CMR INSTITUTE OF TECHNOLOGY

USN | | | | | | | | | |

Internal Assessment Test-II – May 2017

Sub:	**Embedded Systems**							Code:	**10EE665**
Date:	**09/05/2017**	Duration:	**90** mins	Max Marks:	**50**	Sem:	**VI**	Branch:	**EEE**

Note: Answer for FIFTY marks.

		Marks	OBE	
			CO	RBT
1.(a)	Explain the various design metrics of an embedded system.	[5]	CO2	L2
1.(b)	Explain Embedded C language program elements.	[5]	CO5	L1
2.	Explain the software hardware trade-off. Give the advantages of software implementation and hardware implementation.	[10]	CO4	L2
3.	Explain three main design technologies. How are these helpful to designers?	[10]	CO4	L1

. . . .

CMR INSTITUTE OF TECHNOLOGY

USN | | | | | | | | | |

Internal Assessment Test-II – May 2017

Sub:	**Embedded Systems**							Code:	**10EE665**
Date:	**09/05/2017**	Duration:	**90** mins	Max Marks:	**50**	Sem:	**VI**	Branch:	**EEE**

Note: Answer for FIFTY marks.

		Marks	OBE	
			CO	RBT
1.(a)	Explain the various design metrics of an embedded system.	[5]	CO2	L2
1.(b)	Explain Embedded C language program elements.	[5]	CO5	L1
2.	Explain the software hardware trade-off. Give the advantages of software implementation and hardware implementation.	[10]	CO4	L2
3.	Explain three main design technologies. How are these helpful to designers?	[10]	CO4	L1

.

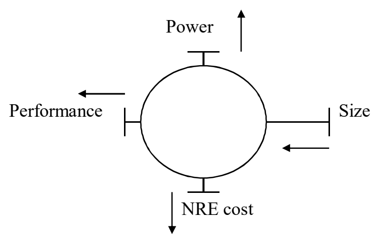
		Marks	OBE	
			CO	RBT
4.(a)	What is a re-entrant function? List the rules to check if a function is re-entrant or not.	[**6**]	CO5	L1
4.(b)	Differentiate functions and macros.	[**4**]	CO5	L1
5.	Explain Round Robin architecture with interrupts with the help of its pseudo-code. Also discuss the drawbacks of this architecture.	[**10**]	CO5	L2
6.	Define the following data structures used in C. When can they be used? Give examples for each i) Queue ii) Stack iii) Array iv) Tree.	[**10**]	CO5	L1
7.	Explain the RTOS task scheduling models.	[**10**]	CO5	L2

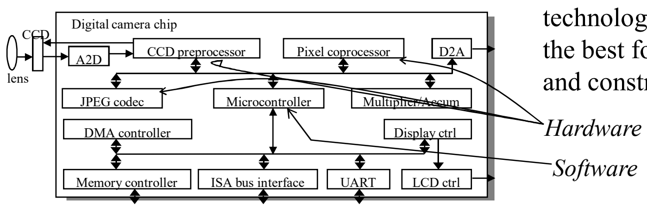
. . . .

		Marks	OBE	
			CO	RBT
4.(a)	What is a re-entrant function? List the rules to check if a function is re-entrant or not.	[**6**]	CO5	L1
4.(b)	Differentiate functions and macros.	[**4**]	CO5	L1
5.	Explain Round Robin architecture with interrupts with the help of its pseudo-code. Also discuss the drawbacks of this architecture.	[**10**]	CO5	L2
6.	Define the following data structures used in C. When can they be used? Give examples for each i) Queue ii) Stack iii) Array iv) Tree.	[**10**]	CO5	L1
7.	Explain the RTOS task scheduling models.	[**10**]	CO5	L2

Q1. A

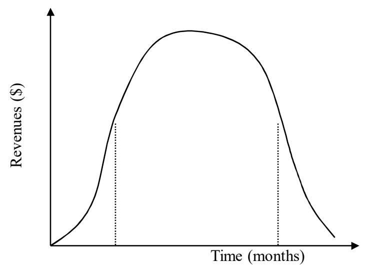
Design metric competition -- improving one may worsen others





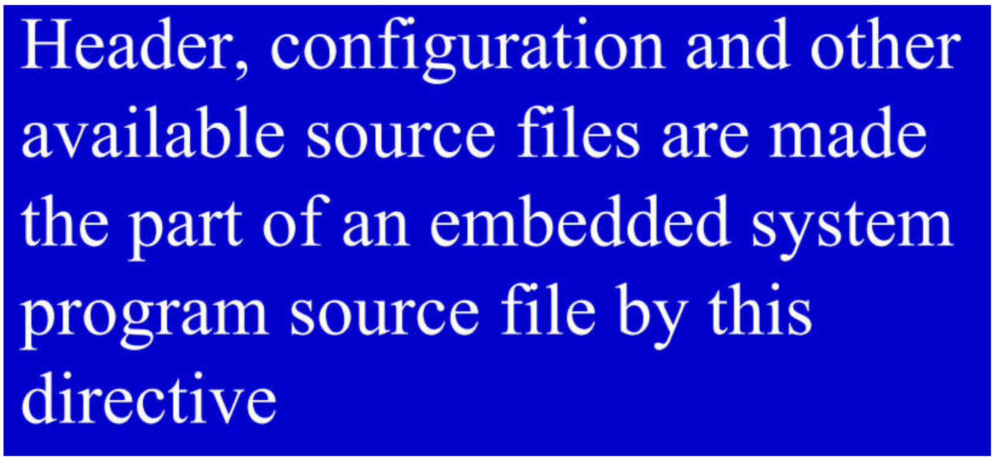
- Expertise with both **software and hardware** is needed to optimize design metrics
  - Not just a hardware or software expert, as is common
  - A designer must be comfortable with various technologies in order to choose the best for a given application and constraints

Time-to-market: a demanding design metric

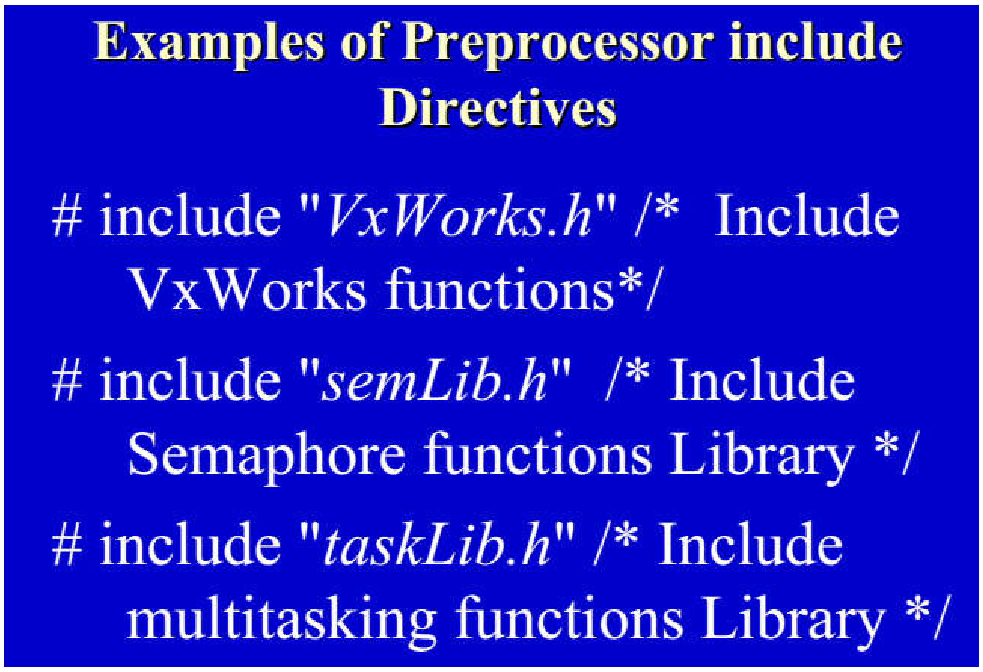


- Time required to develop a product to the point it can be sold to customers
- Market window
  - Period during which the product would have highest sales
- Average time-to-market constraint is about 8 months
- Delays can be costly

Q1.b



Header, configuration and other available source files are made the part of an embedded system program source file by this directive

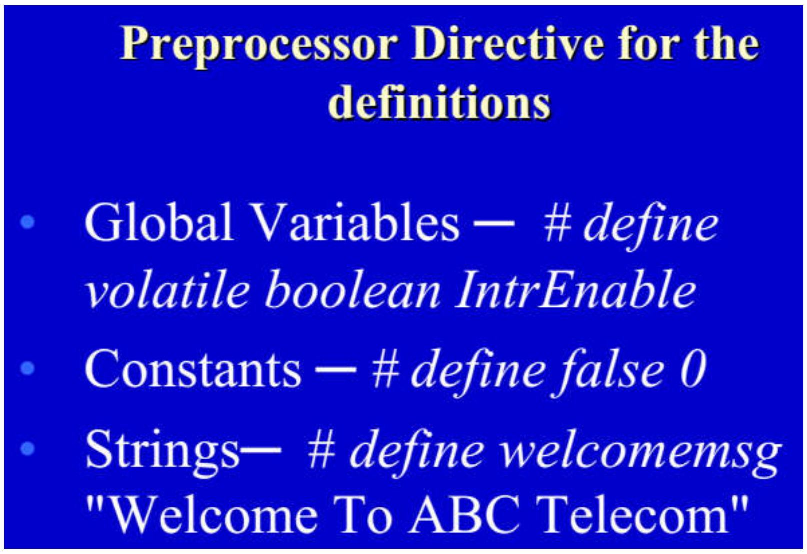


Examples of Preprocessor include Directives

include "*VxWorks.h*" /* Include VxWorks functions*/

include "*semLib.h*" /* Include Semaphore functions Library */

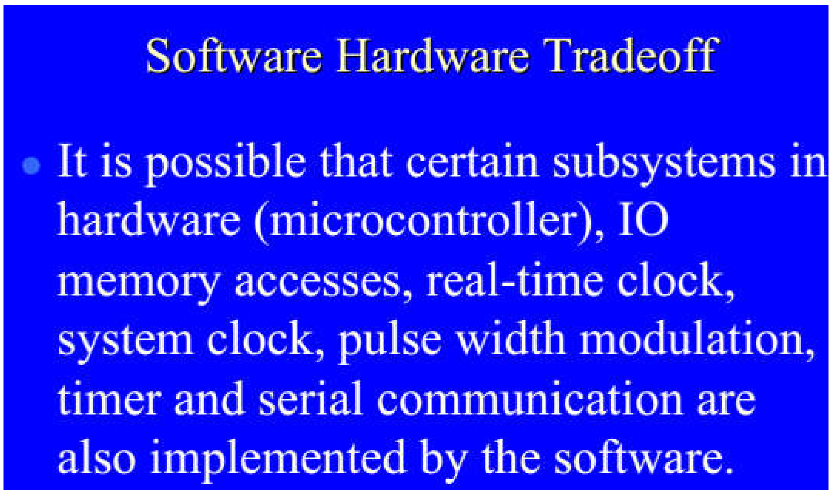
include "*taskLib.h*" /* Include multitasking functions Library */



Preprocessor Directive for the definitions

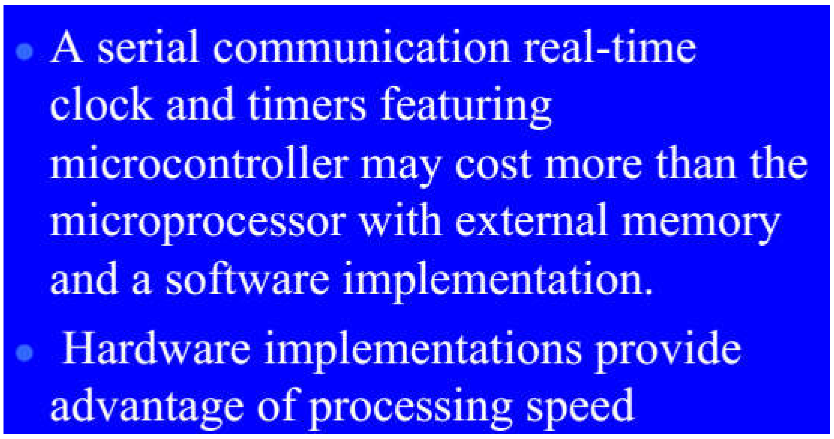
- Global Variables — *# define volatile boolean IntrEnable*
- Constants — *# define false 0*
- Strings— *# define welcomemsg* "Welcome To ABC Telecom"

Q2.

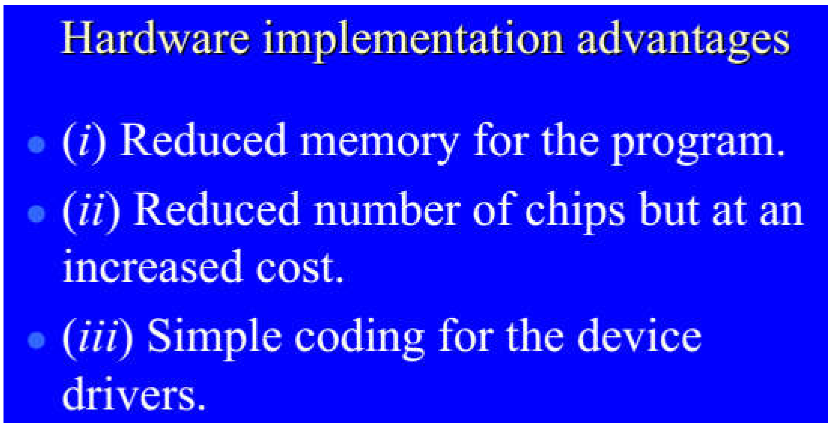


Software Hardware Tradeoff

- It is possible that certain subsystems in hardware (microcontroller), IO memory accesses, real-time clock, system clock, pulse width modulation, timer and serial communication are also implemented by the software.

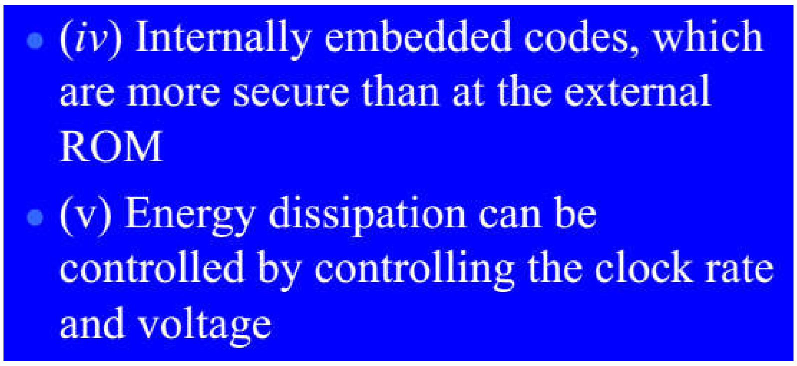


- A serial communication real-time clock and timers featuring microcontroller may cost more than the microprocessor with external memory and a software implementation.
- Hardware implementations provide advantage of processing speed

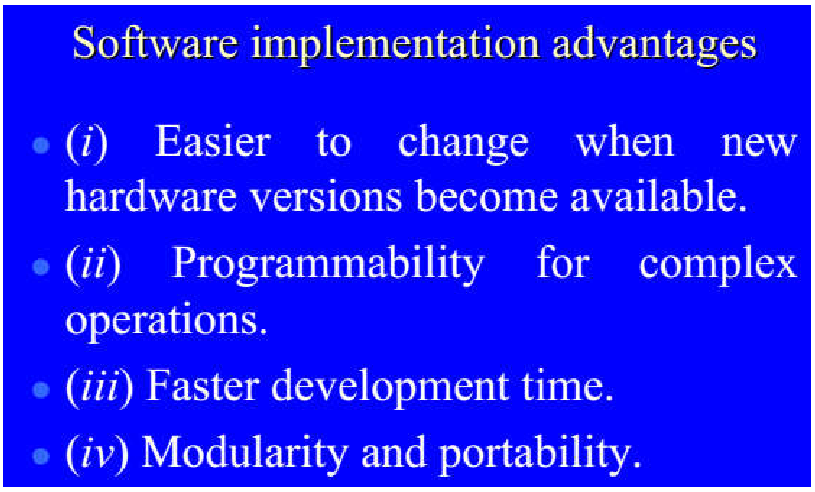


Hardware implementation advantages

- (*i*) Reduced memory for the program.
- (*ii*) Reduced number of chips but at an increased cost.
- (*iii*) Simple coding for the device drivers.

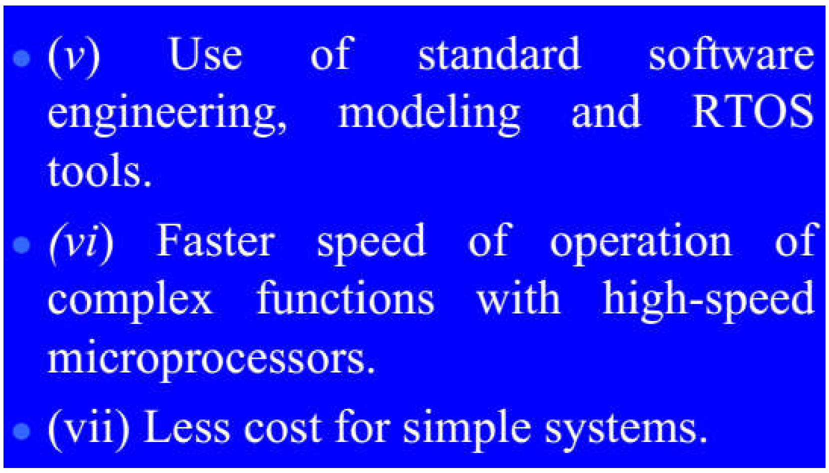


- (*iv*) Internally embedded codes, which are more secure than at the external ROM
- (v) Energy dissipation can be controlled by controlling the clock rate and voltage



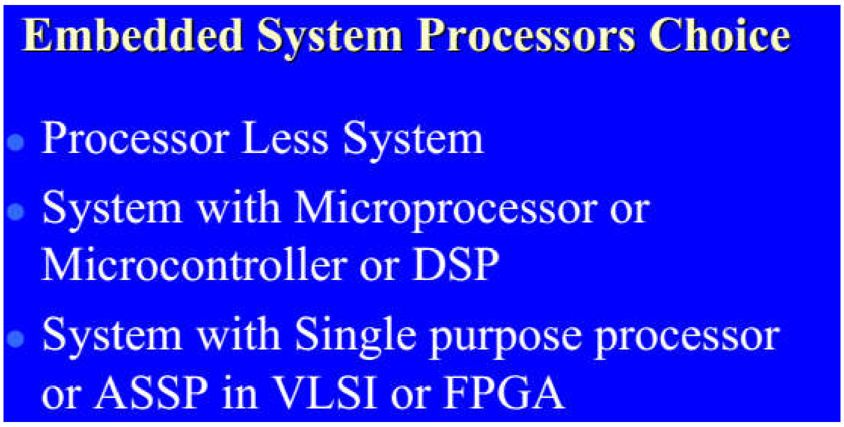
Software implementation advantages

- (*i*) Easier to change when new hardware versions become available.
- (*ii*) Programmability for complex operations.
- (*iii*) Faster development time.
- (*iv*) Modularity and portability.



- (*v*) Use of standard software engineering, modeling and RTOS tools.
- *(vi*) Faster speed of operation of complex functions with high-speed microprocessors.
- (vii) Less cost for simple systems.

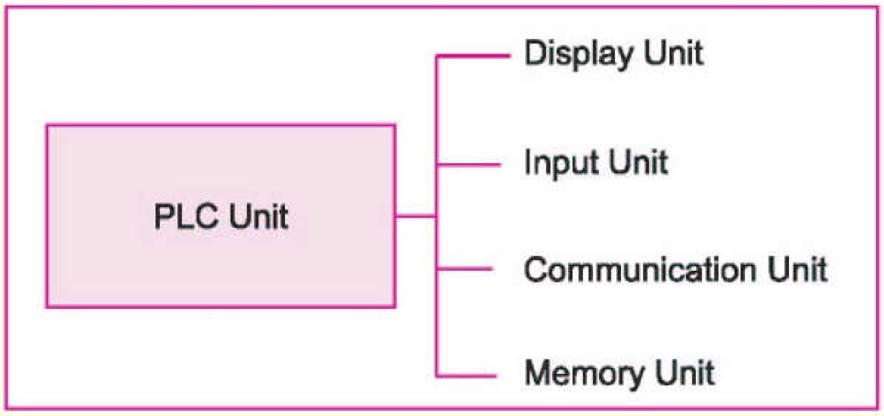
Q3.



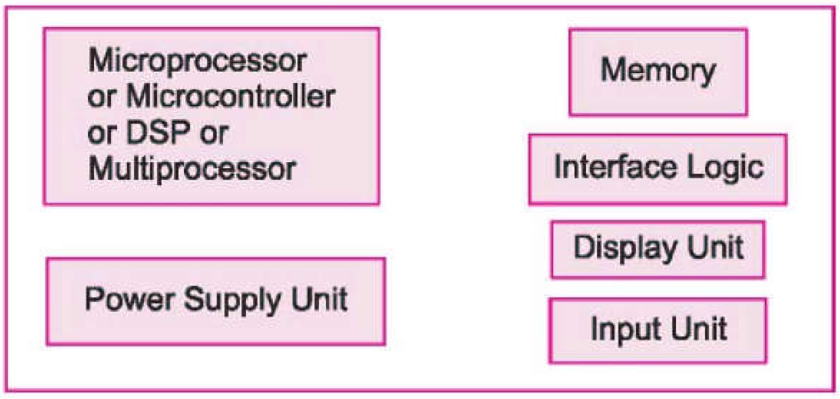
Embedded System Processors Choice

- Processor Less System
- System with Microprocessor or Microcontroller or DSP
- System with Single purpose processor or ASSP in VLSI or FPGA

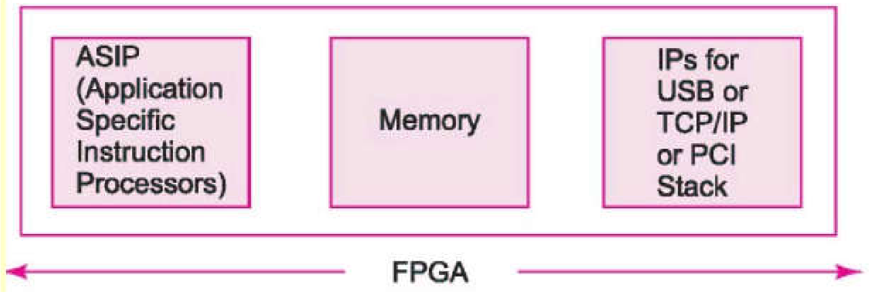
Processor Less System



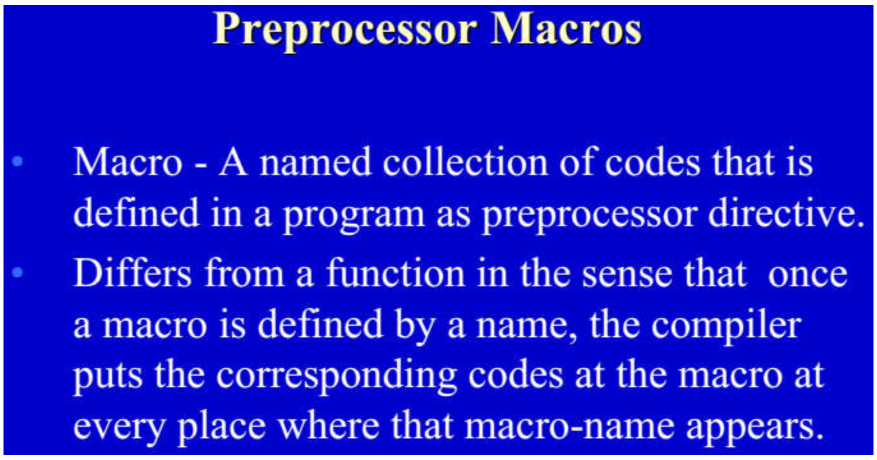
Microprocessor or Microcontroller or DSP



Processing of functions by using IP embedded into the FPGA instead of processing by the ALU

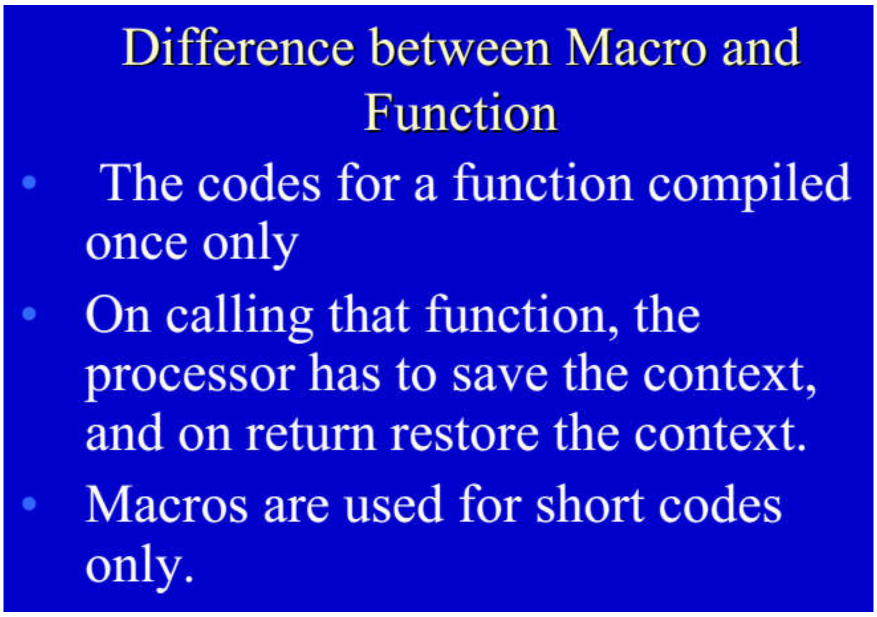

ASIP (Application Specific Instruction Processors)
Memory
IPs for USB or TCP/IP or PCI Stack
FPGA

4.b



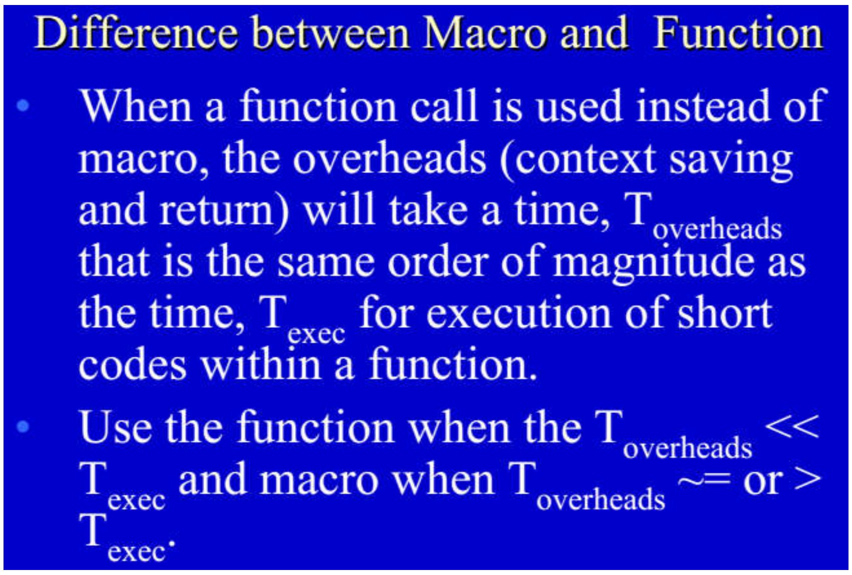
Preprocessor Macros

- Macro - A named collection of codes that is defined in a program as preprocessor directive.
- Differs from a function in the sense that once a macro is defined by a name, the compiler puts the corresponding codes at the macro at every place where that macro-name appears.



Difference between Macro and Function

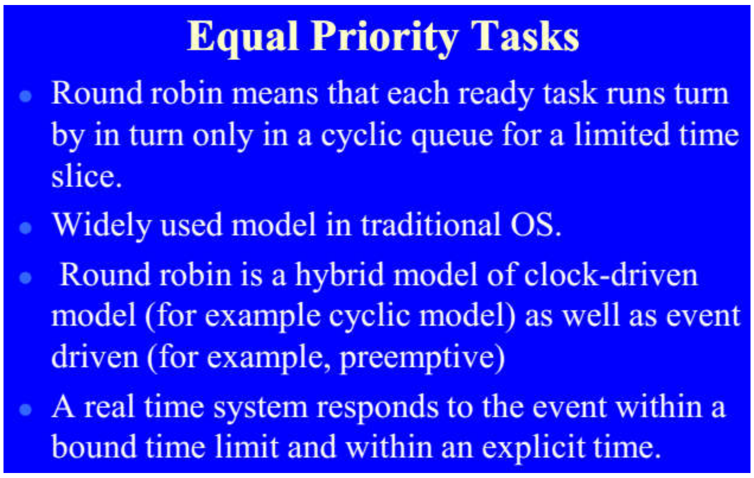
- The codes for a function compiled once only
- On calling that function, the processor has to save the context, and on return restore the context.
- Macros are used for short codes only.



Difference between Macro and Function

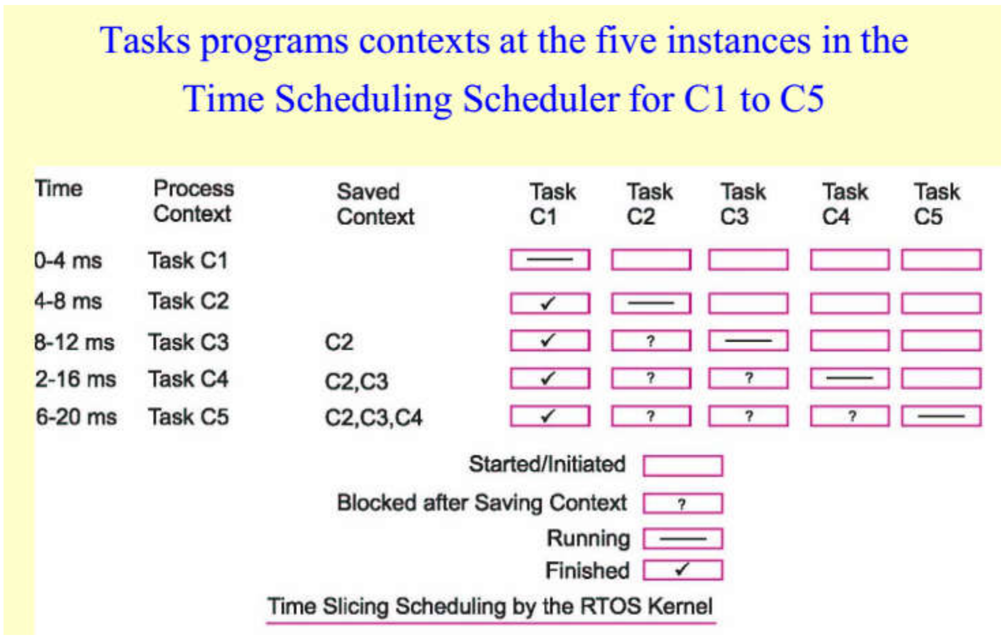
- When a function call is used instead of macro, the overheads (context saving and return) will take a time, $T_{overheads}$ that is the same order of magnitude as the time, T_{exec} for execution of short codes within a function.
- Use the function when the $T_{overheads} << T_{exec}$ and macro when $T_{overheads}$ ~= or > T_{exec}.

Q5.



Equal Priority Tasks

- Round robin means that each ready task runs turn by in turn only in a cyclic queue for a limited time slice.
- Widely used model in traditional OS.
- Round robin is a hybrid model of clock-driven model (for example cyclic model) as well as event driven (for example, preemptive)
- A real time system responds to the event within a bound time limit and within an explicit time.



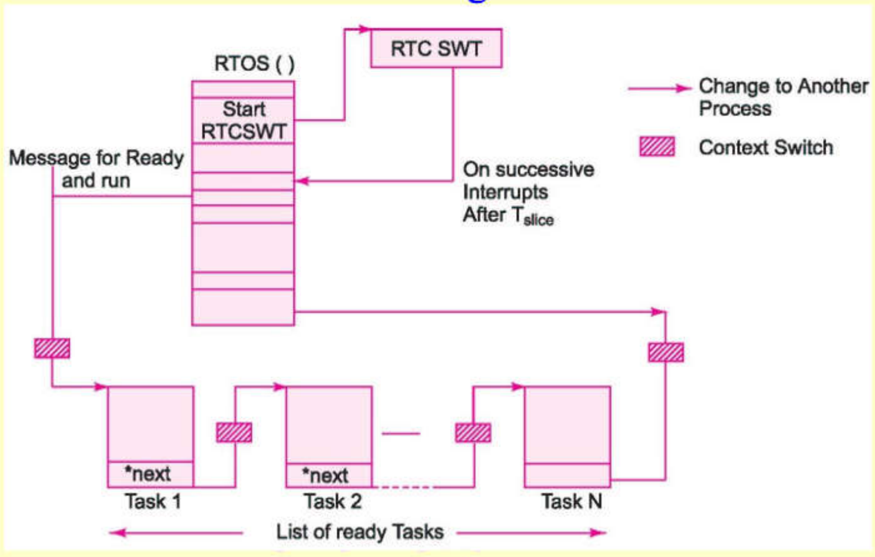
Tasks programs contexts at the five instances in the Time Scheduling Scheduler for C1 to C5

Time	Process Context	Saved Context	Task C1	Task C2	Task C3	Task C4	Task C5
0-4 ms	Task C1		—				
4-8 ms	Task C2		✓	—			
8-12 ms	Task C3	C2	✓	?	—		
2-16 ms	Task C4	C2,C3	✓	?	?	—	
6-20 ms	Task C5	C2,C3,C4	✓	?	?	?	—

Started/Initiated	
Blocked after Saving Context	?
Running	—
Finished	✓

Time Slicing Scheduling by the RTOS Kernel

Programming model for the Cooperative Time sliced scheduling of the tasks


RTOS ( )
Start RTCSWT
RTC SWT
Message for Ready and run
On successive Interrupts After T_{slice}
Change to Another Process
▨ Context Switch
*next
Task 1
*next
Task 2
Task N
List of ready Tasks

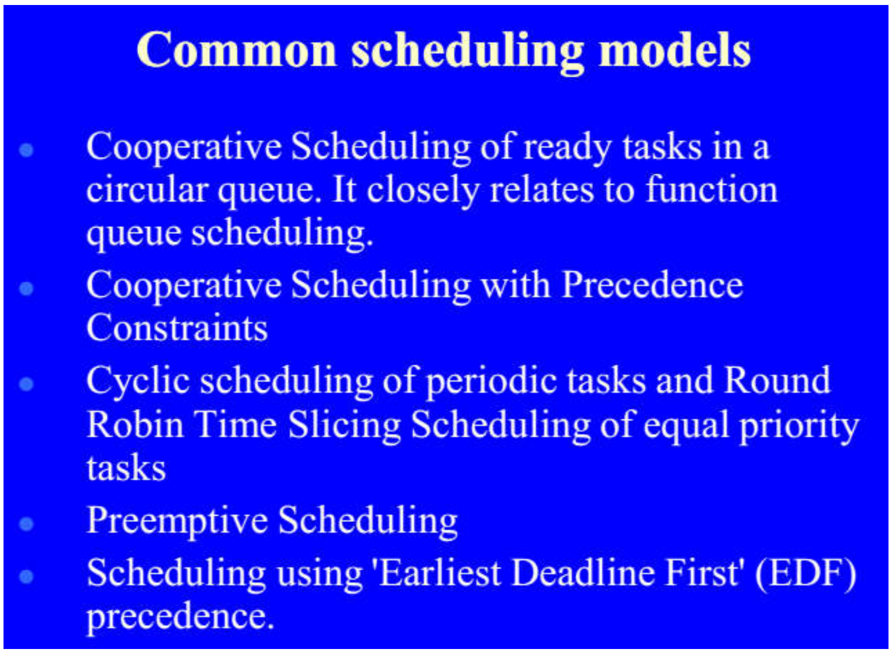
Round Robin

- Case 1: Then each task is executed once and finishes in one cycle itself.
- When a task finishes the execution before the maximum time it can takes, there is a waiting period in-between period between two cycles.
- The worst-case latency for any task is then N × t_{slice}. A task may periodically need execution. A task The period for the its need of required repeat execution of a task is an integral multiple of t_{slice}.

Case 2: Alternative model strategy

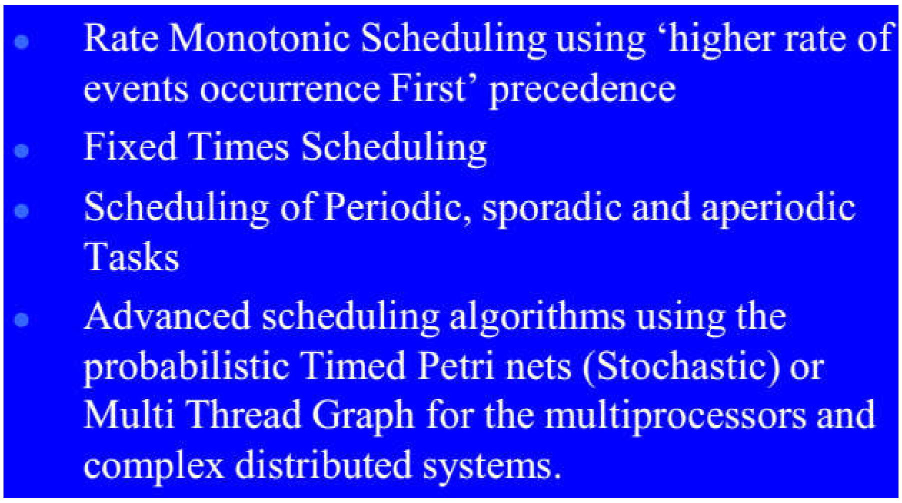
- Case 2: Certain tasks are executed more than once and do not finish in one cycle
- *Decomposition* of a task that takes the abnormally long time to be executed.
- The decomposition is *into two or four or more tasks*.
- Then one set of tasks (or the odd numbered tasks) can run in one time slice, t'slice and the another set of tasks (or the even numbered tasks) in another time slice, t''slice.

Q7.

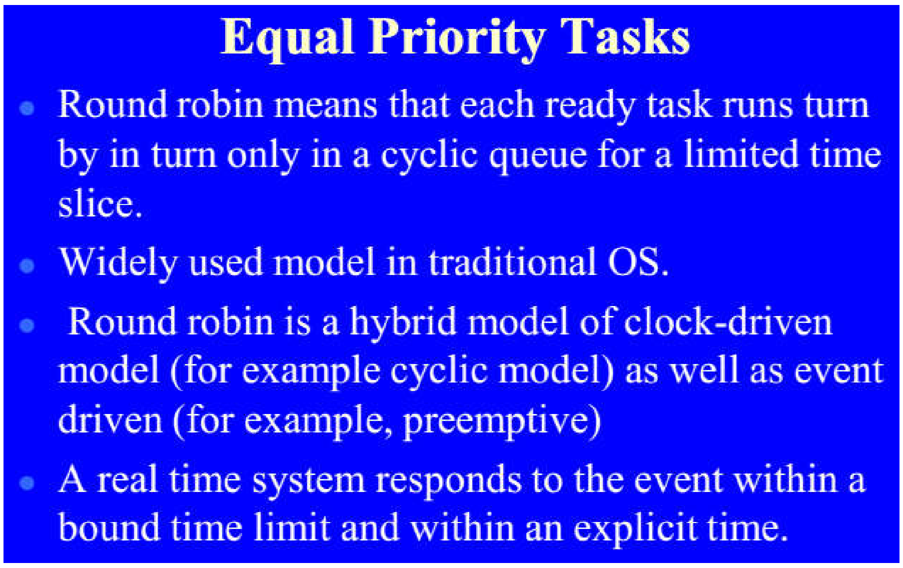


Common scheduling models

- Cooperative Scheduling of ready tasks in a circular queue. It closely relates to function queue scheduling.
- Cooperative Scheduling with Precedence Constraints
- Cyclic scheduling of periodic tasks and Round Robin Time Slicing Scheduling of equal priority tasks
- Preemptive Scheduling
- Scheduling using 'Earliest Deadline First' (EDF) precedence.



- Rate Monotonic Scheduling using 'higher rate of events occurrence First' precedence
- Fixed Times Scheduling
- Scheduling of Periodic, sporadic and aperiodic Tasks
- Advanced scheduling algorithms using the probabilistic Timed Petri nets (Stochastic) or Multi Thread Graph for the multiprocessors and complex distributed systems.



Equal Priority Tasks

- Round robin means that each ready task runs turn by in turn only in a cyclic queue for a limited time slice.
- Widely used model in traditional OS.
- Round robin is a hybrid model of clock-driven model (for example cyclic model) as well as event driven (for example, preemptive)
- A real time system responds to the event within a bound time limit and within an explicit time.