Topics and example questions for

Embedded Systems course

1. The Real-time Environment: Introduction to Real-time Systems

Reference: Kopetz, Ch. 1.
Topics:

- Definition of RT computer system, operator and controlled object, Man-Machine Interface, Instrumentation Interface. Deadlines: soft, hard, firm. HRT system.
- Functional requirements: data collection (sampling RT entities, signal conditioning, alarm monitoring), digital control (DDC), Man-machine interface
- Temporal requirements: object delay, rise time, sampling time constraints, computer delay. Jitter.
- Dependability requirements: Reliability, MTTF, safety, ultra-high reliability, maintainability, MTTR, trade-off between reliability and maintainability. Availability, relationship between MTTF, MTTR, MTBF, A. Security.
- Classification of RT system: Hard, soft. Characteristics (response time, peak-load performance, control of pace, safety, redundancy type). Fail-safe and fail-operational system. Event-triggered and time-triggered systems.

Example questions:

1. What is the difference between availability and reliability? Define ultra-high reliability with MTTF.
2. Define failure rate, MTTF, MTTR and MTBF and their relationship.
3. What is the relationship between reliability and maintainability?

2. Advantages of a distributed solution

Reference: Kopetz, Ch. 2. (parts)
Topics:

- Architecture: distributed system architecture, structure of a node (Host, CNI, CC)
- Event and state messages.

Example questions:

4. Draw the system level block-diagram of a distributed system. What is the structure of a node?
5. What is event and state message?

3. Global Time

Reference: Kopetz, Ch. 3.
Topics:

- Time, orders: temporal, causal and delivery.
- Clocks: physical clock, reference clock, perfect clock, real clock. Clock drift and drift rate. Failure modes: rate and state error.
- Offset between two clocks, precision and accuracy of a set of clocks. Relationship between precision and accuracy. Time standards: TAI, UTC. Internal and external synchronization.

- Definition of global time. Reasonableness condition. Temporal order of events noticed by different nodes. Interval measurement. Dense and Sparse time base. Agreement protocol.
- Internal clock synchronization algorithms. Synchronization condition. Byzantian error. Central, centrally controlled and distributed clock systems. Master-slave algorithm: TEMPO. Distributed algorithms: minimization of max. error, common intervals (slices), FTA algorithm. State-correction vs. rate-correction. External synchronization: gateway node. Controlling faulty gateways: maximizing common drift. Time gateway messages: init, rate correction, actual time for reintegrating nodes. NTP time format.

Example questions:

6. What is the difference between temporal and causal order? What does delivery order means?
7. What is the difference between UTC and TAI? Why UTC might be dangerous for HRT systems?
8. Define granularity, offset, drift, drift rate, precision and accuracy.
9. What is the difference between internal and external synchronization?
10. If we have a set of clocks with an accuracy of 1 msec, can we tell anything about their precision? (and vice versa)
11. What is a *reasonable* global time?
12. What is an agreement protocol? Why do we need it? When can you avoid it?
13. Given a clock synchronization system with a precision of 90 microsec. What is a reasonable granularity for the global time? What are the limits for the observed values for a time interval 1.1 msec?
14. Describe the master-slave TEMPO or FTA or Common slices algorithm.

4. Modeling RT systems

Reference: Kopetz, Ch. 4.
Topics:

- Modeling principles: load and fault hypothesis.
- Structural elements of a model: clusters, FTU, Node (SRU), task. Simple and complex tasks. WCET.
- Interface desing: control, temporal properties, functional intent, data properties. Temporal parameters of the client-server model of interfaces: RESP, WCET, MINT.
- Temporal, logical control. Event and time triggered control. Interrupt and trigger task overhead in a system.
- WCET of S-task: limitations on source code, problems with modern architectures (pipeline, cache). State of practice: testing.

Example questions:

15. Describe the temporal behavior of the client-server model of interfaces (RESP, WCET, MINT).
16. Define simple and complex task.

5. RT Entities and Images

Reference: Kopetz, Ch. 5.
Topics:

- Entities, observation. Observation types: untimed, indirect, state, event.
- Temporal accuracy, temporal accuracy interval. Parametric and phase-sensitive RT images. State estimation. Permanence of messages, action delay w/ and w/o global time base.

Example questions:

17. What is the difference between state and event observation? Discuss their advantages and disadvantages.
18. Define the concept of temporal accuracy.
19. What is a parametric (phase-insensitive) RT image? (Draw a figure)
20. Consider an accelerator pedal in a car. Its temporal accuracy is 10 msec. Transaction from the sensor node to the processing node takes 4 msec. How frequently shall we measure the pedal position to make the image parametric/phase-insensitive?
21. What is state estimation? How can we calculate it if the behaviour of the RT entity is described by a continuous function v(t)?
22. What does permanence means?
23. Calculate the action delay of a distributed system with the following parameters: dmax=20msec, dmin=1msec, and the global time granularity is 20 usec. What happens if we do not have global time?
24. What is the relationship between action delay and temporal accuracy?

6. RT Scheduling

Reference: Kopetz, Ch. 11, Deadline Monotonic Analysis paper (see homepage)

Topics:

- Classification of RT tasks, dynamic, static, preemptive, non-preemptive tasks. Schedulability tests: exact, sufficient, necessary. Task types (periodic, aperiodic, sporadic). States of S- and C-tasks. Necessity test for periodic tasks.
- Simple or static periodic scheduling of tasks with harmonic relation.
- Dynamic scheduling: rate/deadline monotonic algorithm, assumptions. EDF, LLA.
- Systematic test of dependent (C-) tasks: Deadline Monotonic Analysis. $R_i=C_i+I_i$. Calculation of the interference time caused by higher priority tasks. Blocking time by lower priority tasks: B_i. Priority inversion, priority inheritance, deadlock, priority ceiling (instant inheritance) protocols. Definition of blocking time using priority ceiling protocol. Scheduler and task switching cost.
- Practical software implementation of scheduling: (1) Simple/static periodic or round-robin scheduling: infinite loop. WCRT= sum of WCET. Improvements: different rates, started with timer. (2) Cycle + interrupts. Hardware handling: interrupt. WCRT as above. Communication between tasks and interrupts. (3) Scheduled functions. WCRT= max WCET. (4) RTOS.

Example questions:

25. What are the possible states of an S-task and a C-task?
26. What is a periodic, sporadic and aperiodic task?
27. Give a necessary test for independent, periodic tasks to be scheduled on one processor.
28. Describe the RMA/EDF/LL algorithm. What assumptions are made?
29. Given 3 tasks with given period and computational time, schedule them using the RMA/EDF/LL algorithm. (Numerical example can be expected, similar to the lecture)
30. What is priority inversion? Draw an example!
31. What is the difference between the priority inheritance and the priority ceiling (or instant inheritance) protocol?
32. Given 3 tasks with given period and computational time, calculate R_3 using iterative solution. (Numerical example can be expected, similar to the lecture.). Calculate the blocking time for T_1 if priority ceiling protocol is used and semaphore locking times are known.
33. Write a pseudo (C-like) code for static/static with interrupt/scheduled functions. What is the WCRT for any of them?

7. Memory management

Reference: Memory management paper (see homepage)
Topics:

- Static allocation. +: everything is fixed, error free. -: non-recallable (recursion, function pointers, re-entrant code).
- Stack-based management: theory, multitask: 1 stack for each task. Problem: stack overflow. Stack size setting by high watermark testing. Run-time verification of stack.
- Heap-based management: malloc(), free(). Problems: leak (bug), fragmentation (natural). Strategies: first-fit, best-fit, order-of-address, order-of-most-recently-used. Static allocation example: salloc(). Pools, partitions.

Example questions:

34. Describe, how stack watermarking can be carried out. How can we use it for real-time monitoring of the stack?
35. Describe leak and fragmentation. Why is leak dangerous in RT systems?
36. What is first-fit and best-fit strategy for memory allocation?
37. Describe the advantage and disadvantage of salloc() and/or pools compared to malloc().

8. RTOS

Reference: Kopetz, Ch. 10, and the lectures
Topics:

- Basic services: API, timer (scheduling, synch, asynch timers), interrupts.
- Communication between tasks: semaphores (binary, counting, mutex), queues, mailbox, pipe, events, signals.
- The non-blocking write protocol.
- Requirements for safety-critical systems: memory protection, fault-tolerancy, redundancy, access control, guaranteed resource allocation (processor time, memory).

Examples questions:

38. Describe the non-blocking write protocol
39. Describe mutex semaphores. How can we make it recallable by a given task?
40. What is the difference between synchronous and asynchronous timer in a RTOS? What is a watchdog?

9. RT Communication

Reference: Kopetz, Ch. 7, beginning of Ch. 8.
Topics:

- Requirements: protocol latency, jitter, multicast. Flexibility: different configuration, sporadic message support. Error detection: sender notification, blackout management, detection of node errors (membership service), end-to-end acknowledgement. Physical structure: bus.
- Flow control: explicit (Positive acknowledgement or retransmission, PAR), implicit (time-triggered). Operation and action delay of PAR. Thrashing.
- Architecture: RT network, field bus, backbone network.
- Conflicts in protocol design: flexibility vs. error detection
- Media Access protocols: Characterization (bandwidth, propagation delay, bit length, limited efficiency in bus systems). CSMA/CD: LON, back-pressure flow control. CSMA/CA: CAN, fields of CAN bus, arbitration mechanism. Token ring: Profibus, Minislotting: ARINC, TDMA: TTP.

- Performance comparison of ET and TT systems.
- Physical layer: NRZ, Manchester, MFM codes. Synchron and asynch coding.
- TTP protocol: structure, CC, MEDL. Variants: TTP/C, TTP/A, properties.

Examples questions:

41. Describe the characteristic of a media access protocol (bandwidth, propagation delay, bit length, bit cell). What is the limitation in efficiency of a bus system?
42. Describe the operation and action delay of PAR assuming a token ring network. Why is it capable for thrashing? How can we avoid thrashing (LON)?
43. Describe CAN bus operation, arbitration mechanism and the bit-fields.
44. What is the difference between sync and asynh coding? Code the sequence 10001101 using NRZ, Manchester and MFM codes.
45. Describe the structure and the CC of TTP. What are the main differences between TTP/C and TTP/A?
46. Compare the performance of ET and TT system (similar example as in the lecture).