Study Guide

Computer Engineering Written PhD Qualifying Exam

Effective: Fall 2014

1 Introduction

This document defines the subject matter and presents a study guide for the written PhD qualifying examination in Computer Engineering. This document supersedes any and all information published elsewhere on the same subject matter.

2 Exam Format

The written exam will be 3 hours in duration. The exam will comprise 8 equally weighted questions, one question from each of the following subject areas.

1. Digital Logic and Hardware Design Languages
2. Electric Circuits,
3. Electronics,
4. Linear Systems and Control,
5. Discrete Structures
6. Computer Networks
8. Embedded Systems,

Each student is required to answer 5 of the 8 questions. If you choose to answer more than 5 questions, then your exam score will be the sum of the Best-5 scores. Any questions beyond the Best-5 do not count, and their scores will be discarded.

3 Study Guide

The topics covered on the exam will be taken from the lists in the following subsections. Each subject will be covered at up to the levels of detail presented in the cited sections of the listed textbooks.
3.1 Digital Logic and HDLs

- Digital logic at the gate and switch level including both combinational and sequential logic elements.
- Clocking methodologies to manage information flow and preservation of circuit state.
- Digital logic specification methods and the compilation process that transforms these into logic networks.
- Computer aided design tools for implementation with programmable logic devices.
- Advantages/disadvantages between hardware and software implementations of a function.

3.1.1 References


3.1.2 Topics to be Covered

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>I Binary Number System</strong></td>
<td>Wak06: 2</td>
</tr>
<tr>
<td><strong>II Hardware Description Languages</strong></td>
<td>Wak06: 5</td>
</tr>
<tr>
<td>HDL-Based Digital Design, familiarity with <em>either</em> VHDL or Verilog (the choice is yours)</td>
<td></td>
</tr>
<tr>
<td><strong>III Design with Combinational Logic Components</strong></td>
<td>Wak06: 6</td>
</tr>
<tr>
<td>Circuit Timing, Combinational PLDs, Decoders, Encoders, Three-State Devices, Multiplexers, Exclusive-OR Gates and Parity Circuits, Comparators, Adders, Subtractors, and ALUs, Combinational Multipliers</td>
<td></td>
</tr>
<tr>
<td><strong>IV Design with Sequential Logic Components</strong></td>
<td>Wak06: 8</td>
</tr>
<tr>
<td><strong>V Memory FPGA and CPLD</strong></td>
<td>Wak06: 9</td>
</tr>
<tr>
<td>Memory devices, CPLDs, and FPGAs. Memory coverage includes read-only memory and static and dynamic read-write memories from the points of view of both internal circuitry and functional behavior. The last two sections introduce CPLD and FPGA architecture.</td>
<td></td>
</tr>
</tbody>
</table>
3.2 Electric Circuits

This subject will focus on introduction to linear circuit analysis, network theorems, transient analysis, steady-state sinusoidal response, and frequency response. References

3.2.1 References


3.2.2 Topics to be Covered

I. Introduction to Linear Circuit Analysis
   1. Electric Charge and Current Ula10: 1-3
   2. Voltage and Power Ula10: 1-4
   3. Circuit Elements Ula10: 1-5

II. Network Theorems
   1. Ohm’s Law Ula10: 2-1
   2. Kirchoff’s Law Ula10: 2-3
   3. Node-Voltage Method Ula10: 3-1
   4. Mesh-Current Method Ula10: 3-2
   5. Thevenin and Norton Equivalent Circuits Ula10: 3-5
   6. Maximum Power Transfer Ula10: 3-6

III. Transient Analysis
   1. RC and RL First-Order Circuits Ula10: 5
   2. RLC Circuits Ula10: 6

IV. Steady-State Sinusoidal Response
   1. AC Analysis Ula10: 7
   2. AC Power Ula10: 8

V. Frequency Response
   1. Transfer Function Ula10: 9-1
   2. Scaling Ula10: 9-2
   3. Bode Plots Ula10: 9-3
3.3 Electronics

The subjects in this area include the fundamentals of electronic circuits and devices. Students are required to be familiar with the physics and operation of electronic devices such as diodes, op-amps, BJTs and MOSFETS, as well as the analysis of electronic circuits using diodes, op-amps, BJTs and MOSFETS.

3.3.1 References


3.3.2 Topics to be Covered

I. Basics of linear circuits
   Ham99: 1
II. Operational amplifiers and their applications
    Ham99: 2
III. Differentiators and integrators
     Ham99: 2
IV. Diodes and their models
    Ham99: 3
V. Diode applications, rectifiers, PN junction diodes
   Ham99: 3
VI. Bipolar junction transistors, and small signal analysis
    Ham99: 4
VII. Field effect transistors
     Ham99: 5
VIII. MOSFETS physics (PMOS and NMOS) and their switching models
     Ham99: 6
3.4 Linear Systems and Control

This subject introduces the mathematical analysis of signals, systems, and control. Topics include differential equations, Fourier series, Fourier transforms, Laplace transforms, frequency response, Bode plots, state models, and an introduction to control systems.

3.4.1 References


3.4.2 Topics To Be Covered

I. Introduction to continuous-time convolution
   1. LTI system properties, impulse response and total response; Lat05: 2.1-2.6
   2. Steady state response of LTI systems with sinusoidal inputs; Lat05: 6.1-6.4

II. Fourier transform and its properties
   1. LTI system response using the Fourier series; Lat05: 6.1
   2. Fourier series properties and Fourier transform; Lat05: 6.2-6.5
   3. Fourier transform properties; Lat05: 7.1-7.3
   4. LTI response via the Fourier transform; Lat05:7.4
   5. Fourier transform of periodic signals and sampling; Lat05: 8.1-8.6

III. Laplace transform and its properties
   1. Laplace transform; Lat05: 4.1-4.3
   2. properties and applications in circuits and control; Lat05: 4.4
   3. Bock diagrams, signal flow graphs, Mason’s Gain formula; Lat05: 4.5
   4. Control system performance specification; Dor10:5
   5. PD/PID controllers; Dor10:7.6
   6. Feedback control system design; Dor10:10
   7. Bode plots; Lat05:4.9
   8. Filter design via pole/zero placement; Lat05: 4.10

IV. Introduction to continuous state models and their solutions, Lat05: 10.1-10.5
## 3.5 Discrete Structures

This subject focuses on the fundamental concepts in discrete structures used in computation. Topics include sets, trees, graphs, functions, relations, recurrences, proof techniques, logic, and combinatorics.

### 3.5.1 References


### 3.5.2 Topics to be covered

<table>
<thead>
<tr>
<th>I. Logic and Proofs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposition; truth table; logical operators; converse; contrapositive; inverse; logically equivalent; Tautology; logical equivalence laws; Predicate; universal and existential quantifier; Rules of inference: modus ponens, modus tollens, etc.</td>
</tr>
<tr>
<td>Proofs; theorem; propositions; axioms; proof methods: direct proof, proof by contraposition, proof by contradiction, proof with equivalences, etc.</td>
</tr>
<tr>
<td>Ros12: 1.1, 1.3-1.6</td>
</tr>
<tr>
<td>Ros12: 1.7-1.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>II. Sets, Functions, Sequences, Summations and Countability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets; set membership; set equality, subset; proper subset; cardinality; power set; Cartesian product; tuples; Union; intersection; difference; complement; disjoint sets</td>
</tr>
<tr>
<td>Functions; domain; codomain; range; types of functions: 1-to-1, onto, 1-to-1 correspondence, injective, surjective, bijective, inverse function; function composition; mod; div</td>
</tr>
<tr>
<td>Sequences; arithmetic and geometric sequences; recurrence relation; summation; properties of summations; Cardinality; countable; uncountable; Cantor’s diagonalization argument</td>
</tr>
<tr>
<td>Ros12: 2.1-2.2</td>
</tr>
<tr>
<td>Ros12: 2.3</td>
</tr>
<tr>
<td>Ros12: 2.4-2.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>III. Mathematical Induction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Induction; inductive proofs; strong induction; recursion</td>
</tr>
<tr>
<td>Ros12: 5.1-5.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IV. Counting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product and sum rule; inclusion-exclusion principle; pigeonhole principle; permutation; combination; binomial coefficient and theorem; Pascal’s identity; repetition; indistinguishable items</td>
</tr>
<tr>
<td>Ros12: 6.1-6.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>V. Recurrence Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recurrence relations; modeling and solving recurrence relations; characteristic equation/roots;</td>
</tr>
<tr>
<td>Ros12: 8.1-8.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VI. Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary relation; properties of relations; combining relations; composition of relations; powers of relations; n-ary relations; selection, projection, and join operator; Zero-one matrices; directed graphs; diagonal relation;</td>
</tr>
<tr>
<td>Closures of relations: reflexive, symmetric, transitive; inverse relation; Equivalence relation; equivalence classes; partition</td>
</tr>
<tr>
<td>Ros12: 9.1-9.3</td>
</tr>
<tr>
<td>Ros12: 9.4-9.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VII. Graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph; edge; vertex; adjacency; degree; handshaking theorem; types of graph; special graphs; subgraphs; union; Graph representation; Paths; circuits; simple path/circuit; connected component; strongly/weakly connected; cut vertices; cut edge/bridge; Euler path/circuit; Hamilton path/circuit; conditions for Euler/Hamilton paths/circuits</td>
</tr>
<tr>
<td>Planar graphs; Euler’s formula; conditions for planar graph; Kuratowski’s theorem; elementary subdivision; homeomorphic; Graph coloring; dual graph; chromatic number; four color theorem</td>
</tr>
<tr>
<td>Ros12: 10.1-10.5</td>
</tr>
<tr>
<td>Ros12: 10.6-10.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VIII. Trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree; rooted trees; tree terminology; types of trees; Binary search tree; tree traversals; spanning tree; tree search methods</td>
</tr>
<tr>
<td>Ros12: 11.1-11.4</td>
</tr>
</tbody>
</table>
3.6 Computer Networks

This subject focuses on the fundamentals of computer networking including network design and analysis, in the context of computer communications.

3.6.1 References


3.6.2 Topics to be Covered

I. INTRODUCTION

1. Introduction to networks and network layers Pet11: 1
2. Getting connected to a network (ARP and DHCP) Pet11: 3.2.6, 3.2.7

II. APPLICATION LAYER

1. Network application - DNS Pet11: 9.3.1
2. Network application - HTTP Pet11: 9.1.2
3. Network application - Overlay networks Pet11: 9.4

III. TRANSPORT LAYER

1. Error detection, reliable delivery, sliding windows, acknowledgements, TCP, UDP Pet11: 2.4, 2.5, 5
2. Congestion control & quality of service Pet11: 6

IV. NETWORK LAYER

1. IP packets and routing Pet11: 3.2, 3.3, 3.4
2. Internetwork routing Pet11: 4.1, 4.2

V. DATA LINK LAYER

1. Sharing resources: Frequency division multiplexing, time division multiplexing, code division multiplexing, CSMA/CA, CSMA/CD, ALOHA. Pet11: 2.6, 2.7
2. Ethernet and WiFi Pet11: 2.6, 2.7

VI. PHYSICAL LAYER

1. Message encoding Pet11: 2.2

VII. GENERAL NETWORK SECURITY

1. Encryption, public/private key systems, message authentication codes, firewalls. Pet11: 8

VIII. PROGRAMMING

1. Familiarity with writing and reading C code that uses sockets. For an extensive tutorial, see: http://beej.us/guide/bgnet/output/html/singlepage/bgnet.html
3.7 Computer System Architecture

This subject will focus on principles, practices, technology, and performance enhancement, of Computer Systems including both qualitative and quantitative aspects. You do not need to be familiar with case studies of any specific commercial products.

3.7.1 References


Note: Some Appendices are not in the hardcopy of the book. They are available online at the URL listed in the Table of Contents.

3.7.2 Topics to be Covered

<table>
<thead>
<tr>
<th>Topics</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>I. Hierarchical Memory Systems</strong></td>
<td></td>
</tr>
<tr>
<td>1 Generic Memory Hierarchies</td>
<td>Sta08: 4.1, 4.4</td>
</tr>
<tr>
<td>2 Cache Organization and Performance</td>
<td>Sta08: 4.2-4.3</td>
</tr>
<tr>
<td>3 Cache Interfacing &amp; Write Policies</td>
<td>Sta08: 4.3</td>
</tr>
<tr>
<td>4 Optimizing Cache Performance</td>
<td>Sta08: 4.3, App. D</td>
</tr>
<tr>
<td>5 Main Memory Technology and Organization</td>
<td>Sta08: 5.1, App. E</td>
</tr>
<tr>
<td>6 SDRAM Organization &amp; Performance</td>
<td>Sta08: 5.3, App. K</td>
</tr>
<tr>
<td>7 Magnetic Disk Storage Technology &amp; Organization</td>
<td>Sta08: 6.1</td>
</tr>
<tr>
<td>8 Virtual Memory &amp; Virtual Address Translation</td>
<td>Sta08: 8.3</td>
</tr>
<tr>
<td><strong>II. Pipelining and Processor Performance</strong></td>
<td></td>
</tr>
<tr>
<td>1 Introduction to Pipelining</td>
<td>Sta08: 12.3-12.4</td>
</tr>
<tr>
<td>2 Processor Instruction Pipes: Issues &amp; Options</td>
<td>Sta08: 12.4, I</td>
</tr>
<tr>
<td>3 Pipelining in RISC Processors</td>
<td>Sta08: 13.4-13.5</td>
</tr>
<tr>
<td>5 Tomasulo's Algorithm for Pipeline Management</td>
<td>Sta08: App. I</td>
</tr>
<tr>
<td>6 Superscalar Pipelines</td>
<td>Sta08: 14.1-14.2</td>
</tr>
<tr>
<td>7 Handling Control Flow Hazards</td>
<td>Sta08: 12.4, 12.5</td>
</tr>
<tr>
<td><strong>III. Advanced Architecture Topics</strong></td>
<td></td>
</tr>
<tr>
<td>1 Parallel Processing and Metrics</td>
<td>Sta08: 2.5, 17</td>
</tr>
<tr>
<td>2 Cache Coherence Protocols, including MESI</td>
<td>Sta08: 17.3</td>
</tr>
<tr>
<td>3 Multithreading &amp; Multicore Processing</td>
<td>Sta08: 18</td>
</tr>
<tr>
<td>4 RAID Organization &amp; Performance</td>
<td>Sta08: 6.2</td>
</tr>
</tbody>
</table>
3.8 Embedded Systems

This subject focuses on embedded system topics, especially: assembly language programming, interfacing with I/O devices, synchronization schemes and bus architectures. You are not expected to memorize details regarding any particular commercial processor or ISA. However, you may be asked to read, interpret, and write assembly code, and/or translate functions between C code and assembly code.

Any and all assembly language questions will refer to the very simple *Generic Assembly Language* defined in Subsection 3.8.3 below. A copy of Subsection 3.8.3 will be appended to the exam for you. However, it is recommended that you familiarize yourself with the language beforehand, including the purposes and typical uses for each instruction.

3.8.1 References


3.8.2 Topics to be Covered

I. I/O Interfacing

1. I/O Modules and Parallel I/O  Sta08: 7.2
2. Polled I/O  Sta08: 7.3, Cat05: 1
3. Interrupt-Driven I/O  Sta08: 7.4, Cat05: 1
4. DMA  Sta08: 7.5, Cat05: 1
5. I/O Processors  Sta08: 7.6

II. Computer Interconnections

1. Interconnection Structures  Sta08: 3.3
2. Bus Interconnection and Arbitration  Sta08: 3.4
3. Serial Communications  Cat05: 7, Cat05: 9

III. Assembly Language Programming

1. Addressing Modes  Sta08: 11.1, Cat05: 2
2. Instruction Formats  Sta08: 11.3, Cat05: 2
3. Register Organization  Sta08: 12.2, Cat05: 2
3.8.3 **Generic Assembly Language for Embedded Systems Questions**

- The CPU data word, bus data word, and address word are *all* 32 bits in length.
- The address space is byte-addressable, and all word-length data is *aligned* on a word boundary.
- The Processor has 16 Architectural Registers numbered R₀...R₁₅. All are 32 bits wide:  
  - R₀ holds the constant value 0x0000,0000; any data written into R₀ is lost,
  - R₁...R₁₂ are general purpose user registers,
  - R₁₃ is the Frame Pointer (FP); it can be listed as “FP” in an instruction,
  - R₁₄ is the Stack Pointer (SP); it can be listed as “SP” in an instruction,
  - R₁₅ is the Program Counter (PC); it can be listed as “PC” in an instruction.

3.8.3.1 **Basic Syntax:**

- A bracketed expression \([X]\) specifies that the value of \(X\) is being used as a memory address.
- A pound sign (#) specifies a 16-bit *signed* (2s complement) immediate data value.
- A double slash (//}} begins a comment, which continues until the end of the line.

3.8.3.2 **Load/Store (LS) Instructions**

- **LD** \(R_D, [R_A, \#V]\) // Load Register \(R_D\) with the value at memory address \([R_A + V]\).
- **ST** \(R_D, [R_A, \#V]\) // Store the value in Register \(R_D\) to memory address \([R_A + V]\).
- **LDIH** \(R_D, \#V\) // Load immediate value \(V\) into high order 2 bytes of Register \(R_D\).
- **LDIL** \(R_D, \#V\) // Load immediate value \(V\) into low order 2 bytes of Register \(R_D\), and extend the sign bit of \(V\) into high order 2 bytes of \(R_D\).
- **PUSH** \(R_D\) // First execute \(SP = SP - 4\), then store the value in register \(R_D\) at memory address \([SP]\).
- **POP** \(R_D\) // First load register \(R_D\) with the value at memory address \([SP]\), then execute \(SP = SP + 4\).

3.8.3.3 **Arithmetic and Logic (AL) Instructions**

- **ADD** \(R_D, R_i, R_j\) // \(R_D = R_i + R_j\) note: a given Reg may appear in more than one location.
- **SUB** \(R_D, R_i, R_j\) // \(R_D = R_i - R_j\) note: a given Reg may appear in more than one location. 
  - MUL, DIV, AND, NAND, OR, & XOR use the same syntax.
- **SHLZ** \(R_D, N\) // Shift value in \(R_D\) Left by \(N\) bits, shift zeroes into low order bits.
- **SHRZ** \(R_D, N\) // Shift value in \(R_D\) Right by \(N\) bits, shift zeroes into high order bits.
- **SHRS** \(R_D, N\) // Shift value in \(R_D\) Right by \(N\) bits, shift the *sign bit* into high order bits.

3.8.3.4 **Control Transfer (CT) Instructions**

- **BEQ** \(R_D, Label\) // If value in \(R_D = 0\), then branch to instruction at address \([Label]\).
  - BNE, BGT, and BLT use the same syntax.
- **CALL** \(Label\) // Equivalent to: PUSH PC, PC = Label.
- **RET** // Equivalent to: POP PC.