Question Bank on RTSA

Chapter-I: Introduction

1. Consider a payroll processing system for an elevator company. Describe three different scenarios in which the system can be justified as hard, firm, or soft real - time.
2. Discuss whether the following are hard, firm, or soft real - time systems:

a) The Library of Congress print - manuscript database system.
b) A police database that provides information on stolen automobiles.
c) An automatic teller machine in a shopping mall.
d) A coin - operated video game in some amusement park.
e) A university grade - processing system.
f) A computer - controlled routing switch used at a telephone company branch exchange.

3. Consider a real - time weapons control system aboard a fighter aircraft. Discuss which of the following events would be considered synchronous and which would be considered asynchronous to the real - time computing system.
   (a) A 5 - ms, externally generated clock interrupt.
   (b) An illegal - instruction - code (trap) interrupt.
   (c) A built - in - test memory failure.
   (d) A discrete signal generated by the pilot pushing a button to fi re a missile.
   (e) A discrete signal indicating “ low on fuel. ”
4. Describe a system that is completely nonreal - time, that is, there are no bounds whatsoever for any response time. Do such systems exist in reality?
5. For the following systems concepts, fill in the cells of Table as shown below with descriptors for possible events. Estimate event periods for the periodic events.

TABLE Taxonomy of Events and Some Typical Examples

	Periodic	Aperiodic	Sporadic
Synchronous	Cyclic code	Conditional branch	Divide-by-zero (trap) interrupt
Asynchronous	Clock interrupt	Regular, but not fixed-period interrupt	Power-loss alarm

   (a) Elevator group dispatcher: this subsystem makes optimal hall – call allocation for a bank of high - speed elevators that service a 40 – story building in a lively city like Louisville.
   (b) Automotive control: this on - board crash avoidance system uses data from a variety of sensors and makes decisions and affects behavior to avoid collision, or protect the occupants in the event of an imminent

collision. The system might need to take control of the automobile from the driver temporarily.

6. For the example systems introduced (inertial measurement, nuclear - power - plant monitoring, airline reservation, pasta bottling, and traffic -light control) enumerate some possible events and note whether they are periodic, aperiodic, or sporadic. Discuss reasonable response times for the events.
7. In the response – time calculation as shown below, the time from observing a passenger between the closing door blades and starting to reopen the elevator door varies between 305 and 515 ms. How could you further justify if these particular times are appropriate for this situation?

   Sensor Response Time: $t_{S\_min} = 5$ ms, $t_{S\_max} = 15$ ms, $t_{S\_mean} = 9$ ms.

   Hardware Response Time: $t_{HW\_min} = 1$ μs, $t_{HW\_max} = 2$ μs, $t_{HW\_mean} = 1.2$ μs.

   System Software Response Time: $t_{SS\_min} = 16$ μs, $t_{SS\_max} = 48$ μs, $t_{SS\_mean} = 37$ μs.

   Application Software Response Time: $t_{AS\_min} = 0.5$ μs, $t_{AS\_max} = 0.5$ μs, $t_{AS\_mean} = 0.5$ μs.

   Door Drive Response Time: $t_{DD\_min} = 300$ ms, $t_{DD\_max} = 500$ ms, $t_{DD\_mean} = 400$ ms.

   the minimum, maximum, and mean values of the composite response time: $t_{min} \approx 305$ ms, $t_{max} \approx 515$ ms, and $t_{mean} \approx 409$ ms.

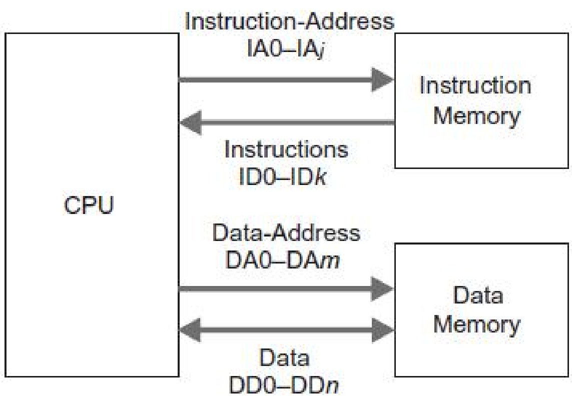
8. A control system is measuring its feedback quantity at the rate of 100 μ s. Based on the measurement, a control command is computed by a heuristic algorithm that uses complex decision making. The new command becomes available 27 – 54 μ s (rather evenly distributed) after each sampling moment. This considerable jitter introduces harmful distortion to the controller output. How could you avoid (reduce) such a jitter? What (if any) are the drawbacks of your solution?
9. Consider the CPU utilization factor, how short could the execution period of Task 1, $e1$, be made to maintain the CPU utilization zone no worse than " questionable "? Refer to table given below.

TABLE CPU Utilization (%) Zones

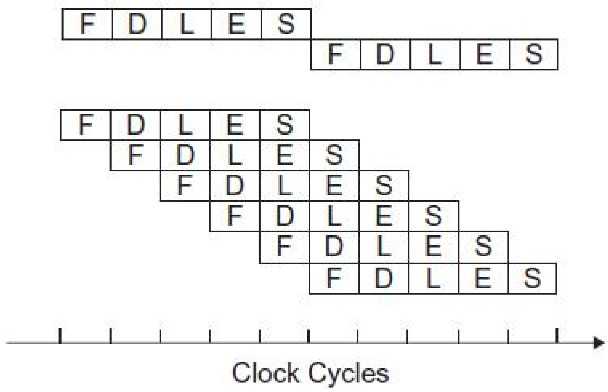
Utilization (%)	Zone Type	Typical Application
<26	Unnecessarily safe	Various
26–50	Very safe	Various
51–68	Safe	Various
69	Theoretical limit	Embedded systems
70–82	Questionable	Embedded systems
83–99	Dangerous	Embedded systems
100	Critical	Marginally stressed systems
>100	Overloaded	Stressed systems

Chapter- II: Hardware

1. Compose a table providing the available memory spaces for the following address - bus widths: 16, 20, 24, and 32 bits.
2. It is common practice for programmers to create continuous test - and -loop code in order to poll I/O devices or wait for interrupts to occur. Some processors provide an instruction (WAIT or HALT) that allows the processor to hibernate until an interrupt occurs. Why is the latter form more efficient and desirable?
3. In general terms, suggest a possible scheme that would allow a machine - language instruction to be interruptible. What would be the overall effect on instruction ' s execution time and CPU ' s throughput and response times?
4. Figure 2.5 illustrates the interface lines of a generic memory component. Assume $m = 15$ and $n = 7$. The address bus of your microprocessor is 24 bits wide. How, in principle, could you locate this particular memory block to begin from the address 040000 (hexadecimal)? What is the corresponding end address?
5. Compare and contrast the different memory technologies discussed in this chapter as they pertain to embedded real - time systems.
6. How would you test the validity and integrity of factory parameters stored in EEPROM? Sketch a suitable procedure for that purpose.
7. Assume a hierarchical memory system having a joint instruction/data cache with a memory - access cost of 10 ns on a hit and 90 ns on a miss. An alternative design without hierarchical memory architecture has a memory - access cost of 70 ns. What is the minimum cache - hit percentage that would make the hierarchical memory system useful?
8. The Harvard architecture as shown in figure below offers separate address and data buses for instruction codes and data. Why is not it feasible to have separate buses for programmed I/O as well?



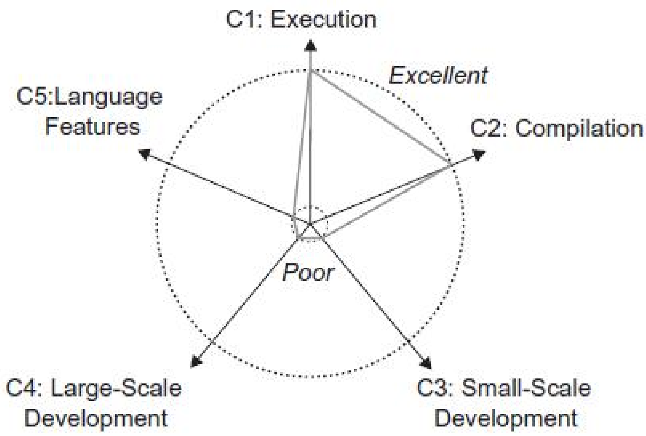
9. Show with an illustrative example how the five - stage pipeline as shown in figure below, what could the benefit from the Harvard architecture.



10. What special problems do super-pipelined and superscalar architectures pose for real - time system designers? Are they any different for nonreal - time systems?
11. In CISC - type processors, most instructions are having memory operands, while RISC - type processors access memory by LOAD and STORE instructions only. What are the advantages and disadvantages of both schemes?
12. You are designing the architecture of a high - performance CPU for hard real - time applications. List and justify the principal architectural selections that you would make.
13. Discuss the relative advantages and disadvantages of memory – mapped I/O, programmed I/O, and DMA as they pertain to real - time systems.
14. Why is DMA controller access to main memory in most systems given higher priority than CPU access to main memory?
15. An embedded system has a 12 - bit A/D converter for measuring voltages between − 10 V and + 10 V. What is the digital value corresponding to + 5.6 V?
16. Find a microcontroller with unique, special instructions and, considering the application area for that processor, discuss the need for those special instructions.
17. What are the advantages of systems on chip over computers on chip? Find a few examples of commercial systems on chip from the Web.
18. A watchdog timer is used for supervising the operation of an embedded system in a high - EMI environment. Why is it practical to connect the watchdog - circuit ' s output to the CPU 's non-maskable (instead of maskable) interrupt input?
19. List the different data - transmission media mentioned in this chapter and give typical applications for each.

Chapter- III: Programming Language

1. It can be argued that in some cases there exists an apparent conflict between good software engineering practices and real - time performance. Consider the relative merits of recursive program design versus interactive techniques, and the use of global variables versus parameter lists. Using these topics and an appropriate programming language for examples, compare and contrast real - time performance versus good software engineering practices as you understand them.
2. What programming restrictions should be used in a programming language to permit straightforward analysis of real - time applications?
3. Write a set of coding standards for use with any of the real - time applications for the programming language of your choice. Document the rationale for each provision of the coding standard.
4. Why is it very important to cite the reference for any computational algorithm that is used in a real - time program in the program 's annotation?
5. In a procedural language of your choice, develop an abstract data type called "image" with associated functions. Be sure to follow the principle of information hiding. Make any assumptions that you need to about the properties of the images.
6. In the object - oriented language of your choice, design and code an " image " class that could be useful across a wide range of projects. Be sure to follow the best principles of object - oriented design.
7. How can misuse or misunderstanding of a software technology impede a software project? For instance, writing structured C code instead of classes in C ++ , or reinventing a tool for each project instead of using a standard one.
8. Java has been compared with Ada 95 in terms of "hype" and "unification " — defend or refute the arguments for this comparison.
9. By using the five metrics of Cardelli, compare the fitness of C and C ++ languages for real - time programming; use pentacle diagrams as shown below for visualizing your justified comparison.



10. Are there any language features that are exclusive to C/C ++ ? Do these features provide specific advantage or disadvantage in embedded environments?
11. You are hired to defi ne a set of principal requirements for a new real -time programming language for embedded control applications. What are the most important requirements that your defi nition would contain? Justify your answer.
12. What compiler options are available in your favorite C compiler and what do they specifically do?
13. Develop a set of tests to exercise a compiler to determine the best use of the language in a real - time processing environment. For example, your tests should determine such things as when to use case statements versus nested if-then-else statements; when to use integers versus Boolean variables for conditional branching; whether to use while or for loops, and when; and so on.
14. Use standard compiler - optimization methods and multiple optimization phases to optimize the following C code by hand:

```
#define UNIT 1
#define FULL 1
void main(void)
{
int a,b;
a=FULL;
b=a;
if ((a ==FULL) && (b ==FULL))
{
if (debug)
printf("a=%d b =%d",a,b);
a=(b*UNIT)/2;
```

```
a=2.0*a*4;
b=b*sqrt(a);
}
}
```