CS250
VLSI Systems Design
Lecture 2: Introduction

Fall 2011
Krste Asanovic', John Wawrzynek
with
John Lazzaro
and
Brian Zimmer (TA)

So what has changed in 30 years?
Moore’s Law Growth and Effects
Secondary driver: Wafer size


Wafer size conversions offset trend of increasing wafer processing cost

Processing advances

4µm

45nm
IC Technology Stuff (1)

- **Feature size:**
  - then: 4 µm, now: 0.32 µm, moving to: 0.28 µm

- **Interconnect:**
  - then: 2 layers, now: ~10 layers, then: aluminum, now: copper

- **Transistors:**
  - then: planar MOSFET, now: same

- **Layout and GDRs:**

- **Circuits:**
  - then: clocked static CMOS, now: same (lots of crazy stuff in between)
  
  Interesting, though, most CMOS circuits and layouts designed in 1980 would work if fabricated on today’s IC process.

IC Technology Stuff (2)

- **Transistors:**
  - then: near perfect switch, now: leaky

- **Power consumption:**
  - then: dynamic (switching) energy, now: approaching 50% static leakage (back to the future - nMOS has similar problem)

- **New improved devices coming soon:** FinFETs

- **Chip Input/Output**
  - then: parameter pads, now: often area pads

- **Lithographic Mask Costs:**
  - then: few $k, now: $M (full die, 65, 45, 28nm)
IC Technology Stuff (3)

- **Device reliability:**
  - *then:* devices nearly never fail
  - *future (<65nm):* high soft and hard error rates

- **Process variations across die, die-to-die:**
  - Statistical variations in processing (wire widths/resistivity, transistor dimensions/strengths, doping inconsistencies) become apparent at smaller geometries.
  - Some circuits fast, others slow. Some high-power, some low.

- **Yield on leading edge processes dropping dramatically**
  - IBM quotes yields of 10 – 20% on Cell processor

Design Stuff

- **Chip functionality:**
  - *then:* limited by area
  - *now:* usually limited by energy dissipation

- **Design cost:**
  - *now:* design costs in $50M range for full-die custom designs (high percentage in verification)

- **Implementation Alternatives:** more alternatives that trade up-front design costs for per unit costs.

- **FPGA compete aggressively with custom silicon**
  - *then:* most custom designs implemented at silicon level
  - *now:* many more custom designs implemented with FPGAs

- **Standard design abstraction:**
  - *then:* transistors circuits
  - *now:* RTL in HDLs, standard "cores" and standard cells (higher productivity, somewhat less area/energy efficient)
**Implementation Alternatives**

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full-custom:</td>
<td>All circuits/transistors layouts optimized for application.</td>
</tr>
<tr>
<td>Standard-cell:</td>
<td>Arrays of small function blocks (gates, FFs) automatically placed and routed.</td>
</tr>
<tr>
<td>Gate-array (structured ASIC):</td>
<td>Partially prefabricated wafers customized with metal layers or vias.</td>
</tr>
<tr>
<td>FPGA:</td>
<td>Prefabricated chips customized with loadable latches or fuses.</td>
</tr>
<tr>
<td>Microprocessor:</td>
<td>Instruction set interpreter customized through software.</td>
</tr>
<tr>
<td>Domain Specific Processor:</td>
<td>Special instruction set interpreters (ex: DSP, NP, GPU).</td>
</tr>
</tbody>
</table>

By “ASIC”, most people mean “Standard-cell” based implementation.

**What are the important metrics of comparison?**

**The Important Distinction**

- **Instruction Binding Time**
  - When do we decide what operation needs to be performed?

- **General Principles**
  - *Earlier the decision is bound, the less area, delay/energy required for the implementation.*
  - *Later the decision is bound, the more flexible the device.*
Full-Custom

- Circuit styles and transistors sizes are customized to optimize die, size, power, performance.
- High NRE (non-recurring engineering) costs
  - Time-consuming and error prone layout
- Optimizing for small die can result in low per unit costs, extreme-low-power, or extreme-high-performance.
- Common for analog design.
- Requires full set of custom masks.
- High NRE usually restricts use to high-volume applications/markets or highly-constrained and cost insensitive markets.

Standard-Cell*

- Based around a set of pre-designed (and verified) cells
  - Ex: NANDs, NORs, Flip-Flops, buffers, ...
- Each cell comes complete with:
  - layout (perhaps for different technology nodes and processes),
  - Behavioral simulation, delay, & power models.
- Chip layout is automatic, reducing NREs (usually no hand-layout).
- Requires full set of masks - nothing prefabricated.
- Non-optimal use of area and power, leading to higher per die costs than full-custom.
- Commonly used with other design implementation strategies (large blocks for memory, I/O blocks, etc.)
Gate Array

- Store prefabricated wafers of "active" & gate layers & local interconnect, comprising, primarily, rows of transistors. Customize as needed with "back-end" metal processing (contact cuts, vias, metal wires). Could use a different factory.

- Two-step manufacture:
  - First (deep) processing steps
  - Standard masks
  - Base wafers

- Customization:
  - Contacts & metal layers
  - Custom masks
  - ASIC

Gate Array

- Shifts large portion of design and mask NRE to vendor.
- Shorter design and processing times, reduced time to market.
- Highly structured layout with fixed size transistors leads to large sub-circuits (ex: Flip-flops) and higher per die costs.
- Memory arrays are particularly inefficient, so often prefabricated, also:
  - Sea-of-gates, structured ASIC, master-slice.
Field Programmable Gate Arrays

- Two-dimensional array of simple logic- and interconnection-blocks.
- Typical architecture: LUTs implement any function of n-inputs (n=3 in this case).
- Optional Flip-flop with each LUT.

- Fuses, EPROM, or Static RAM cells are used to store the "configuration".
  Here, it determines function implemented by LUT, selection of Flip-flop, and interconnection points.

- Many FPGAs include special circuits to accelerate adder carry-chain and many special cores: RAMs, MAC, Enet, PCI, SERDES, ...

Traditional FPGA versus ASIC argument (circa 2000)

- ASIC: High NRE costs (∼2M for 0.35um chip). Relatively Low cost per die.
- FPGAs: Very low NRE costs. Relatively low silicon efficiency ⇒ high cost per part.
- Cross-over volume from cost effective FPGA design to ASIC in the 10K range.
Cross-over Point has Moved Right

- **ASIC:** Increasing NRE costs ($40M for 90nm chip\(^1\)) (verification, mask costs\(^2\), etc.)
  - Fewer silicon designs becomes inevitable.
- **FPGAs:** Move in to fill the need, furthermore, FPGAs better able to follow Moore's Law, relatively cheaper to test.
- **Cross-over volume now >100K.**

\(^1\) Vahid Manian, VP manufacturing and operations, Broadcom Corp.
\(^2\) Roger Minear, Agere Systems Inc, 30-35-layer mask set = $650,000 for 130nm and $1.4M for 90nm.

Post-fabrication Customization

- **Gate Array** like devices (structured ASICs) return to fill the gap. Post-fab customization with limited mask layers.
  - Lower NREs than ASICs, more silicon efficiency than FPGAs.
Hybrids Chip Implementations Abound

- Ex: standard practice in microprocessors that data-paths are full-custom and control (instruction decode, pipeline control) in standard-cells. (Less common recently)

Control (“random”) logic difficult to “regularize”. Relatively small percentage of die area/power. Permits late binding of design changes.

Extra NAND or NOR gates were often added to control section, and some wafers left without metallization, to permit late design fixes through metal mask revisions (gate-array idea).

System-on-chip (SOC)

- Brings together: standard cell blocks, custom analog blocks, processor cores, memory blocks, embedded FPGAs, ...
- Standardized on-chip buses (or hierarchical interconnect) permit “easy” integration of many blocks.
  - Ex: AMBA, Sonics, ...
- “IP Block” business model: Hard- or soft-cores available from third party designers.
- ARM, inc. is the shining example. Hard- and “synthesizable” RISC processors.
- ARM and other companies provide, Ethernet, USB controllers, analog functions, memory blocks, ...

- Pre-verified block designs, standard bus interfaces (or adapters) ease integration - lower NREs, shorten TTM.

SIP, SOP, MCM interesting alternatives.
Early ’80’s Design Methodology and Flow

- **Schematic + Full-Custom Layout**
  - SPICE for critical path,
  - switch-level simulation for overall functionality,
  - hand layout,
  - no power analysis,
  - layout verified with LVS and GDRC

Modern ASIC Methodology and Flow

- **RTL Synthesis Based**
  - HDL specifies design as combinational logic + state elements
  - Cell instantiations needed for blocks not inferred by synthesis (typically RAM)
  - Event simulation verifies RTL
  - “Formal” verification compares logical structure of gate netlist to RTL
  - Place & route generates layout
  - Timing and power checked statically
  - Layout verified with LVS and GDRC
Design Representations

Engineering Challenge

Application

Gap usually too large to bridge in one step, but there are exceptions...

Physics
Properties of a Useful Abstraction

- Hides less important details
  - e.g., for RTL, don’t worry how combinational logic is decomposed into logic gates
- Allows control of more important details
  - e.g., RTL designer still controls how much logic is performed between any two registers
- If done right, provides portable efficiency
  - i.e., same RTL can be implemented as custom logic, standard cells, FPGA, or even vacuum tube logic, with reasonably good results
Logic Synthesis

- Verilog and VHDL started out as simulation languages, but quickly people wrote programs to automatically convert Verilog code into gate level netlists.

- Synthesis converts Verilog (or other HDL) descriptions to implementation technology specific primitives:
  - For FPGAs: LUTs, flip-flops, and RAM blocks
  - For ASICs: standard cell gate and flip-flop libraries. Memory blocks built with special "memory generator" and then hand-instantiated.
**Why Logic Synthesis?**

1. Automatically manages many details of the design process:
   - Fewer bugs
   - Improved productivity
2. Abstracts the design data (HDL description) from any particular implementation technology.
   - Designs can be re-synthesized targeting different chip technologies. Ex: first implement in FPGA then later in ASIC.
3. In most cases, leads to a more optimal design than could be achieved by manual means (ex: logic optimization)

**Why Not Logic Synthesis?**

---

**Main Logic Synthesis Steps**

- **foo.v**
  - Load in HDL file, run macro preprocessor for `define, `include, etc..
  - Compute “parameter” expressions, process generates, create instances, connect ports.
  - Recognize and insert special blocks (arithmetic structures, ...)
  - Expand combinational logic to primitive Boolean representation.
  - Apply Boolean algebra and heuristics to simplify and optimize under constraints.
  - Map generic logic representation to cell instances from chosen cell library.
  - Modern tools incorporate preliminary layout & timing constraints, and attempt timing driven synthesis.
Operators and Synthesis

- Logical operators map into primitive logic gates
- Arithmetic operators map into adders, subtractors, ...
  - Unsigned 2s complement
  - Watch out for *, %, and /
- Relational operators generate comparators
- Shifts by constant amount are just wire connections
  - No logic involved
- Variable shift amounts a whole different story --- shifter
- Conditional expression generates logic or MUX

Y = \sim X <<

CMOS From the Bottom, Up
IC Fabrication and Layout Representation

"Mask" drawings sent to the fabrication facility to make the chips.

Mask set for an n-Fet (circa 1986)

Top-down view:

- Vd = 1V
- Vg = 0V
- Vs = 0V
- I ≈ nA

Dielectric

Layers to do p-Fet not shown. Modern processes have 6 to 10 metal layers (or more) (in 1986: 2).

Masks
- #1: n+ diffusion
- #2: poly (gate)
- #3: diff contact
- #4: metal

Modern processes have 6 to 10 metal layers (or more) (in 1986: 2).
“Design rules” for masks, 1986 ...

Poly overhang. So that if masks are misaligned, channel doesn’t short out.

Minimum gate length. So that the source and drain depletion regions do not meet!

Metal rules: Contact separation from channel, one fixed contact size, overlap rules with metal, etc ...

#1: n+ diffusion
#2: poly (gate)
#3: diff contact
#4: metal

Fabrication
Mask set for an n-Fet ...

Vd = 1V  
Vg = 1V  
Vs = 0V

Top-down view:

Masks
#1: n+ diffusion
#2: poly (gate)
#3: diff contact
#4: metal

How does a fab use a mask set to make an IC?

Start with an un-doped wafer ...

UV hardens exposed resist. A wafer wash leaves only hard resist.

Steps
#1: dope wafer p-
#2: grow gate oxide
#3: deposit undoped polysilicon
#4: spin on photoresist
#5: place positive poly mask and expose with UV.

CS250, UC Berkeley Fall '11
Lecture 02, Introduction 1
Wet etch to remove unmasked ...

HF acid etches through poly and oxide, but not hardened resist.

After etch and resist removal

Use diffusion mask to implant n-type

accelerated donor atoms

Notice how donor atoms are blocked by gate and do not enter channel.

Thus, the channel is "self-aligned", precise mask alignment is not needed!
Metallization completes device

- Grow a thick oxide on top of the wafer.
- Mask and etch to make contact holes.
- Put a layer of metal on chip. Be sure to fill in the holes!

Final product ...

"The planar process"
Jean Hoerni, Fairchild Semiconductor 1958
p-channel Transistors

**p-Fet: Change polarity of everything**

\[ V_{well} = V_s = 1V \quad V_g = 0V \quad V_d = 0V \]

- New "n-well" mask

"Mobility" of holes is slower than electrons.

p-Fets drive less current than n-Fets, all else being equal.
**Bulk versus SIO Processing**

- "Silicon on Insulator"

- Lower parasitic capacitance -> lower energy, higher-performance

- Also used for "radiation hard" application (space craft) - sapphire instead of Oxide.

- 10 - 15% increase in total manufacturing cost due to substrate cost.

**Lithography**

- Optical proximity correction (OPC) is an enhancement technique commonly used to compensate for image errors due to diffraction or process effects.

- Current state-of-the-art photolithography tools use deep ultraviolet (DUV) light with wavelengths of 248 and 193 nm, which allow minimum feature sizes down to 50 nm.
Modern Processing Parameters

From 2009 ITRS Roadmap

<table>
<thead>
<tr>
<th></th>
<th>2010</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td># Mask Levels—MPU</td>
<td>35</td>
<td>37</td>
</tr>
<tr>
<td># Mask Levels—DRAM</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>Maximum Lithography Field Size—area (mm²)</td>
<td>858</td>
<td>858</td>
</tr>
<tr>
<td>Maximum Lithography Field Size—length (mm)</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>Maximum Lithography Field Size—width (mm)</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>Bulk or epitaxial or SOI wafer size (mm)</td>
<td>300</td>
<td>450</td>
</tr>
</tbody>
</table>

International Technology Roadmap for Semiconductors
http://www.itrs.net/

Processing Enhancements

‣ Trench isolation: Shallow trench isolation (STI), a.k.a. Box Isolation Technique, prevents current leakage between n-well and p-well devices.

‣ High-K dielectrics / Metal gate: Replacing the silicon dioxide gate dielectric with a high-κ material allows increased gate capacitance without the concomitant leakage effects.

‣ Strained Silicon: A layer of silicon in which the silicon atoms are stretched beyond their normal interatomic distance leading to better mobility, resulting in better chip performance and lower energy consumption.

‣ “Gate Engineering”: for within-die choice of multiple transistor threshold voltages (Vt) to optimize delay or power.
End of Introduction part 2