateonthedurationof'the developmentanddeploymentactivity should begiven tothe enduser/client..

	• Resourceallocationiscritical-havingadirectimpactoninvestment. • Projectmanagementalsocoversactivitiesliketaskallocation, scheduling,monitoringandprojecttracking. • Productivity-measureofefficiencyaswellasReturnonInvestment(ROT). • Productivitymeasurementisbasedontotalmanpowerefficiency. • Usingresourceswith specificskillsetswhichmatchestheexactrequirement. • Recruiting peoplewithdesiredskillsetsfor thecurrentproductdevelopment.
3.	**Generalize the LinearorWaterfallmodel in embedded design.(13M)** BTL2 **Answer: Page 3.24 –Dr.G. Ramprabu** • **Diagram (6M)** • **Explanation (7M)** LinearorWaterfallmodel 

	• Adoptedinmostoftheoldersystems - executedinsequence • linearmodelestablishesaformal analysisanddesignmethodology • Theflowisunidirectionalwithoutput ofonephaseservingastheinputtothenextphase. • Welldocumented,givinganinsightintowhatshouldbedoneinthenextphase. • Feedbackofeachphaseisavailablelocallyandonlyaftertheyareexecuted. • Fixesforthe bugsarepostponedtill thesupportphase. • Easyprojectmanagement andgoodcontrolovercostandschedule. • Therisk analysisisperformedonlyoncethroughoutthedevelopment.
4.	**Generalize the Iterative/Incremental orFountainmodel in embedded design.**(13M)BTL1 **Answer: Page 3.26–Dr.G. Ramprabu** • **Diagram (6M)** • **Explanation (7M)** Iterative/Incremental orFountainmodel  • Iterativemodelcanbeviewedasacascadedseriesof linearmodels. • therequirementsare knownatthebeginningand theydividedinto differentgroups. • eachcycleisinterconnectedinasimilarfashion ofafountain. • very gooddevelopmentcycle feedbackateach function/featureimplementation. • datacanbeused asareferenceforsimilarproduct developmentinfuture. • iterativemodelprovidesaworkingproduct modelwithatleastminimumfeaturesatthefirstcycleitself.

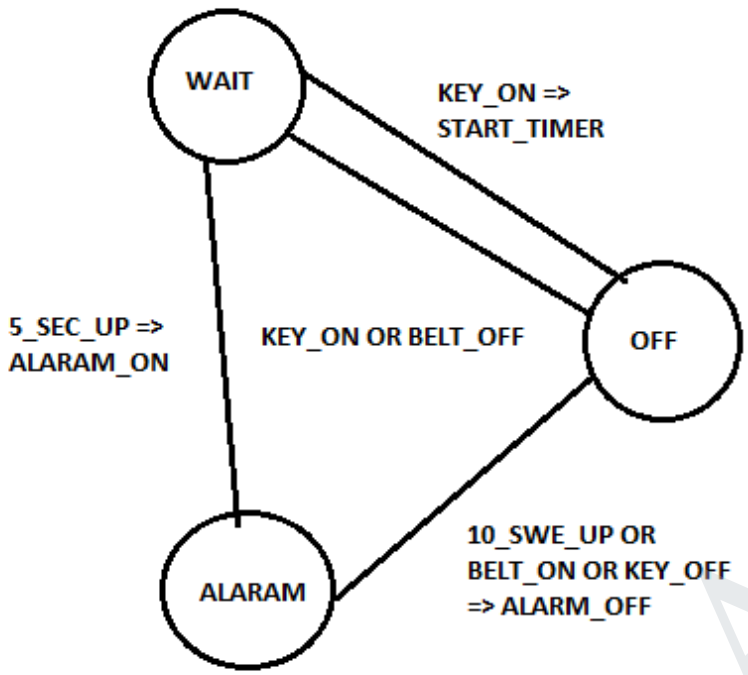
	• eachnewdeploymentat the end ofeach developmentcycle. • model -bestsuitedforproductdevelopmentswherethecontinuedfundingforeachcycle-not assured.
6.	**Describe the object oriented model.(13M)** BTL1 **Answer: Page 3.48–Dr.G. Ramprabu** • Object oriented model – Object based model for system. • Complex software – to simple defined process called object • Object oriented model – re usability – maintainability – productivity • Entity – representing or modeling a system • Character – unique behavior • Abstract description – blue print of object • Class – state of an object • Member variable, member function – private, protected and public • Public – access outside the class • Private – access only inside the class • Protected – access from derived class
Part*C	
SL NO	**QUESTIONS**
1.	**With a suitable example, explain about the State Machine Model of Chocolate Vending Machine (ACVM).(May/June 2016, Nov/Dec 2016)**(15M). BTL6 **Answer: Notes** • **Diagram (6M)** • **Explanation (9M)**



- state transition functions - changes a state to its next state.
- States - Idle, Running, Blocked, Finished
- A transition may be also be interrupt flag driven
- state can receive multiple tokens - inputs, messages, flags interrupts or semaphores
- coin inlet collects – coin from customer
- Mechanical coin sorter- sorts the coin according customer requirement.

2. **Explain the State Machine Model for seat belt warning system. (15)**BTL2
Answer: Page 3.40–Dr.G. Ramprabu

- **Diagram (6M)**
- **Explanation (9M)**
- State machine model – state, event, Action and transition
- State – current situation (Alarm OFF, Alarm ON and waiting)
- Event – an input (Ignition key ON, Ignition key OFF, Time expire)
- Action – Stimuli for state transition (ready for ON and OFF)
- Transition – one state to other (ON and OFF)
- Vehicle ignition turned ON – seat belt ON with in 10 S – if not Alarm ON
- Alarm OFF – seat belt ON or Time Expire



3. **Explain the Sequential Model Program for seat belt warning system.(April/May 2017) (15M)**BTL1

Answer: Page 3.43–Dr.G. Ramprabu

- **Explanation (6M)**
- **Program (9M)**
- Vehicle ignition turned ON – seat belt ON with in 10 S – if not Alarm ON
- Alarm OFF – seat belt ON or Time Expire
- If the driver - turns on the key - does not fasten the seat belt within 5 seconds
- then sound the alarm - for 5 seconds - until the driver fastens the seat belt - until the driver turns off the key
- Program

```
# define ON 1
# define OFF 0
# define YES 1
# define NO 0
Void seat_belt_warn()
{
Wait_10 sec();
If(check_ignition_key()== ON)
{
If(check_seat_belt()==OFF)
{
```

```
Set_timer(5);
Start_Alarm();
While((check_seat_belt()==OFF) && (check_ignition_key()== OFF) &&
(timer_expire()==ON));
Stop_alaram();
}
}
}
```

UNIT IVRTOS BASED EMBEDDED SYSTEM DESIGN

Introduction to basic concepts of RTOS- Task, process & threads, interrupt routines in RTOS, Multiprocessing and Multitasking, Preemptive and non-preemptive scheduling, Task communication shared memory, message passing-, Inter process Communication – synchronization between processes-semaphores, Mailbox, pipes, priority inversion, priority inheritance, comparison of Real time Operating systems: VxWorks, чC/OS-II, RT Linux

Part*A

SL NO.	Questions
1.	**Compare preemptive and non preemptivescheduling.(May/June 2016, Nov/Dec 2016)** BTL2 **Non-preemptive Scheduling** is one which can be applied in the circumstances when a process terminates, or a process switches from running to waiting state. In **Non-Preemptive Scheduling**, once the resources (CPU) is allocated to a process, the process holds the CPU till it gets terminated or it reaches a waiting state.
2.	**Define thread and process.(May/June 2016)** BTL2 Thread : A thread is a sequential flow of control with a process. A thread is the preemptive that can execute code, Thread is known as light weight process Process :A process is a program, or a part of it, in execution. Process is also known as an instance of a program in execution.
3.	**What are the functions of RTOS? (Nov/Dec 2016, April/May 2017)** BTL1 • Task/Process management • Task/process scheduling • Task/process synchronization • Error/Exception handling • Memory management • Interrupt handling • Time management
4.	**Define Multi threading.(April/May 2017)** BTL2 Multithreading is a type of execution model that allow multiple threads to exist with in the context of process such that they execute independently but share their process resource.
5.	**Whatissophisticatedmultitaskingembeddedsystem?** BTL1 Multitaskingprovidesthefundamentalmechanismforanapplicationtocontrolandreacttomultiple,discretereal-

	worldevents.Multitaskingcreatestheappearanceofmanythreadsofexecutionrunningconcurrentlywhen,infact,thekernelinterleavestheirexecutiononthe basisofaschedulingalgorithm.
6.	**Explainmultitaskandtheirfunctionsinembeddedsystem.**BTL2 Thissystemimplementscooperativeandtime-sliced multitasking, provides resource lockingandmailboxservices,implementsanefficientpagedmemorymanager, traps and reportserrors,handlesinterrupts,andautostartsyourapplicationatsystemstartup.Byfollowingsomesimple codingpracticesasshowninthe documentedcoding examples, you can takeadvantageof these sophisticated features without having toworry abouttheimplementation details.

7.	**Givethefunctionforsendingaqueue.**BTL1
	Eachmessageismadeupoftwoparts,whicharedefinedinthetemplatestructurestruct msgbuf,asdefinedinsys/msg.h:
	structmsgbuf{
	long mtype;
	char mtest
	};
	Thefieldmtypeisusedlaterwhenretrievingmessagesfromthequeue,andcanbesettoany positivenumber.mtextisthedatathiswillbeaddedtothequeue

8.	**Givetheneedsformemorymanagement.**BTL1 Eachnewmodelofcomputerseemstocome withmoremainmemorythan thelast,but,sincethe memory requirementsof the software rise just as fast, memoryis always a precious commodity,hencetheneedformemorymanagement. Memoryisallocatedtoaprocesswhenneeded Memoryisdeallocatedwhennolongerinuse Swappingallowsthetotalmemoryusedbyalltherunningprocessestoexceedmain memory Virtualmemorymakesitpossibletorunasingleprogram thatuses morememorythanthe mainmemory(normallyRAM)availableonthesystem.Virtualmemoryisnormally dividedintopages. Programs refertoparts ofmemoryusingaddresses.Inavirtualmemorysystem,thesearevirtualaddresses Thevirtualaddressismappedonto physicaladdressesbyamemory management unit (MMU)
9.	**Namesomeapplicationforthe VxWorksRTOS.**BTL1 • Automobiles • Avionics • Consumerelectronics • Medicaldevices • Military • Aerospace • Networking
10.	**WhatarethevariousfeaturesofVxWorks?**BTL1 • Highperformance • Hostandtargetbaseddevelopmentapproach • Supportsadvancedprocessorarchitecture • Hardreal-timeapplications
11.	**WhatarethebasicfunctionsofVxWorks?**BTL1 • Systemlevelfunctions

	• Taskservicefunctions • Taskcontrolfunctions • IPCs • NetworkandIOfunctions
12.	**WhatarethetaskservicefunctionssupportedbyVxWorks?**BTL1 Taskcreationandactivationdistinctstates. Functionsforthetaskcreating,running,waiting,suspending. and resuming, task pendingcumsuspendingwithandwithouttimeouts.
13.	**Whatarethe differenttypes ofsemaphoresinvxworks? Whichisthefastest?**BTL4 VxWorkssupportsthreetypesofsemaphores.Binary,mutualexclusion,andcounting semaphores.Binaryisthefastestsemaphore.
14.	**Whatissignalservicingfunction?**BTL1 VxWorkssupportsasoftwaresignalfacility.Signalsasynchronouslyalterthecontrol flowofatask.AnytaskorISRcanraiseasignalforaparticulartask.Thetaskbeing signaledimmediatelysuspendsitscurrentthreadofexecutionandexecutesthe task specifiedsignalhandlerroutinethenext timeitisscheduled torun. Thesignal handler executesin thereceivingtask'scontext andmakesuseofthattask'sstack.Thesignal handlerisinvokedevenifthetaskisblocked.
15.	**DefineMicroC/OSII**.BTL1 MicroC/OSII(commonlytermeduC/OSIIor mC/OS-Il),isalow-costpriorty-basedpreemptiverealtimemultitaskingoperatingsystemkemelfor microprocessors,mainlyintheCprogramminglanguage.Itismainlyintendedforuseinembeddedsystems.
16.	**WhatarethetaskstatesinMICROC/OS-II?**BTL1 Taskstates: mC/OS-[isamultitaskingoperatingsystem.Each taskisan infiniteloopandcanbe inanyoneofthefollowing5states: • Dormant • Ready • Running • Waiting • ISR
17.	**What arethe2 sourcefilesinMicroC/OS-II?**BTL1 • Preprocessordependentsourcefile • Preprocessorindependentsourcefile
18.	**Whatarethebasic functionsofMUCOS?**BTL1 Systemlevel:OSinitiate,start,systemtimerset,ISRenterandexit • Taskservicefunction:create,run,suspend,resume • Taskdelay • Memoryallocationandpartitioning • IPCs,mailboxandqueues
19.	**Define software or soft modem**. BTL2 A software modem or a soft modem is a modem with minimal hardware capacities, designed to use a host computer's resources to perform most of the task by a dedicated hardware in a traditional modem.
20.	**Givethestepstodestroyamessagequeue.**BTL1

	There are two ways: • UsetheUnixcommandipcstogetalistofdefinedmessagequeues,thenusethecommandipcrmtodeletethequeue. • Writeaprogramtodoitforyou
21.	**Define Alarm Clock.** BTL1 An alarm clock is a clock that is designed to make a sound or some other signal at a specific time. Microprocessors are used to read the clock's buttons and update the time displays.
22.	**Define Audio players.** BTL1 Audio players are usually referred as MP3 players after the well-known audio data format. The earliest portable MP3 players were based on compact disc mechanisms. Modern MP3 use either flash memory or disc drives to store music.
23.	**Define Video Accelerator.** BTL1 A video accelerator significantly speeds up the updating of the images on a screen, which also frees the CPU totake care of other tasks.
	Part*B
SL NO.	**Questions**
1.	**Explain preemptive and non preemptive scheduling in RTOS .(13M)** BTL2 **Answer: Page 4.20 –Dr.G. Ramprabu** • **Preemptive (6M)** • **Non-Preemptive (7M)** Preemptive scheduling • one which can be applied in the circumstances when a process terminates • a process switches from running to waiting state • Basic -The resources are allocated to a process for a limited time. • Interrupt - Process can be interrupted in between • Starvation - If a high priority process frequently arrives in the ready queue, low priority process may starve. • Overhead - Preemptive scheduling has overheads of scheduling the processes • Flexibility - Preemptive scheduling - flexible. • Cost - Preemptive scheduling - cost associated Non Preemptive scheduling • once the resources (CPU) - allocated to a process • the process holds the CPU till it gets terminated or it reaches a waiting state. • Basics -Once resources are allocated to a process, the process holds it till it completes its burst time or switches to waiting state. • Interrupt - Process can not be interrupted till it terminates or switches to waiting state. • Starvation - If a process with long burst time is running CPU, then another process with less CPU burst time may starve.

	• Overhead - Non-preemptive scheduling does not have overheads. • Flexibility - Non-preemptive scheduling is rigid. • Cost - Non-preemptive scheduling is not cost associative.
2.	**Explain in detail about Inter Process communication and context switching .(Nov/Dec 2016) (13M)** BTL2 **Answer: Page 4.36 – Dr.G. Ramprabu** • **Inter Process communication (6M)** • **Context switching (7M)** Inter processor communication • Used to generate information about certain sets of computations finishing on one processor • let the other processors waiting for finishing the computations take note of the information • need to send through the kernel an output data -for processing • Global variables problems ─ shared data and no encapsulation of the data • IPC - scheduler, task or ISR • generates an output so that it lets another process take note • generate message from the certain sets of computations finishing on one task • let the other tasks take note of signal or get the message Context Switching • Context switching means saving the context of interrupted routine - task and retrieving • loading the new context • The multitasking and multiple ISRs execute even though there is only one processor • first saving the one program context and retrieving another program context • Before executing new instructions of the new function, the current program's program counter – saved • Also status word, registers, and other program-contexts – saved - needed by the newly called function. • Getting an address - loading that address into the program counter - executing the called function's instructions • Program Counter- a part of the context of presently running program • A context of a program must include program counter as well as the program status word, stack pointer • A register set or memory block can hold context information
3.	**Explain the terminologies semaphores, Mail box, pipes and shared memory in RTOS.(May/June 2016)(13M)** BTL2 **Answer: Page 4.38, 4.29–Dr.G. Ramprabu** Semaphore (4M) • A semaphore is a kernel object that one or more tasks can acquire - synchronization or mutual exclusion. • Mutual exclusion - a provision by which only one task at a time can access a shared

	resource. • semaphore as a key - task can make a request for the key - available, your task can check • Multiple semaphores can be used if desired - multiple tasks that are waiting • either the oldest task on the queue or the highest priority task is given the semaphore • Semaphores - global resources – task sync Mail Box (3M) • provide the mailbox and queue both IPC functions • mailbox - not provided by an OS, then the OS employs queue for the same purpose • mail box - an IPC through a message-block at an OS that can be used only by a single destined task. • include a header to identify the message-type • Deleting means message-pointer pointing to Null Pipes (3M) • Pipe - a device used for the inter process communication • has the functions create, connect and delete - open, write, read, close • a device for inserting (writing) and deleting (reading) • fwrite with a file name to write into a named file • fread with a file name to read into a named file • limited and have a variable number of bytes per message between the initial and final pointers. Shared memory (3M) • A shared memory - an extra piece of memory that is attached to some address spaces for their owners to use • all of the processes share the same memory segment • feature supported by UNIX System V, including Linux, SunOS and Solaris. • Server - One process must explicitly ask for an area, using a key, to be shared by other processes. • Client - the shared area can access
4.	**Explain how the interrupt is handled by RTOS and illustrate the features of RTLinux RTOS. (13M)**BTL2 **Answer: Page 4.53 – Dr.G. Ramprabu** • Time sharing software: switch between different tasks fast enough to create the illusion that all are going forwarded • Realtime software: switch between different tasks in time to meet deadline • Hard real time : Predictable performance at each moment in time: not as an average. • Low latency response to events. • Precise scheduling of periodic tasks. • Soft real time :Good average case performance • Low deviation from average case performance • Traditional problems with soft real time - The chips are usually placed on the solder dots • The machine tool generally stops the cut as specific

	• The power almost always shuts off before the turbine explode • The e cell-phone connect won't drop your Internet handset during a handoff unless there is heavy traffic
Part*C	
SL NO.	**Questions**
1.	**Explain how the interrupt routines are handled by RTOS and illustrated the features of VxWorks.(May/June 2016)**(15M) BTL2 **Answer: Page 4.41–Dr.G. Ramprabu** • High -performance, Unix performance - Unix -like, multitasking Environment scalable and scalable and hierarchical RTOS • Host and target based development approach • Device Software Optimization - a new methodology that enables development and running of device • VxWorks 6.x processor abstraction layer - enables application design for new versions later • Supports advanced processor architectures ─ ARM, ColdFire, MIPS, Intel, SuperH • Supports kernel mode execution of Supports kernel mode execution of tasks • Provides for the preemption points at kernel - as well as round robin scheduling • Schedules the ISRs separately and has special functions for interrupt handling • Watchdog timers - Virtual I/O devices including the pipes and sockets • Virtual Memory Management functions • Power management functions - the ability to control power consumption • Automatic detection and reporting of common memory and other errors • Interconnect functions that support large number of protocols • System Level – OS initiate, start, system timer clock rate set, ISR enter and exit, enable and disable • Task Service Functions – initiate, resume, activate, run, suspend.
2.	**Explain how the interrupt is handled by RTOS and illustrate the features of µC/OS-II RTOS.(Nov/Dec 2016)(15M)** BTL2 **Answer: Page 4.48 –Dr.G. Ramprabu** • Preemptive RTOS • Different Platforms support • Multitasking • Portable as ROM image • Scalable - only needed OS functions become part of the application codes • software component for safety critical systems • Applications - Automotive, avionics, consumer electronics, medical devices, military, aerospace, networking, and systems-on-a-chip. • OSTaskCreate( ) ─ a MUCOS function that creates a task • OS_NO_ERR─ a MUCOS macro that returns true in case no error is reported

	• OS_MAX_TASKS ─ user definable constant for specifying maximum number of tasks • System Level – OS initiate, start, system timer set, ISR enter and exit • Task Service Functions – create, run, suspend, resume • IPCs – Semaphore, Queue and Mailbox • Same Semaphore function usable as event flag
3.	**Explain the terms Priority inversion and priority Inheritance. (13M)** BTL2 **Answer: Page 4.39– Dr.G. Ramprabu** Priority inversion • Priorities of tasks be in an order such that task I highest priority • task J a lower, and task K the lowest priority • Only tasks I and K share the data and J does not share data with K • let tasks I and K alone share a semaphore s_{ik} and not J • latency becomes too high and may exceed the deadline if all tasks are blocked • small only if the time taken by the tasks that share the resource • an instant t0, suppose task K takes sik , it does not block task J - blocks only the task I. • happens because only tasks I and K share the data and J does not share Priority Inheritance • a method for eliminating unbounded priority inversion • the priority inheritance protocol is that when a job blocks one or more high-priority jobs • executing its critical section and releasing its locks • the process returns to its original priority level • First Come First serve – method • Job 1 at 2 m – job 2 at 3 m • It will take job 1 first

Unit V EMBEDDED SYSTEM APPLICATION DEVELOPMENT	
Case Study of Washing Machine- Automotive Application- Smart card System Application,.	
Q.No	**Questions**
1.	**Define Alarm Clock.** BTL2 An alarm clock is a clock that is designed to make a sound or some other signal at a specific time. Microprocessors are used to read the clock's buttons and update the time displays.
2.	**Define Audio players**. BTL2 Audio players are usually referred as MP3 players after the well-known audio data format. The earliest portable MP3 players were based on compact disc mechanisms. Modern MP3 use either flash memory or disc drives to store music.

3.	**Define Video Accelerator.** BTL2 A video accelerator significantly speeds up the updating of the images on a screen, which also frees the CPU to take care of other tasks.
4.	**Define software or soft modem.** BTL2 A software modem or a soft modem is a modem with minimal hardware capacities, designed to use a host computer's resources to perform most of the task by a dedicated hardware in a traditional modem.
5.	**What are the goals of design process? (Apr/May 2011)**BTL1 A design process has several important goals beyond function, performance, and power. They are time to market, design cost and quality
6.	**What is prototype?**BTL1 Prototype is the model of the system being designed. Prototypes are a very useful tool when dealing with end usersrather than simply describe the system to them in road, technical terms, a prototype can let them see, hear, and touch at least some of the important aspects of the system
7.	**What is design technology?**BTL1 Design technology involves the manner in which we convert our concept of desired system functionality into an implementation. Design methodologies are used in taking the decisions at the time of designing the large systems with multiple design team members.
8.	**List some application of embedded system.(May/June 2016, Nov/Dec 2016, April/May 2017)**BTL1 • Washing machine • Digital camera • Automotive • Robotics • Smart card system
9.	**What are the events involved in smart card application? .(May/June 2016,Nov/Dec 2016)**BTL1 • Start up event • Battery monitoring and charge controlling event • Card read and write event • Communication event • Keyboard scanning event • LCD update event • Watch dog timer update event
10.	**Define spin phase.**BTL2 In the second phase of washing , water is pumped out from the tub and the inner tub uses centrifugal force to wring out more water from the cloth by spinning at several hundred rotations Per minute. This is called spin phase
11.	**Define engine control unit.**BTL2 It control the operating parameters, to make sure that the engine gets proper inputs. Thus they help protect the engine against damage. It is the brain of the engine.
12.	**What are the input parameters are used in engine control uni**t?BTL1 • Vehicle speed • Vehicle acceleration or deacceleration

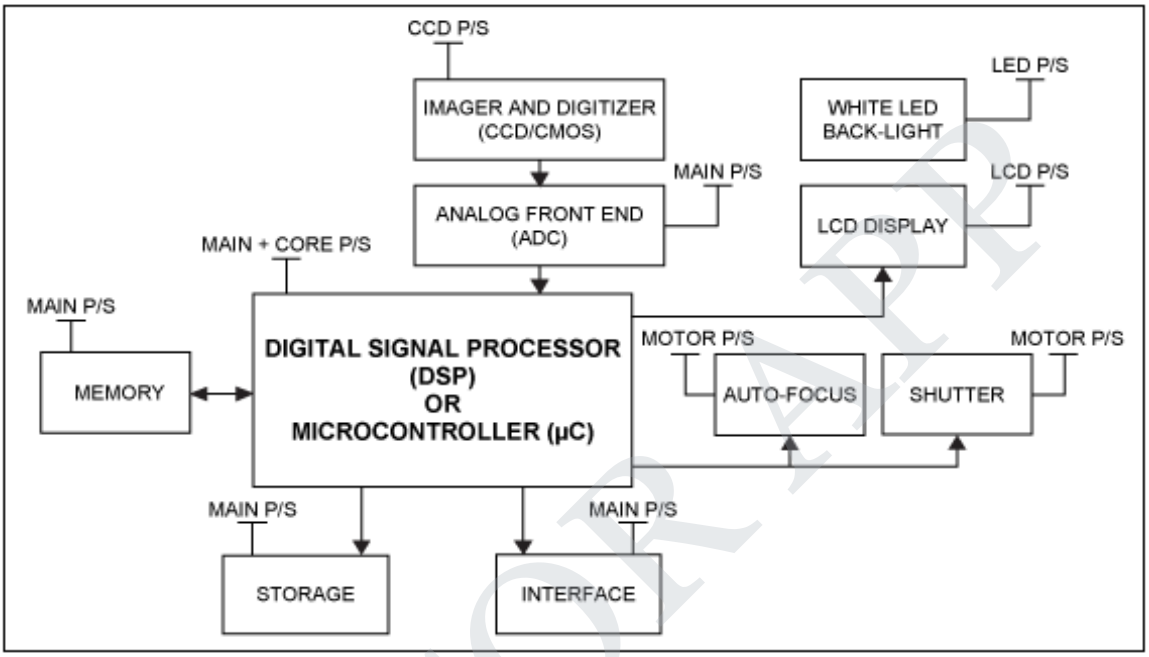
	• Temperature • Air pressure • Throttle position center
13.	**What are the constructional parts of washing machine?**BTL1 • Inner tub • Outer tub • Water inlet pipe • Level sensor • Temperature sensor • Water outlet • Keypad - LCD • Control panel interface
14.	**Define testing.**BTL2 Testing is the process of make sure the program module will work properly. Testing can be performed in different kind of ways relayed with the application
15.	**What is high speed electronic unit?**BTL1 It is having fast response, Like fuel injection system, anti clock break system, engine control, electronic throttle, steering control, transmission control and central control unit
16.	**What is low speed electronic unit?**BTL1 It is deployed in applications not critical. They are low cost microprocessor or micro controllers and digital signal processors , Audio controllers, driver door locks, door glasses control etc.
17.	**Define controller area network(CAN).**BTL1 The CAN bus was originally proposed by Robert Bosch, pioneer in automotive embedded solution providers. It supports medium speed and high speed data transfer. CAN is an event driven protocol interface with support for error handling in data transmission. It generally employed is safety system like air bag control, engine control and antilock break system (ABS)
18.	**Define Local Interconnect Network(LIN).**BTL2 Lin bus is a single master multiple slave communicationinterface. LIN is a low speed, single wire communication interface with support for data rates up to 20 Kbps and is used for sensor/actuator interfacing. LIN bus follows the master communication triggering technique to eliminate the possible bus arbitration problem that can occur by the simultaneous taking off different slave nodes connected to a single interface bus. LIN bus is employed in application like mirror control, Fan controls, seat positioning controls, window controls and position control where response time is not a critical issue
19.	**Define Media – Oriented System Transport (MOST) Bus.**BTL2 The MOST is the targeted for automotive audio/video equipment interfacing, used primarily in European cars. MOST bus is an optical fiber cables. The MOST bus specifications define the physical layer as well as the application layer, network layer and media access control.
20.	**What is the basic function of audio players?**BTL2 • Audio decompression • User interface • Audio storage
Part*B	
Q.No	**Question**

1. **With suitable diagram explain in detail about the concept of DigitalCamera application in embedded system. (13M)** BTL2

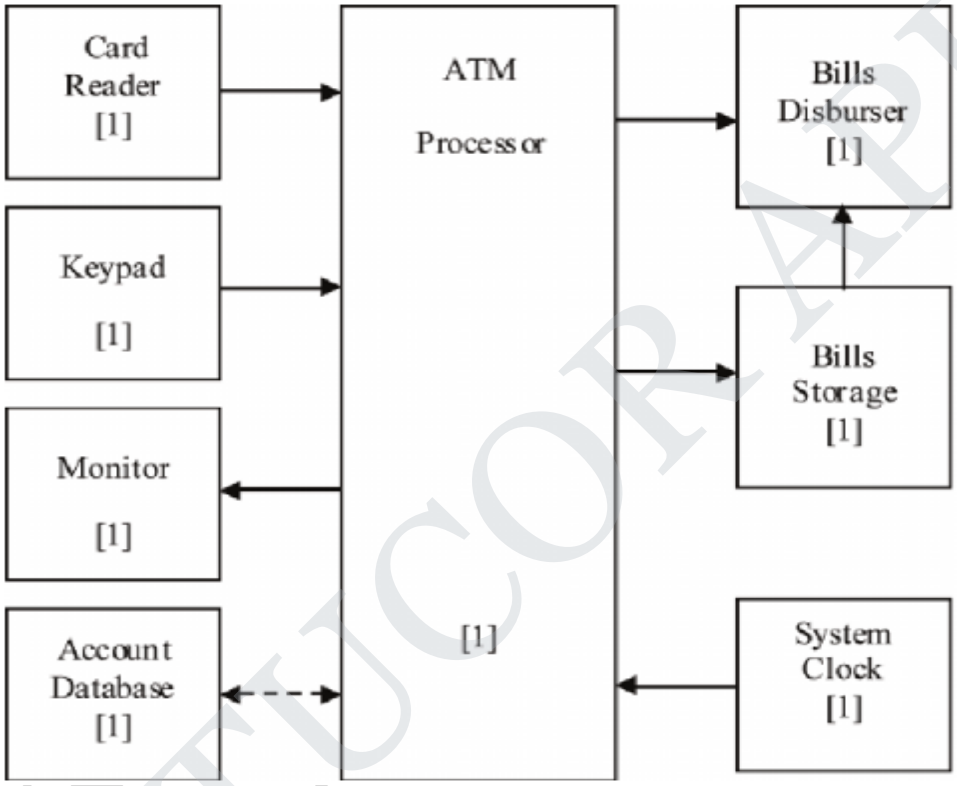
Answer: Page 5.49–Dr.G. Ramprabu

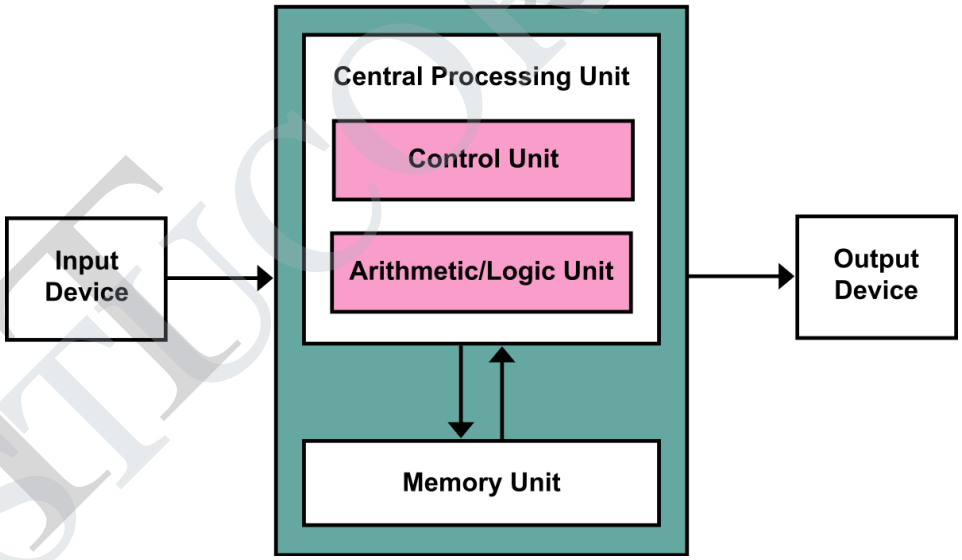
- **Diagram (6M)**
- **Explanation (7M)**

Digital Camera in Embedded System



- General-purpose processor
- Camera records the pictures using a charge coupled devices (CCD) array
- The array consisting of large number of horizontal rows and vertical columns of CCD cells for the picture
- A number of CCD cell unexposed to the picture but used for off-set corrections in the each-row output.
- Each set of pixel has three cells, for the red, green and blue components in a pixel.
- Each cell gets exposed to a picture when shutter of camera opens on a user command.
- A set of controllers – to control shutter, flash, auto focus and eye-ball image control.
- User gives commands for switching on the camera, flash, shutter, adjust brightness, contrast, color, save and transfer.
- shutter is pressed, a flash lamp glows and a self-timer circuit switches off the lamp automatically.
- JPEG file for a picture can be copied or transferred to a memory stick using a controller
- A picture jpg can be copied to a computer connected through USB port controller.
- Digital recording and display of pictures Processing to get the pictures of required brightness, contrast and color
- Transfer files to a computer and printer through a USB port
- Intensity and color values for each picture horizontal and vertical rows and columns of pixels in a

	picture frame. • Special-purpose processor Custom or Standard • Memory • Interfacing • Encodes a digital images
2.	**With suitable diagram explain in detail about the concept of ATM application in embedded system. (13M)** BTL2 **Answer: Notes** • **Diagram (6M)** • **Explanation (7M)**  • ATM standard (defined by CCITT) is widely accepted by common carriers as mode of operation for communication – particularly BISDN. • ATM is a form of cellswitching using small fixed-sized packets. • ATM network will be organized as a hierarchy. • Two levels of ATM connections:-virtual path connections-virtual channel connections • Vast majority of ATM networks will run on optical fiber networks with extremely low error rates.

	• ATM must supports low cost attachments • ATM Adaptation Layer (AAL) – the protocol for packaging data into cells - collectively referred to as AAL • Must efficiently package higher level data such as voice samples, video frames and datagram packets into a series of cells. • ATM provides permanent virtual connections and switched virtual connections • Permanent Virtual Connections (PVC)- permanent connections set up manually by network manager • Switched Virtual Connections (SVC)-set up and released on demand by the end user via signaling procedures.
3.	**With suitable diagram explain in detail about the concept of Engine control Unit in embedded system. (13M)** BTL2 **Answer: Page 5.67– Dr.G. Ramprabu** • **Diagram (6M)** • **Explanation (7M)**  • A system designed to perform a single well defined function life-long- embedded system • Applications of embedded systems range from home to office, to automotive and avionics industries. • Traditional design methodologies for designing embedded systems are generally based only on past experiences • The hardware and software components are designed in a manner that ignores the interdependence between them

	• The Engine Control Unit (ECU) - the brain of the engine • Inputs - speed, temperature, pressure and pilot throttle • advanced micro-processors and comprehensive software helps increase the engine life and ensure safety. • An ECU consists of a set of sensors, a processing unit and a set of actuators. • sensors periodically measure the engine status and provide input to the processing unit which processes this data • actuators execute the commands received from the control unit. • optimize the fuel injection and ignition so that it minimizes fuel consumption and emissions of pollutants and maximizes the torque and power. • Injection : In order to burn the fuel completely and correctly, the ratio between air and the fuel which go into the piston should be kept constant. • This ratio is maintained by the ECU by controlling the opening time of each injector. • Ignition : The fuel should get enough time to burn completely. To let this happen, the spark has to be fired in advance with respect to the instant when the piston is at its highest point. • Parameter also affects the emissions since unburnt fuel - pushed out of cylinder as emission or pollution. • The timing difference between fuel injection and firing of spark is maintained by monitoring engine RPM
4.	**With suitable diagram explain in detail about the concept of Automotive application in embedded system. (13M)BTL2** **Answer: Page 5.5 – Dr.G. Ramprabu** • **Diagram (6M)** • **Explanation (7M)**

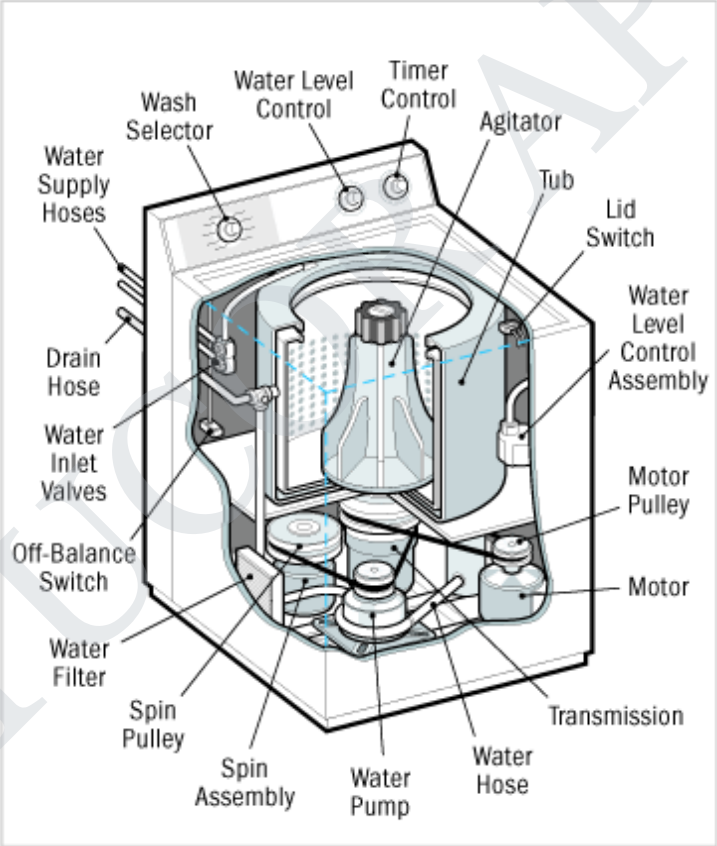
- 8-32bit 40 MHz microprocessor
- Analog-digital converter
- High-level digital outputs
- Digital-analog converter
- Signal conditioner
- Communication chips
- Each module communicates errors to a central module
- Can communicate errors to a diagnostic tool

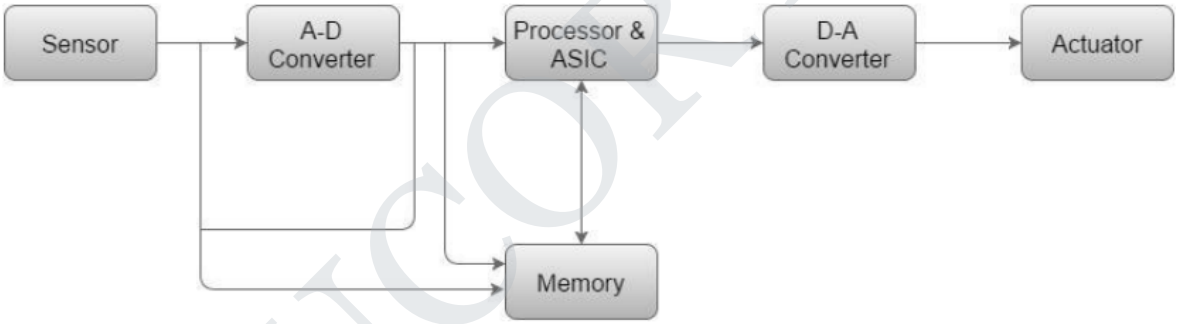
Instrument Cluster

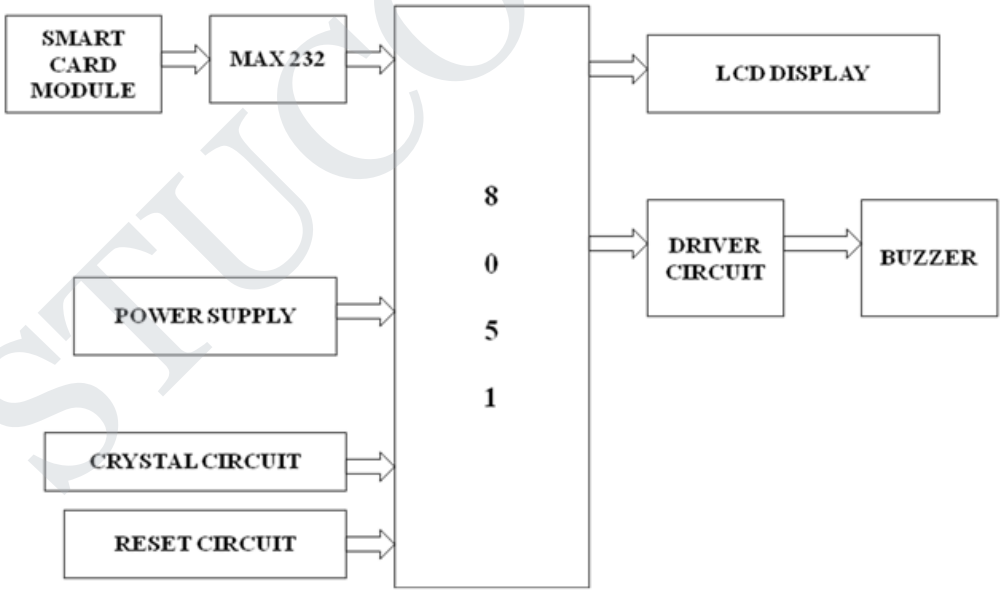
- Displays data about the vehicle in its current state
- Various modules send data to ECU
- ECU send a packet of info
- Cluster module looks for specific headers
- Cluster is updated

Other modules

- Antilock Brakes
- Airbags
- Security systems
- Keyless entry
- Media center
- Cruise control
- Seat position and temperature
- Brake assist

	• Stability control • Anti-collision • Reverse assist • Traction control • Self-parking
Part*C	
Q.No	**Questions**
1.	**With suitable diagram explain in detail about the concept of washing machine application.(May/June 2016, Nov/Dec 2016)(15M)**BTL4 **Answer: Page 5.1–Dr.G. Ramprabu** • **Diagram (6M)** • **Explanation (9M)**  • Washing machine supports three functional modes • The system should provide fully automatic mode, semi-automatic mode and manual mode. Modes should be selectable by a keypad • Under fully automatic mode user intervention requirement should be zero • after the completion of work it should notify the user about the completion of work • Semi-automatic mode also user requirement should be nil - But user has to choose

	any one of the semi-automatic mode. • manual mode continuous intervention of user is required • When the lid - open system should not work • basic features of a washing machine - washing, rinsing, spinning, drying, cold wash, hot wash • PWM feature of the microcontroller controls motor speed. • PWM output is fed to driver circuit and then to motor • Rotate the motor in two different directions 'forward' and 'reverse' direction. • Microcontroller reads the speed of the motor and appropriately controls the speed of the motor in different phases of washing. • Part of home automation
2.	**Elucidate the selection of processor and memory for any one embedded system applications with suitable diagram in detail.(May/June 2016, April/May 2017) (15M)** BTL6 **Answer: Notes** • **Diagram (6M)** • **Explanation (9M)**  • An embedded system has three components • It has hardware. • It has application software. • Real Time Operating system (RTOS) that supervises the application software and provide mechanism to let the processor run a process - plan to control the latencies. • Single-functioned − An embedded system usually performs a specialized operation and does the same repeatedly - A pager always functions as a pager. • Tightly constrained − All computing systems have constraints on design metrics - but those on an embedded system can be especially tight • Reactive and Real time − Many embedded systems must continually react to changes in the system's environment and must compute certain results in real time without any delay. • Microprocessors based − It must be microprocessor or microcontroller based. • Memory − It must have a memory, as its software usually embeds in ROM. • Connected − It must have connected peripherals to connect input and output devices • HW-SW systems − Software is used for more features and flexibility. Hardware is used for

	performance and security. • Advantages - Easily Customizable - Low power consumption - Low cost – Enhanced performance • Disadvantages - High development effort - Larger time to market • Sensor − It measures the physical quantity and converts it to an electrical signal which can be read by an observer or by any electronic instrument like an A2D converter. A sensor stores the measured quantity to the memory. • A-D Converter − An analog-to-digital converter converts the analog signal sent by the sensor into a digital signal. • Processor & ASICs − Processors process the data to measure the output and store it to the memory. • D-A Converter − A digital-to-analog converter converts the digital data fed by the processor to analog data • Actuator − An actuator compares the output given by the D-A Converter to the actual (expected) output stored in it and stores the approved output.
3.	**With suitable diagram explain in detail about the concept of smart card system application.(Nov/Dec 2016, April/May 2017) (15M)**BTL2 **Answer: Page 5.9–Dr.G. Ramprabu** • **Diagram (6M)** • **Explanation (9M)**  • Enabling authentication and verification of card and card holder by a host • Enabling GUI at host machine to interact with the card holder

	• Received header and messages at IO port Port_IO from host through the antenna • powered charge pump supply of the card activated signal to start • Transmitted headers and messages at Port_IO through antenna • No control panel - at the card – in host • radiations from the host activate a charge pump at the card. • task_ReadPort sends requests for host identification and reads through the Port_IO the host-identification message • All transactions between cardholder/user now takes place through GUIs • Code size: optimum. card system memory needs should not exceed 64 kB memory • Limited use of data types; multidimensional arrays, long 64-bit integer and floating points • File system(s): Three-layered file system for data. • File management: fixed length file management • Microcontroller hardware: Generates distinct coded physical addresses for the program and data • Validity: System is embedded with expiry date • Extendibility: The system expiry date is extendable by transactions and authorization of master control unit • Performance: Less than 1s for transferring control from the card to host machine. • User Interfaces: At host machine, graphic at LCD or touch screen