VLSI design

Course overview

P.J. Drongowski
SandSoftwareSound.net

Course overview

- Technological alternatives
- Basic CMOS design
  - Logic and storage (irsim)
  - Electrical behavior (spice)
  - Building blocks
  - Fabrication process
  - Design rules
  - Chip planning and layout
- Example: A systolic array processor
- Design synthesis
  - Modular design and communication
  - Estimation (again)
  - High level modelling
  - Datapath
  - Control
    - Microcode
    - Finite state machines (FSM)
- Automated synthesis
- Packaging
- Testing

COS / ELE 420

- Design of VLSI systems
  - Room 103
  - Tuesday and Thursday 1:30 - 2:45pm
- Instructor
  - Paul J. Drongowski
  - Room 211, Tuesday and Thursday 2:45 - 3:45pm
  - Extension: X-YYYY
  - Phone: (609) XXX-YYYY
  - E-Mail:
- Teaching assistant one
  - TBD
- Teaching assistant two
  - TBD - Will handle software tools

Course work

- Assignments
  - Spice simulation
  - Logic / switch level simulation
- Midterm examination
- Project
  - Proposal
  - Progress report
  - FInal report

Grading

Assignments	10%
Midtern examination	30%
Project	60%

Textbook and notes

- Drongowski, VLSI system design notes (required)
- Weste, Principles of CMOS VLSI Design (recommended)
- Wolf, Modern VLSI Design (recommended)
- User manuals, etc. (on demand)

Tools

- C language (high level modelling)
- irsim / cosmos (switch level simulation)
- Spice (electrical simulation)
- fsm2oct and oct (FSM generation)
- ITD dlmv2.2 CMOS standard cells

Project

- Design is an activity - not a spectator sport!
- Major portion of the final grade
- Will consume a lot of effort and time
- Deliverables
  - Proposal
    - Brief description of system to be designed
    - Right to the point; emphasize functionality
    - How will you test the system?
    - Suggest early development of C model
    - Three pages
  - Progress report
    - Work completed; work to be performed
    - Major problems encountered
    - Two pages + C language system model
  - Final report
    - Fully exercised irsim model
    - Spice simulation of custom cells
    - Complete layout
    - Develop strong conclusions
- Typical report format
  - Problem statement
  - Approach
  - High level, abstract model
  - Concept of operation and block diagram
  - Chip plan
    - Floor plan (relative size and placement)
    - Pin count and assignment
    - Speed and power estimation
  - Testing and validation
  - Conclusions
  - References

Project ideas

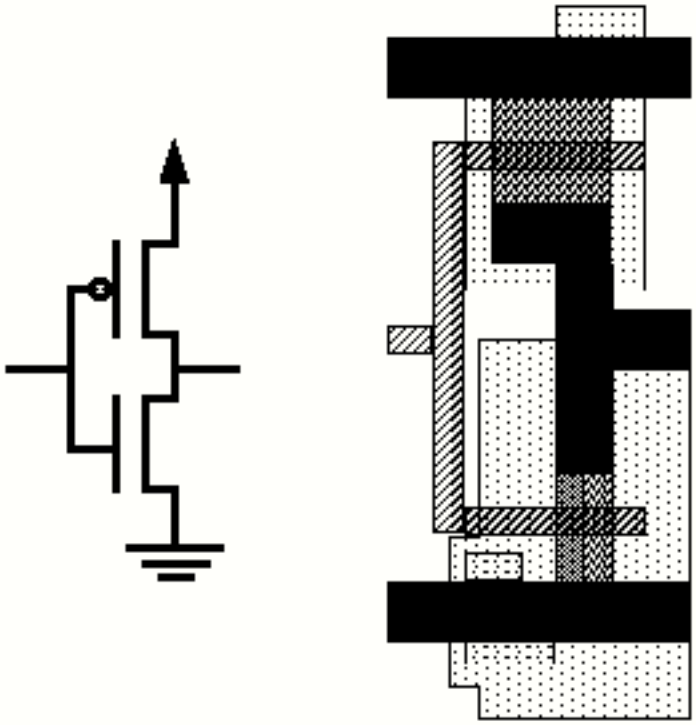
- General guidelines
  - Small, but significant
  - Completely design and simulate in one semester
  - Essential feature set, not bells and whistles
  - Simple, well-executed design and implementation
  - No ``unfinished masterpieces''
  - Use the library
  - Think about the interface, signalling, timing
  - Plan ahead for testing
  - Be pessimistic, be self-critical
  - Be prepared to explain and justify your work
- Example: encryption engine
  - Encrypt and decrypt stream of text
  - Linear feedback shift register
  - Basic engine is simple
  - Interface to microcomputer bus is the challenge
- Example: telegraph keyer
  - Simple finite state machine
  - Again, external interface is the real work
- Example: systolic array
  - Simple processing element replicated many times
  - Try to achieve high density and speed
  - Scan-in, scan-out testing of PE's
- Example: generalized logic block
  - Similar to field programmable gate array cell
  - Programmability (download control info)
  - Interconnection network to form large systems

The times we design in

- Time to market is very short
- Pricing is very competitive
- High quality is critical
- Future designs
  - Problems at the system level
  - Combination of software and hardware
  - Major building blocks plus glue
  - System integrators
- Tall, thin designers
  - System engineering
  - Get the functionality right
  - Technological implications and constraints
- Concurrent engineering
  - Multi-disciplinary design
  - Team effort
  - Balance many design concerns

Levels of abstraction

- Properties to emphasize and specify
- Properties to hide



Level to level correspondence

- Map design objects and semantics
  - Transistor -> physical transistor site
  - Interconnection -> physical wire
  - Zero or one -> voltage
- Add or suppress details
- Inter-level testing and validation
  - Different fault models
  - More (less) design information to generate tests