Digital Design (VIMIAA01)
Requirements and schedule

Tamás Raikovich
BUTE DMIS
Lecturers

• Mr. Béla Fehér
  – Room: I.E.417.
  – E-mail: feher@mit.bme.hu

• Mr. Tamás Raikovich
  – Room: I.E.335.
  – E-mail: rtamas@mit.bme.hu
  – Homepage: http://home.mit.bme.hu/~rtamas
Requirements

- **Lecture: 3 hours/week**
  - Monday 9:15 – 12:00, room IB.144

- **Classroom practice: 1 hour/week**
  - Tuesday 9:15 – 10:00, room IL.405
  - Problem solving and preparing for the laboratory

- **Laboratory practice: 2 hours/week**
  - Thursday 12:15 – 14:00, room IL.405
  - Implementing simple designs on real hardware

- **7 credits: about 210 hours work is required**
Requirements

- **Mid-term requirements for the signature:**
  - *Participating in the lectures, classroom practices and laboratory practices is compulsory*
  - 2 homeworks and 1 test
  - You won’t get the signature if you
    - Miss more than 4 (30%) lectures or classroom practices
    - Miss more than 2 laboratory practices
    - Don’t have enough points from the homeworks and test

- **Examination period:**
  - Exam (if you have the signature)
  - Final mark: 75% exam points + 25% mid-term points
Textbook

- The lecture presentations can be downloaded from my homepage
Schedule

- **Chapter 1: Introduction**
  - Introduction to the digital systems
  - Binary and hexadecimal number systems
  - *Binary number representations*
  - *Source coding (data compression)*
  - *Channel coding (error detection/correction)*
  - *Position codes (Johnson code and Gray code)*

- **Chapter 2: Combinational logic design**
  - Boolean algebra
  - Logic gates: AND, NAND, OR, NOR, XOR, XNOR, NOT
  - Representation of Boolean functions
    - Truth table, sum of minterm form
  - Functional elements: multiplexers, decoders
Schedule

• Chapter 3: Sequential logic design / Controllers
  – Bit storage elements: D flip-flop, register
  – Describing the behavior of sequential circuits: FSMs
  – Implementing sequential circuits: controllers

• Chapter 6: Optimizations
  – Two-level combinational logic size minimization
    • Algebraic method
    • Graphical method: Karnaugh maps
  – Mealey and Moore finite state machines
  – State encoding: one-hot, Gray and output encoding
  – State reduction: implication table, partition method
Schedule

- Chapter 4: Datapath components
  - Building blocks for storing and transforming data
- Chapter 5: Register-Transfer Level (RTL) design
  - Combining controllers and datapaths to create custom data processors
- Chapter 7: Physical implementation
- Chapter 9: Hardware description languages
- Chapter 8: Programmable processors
- The order of the chapters may change