Review Questions and Problems Chapter 1

1. What makes a computer system a *real-time* computer system?

2. What are *typical functions* a real-time computer system must perform?

3. Where do the *temporal requirements* come from? What are the parameters that describe the temporal characteristics of a controlled object? Draw a picture illustrating these parameters.

4. Give a „rule of thumb" that relates the *sampling period* in a quasi-continuous system to the rise time of the step-response function of a controlled object. Draw a picture to illustrate this.

5. What are the effects of delay and delay jitter on the quality of control? Compare the error-detection latency in systems with and without jitter.

6. What does *signal conditioning* mean?

7. Consider an RT entity that changes its value periodically according to $v(t) = A_0 \sin(2\pi t/T)$ where T, the period of oscillation, is *100* msec.
   What is the maximum change of value of this RT entity within a time interval of *1* msec (express the result in percentage of the amplitude A_0).

8. Consider an engine that rotates with *3000* rpm. By how many degrees will the crankshaft turn within *1* msec?

9. What is a real-time entity? Give an example for a real-time entity that is not in the sphere of control of a related subsystem.

10. What is a *critical failure mode*? Give examples.

11. What is the difference between *availability* and *reliability*? What is the relationship between *maintainability* and *reliability*?

12. When is there a simple relation between the MTTF and the failure rate?

13. Assume you are asked to certify a safety-critical control system. How would you proceed?

14. What are the main differences between a soft real-time system and a hard real-time system?

15. Estimate and discuss the relation (development cost)/(production cost) in an embedded application and in a plant automation system.

16. What knowledge areas are involved in the design of real-time systems? Give an example for each area.

17. Discuss the advantages and disadvantages of event triggered vs. time triggered systems.

(Most questions taken from the text of Kopetz)

Review Questions and Problems Chapter 2

1. What are the advantages of an architecture where every logical function is implemented in a self-contained hardware unit? What are the disadvantages?

2. Why is it important to have stable and testable interfaces among the subsystems of a large system? What is the cost associated with such interfaces?

3. What are the differences between *event-message semantics* and *state-message semantics*?

4. Why is it not possible to specify the temporal properties of the CNI in event-triggered communication systems? What are the consequences of the missing temporal specifications of the CNI on the temporal composability?

5. Discuss the *responsiveness* of a CNI that is supporting *state messages*. Compare this *responsiveness* to that of a CNI that supports *event messages*.

6. Why is it difficult to define *error-containment regions* in a centralized architecture?

7. What is a "*safety case*"?

8. Give examples of faults of different *criticality* in control systems of a car.

9. Discuss the requirements of Safety Integrity Levels SIL1 to SIL 4 according to standard IEC61508.

10. How can an architecture support certification of safety-critical real-time systems?

11. Discuss the advantages/disadvantages of distributed real-time system from the viewpoint of physical installation.

12. What is "*mechatronics*"?

Review Questions and Problems Chapter 3

1. What is the difference between an *instant* and an *event*?
2. What is the difference between *temporal* order, *causal* order, and a consistent *delivery* order of messages? Which of the orders imply each other?
3. How can clock synchronization assist in finding the *primary event* of an alarm shower?
4. What is the difference between *UTC* and *TAI*? Why is TAI better suited as a time base for distributed real-time systems than UTC?
5. Define the notions of *offset*, *drift*, *drift rate*, *precision*, and *accuracy*.
6. What is the difference between *external synchronization* and *internal synchronization*?
7. What are the fundamental limits of time measurement?
8. When is an event set π/Δ-*precedent*?
9. What is an *agreement protocol*? Why should we try to avoid agreement protocols in real-time systems? When is it impossible to avoid agreement protocols?
10. What is a *sparse time base*? How can a sparse time base help to avoid agreement protocols?
11. Give an example that shows that, in an ensemble of three clocks, a *Byzantine clock* can disturb the two good clocks such that the synchronization condition is violated.
12. Given a *clock synchronization system* that achieves a precision of 90 microseconds, what is a reasonable *granularity* for the global time? What are the limits for the observed values for a time interval of 1.1 msec?
13. What is the role of the *convergence function* in internal clock synchronization?
14. Given a latency jitter of 20 µsec, a clock drift rate of 10^{-5} sec/sec, and a resynchronization period of 1 second, what *precision* can be achieved by the *central master algorithm*?
15. What is the effect of a Byzantine error on the quality of synchronization by the *FTA algorithm*?
16. Given a latency jitter of 20 µsec, a clock drift rate of 10^{-5} sec/sec, and a resynchronization period of 1 second, what *precision* can be achieved by the *FTA algorithm* in a system with 10 clocks where 1 clock could be malicious?
17. Discuss the consequences of an error in the *external clock synchronization*. What effect can such an error have on the internal clock synchronization in the worst possible scenario?

Review Questions and Problems Chapter 4

1. What is *assumption coverage*? How can you determine a quantitative value for the assumption coverage? What do we mean by *load hypothesis* and *fault hypothesis*?

2. List the properties that must be part of an architectural model of a real-time system and the properties that can be disregarded in such a model?

3. Describe the structure of a node. Why is it important to distinguish between the *i-state* and the *h-state* in an embedded system?

4. Describe the elements of an interface. What is the difference between *functional intent* and *function*? What are the characteristics of *world interfaces* and *message interfaces*? Give examples of standardized message interfaces.

5. What are the *temporal obligations* of clients and servers at a *client-server interface* in a real-time system?

6. What is the difference between *temporal control* and *logical control*?

7. Calculate the *overhead* of a *trigger task* if the *WCET* of the trigger task is 200 µsec and the laxity of an RT transition is 10 msec. Discuss the advantages and disadvantages of an application-task activation by an interrupt versus that by a trigger task.

8. What are the effects of *pipelining* and *caching* on the *WCET*? Assume that an interrupt must be serviced during the execution of a task. How is the WCET of the task affected?

9. Assume that there is a large difference between the *experimentally observed WCET* and the *analytically calculated WCET*. What can you learn from this difference? How can you reduce the difference? What are the problems with the experimental measurement of the WCET?

10. Assume that an instruction cache has a cycle time of 20 nsec. If an instruction resides in the cache the access time is one cycle, while the penalty of a cache miss is 8 extra cycles. The cache size is 256 instructions. What is the worst-case variability of the microarchitecture delay caused by cache reloading?

    Assume that a processor has an instruction “Perform the operation without using the cache” and that the time for the two context switches and the interrupt service by task B on slide 4.22 is 50 µsec if the caches are bypassed. What would be the effect of such a microarchitecture on the WCET?

Review Questions and Problems Chapter RT02-RT-Entities.pdf

1. Give examples of *RT entities* that are needed to control an automotive engine. Specify the static and dynamic attributes of these RT entities, and discuss the *temporal accuracy* of the *RT images* associated with these RT entities.

2. What is the difference between a *state observation* and an *event observation*? Discuss their advantages and disadvantages.

3. What are the problems with *event observations*?

4. Give an informal and a precise definition of the concept of *temporal accuracy*. What is the *recent history*?

5. What is the difference between a *parametric RT image* and a *phase-sensitive RT image*? How can we create parametric RT images?

6. What are the inputs to a *state estimation model*? Discuss state estimation in a system with and without a global time-base.

7. Discuss the interrelationship between *state estimation* and *composability*.

8. What is a *hidden channel*? Define the notion of *permanence*.

9. Calculate the *action delay* in a distributed system with the following parameters: dmax=20 msec, dmin=1 msec, and
   (a) no global time available, and the granularity of the local time is 10 µsec,
   (b) granularity of the global time 20 µsec.

10. What is the relationship between *action delay* and *temporal accuracy*?

11. Define the notion of *replica determinism*. Give an example of a *major decision point* that can be found in almost any application.

12. Give an example that shows that a local time-out can lead to *replica non-determinism*. Why can dynamic preemptive scheduling cause replica non-determinism?

13. What mechanism may lead to *replica non-determinism*?

14. How can we build a *replica-determinate system*?

15. Why should explicit *inter-replica coordination* be avoided?

Review Questions and Problems Chapter 6

1. Give the precise meaning of the terms *failure*, *error*, and *fault*.

2. What are the typical *permanent* and *transient failure rates* of VLSI chips?

3. The following fault is observed in the field: before installation, the proper operation of each of a batch of single chip microcontrollers was tested at the usual test points of -20C˚C, 0˚C, +20˚C, +40˚C, +60˚C, and at +80˚C. During operation, every fifth chip from the batch failed at about -12˚C, although it operated correctly at -20˚C and 0˚C.

   How would you classify this fault? How can this fault be detected if this chip is part of a large distributed system? What is the probability that a TMR system built out of three such microcontrollers would fail at -12˚C?

4. Why is a short *recovery time* from transient faults important?

5. What are the basic techniques for *error detection*? Compare ET systems and TT system from the point of view of error detection.

6. Discuss the topic of *exception handling* in real-time systems.

7. Discuss the different types of *errors* that can be *detected* by *redundant computations*.

8. What is a *membership service*? Give a practical example for the need of a *membership service*. What is the *quality parameter* o the membership service? How can you implement a membership service in an ET architecture?

9. What is the most serious *error* in a distributed system with a shared communication channel, e.g., a bus? Why?

10. Assume a computer system that can control three concurrently operating trains running on a model railway track, containing 10 switches and 15 signals.

    Identify the *h-state* at the reintegration point. Which part of the *h-state* can be enforced on the environment at the reintegration point? What is the minimal remaining *h-state* at the reintegration point?

11. What is a *restart vector*? Give an example.

12. *Fault tolerance* can be implemented by *two fail-silent components* or by *TMR*. Discuss the advantages and disadvantages of each one of these methods.

13. What are the arguments for, and against, using *diverse hardware units* in a safety-critical real-time application with replicated hardware channels?

14. What are the advantages and limits of *design diversity*? Why is it easier to deploy design diversity in fail-safe applications than in fail-operational applications?

Review Questions and Problems Chapter 7

1. Compare the *requirements* of *real-time communication systems* with those of non real-time communication systems. What are the most significant differences?

2. What are special *requirements* of a communication system for a *safety critical application*? Why should the *SRUs* forming an *FTU* be physically separated?

3. Why are *end-to-end protocols* needed at the interface between the computer system and the controlled object?

4. Which subsystem controls the *speed of communication* if an explicit flow control schema is deployed?

5. Calculate the *latency jitter* of a high level *PAR protocol* that allows three retries, assuming that the lower level protocol used for this implementation has a dmin of 2 msec and a dmax of 20 msec. Calculate the *error detection latency* at the sender.

6. Compare the *efficiency* of event-triggered and time-triggered communication protocols at low load and peak load.

7. What mechanism can lead to *trashing*? How should you react in an event-triggered system if trashing is observed?

8. What are the characteristics of *OSI based protocols*? How do they match with the requirements of hard real-time systems?

9. How is the information organized in an *ATM* system? Discuss the suitability of ATM systems for the implementation of wide-area real-time systems.

10. What are the main *differences* between a *field bus*, a *real-time network*, and a *backbone network*?

11. Discuss the fundamental *conflicts* in the requirements imposed on a *real-time protocol*.

12. Given a bandwidth of 500 MBits/sec, a channel length of 100 m and a message length of 80 bits, what is the *limit of the protocol efficiency* that can be achieved at the media access level of a bus system?

13. How do the nodes in a *CAN system* decide which node is allowed to access the bus?

14. Explain the role of the three time-outs in the *ARINC 629 protocol*. Is it possible for a collision to occur on an ARINC 629 bus?

15. Compare the *requirements* of *real-time communication systems* with those of non real-time communication systems. What are the most significant differences?

16. What are special *requirements* of a communication system for a *safety critical application*? Why should the *SRUs* forming an *FTU* be physically separated?

17. Why are *end-to-end protocols* needed at the interface between the computer system and the controlled object?

18. Which subsystem controls the *speed of communication* if an explicit flow control schema is deployed?

19. Compare the *efficiency* of event-triggered and time-triggered communication protocols at low load and peak load.

20. What mechanism can lead to *trashing*? How should you react in an event-triggered system if trashing is observed?

21. What are the characteristics of *OSI based protocols*? How do they match with the requirements of hard real-time systems?

22. What are the main *differences* between a *field bus*, a *real-time network*, and a *backbone network*?

23. Discuss the fundamental *conflicts* in the requirements imposed on a *real-time protocol*.

24. How do the nodes in a *CAN system* decide which node is allowed to access the bus?

25. Explain the role of the three time-outs in the *ARINC 629 protocol*. Is it possible for a collision to occur on an ARINC 629 bus?

Review Questions and Problems Chapter 8

1. What *services* are provided by the *TTP/C protocol*?

2. How is the *regularity* inherent in the *TDMA access strategy* used to increase the data *efficiency* of the protocol and to improve the *robustness* of the protocol?

3. Explain the *programming interface* of a *TTP controller*. What are the contents of the status area and the control area of the CNI? What are the contents of the status byte of each message?

4. How is the *consistency* of the data transfer across the CNI enforced by the TTP protocol?

5. Why is the control data structure that controls the protocol operation stored in the TTP controller and not in the host?

6. What mechanism helps to ensure the *fail-silence* of a TTP controller in the *temporal domain*?

7. What system must implement the *fail-silence* in the *value domain*?

8. What are the *differences* between the *TTP/C* protocol and the *TTP/A* protocol?

9. What is the *controller state* (C-state) of a TTP/C controller? How is the agreement of the C-state enforced within an ensemble?

10. Explain the operation of the *membership service* of the TTP/C protocol. How is the situation that a node does not receive a message from its immediate predecessor resolved? (In this scenario the node does not know if its incoming link is faulty or the predecessor has not sent a correct message).

11. Explain the *clock synchronization* of the *TTP/C* protocol.

12. Sketch the contents of the *Message Descriptor List* (MEDL) that controls the protocol operation.

13. What is the difference between an *immediate mode change* and a *delayed mode change*?

14. What is the *frame format* of a *TTP/C frame* on the network? What are the contents of the header byte?

15. Explain the *principle of operation* of the *TTP/A protocol*. Describe the concept of a "*round*".

16. How can one distinguish between a *Fireworks byte* and a *data byte* in the *TTP/A* protocol?

17. Estimate the *average* and *worst-case response time* of a *TTP/C* system with 5 FTUs, each one consisting of two nodes that exchange messages with 6 data bytes on a channel with a bandwidth of 1 Mbit/sec. Assume that the interframe gap is 8 bits.

18. Calculate the data efficiency of a TTP/A system that consists of 8 nodes where each node sends periodically a two byte message (user data). Assume that the intermessage gap between the Fireworks byte and the first data byte is 4 bitcells, and the intermessage gap between two successive data bytes is two bitcells. The gap between the end of one round and the start is 6 bitcells. What is the data efficiency of a functionally equivalent CAN system with a two byte data field? Assume that the intermessage gap in the CAN system is 4 bitcells.

Review Questions and Problems Chapter 12

1. What is a "*safety case*"?
2. What properties of the *architecture* support the design of a "*safety case*"?
3. List some *causes* for *common-mode failures* in a distributed system.
4. Discuss the different steps that must be taken to investigate a real-world phenomenon by a *formal method*. Which one of these steps can be formalized, which cannot?
5. In the lecture three different levels of formal methods have been introduced. Explain each one of these levels and discuss the costs and benefits of applying formal methods at each one of these levels.
6. What is the "*probe effect*"?
7. How can the *testability* of a design be improved?
8. What is the *role of testing* during the *certification* of an ultra-dependable system?
9. Which are the purposes of *fault-injection* experiments?
10. Compare the characteristics of hardware and software *fault-injection methods*.
11. Explain the notions of "*risk*" and "*hazard*".
12. Design a *fault-tree* for the brake-system of an automobile.