DESIGN FOR TESTABILITY

This course is an introduction to **fault diagnosis and self-testing of digital systems**. More difficult than to create a system is to guarantee that the designed and implemented system is correctly functioning.

Fault diagnosis is a reverse task to design, and has the target to answer the question, why a system is not working properly.

Self-testing system is the prerequisite of dependability of the technical world around us.

The difference between a programmer and a test engineer is that the programmer can write a software only for a computer that is working, whereas the test engineer must be able to program any faulty computer while the number of possible faults is infinite.

This course is about how to program a faulty computer.

The course consists of lectures, laboratory tasks, independent course work with report, and exam.

The lecture course will target the following topics:

INTRODUCTION

1. Introduction. Design criteria: testability, reliability, fault-tolerance, dependability and availability. Design for testability economics.

OVERVIEW ABOUT TESTING

2. Overview about testing. Fault models, test generation, fault simulation, fault diagnosis.
3. Decision diagrams as a powerful graph-model for diagnostic modeling both, of gate-level digital circuits and high-level described complex systems

TESTABILITY EVALUATION

4. System Quality policy. Yield and defect level. Economic tradeoff between test and design for testability.
5. Testability criteria. Testability of different design types. Easy-testable circuits. Reasons of bad testability.
6. Testability measures. Controllability and observability. Heuristic measures. Probabilistic measures. Probabilistic estimation of testability at the logic level: McCluskey method, cutting method and method of conditional probabilities. Testability evaluation at higher register-transfer and behavioral levels. Calculation of testability using BDDs and HLDDs

DESIGN FOR TESTABILITY

7. Ad-hoc methods of design for testability. Test points. Improving of observability: multiplexing and time sharing. Improving of controllability: de-multiplexing and time sharing. Fault redundancy and testability. Partitioning for testability. New design

standards: Scan Path based design. Scan architectures: multiple scan, partial scan, random access scan.

8. Selection and insertion of test points. Classification of not detectable faults and their impact on design.
9. Boundary scan and interconnect testing. Boundary-scan architecture: test access port, registers, TAP controller, modes of operations, Boundary-scan languages. Interconnect diagnosis: fault isolation and diagnostic synthesis.

BUILT-IN SELF-TEST

10. Built-in Self-test (BIST). Test per scan and test per clock. Linear feedback shift registers (LFSR): standard and modular schemes. Pseudorandom test generation. Reconfigurable LFSR. Theory of LFSR. Primitive and non-primitive polynomials. Hard to test faults. Deterministic synthesis of LFSR. Weighted pseudorandom test. Calculation of weights.
11. Signature analysis. Probability of aliasing. BIST architectures: LOCST, STUMPS, CSTP, BILBO. Software BIST. Hybrid BIST. Optimization of BIST. Hybrid BIST with reseeding. Store and generate architecture. Functional self-test. Hybrid functional BIST. Hybrid BIST for multiple cores.

Course work: Design of a circuit by CAD tools, analysis of testability of the circuit, improving the design for testability, design for self-test, analysis of self-test quality by BIST simulation.

Laboratory support: Cadence and Synopsys design tools, Turbo-Tester ATPG, fault simulation and BIST simulation tools.

Literature:

1. M.Bushnell,V.Agrawal. Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits. Springer Publishing Comp., Incorp., 2013.
2. H.-J.Wunderlich (ed.). Models in Hardware Testing. Springer, 2010, 255 p
3. L.T.Wang. C.-W.Wu, X.Wen. VLSI Test Principles and Architectures. Design for Testability, 2006, Elsevier, 777 p.
4. D.Gizopoulos (ed.). Advances in Electronic Testing: Challenges and Methodologies. Springer, January 2006
5. N.Jha, S.Gupta. Testing of Digital Systems. Cambridge Univ. Press, 2003, 1000 p.
6. **R.Ubar**, A.Jasnetski, A.Tsertov, A.Oyeniran, Software-Based Self-Test with Decision Diagrams for Microprocessors. Lambert Publishing House, 2018, 171 p.
7. **R.Ubar**, J.Raik, H.-T.Vierhaus. Design and Test Technology for Dependable Systems-on-Chip. IGI Global, 2011, 550 p.
8. O.Novak, E.Gramatova, **R.Ubar**. Handbook of Testing Electronics Systems. Czech TU Publishing House, 2005, 400 p.