Introduction to ASIC Design

Dr. Paul D. Franzon

Outline
1. The wonderful world of Silicon
2. Application Specific Integrated Circuits (ASICs)
   • Typical applications, types, decision making
3. ASIC Design Flow

References
See Web page

The Wonderful World of Silicon

About every two years, the number of transistors on a CMOS silicon chip doubles and the clock speed doubles....This rate of improvement will continue for the next 15-20 years.

Technology Drivers:
• Decreasing lithographic feature size, typically measured by the transistor gate length:
  0.35 µm .... 0.18 µm 0.15 µm .... etc.... 0.050 µm .... 0.01 µm(?)
• Increasing wafer size:
  6 inch diameter ..... 8 inch diameter..... 12 inches
• Increasing number of metal interconnect layers:
  4 ..... 6 ..... 8 ...... 9 (?)
• Approximately constant cost per wafer to manufacture:
  About $2,000 - $4,000 per wafer
• Increasing IC yields for ‘large’ (> 1 sq. cm.chips): 60% .... 90%
Semiconductor Roadmap

Projections for ‘leading edge’ ICs: (www.sematech.org)

<table>
<thead>
<tr>
<th>Year</th>
<th>2002</th>
<th>2003</th>
<th>2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gate Length (nm)</td>
<td>53</td>
<td>45</td>
<td>25</td>
</tr>
<tr>
<td>Clock (PDA)</td>
<td>150 MHz</td>
<td>450 MHz</td>
<td></td>
</tr>
<tr>
<td>Clock (HP)</td>
<td>100-10,000 MHz</td>
<td>600 MHz+</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>1-2 GHz</td>
<td>2-4 GHz</td>
<td></td>
</tr>
</tbody>
</table>

Increasingly chip size is constrained by design technology.

Integrated Circuit Design Trends

Moore’s Law
- Transistors/Chip increasing by 50% per year (by 4x in 3.5 years)

Feature Size and Delay
- Gate Delay decreasing by 13% per year (by ½ in 5 years)
  - 2 input NAND with fanout of 4 (wiring delays becoming a problem)
- Gate Length decreasing by 13% per year (by ½ in 5 years)
- Chip Edge increasing by 6% per year (by 2x in 10 years)
- Wire Pitch decreasing by 13% per year (by ½ in 5 years)
  - distance from center to center of two adjacent wires
- Wire Tracks increasing by 22% per year (by 2x in 3.5 years)
  - number of wires that can be placed side by side for on-chip routing
- Grid Squares increasing by 50% per year (by 4x in 3.5 years)
  - good measure of functionality that can be implemented on a chip ⇒ complexity management
- Grids/Gate_Delay increasing by 70% per year (by 10x in 4 years)
  - capability = functionality/delay

**Gate Length and Gate Delay Trends**

![Chart showing gate length and gate delay trends over time.](chart1)


---

**Wire Pitch and Chip Size Trends**

![Chart showing wire pitch and chip size trends over time.](chart2)

**Wire and Functional Density Trends**

![Graph showing wire and functional density trends over time.](image)


---

**ASICs vs. what?**

**Application Specific Integrated Circuit**
- A chip designed to perform a particular operation as opposed to General Purpose integrated circuits
- An ASIC is NOT software programmable to perform a wide variety of different tasks
- However an ASIC will often have an embedded CPU to manage book-keeping tasks
- An ASIC may be implemented as an FPGA (see later)
  - Sometimes considered a separate category

**General Purpose Integrated Circuits:**
- Programmable microprocessors (e.g. Intel Pentium Series, Motorola HC-11)
  - Used in PCs to washing machines
- Programmable Digital Signal Processors (e.g. TI TMS 320 Series)
  - Used in many multimedia, sensor processing and communications applications
- Memory (dRAM, SRAM, etc.)

©2003, Dr. Paul D. Franzon, www.ece.ncsu.edu/erl/faculty/paulf.html
Examples of ASICs

- Video processor to decode MPEG-2 digital TV signals
- Audio processor to perform Dolby AC3 encoding
- Low power dedicated DSP/controller for cell phone
- Non-CPU portion of network processors
- Encryption processor for security
- Many examples of graphics chips

ASIC Styles

1. Full Custom ASICs
   - Every transistor is designed and drawn by hand
   - Typically only way to design analog portions of ASICs
   - e.g. part of a custom DES (encryption) processor designed as part of a research project at NCSU:

   ![Diagram of a Full Custom ASIC]

   Gives the highest performance but the longest design time
   - Typically only used for analog portions and for very high volume parts (e.g. microprocessors) or for small parts to be used in many different designs
2. Standard-Cell-Based ASICs

- or ‘Cell Based IC’ (CBIC) or ‘semi-custom’
- **Standard Cells** are custom designed and then inserted into a library

  - These cells are then used in the design by being placed in rows and wired together using ‘place and route’ CAD tools
  - Some standard cells, such as RAM and ROM cells, and some datapath cells (e.g. a multiplier) are tiled together to create macrocells
  - Custom designed blocks (e.g. microprocessors) might be mixed in as well (sometimes called megacells or hard macros.)

Sample ASIC floorplan:

- Standard Cell designs are usually synthesized from an RTL (Register Transfer Language) description of the design
- ASIC design is much quicker than full custom design
- A full set of masks is still required for fabrication

Fabless semiconductor company model

- Company does design only. Fab performed by another company (e.g. VLSI, TSMC, UMC).
- Back-end (place and route, etc.) might be performed at that company or with their assistance
3. Gate-Array Based ASICs
   - In a gate array, the transistors level masks are fully defined and the designer can not change them
   - The design instead programs the wiring and vias to implement the desired function
   - Gate array designs are slower than cell-based designs but the implementation time is faster as less time must be spent in the factory
   - RTL-based methods and synthesis, together with other CAD tools, are often used for gate arrays.

4. Programmable Logic Devices (PLDs and FPGAs)
   - FPGA = Field Programmable Gate Array
   - are off-the-shelf ICs that can be programmed by the user to perform various functions (usually just combinational logic functions)
   - There are no custom mask layers so final design implementation is a few hours instead of a few weeks
   - Simple PLDs are used for simple functions. FPGAs are increasingly displacing standard cell designs.
5. Field Programmable Gate Arrays (FPGAs)

- Off-the-shelf chips that the user programs to perform simple functions
- Can be quite complex, capable of implementing 100,000s of gates
- Some companies call them ‘complex PLDs’
- e.g. XILINX:

Programmable Interconnect Array:

Configurable Logic Block (CLB):

Source: XILINX Application notes

ASIC Market

<table>
<thead>
<tr>
<th>MOS Std Cell</th>
<th>MOS PLD</th>
<th>MOS Gate Array</th>
<th>Full Custom</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 (20%)</td>
<td>20 (16%)</td>
<td>16 (18%)</td>
<td>2 (64%)</td>
<td></td>
</tr>
<tr>
<td>36 (10%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 (25%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: WSTS, IC Insights

Forecast

©2003, Dr. Paul D. Franzon, www.ece.ncsu.edu/erl/faculty/paulf.html
Decision Making

Alternative implementation approaches:

- Program a PC or a microcontroller
  - (some microcontrollers have high-performance 32- or 64-bit CPUs)
- Build a full custom, cell-based or gate array ASIC
- Use a PLD or FPGA

<table>
<thead>
<tr>
<th>Approach</th>
<th>Design Productivity (gates/day)</th>
<th>Manufacturing NRE + lead time</th>
<th>Manufacturing cost/unit</th>
<th>Clock frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell-based</td>
<td>200</td>
<td>$50k (full mask set) 8 weeks</td>
<td>1x</td>
<td>1x</td>
</tr>
<tr>
<td>Gate Array</td>
<td>200</td>
<td>$10k &lt; 1 week</td>
<td>~1.2x</td>
<td>~0.8x</td>
</tr>
<tr>
<td>FPGA</td>
<td>500</td>
<td>$0 1 day</td>
<td>~5x</td>
<td>~0.3x</td>
</tr>
</tbody>
</table>

Other Issues:

- Design tools and design training
  - Cell-based ICs require more investment and greater technical sophistication but give more capable designs
- Time to Market
  - Profits increase dramatically with every month shaved off time-to-market (about 30%/3-months, according to one study)
  - Approaches:
    - Consider using an FPGA for product introduction, or at least prototyping
    - Enhance design productivity as much as possible
      - people, tools, training, methods and techniques
    - Good design verification is very important for Cell-based ICs
Some General Notes

- Dominated by standard cell ASICs and FPGAs
  - Ideally standard cell designs would be used for higher volume applications that justify the NRE
- Many consider FPGAs separate from ASICs
- Why? FPGA design characteristics:
  - Different level of design skills required, especially in “back end” (place and route or physical design)
  - Reduced level of verification required before sending to factory
    - Again reduces sophistication required of team
  - Low-cost (barrier) of entry
    - Often different, lower cost Design Automation (CAD) tools
  - Lower performance
- However, front-end design (RTL coding) is virtually identical for each implementation style
- Sometimes FPGA done first and standard cell ASIC done later

ASIC Design Methodology

Most ASICs are designed using a RTL/Synthesis based methodology

- Design details captured in a simulatable description of the hardware
  - Captured as Register Transfer Language (RTL)
  - The two most common RTLs are Verilog and VHDL
- Automatic synthesis is used to turn the RTL into a gate-level description
  - ie. AND and OR gates, etc.
  - Chip-test features are usually inserted at this point
- Physical Design tools are then used to turn the gate-level design into a set of chip masks (for photolithography) or a configuration file for downloading to an FPGA
- At each stage, extensive verification is performed.
ASIC Design Flow
(at a 10,000 foot view)

Specify Design Problem
Develop C Model
Design Hardware
Develop Timing Diagram
Write Verilog Code
Verify Design
  Create Test Fixture
    Perform Pre-synthesis Simulation
Synthesize Verilog into Hardware
Perform Post-synthesis Simulation
  or Equivalency Check
Timing Verification
Back-end Processing...

Iterate steps 1-2, 2-3, 4-6, 5-6, 2-6, 7-8, 2-8, (1-8)

Focus For This Class...

Develop C Model
Design Hardware
Develop Timing Diagram
Write Verilog Code

- It's often useful to write a behavioral simulatable specification
  - Captures intent well
  - Gives an independent source of verification vectors
- Note: Design THEN Code
- Code simply captures the design in a simulatable and synthesizable format.
**Visual view of Tool Flow for ECE 520**

**ECE 464 / ECE 520 ASIC Design**

**Objective:** Learn how to design a digital ASIC using Register Transfer Languages (Verilog), modern synthesis tools (Synopsys) and modern verification and back end tools (Cadence). Learn how the tools constrain and permit modern design. Survey ASIC architectures and conduct a project in multimedia or communications.

---

**Some Major Steps**

Verilog Code, capturing hardware design

```
module LFSR_TSK (clock, Reset, Y, Y2);
input [7:0] clock, Reset;
output [7:0] Y, Y2;
reg [7:0] Y, Y2;
parameter [7:0] x0 = 8’s00000000;
parameter [7:0] x1 = 8’s11111111;
parameter [7:0] Taps1 = 8’s10000000;
parameter [7:0] Taps2 = 8’s00000000;

assign LFSR_TSK_XNOR;
input [7:0] x;
input [7:0] taps;
output [7:0] next_LFSR_Reg;
integer Tap;
reg [7:0] next_LFSR_Reg;

begin
  if (clock && !Reset)
    Y <= x;
  else
    if (Taps2[p] && Taps1[p] && Taps1[p])
      next_LFSR_Reg[p] <= x[p] & x[p];
  next_LFSR_Reg[p+1] <= Feedback;
end

```

The flow from high level design to the results of synthesis is often called the “Front End” of the design flow.
Result of Synthesis

Netlist

- I.e. Schematic showing gates and how they are connected

Visual:

Textual Form:

NAND2 U37 ( .Y(n109), .A0(n105), .A1(n103)
NAND2 U38 ( .Y(n114), .A0(value[1] ), .A1

‘n107’, etc. are nets, I.e. wires that connect the gates together.

The netlist format will usually meet the specified timing (performance) assuming a simple model for wire capacitance (delay).

“Back End” design flow

“Back End” design occurs after synthesis

- Most important steps are Place & Route, Extraction, and Timing Verification
- Sometimes done by the Fab
- Getting increasingly difficult, as wire delays get more important
- Result is a “layout”, I.e. Detailed mask information that can be sent to the mask maker

Standard cells arranged in rows (‘placed’) and wired (‘routed’) together.

This format will meet timing and noise requirements, with actual wire R, C & L taken into account.
Future Tool Issues

Growing Chip Complexity
- 10M+ transistors/chip very close
- combined with a shortage of designers, and need for shorter design turnaround times
  - Design Reuse
    - Components designed to be reused in other designs
    - A methodology for fast integration of library-based designs

Increasing importance of wiring
- Wiring parasitic capacitance increasing faster than gate capacitance
  - leads to wiring having a strong effect on delay and noise
  - Often referred to as the “deep sub-micron” problem because its importance becomes pronounced at gate sizes below 0.18 µm.

**Solution:**
- Short term: More tightly integrated tool flows, better estimators
- Medium term: Combined logical and physical synthesis

Questions
- What basic technological trend drives the Semiconductor Industry?
- What is the key difference between a standard cell ASIC and an FPGA?
- What will be important challenges to future design houses?
- What is a “fabless semiconductor vendor”?
Summary

Over the next ten years, product growth will be driven by:

- Underlying technology push
- High demand for graphics, multimedia and wireless connectivity
- Insidious insertion of electronics and computers into our everyday lives

Many of the resulting products will require specialized silicon chips to meet performance (speed/size/weight/power/cost) demands - ASICs

To match this product need, the capability of a silicon CMOS chip will continue doubling every 2-3 years until after 2015.

- To sell a product at $300-$1,000, it can only include one high value chip
  - Thus product performance is determined by the performance of that one chip
- AND talk about planned obsolescence!!

ASIC styles include full custom (for analog) and RTL-based design: Cell based (semi-custom), Gate Array or FPGA implementation

ASIC design methodology includes logic, timing, and physical design
- Unfortunately, design productivity is not keeping up with chip performance growth