E 3.05
Digital System Design

Peter Cheung
Department of Electrical & Electronic Engineering
Imperial College London

URL: www.ee.ic.ac.uk/pcheung/
E-mail: p.cheung@ic.ac.uk

Aims and Objectives

- How to go about designing complex, ***high speed*** digital systems (not just circuits)?
- How to use some of the modern ***CAD tools*** to help with the design?
- How to implement such designs using ***programmable logic*** (e.g. FPGAs)?
- How to read ***data sheets*** and make sense of them?
- How do ***digital building blocks*** (such as memory chips, processing elements, arithmetic circuits etc.) work?
- How to interface to processors and computers (from hardware point of view)?
- How to deal with testing of complex systems?
- Have fun!

Course Syllabus

- The course syllabus is divided into five main sections:
  - **Programmable Logic**
  - **Arithmetic Circuits**
  - **Data Encoding & communication**
  - **Architectures**
  - **Testing**

Course Syllabus (1) – Programmable Logic

- Technologies behind programmable logic
- Programmable Logic architectures in general
- Complex Programmable Logic Devices (CPLDs)
- Field Programmable Gate Arrays (FPGAs)
- Recent advances in FPGAs
- Designing with FPGAs
- Design Flow, Design Tools, Design Libraries
- Future of programmable logic

Course Syllabus (2) – Arithmetic Circuits

- Adders architectures
- Multipliers circuits
- Floating point arithmetic circuits
- Other computational building blocks

Course Syllabus (3) – Data Encoding & Communication

- Logic interface standards
- Clocking for high speed digital design
- Metastability issues
- Clock synchronisation
- Data encode and error correction
- On-chip and On-board communication

Course Syllabus (4) – Hardware Architectures

- Parallel vs serial
- Systolic and other array architectures
- Distributed arithmetic
- Cordic based architecture

Course Syllabus (5) – Digital Test

- Modern packaging
- Board testing issues
- JTAG Boundary Scan

Recommended Books

No perfect textbook for this course. Here are four reasonable possibilities:

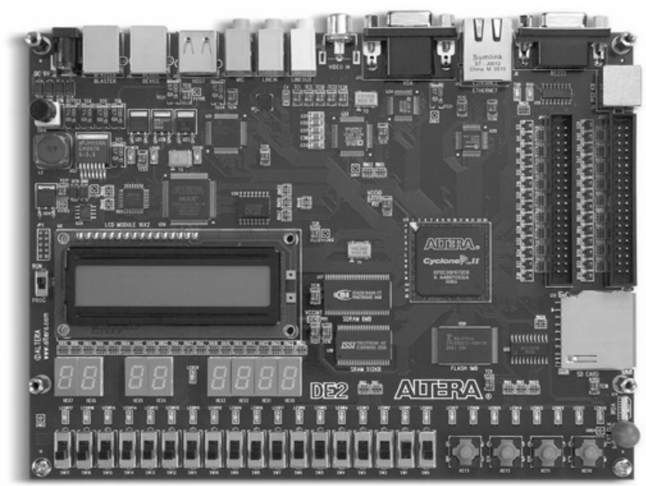
- *"Digital Design Principles and Practices", 4th Edition (Sept 2005), John F. Wakerly, Prentice Hall.*
  - This is a new edition of a well established textbook. It covers a significant portion of the materials taught on this course. At ~£45, this a bargain. Recommended purchase if you have not already done so!
- *"Contemporary Logic Design", Gaetano Boriello, Randy H. Katz, August 2004, Prentice Hall.*
  - Good coverage on finite state machines and computer architectures. (~£45)
- *"High-Speed Digital Design - A handbook of black magic", Howard G. Johnson*, Prentice Hall, 1993; ISBN 0-13-395724-1 (£61).
  - The best practical guide to designing and building very high speed digital circuits. Expensive reference for your company to buy (not you).
- *"FPGA-based System Design", Wayne Wolf, Prentice Hall, 2004, ISBN 0131424610 (£75)*
  - *Contemporary book based on FPGA; possibly too expensive for what it covers*

Coursework

- Best way to learn DSD is to do it!
- Unassessed coursework:
  - 2 to 8 lab exercises using DE2 Board (from Altera) to learn the system
- Assessed coursework:
  - Design of a cordic based processor to add ripple effect on an image
  - Work in pairs – one deliverable between the pair
- Deliverables:
  - Working design and demonstrator
  - Design document (effectively a no-nonsense report)
  - Deadline: 1st day of the Summer Term
- Quartus-II software has a web-edition that can be downloaded (free) from Altera website after you register
- Software also available on all Level 5 & Level 1 machines
- DE2 Boards available on Level 5 and Level 1 Labs

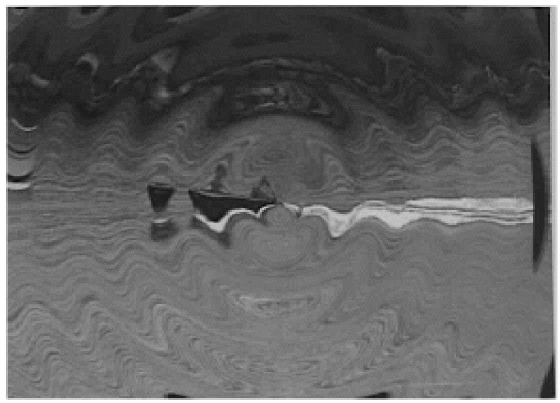
Coursework – DE2 Board

- http://www.altera.com/education/univ/materials/boards/unv-de2-board.html



Coursework – Demo

- This shows an example of the ripple video effect which is the goal of this coursework



Topic 1

Design Methodologies & Implementation Technologies

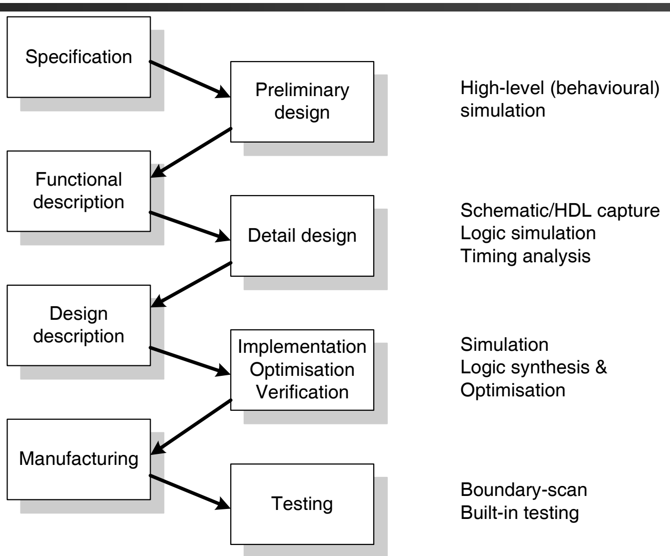
Peter Cheung
Department of Electrical & Electronic Engineering
Imperial College London

URL: www.ee.imperial.ac.uk/pcheung/
E-mail: p.cheung@imperial.ac.uk

Levels of Design Abstractions

Design Levels	*Design Descriptions*	*Primitive Components*	*Theoretical Techniques*
Algorithmic	Specifications High-level lang. Math. equations	Functional blocks 'black boxes'	Signal processing theory Control theory Sorting algorithm
Functional	VHDL, Verilog FSM language C/Pascal	Registers Counters ALU	Automata theory Timing analysis
Logic	Boolean equations Truth tables Timing diagrams	Logic gates Flip-flops	Boolean algebra K-map Boolean minimization
Circuit	Circuit equations Transistor netlist	Transistors Passive comp.	Linear/non-linear eq. Fourier analysis

The Design Process



The Design Process (cont')

- **Top-down design strategies**
  - Refine Specification successively
  - Decompose each component into small components
  - Lowest-level primitive components
  - Over-sold methodology - only works with plenty of experience
- **Bottom-up design strategies**
  - Build-up from primitive components
  - Combined to form more complex components
  - Risk wrong interpretation of specifications
- **Mixed strategies**
  - Mostly top-down, but also bits of bottom-up
  - Reality: need to know both top level and bottom level constraints

Design Descriptions

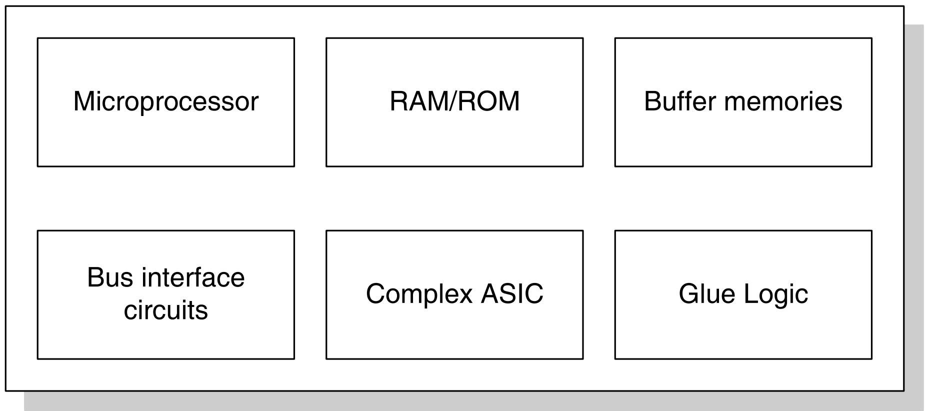
Schematic capture	*Hardware Description Languages*
Good for multiple data flow	Flexible & parameterisable
Give overview picture	Excellent for optimisation & synthesis
Relates to hardware better	Direct mapping to algorithms
Doesn't need to be good in computing	Excellent for datapaths
High information density	Readily interfaced to optimiser
Back annotations possible	Easy to handle and transmit (electronically)
Mixed analogue/digital possible	
Not good for algorithms	*Essentially serial representation*
Not good for datapaths	*May not show overall picture*
Doesn't interface well in optimiser	*Often need good programming skill*
No good for synthesis software	*Divorce from physical hardware*
Difficult to reuse	*Need special software*
Not parameterisable	

Design Tools

Modern digital design systems contain the most of the following features:

- Schematic Capture
- Hardwar Description Language
- Logic Synthesis & Optimisation
- Timing Analysis
- Parameterised Library Components
- Hierarchical Design Management
- Symbol Editing
- Simulation with Timing
- Autoplacement and Routing
- Floorplan Editing
- Reporting

A typical digital system



Implementation Technologies