Topics for the Final Exam
Design and Integration of Embedded Systems (VIMIMA11)

Design Process part:

1. Introduce CMMI and its Continuous and Staged representation. Describe the maturity levels of Staged representation and the Capability levels of Continuous representation. Present the main process categories of CMMI and show their relationship to each other.
2. Present the processes in the CMMI's Project Management Category. Introduce the specific goals of Project Planning process. What are the commonly used tools and methods to achieve the specific goals of Project Planning process?
3. Introduce the specific goals of Configuration Management CMMI Process Area. What are the typical paths of a Configuration Management Lifecycle? What are the basic components of a Version Control Systems? Introduce the lock-modify-unlock and the copy-modify-merge approach. Compare the properties of Centralized and Distributed version control systems.
4. Introduce the V-Model. Show the real-life V-model usage with iterations and branching. Introduce the steps and methods used for defining user requirements. Show the need for the Requirement Management Process and show its methods and tools. Present the Design of the Logical System Architecture.
5. Present the steps of creating the Technical System architecture from the Logical System Architecture. Explain the order of the steps and their contents.
6. Show the typical steps of Hardware Architecture Design. What is IP rating? What is the EMC problem? What do the emission and immunity mean? Present the immunity classes. What is the Inter-system and intra-system EMC? Present the typical power supply architectures and layouts.
7. Introduce the typical steps of Hardware Module Design. What are the typical problems of component placements? What is the basis for determining the degree of warming of an IC? What kind of noise coupling can be created during the PCB wiring, and how can a designer avoid these couplings?
8. Introduce the typical steps for Designing Software Architecture. What is the typical layered structure of embedded system software? Present the typical operating modes of an embedded system. Introduce the typical steps of designing a software component. What are the methods for determining the real-time model of software systems?
9. Introduce the Static Testing Technologies: reviews and analyses. Present the Structure-based testing methods: Determine the role of different coverage metrics and determine the amount of coverage they provide. Show the methods and tools of structure-based code coverage measurement.
10. Introduce the Specification-based Testing Techniques. Introduce the features and difficulties of Module testing. Show an example tool for Module testing. Present the role of MIL, SIL, HIL tests in the embedded development process. Introduce a typical HIL test environment.

Safety Critical Systems part:

11. The basic concepts of system and software safety: Introduce the notions of accident, risk, safety, and safety integrity level (SIL). Present the relation of safety integrity levels and the development process required by safety standards.
12. Introduce the definitions and measures of reliability and availability: MTFF, MTTF, MTTR, MTBF, r(t), a(t), A.
13. The architecture of safety-critical systems: Present the typical architecture solutions (single-channel architecture, two-channel architectures) used in case of fail-stop behaviour.
14. The architecture of safety-critical systems: Present the typical architecture solutions in case of fail-operational behaviour (fault tolerance for permanent and transient hardware faults).
15. The architecture of safety-critical systems: Present the typical fault tolerance techniques (N-version programming, recovery blocks) used in case of software design faults. Compare the solutions from the point of view of redundancy, execution time, and number of tolerable faults.
16. Hazard analysis techniques: Provide an overview of the typical hazard analysis techniques. Present the checklists, fault tree analysis (FTA), and event tree analysis (ETA) techniques.
17. Hazard analysis techniques: Present the cause-consequence analysis (CCA) and the failure modes and effects analysis (FMEA) techniques. Show the construction of a risk matrix on the basis of the results of the hazard analysis.
18. Dependability analysis: Introduce the elements of a reliability block diagram (RBD). Show how to construct an RBD in case of serial, parallel and voting architectures.
19. Formal modelling of time-dependent behaviour: Introduce the timed automaton formalism. Present the extensions of timed automata to support the modelling of distributed systems.
20. Formal verification: Show how to formalise safety requirements using temporal logics. Introduce the concept of formal verification with model checking.