Computer Science Engineering BSc (2017) - Final Exam topics

1. The processor implementation options: Processor technology, implementation techniques and design technologies. Typically peripherals for embedded systems. Communication protocols.

   Program units. Subprograms. Parameter evaluation. Parameter passing methods. Block. Scoping, accessibility. Abstract data type. Generic programming. I/O tools of programming languages, file handling. Exception handling. Parallel programming.

2. Synthesis of continuous time control systems. The gain and phase margin. Linear systems and their description in time- and frequency domains. Signal transfer in control systems.

   Explain the data elements of TCP and UDP transport layer protocols, and the differences between their mechanisms.

3. Combinational logic design. Multiplexers/Demultiplexers. Encoders/Decoders. Comparators. Parity generators/checkers. Arithmetical logical units.

   Present the general problem solving methods and compare them with the methods for solving constraint satisfaction problems.

4. The SSH protocol, key generation, configuration of user settings

   The principles of control, feedback control and open loop control. Set point control and reference signal tracking, the role of negative feedback. Requirements for control systems.

5. Present the adversarial searches and the conditions necessary for the existence of a winning strategy.

   MOS transistor: large signal model and characteristics. The MOS transistor as a switch. CMOS inverter, basic logic gates. The operational amplifier. Negative feedback. Basic applications.

6. Sequential logical: Latches and Flip-Flops. Counters. Shift registers. Memories.

   New elements of HTML5. New features of CSS3. Control structures in web scripts. Sensor through a web page. Providing remote management systems through a web page.

7. Provide the necessary steps and technologies for developing a sample software product on a choosen platform. Describe the benefits and difficulties of the platform, the implementation steps, and the most widely used current technologies.

   Implementation of control structures in assembly (control program flow, branching, looping)

8. Concept, typical applications and requirements of embedded systems. Real-time and reactive systems. Embedded systems architecture. Hardware and software layers. Embedded software: system software layer and application software layer.

   Functions and services of the MRTG and Nagios network management systems.

9. Programmable logic devices. Designing a digital system in hardware description language, and implementing it in FPGA devices.

   Basic concepts of system engineering, different paradigms; Characteristics of classical methodologies, waterfall (structured) model; Iterative (evolutionary, incremental) models; Agile software development methodologies, tools; OO design principles and important design patterns, MVC architectural pattern.

10. Configuration of a web server using SSL, the OpenSSL cryptographic library: authentication, encryption.

    The instruction set architecture (ISA) of Intel X86 processors (registers, addressing, instructions, memory architecture, interrupt system)

11. Interprocess communication (file, signal, pipe, socket)

    Time complexity of algorithms: insertion sort, merge sort, searching in linear and logarithmic time. Quick sort, the minimal number of necessary comparisons. Sorting in linear time: radix sort, bucket sort.

12. Entity-relationship (ER) model, design with ER diagrams. Relational data model, relation, scheme, attribute. Building up a relational scheme from an ER-diagram.

    Diodes. Rectifiers. DC to DC converters. Voltage regulators. Current regulators.

13. Modern processor solutions (pipeline, hazard, out-of-order execution, speculative execution, superscalar-, VLIW- and vector processors)

    Optimization and evaluation of relational queries. Tree-based optimization in relational algebra. Cost-based optimization.

14. Explain the NAT/PAT address translation mechanisms.

    Basic notions concerning data structures: modelling, abstraction, abstract data types. Elementary data structures: lists, stacks, queues. Sets, multisets, arrays. The representation of trees, tree traversal, deletion and insertion.

15. Basic concepts of object-oriented paradigm. Class, object, instantiation. Inheritance, class hierarchy. Polymorphism, method overloading. Scoping, information hiding, accessibility levels. Abstract classes and interfaces. Class diagram of UML.

    Compare the SNMP and RMON network management systems