Testing and Fault Tolerance (CE-TFT)

CE-TFT0. History and overview of testing and fault tolerance [core]
CE-TFT1 – Faults and Fault Models in Digital Circuits
CE-TFT2 - Test generation methods
CE-TFT3 - Design for testability
CE-TFT4 - Testing non-stuck-at faults
CE-TFT5 - System-level test and diagnosis
CE-TFT6 - Reliability and fault tolerance definitions
CE-TFT7 - Error detecting and correcting codes
CE-TFT8 – Fault Tolerant System Design
CE-TFT9 – Software approaches and software fault tolerance

Fault models in digital circuits, test generation algorithms, test generation for sequential circuits, fault simulation, testability measures, fault coverage, yield and defect levels, design-for-testability approaches, scan and boundary scan, IDDQ testing, mixed signal testing. Architecture and design of fault tolerant computer systems using protective redundancy, estimation of the reliability and availability of fault tolerant systems, error recovery, and fault diagnosis.

CE-TFT0. History and overview of testing and fault tolerance [core]

Suggested time: 1 hour

Topics:

- Indicate some reasons for studying testing and fault tolerance.
- Highlight some people that influenced or contributed to the area of testing and fault tolerance.
- Indicate some important topic areas such as fault models, test generation, testability, system-level tests, reliability, error detection and correction codes, and approaches to fault tolerance designs.
- Describe a fault model.
- Indicate some methods for test generation.
- Describe the meaning of non-stuck-at faults.
- Describe and contrast reliability and fault tolerance.
- Mention some error-detecting codes.
- Mention some error-correcting codes.
- Highlight the importance of fault tolerant system design.
- Explore some additional resources associated with testing and fault tolerance.
- Explain the purpose and role of testing and fault tolerance in computer engineering.

Learning objectives:

- Identify some contributors to testing and fault tolerance and relate their achievements to the knowledge area.
- Define a fault model.
- Give an example of test generation.
Know the difference between reliability and fault tolerance.
Identify an error detecting code.
Identify and error correction code.
Articulate reasons for the importance of fault tolerant system designs.
Describe how computer engineering uses or benefits from testing and fault tolerance.

CE-TFT1 – Faults and Fault Models in Digital Circuits
- Logical (stuck-at) faults (single and multiple)
- Bridging faults and opens
- Delay faults
- Fault equivalence and dominance
- Yield and defect levels
- Test coverage
- Fault simulation and fault grading

CE-TFT2 - Test generation methods
- The D algorithm
- PODEM, FAN, Learning Algorithms
- Automatic Test Pattern Generation (ATPG)
- Pseudorandom techniques
- Deterministic test pattern generation
- Test generation algorithms for sequential circuits
- Memory testing
- PLA testing

CE-TFT3 - Design for testability
- Testability measures (controllability, observability)
- Scan and partial scan design
- BIST and other design for testability techniques
- Boundary scan and the IEEE 1149.1 testability standard
- Ad-hoc methods

CE-TFT4 – Testing non-stuck-at faults
- CMOS opens testing
- Performance and delay testing
- IDDQ and other current based tests
- Mixed signal testing

CE-TFT5 - System-level test and diagnosis
- Printed circuit board testing
- MCM and core based testing
- System testing
- Verification Testing
CE-TFT6 - Reliability and fault tolerance definitions
  Reliability and availability modeling

CE-TFT7 - Error detecting and correcting codes
  Single Error Detection
  Single Error correction/Double Error Detection
  Burst Error detection codes
  Hamming and Reed/Solomon Codes
  Test compaction and aliasing

CE-TFT8 – Fault Tolerant System Design
  Hardware redundancy approaches
  Fault tolerance in VLSI devices
  Aerospace systems
  Telecommunications systems
  Industrial control applications
  Fault-tolerant transaction processing systems
  Software approaches

CE-TFT9 – Software fault tolerance
  Software reliability models
  Software fault-tolerance methods (N-version programming, recovery blocks, rollback and recovery )
  Fault tolerance in operating systems and data structures
  Fault tolerance in database and distributed systems

CE-TFT10 – Software Testing (this may belong in Software Engineering)
  Test specifications
  Black box testing
  White box testing
  Random tests
  Test coverage