Course Description
Computer systems play an increasingly roll in our daily life, where some of them are safety-critical. Examples of safety-critical applications are flight control, train control, avionics control, medical systems, satellites, and plant power systems. A failure in such systems may lead to catastrophic consequences. Therefore, reliability and correct operation of these systems are of decisive importance. These systems must be able to tolerate faults/ errors and continue to deliver correct results in the presence of hardware and software faults. This course provides knowledge on the design or reliable and fault-tolerant computer systems.

Outline
1. Why do we need fault tolerance?
2. Applications of fault-tolerant computer systems
3. Basic Terminologies: Reliability, Availability, Safety, Maintainability, Confidentiality, Integrity, Security, Testability, Dependability
4. Basic definitions: fault, error, failure
5. Fault characteristics
6. Fault / Error models
7. Fault / Error manifestation
8. Design techniques to achieve fault tolerance:
   - Hardware redundancy: TMR, NMR, etc.
   - Information redundancy: parity codes, m-of-n codes, etc.
   - Time redundancy: re-computation, etc.
   - Software redundancy: consistency checks, etc.
9. Evaluation techniques:
   - Quantitative evaluation methods: Failure rate, Reliability function, Coverage, MTTF, MTTR, MTBF, etc.
   - Reliability modeling: Combinational models, m-of-n systems and Markov models
   - Reliability estimation using the SHARPE software
10. Estimation of failure frequency, MIL HDBK 217F
11. Design of practical fault-tolerant systems
12. Some examples of fault-tolerant systems

References
Advanced Storage Systems

Outline
1. Introduction to Data Storage Systems
   a. Storage History
   b. Performance trend of disk drives and microprocessors
   c. Amdahl Law and its implication to storage systems
   d. Architecture of server-centric storage
2. Architecture of Storage-Centric IT Infrastructure
3. I/O Architecture & Configuration in Disk Subsystem
4. Qualitative & Quantitative Metrics in Storage Systems
   a. Throughput, response time, availability, serviceability, and scalability
5. Disk Configuration in Storage Systems
   a. RAID1, RAID10, RAID5, RAID6
   b. Read performance, write performance, and availability
6. Design of an Advanced Storage System
   a. Backend design
   b. Front-end design
   c. Memory system design
7. I/O Flow in Storage Systems
   a. Read, write, and copy
8. Advanced Features of Data Storage Systems
   a. Remote Mirroring
   b. Instant Copies
   c. Data Migration
   d. LUN Masking
9. Cache Memory in Storage Systems
   a. Structure of cache memory in storage systems
   b. Comparison of cache memory in storage systems and microprocessors
   c. Cache replacement algorithms used in storage systems
10. Architecture of Off-The-Shelf Storage Systems
    a. IBM, HP, and EMC
11. Design & Implementation of SAN & NAS
    a. Storage Area Network (SAN) and Network Attached Storage (NAS)
12. I/O Techniques in Storage Systems
    a. SCSI, iSCSI, Fibre Channel, SAS
13. Design & Architecture of Emerging Technologies used in Storage Systems
    a. Architecture of NAND & NOR chips
    b. Design & architecture of Solid-State Disk Drives (SSDs)

References
Course Description

Computer systems play an increasingly roll in our daily life, where some of them are safety-critical. Examples of safety-critical applications are flight control, train control, avionics control, medical systems, satellites, and plant power systems. A failure in such systems may lead to catastrophic consequences. Therefore, reliability and correct operation of these systems are of decisive importance. These systems must be able to tolerate faults/errors and continue to deliver correct results in the presence of hardware and software faults. This course reviews different research areas, including past and current research, and provides knowledge on the design of dependable and fault-tolerant computer systems.

Outline

1. Behavior, propagation, and effects of faults/errors in computer systems:
   - Faults classification
   - Data errors, program errors
   - Data error detection techniques
   - Control flow error detection techniques
   - Control flow checking
   - Watchdog processors
2. Evaluation techniques for fault-tolerant computer systems:
   - Physical fault injection techniques
   - Simulation-based fault injection techniques
   - Emulation-based fault injection techniques
   - Comparison of fault injection techniques
   - Probability techniques to analyze fault injection results
   - Estimation techniques to fault coverage
3. Fault-tolerant techniques in microprocessors
4. Dependability in embedded systems
5. Dependability in NOCs
6. Dependability in computer networks
7. Dependability in embedded systems
8. Dependability in distributed systems
9. Dependability in real-time systems
10. Dependability in e-commerce

References

1. Selected Papers.
Course name: Advanced Microprocessor

<table>
<thead>
<tr>
<th>Course ID: 40722</th>
<th>Credits: 3</th>
<th>Program: Msc. H/W Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisites:</td>
<td></td>
<td>Co-requisites: -</td>
</tr>
<tr>
<td>Prepared by:</td>
<td>Amir Hossein Jahangir</td>
<td></td>
</tr>
</tbody>
</table>

Outline
1- Introduction (Definition of superscalar processors, static and dynamic scheduling, pipeline architectures, modern processors characteristics and their ISA :Instruction Set Architecture)
2- Description of Scoreboarding and Tomasulo algorithms in CISC processors
3- Branch prediction techniques, speculative execution.
4- VLIW architectures (+ predicative execution), Memoization and Value prediction.
5- Multiprocessing issues in modern processors (cache consistency protocol, and arbitration mechanisms + case study: Pentium)
6- Multithreading techniques and examples.
7- Advanced Bus and I/O architectures in modern processors

References:
4- Several papers from literature.
Course name: Advanced Computer Architecture

<table>
<thead>
<tr>
<th>Course ID:</th>
<th>40723</th>
<th>Credits:</th>
<th>3</th>
<th>Program:</th>
<th>Graduate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisites:</td>
<td></td>
<td></td>
<td></td>
<td>Co-requisites:</td>
<td></td>
</tr>
<tr>
<td>Prepared by:</td>
<td></td>
<td></td>
<td></td>
<td>Amir Hossein Jahangir</td>
<td></td>
</tr>
</tbody>
</table>

Outline

1- Introduction
   Definition of Speedup, Efficiency, Amdahl’s law, For/ against parallel architectures, Classification of high performance architectures...

2- Memory system architecture for advanced computer architectures
   interleaved memory, cache

3- Pipeline architecture
   Instruction and arithmetic pipeline control (collision vector, reservation stations), speeding up pipeline with delay, eliminating data dependency in recursive operations...

4- Vector computers
   Array processors, Pipelined vector computers, Memory stride for high bandwidth access...

5- Interconnection networks and Multicomputers
   Hypercube, k-ary cube, mesh, butterfly, pyramid...

6- Multiprocessors
   Analysis of Run time to communication ratio, Role of interconnection network in the performance, cache consistency protocols

7- Software issues and speedup
   Synchronization, Communication, Code optimization for superscalar and parallel architectures...

References


Course name | Low Power Design
---|---
Course ID: | 40727
No of units: | 3
Program: | Graduate
Prerequisites: | VLSI Circuit Design
Co-requisites: | -
Prepared by: | Alireza Ejali

Outline


**Topic 2: On-chip Interconnects: Reduced Voltage Swing, Level Shifters, Low Power Encoding, Bus Inverting,**

- Partitioned Bus-Inverting, Data Compression Encoding, Transition Signaling, Limited Weighted Codes (LWC), Bus-Inverting vs. LWC

**Topic 3: Circuit-Level Techniques: Dual-Threshold Circuits, Design Issues in Dual-Threshold Circuits, Dual-VDD circuits, Static Short-Circuit Power in Dual-VDD, SDCVSL Converters, CVS Structure, Optimum VDDL value**

**Topic 4: Gate-Level Techniques: Technology Mapping and Decomposition, Activity Estimation, Problem of Re-convergent Fan-outs, Input Reordering, Activity Postponement, Transistor Reordering, Concurrency and Redundancy**

**Topic 5: RT-Level Techniques: Clock Gating, Clock Skew Problem in CG, Glitch Problem in CG, Operand Isolation,**

- RT-Level Concurrency and Redundancy, Pre-Computation, ODC, Glitch Reduction, Block-Level Control, Pipeline for Low Power

**Topic 6: FSM: FSM Partitioning, Activity Estimation for FSMs (Using DTMC), State Encoding**

**Topic 7: Adiabatic Circuits: Principle of Energy Recovery, Adiabatic-Charging Principle, Constant Current Generator, APS (Voltage Ramp), Clock-Power Signals, Cascading Problem, Retractile Cascading,**

- Reversibility, 8-Phase Reversible Logic Family, 4-Phase Reversible Logic Family

**Topic 8: System-Level Techniques: Dynamic Voltage Scaling, Dynamic Power Management (DPM), Adaptive Body Biasing (ABB), System-level Methods in Real-time systems.**

**Topic 9: Temperature-Aware Design: Temperature Modeling, DVS, DFS (Dynamic Frequency Scaling), Fetch Gating, Clock Gating, Computation Migration.**

References

6) Published conference and journal papers.
Course name: DSP Architecture

<table>
<thead>
<tr>
<th>Course ID:</th>
<th>Credits:</th>
<th>Program:</th>
</tr>
</thead>
<tbody>
<tr>
<td>40732</td>
<td>3</td>
<td>Graduate</td>
</tr>
</tbody>
</table>

Prerequisites: -

Co-requisites: -

Prepared by: Amir Hossein Jahangir

Outline
1- Introduction: DSP algorithms and processor architectures.

2- Arithmetic algorithms used in DSP processors
   - Number representation (Fixed point and floating point)
   - Adders
   - Multipliers
   - Function evaluation

3- DSP processor architecture
   - ALU and Processing Elements
   - Memory organization
   - I/O interfaces

4- DSP algorithm implementations
   - DFT, FFT
   - FIR, IIR, Decimation
   - DDC

5- Real DSP processor architectures (from TI, Analog Devices etc.) and applications (Software defined radio, Multimedia etc.)

6- Simulation or implementation project.

References:
3. Digital Signal Processors, B Venkataramani and M Bhaskar 2002 (TMH)
1- Introduction (Definition of superscalar processors, static and dynamic scheduling, pipeline architectures, modern processors characteristics and their ISA : Instruction Set Architecture)
2- Description of Scoreboarding and Tomasulo algorithms in CISC processors
3- Branch prediction techniques, speculative execution.
4- VLIW architectures (+ predicative execution), Memoization and Value prediction.
5- Multiprocessing issues in modern processors (cache consistency protocol, and arbitration mechanisms + case study: Pentium)
6- Multithreading techniques and examples.
7- Advanced Bus and I/O architectures in modern processors

References:
4- Several papers from literature.
Course name: Embedded System Design

<table>
<thead>
<tr>
<th>Course ID:</th>
<th>40747</th>
<th>No of units:</th>
<th>3</th>
<th>Program:</th>
<th>Graduate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisites:</td>
<td>-</td>
<td>Co-requisites:</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepared by:</td>
<td>Alireza Ejlali</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Outline


Topic 2: Automata-Based Programming: Automata-Based Programming of Reactive Systems, Hierarchical Design in ABP, Mealy and Moore ABP.


Topic 6: Embedded System Hardware: Actuators and Sensors, A/D, D/A and Sample and Hold Circuit, Processing Units

Topic 7: Distributed Embedded Systems: Requirements, Real-Time Communication, Robustness, Maintainability and Diagnosability, Electrical Robustness, CSMA/CD vs. CSMA/CA, CAN and TTP, Error Detection and Error Handling in CAN, Non-Destructive Arbitration in CAN.


Topic 10: Memory Organization: Code-size efficiency, Code Compression in ARM processors, Dictionary-Based Methods, Energy/Performance Trade-off in Cache and Memory, Scratch Pad Memory (SPM), Cache vs. SPM.


References


4) Published conference and journal papers.
Course name: System-on-Chip Design

<table>
<thead>
<tr>
<th>Course ID:</th>
<th>Credits:</th>
<th>Program:</th>
<th>Prerequisites:</th>
<th>Co-requisites:</th>
</tr>
</thead>
<tbody>
<tr>
<td>40757</td>
<td>3</td>
<td>Graduate</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Prepared by: Shaahin Hessabi

Outline


Topic 2: Overview of ASICs (Methodology and Design Flow, Programmable ASICs: CPLDs and FPGAs, FPGA to ASIC Conversion, Verification)


Topic 4: Design Methodology for Memory and Analog Cores (Design Methodology for Embedded Memories, Specifications of Analog Circuits, High-Speed Circuits)

Topic 5: Platform Based Design

Topic 6: Multi-Processor SoC (MPSoC)

Topic 7: On-Chip Interconnection Networks (SoC Bus Architectures, Network-on-Chip)

Topic 8: SoC Testing (Digital Logic Cores, Embedded Memories, Analog and Mixed-Signal Cores)

References


**Course name**

**Electronic System Level Design**

<table>
<thead>
<tr>
<th>Course ID:</th>
<th>40843</th>
<th>No of units:</th>
<th>3</th>
<th>Program:</th>
<th>Graduate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisites:</td>
<td>-</td>
<td>Co-requisites:</td>
<td>-</td>
<td>Prepared by:</td>
<td>Maziar Goudarzi</td>
</tr>
</tbody>
</table>

**Outline**

1. Evolution of design flow of hardware and software of embedded systems design up to Electronic System Level (ESL)
   a. Evolution of hardware modeling and design methodologies, Gate-level design, Register-Transfer Level (RTL) design, Behavioral level design, ESL design.
   b. Brief review and comparison of hardware and software description languages.
   c. Motivations for the move toward ESL design
2. Electronic System Level Design Flow
   a. Specification and Modeling, pre-partitioning analysis, partitioning, post-partitioning verification, post-partitioning analysis, software implementation, hardware implementation, implementation verification
3. Digital system specification using SystemC
   b. Modules and hierarchy in SystemC: ports, signals, data storage, processes, module constructor, positional port mapping, named port mapping, hierarchical design, port connection rules
   d. Data types in SystemC: bit-accurate data types, single-bit types (sc_bit, sc_logic), integer types (sc_int, sc_uint, sc_bigint, sc_bignint), bit-vector types (sc_fixed, sc_ufixed, sc_fix, sc_ufix), resolved logic data types, tracing signal and port values, speed issues, user-defined data types.
   e. Combinational logic modeling and recommendations for synthesis, local variables vs. signals and ports, delta delay concept, logical operators, arithmetic operators, relational operators, vectors and ranges, reading from vectors, writing to vectors, if statement, switch statement, loops, methods, structures, multiple processes per module.
   f. Synthesis of SystemC models, concepts, recommendations, and pitfalls.
   g. Finite State Machine (FSM) design in SystemC.
4. A systematic methodology for hardware implementation of software programs
   a. Introduction to and significance of FSM with Datapath (FSMD) model.
   b. A classification of statements in software languages.
   c. Hardware implementation of a single basic-block using FSMD model.
   d. Hardware implementation of a complete software program using FSMD model.
   e. Introduction to Behavioral (or High Level) Synthesis (HLS).
   f. Basic topics in HLS: Allocation, Scheduling, Binding.
   g. Review of famous algorithms in HLS allocation, scheduling, and binding.
5. Hardware-Software communication mechanisms and their significance.
a. A case study of full-software and hardware-software implementation of AES encryption and analysis of performance loss due to hardware-software communication overhead.
b. Hardware-software and software-software communication mechanisms in uniprocessor and multiprocessors.

   a. Significance of separating communication from computation.
   b. The Channel concept, its features and importance.
   c. Gajski’s classification of TLM models, their components, and their usage in a top-down ESL design flow.
   d. SystemC mechanisms and features to define Channels and to model at TLM level.

7. Co-synthesis Algorithms
   a. Hardware-Software partitioning algorithms. Primal and Dual approaches, case study of two projects: Vulcan and Cosyma, performance and cost estimation techniques for hardware or software implementation of components, Simulated Annealing optimization algorithm.
   b. Multiprocessor co-synthesis. Integer Linear Programming technique to obtain the global optimal solution, a MILP model for multiprocessor co-synthesis problem, heuristic algorithms, Wolf’s heuristic algorithm for ordinary task graphs, Wolf’s heuristic algorithm for object-oriented applications.

8. Latest platforms for digital system implementation.
   a. Evolution of semiconductor technology up to System-on-Chip (SoC).
   b. Evolution of FPGAs up to Programmable SoC (PSoC) devices.
   c. Challenges and opportunities by the above evolutions.
   d. System design using PSoC devices.

   a. Hardware-software co-simulation techniques.
   b. Introduction to formal verification of hardware-software systems.

10. Software-level optimization techniques
    a. Process variation and its significance.
    b. A variation-aware online power management algorithm to optimize power consumption and performance of Chip Multiprocessors.
    c. Variation-aware power-yield optimization by task scheduling in Multiprocessor SoC devices.
    d. Code and data placement algorithms and their usages in system optimization.

References

# Reconfigurable Computing

<table>
<thead>
<tr>
<th>Course ID:</th>
<th>Credits:</th>
<th>Program:</th>
</tr>
</thead>
<tbody>
<tr>
<td>40844</td>
<td>3</td>
<td>Graduate</td>
</tr>
</tbody>
</table>

**Prerequisites:** VLSI Design (40353), Digital System Design (40223)

**Co-requisites:**

**Prepared by:** Hossein Asadi

## Outline

1. **Introduction to Reconfigurable Computing**
   - FPGA technology
   - Logic blocks
   - Basics of COTS reconfigurable devices: Xilinx, Altera, Lattice, and Actel

2. **Design Mapping**
   - FPGA technology mapping
   - Placement & routing algorithms considering area, delay, power, and reliability
   - Simulated annealing, FD relaxation, and macro-based methods

3. **Architecture of Reconfigurable Devices**
   - Logic block architectures
   - Interconnect and routing matrix architectures
   - Design tradeoffs in a reconfigurable logic block
   - Design tradeoffs in a reconfigurable interconnect
   - Area, delay, power, and reliability optimization techniques using VPR toolset
   - Architecture of the state-of-the-art reconfigurable devices

4. **Dynamic Reconfiguration**
   - Reconfiguration and scheduling algorithms
   - Limitations of reconfigurable approaches
   - Hardware-support for reconfiguration

5. **Reconfigurable Systems**
   - Multi-FPGA system topologies
   - Logic emulation using Multi-FPGA systems
   - Partitioning in multi-FPGA systems
   - Interconnect of multi-FPGA systems
   - Architecture of modern multi-FPGA systems
   - Hybrid reconfigurable systems (LSI logic) vs. FPGAs vs. processors

6. **Reconfigurable Applications**
   - Arithmetic operations
   - Systolic machines
   - Partially reconfigurable machines
   - Data acquisition systems

7. **System Prototyping Using Reconfigurable Devices**
   - System validation & verification using prototyping

8. **Advanced Topics on Reconfigurable Computing**
   - Reconfigurable co-processors
   - Hardware cores in reconfigurable devices
   - Emerging reconfigurable technologies
Textbooks


References

**Course name** | **Interconnection Networks**
---|---
**Course ID:** | 40853
**No of units:** | 3
**Program:** | Graduate
**Prerequisites:** | Computer Architecture (BSc course)
**Co-requisites:** | None
**Prepared by:** | Hamid Sarbazi-Azad

**Outline**

1. **Introduction** (1 session: The Evolution of Computer Architecture; Multicomputers/Multiprocessors and their Interconnection Networks (INs); Basic definitions and notation)

2. **Topology** (8 sessions: Topological factors; Popular Topologies and their Characteristics; Complex Topologies; Embedding; Hamiltonian Properties; Combinatorial Properties)

3. **Switching Methods** (3 sessions: Packetization/depacketization; Circuit Switching; Packet Switching; Wormhole Switching and VCT Switching; Mad Postman Switching; Virtual Channels; Combined Switching Techniques: Pipelined Circuit Switching, Buffered Wormhole Switching)

4. **Routing Algorithms** (8 sessions: Deadlock and Livelock Prevention; Deterministic Routing Algorithms in Popular INs; Partially Adaptive Routing Algorithms in Popular INs; Fully Adaptive Routing in Algorithms Popular INs)

5. **Multicast Routing** (6 sessions: Basic Definitions; Hardware Tree-Based and Path-based Multicast Routing Algorithms; BRCP Model; Software Multicast Algorithms: Dimension Order and Dimension Ordered Chains, Software Multicast in Popular INs)

6. **Performance Evaluation** (3 sessions: Performance Evaluation Methods; Technological Constraints; Traffic Models; Delay Models; Discrete Event Simulation; Xmulator Package)

7. **Hot Topics** (1 session: New topics on interconnection networks introduced in the last two year in related conferences and journals)

**References**

5. Papers from IEEE TPDS, JPDC, PC, JOIN, IJPDEP journals (and other related journals) and IPDPS, ICPADS, ICPP, HiPC, HPCA, NOCS, HPCA Conferences (and other related parallel and network-based conferences).