Digital Integrated Circuits
A Design Perspective

Jan M. Rabaey
Anantha Chandrakasan
Borivoje Nikolic
Modified and integrated by Davide Bertozzi

Design Methodologies:
Standard cell design
Impact of Implementation Choices

Three orders of magnitude
Higher efficiency

Energy Efficiency (in MOPS/mW)

None

Hardwired components

100-1000

Configurable/Parameterizable

10-100

Domain-specific processor (e.g. DSP)

1-10

Embedded microprocessor

0.1-1

Providing programmability adds overhead to the implementation

• Late binding
• Re-use across multiple applications
• Software upgrade

0.25um CMOS process

Flexibility comes at a cost in terms of power and performance
Heterogeneous Parallel Computing

- There is not only one perfect mapping solution!
- Architectural heterogeneity and many-cores are THE design paradigm for embedded SoCs
- This is today the way to pursue high performance with energy efficiency, that is high performance-per-watt

Host Processor (multi-core processor)
- Cache-coherent interconnect

Parallel threads heavily dependent on local data content
- Many, truly independent parallel computations (MIMD)
- Branch divergence between threads

General Purpose Parallel Programmable Accelerator
- 2nd-level NoC

- Leading core count
- Single-Instruction Multiple-Data (SIMD)
- OK with thousands of small threads (better if almost identical) to expose massive HW multithreading

Hardware accelerators
- Highest GOPS/W
- No flexibility (lower yield)

FPGA-like fabric
- Power-efficient for some workloads
- Flexible acceleration

DRAM memory controller
- 2nd-level NoC
- Embedded GPU

© Digital Integrated Circuits 2nd Design Methodologies
Implementation Choices for Hardware: both Hardware Accelerators and Microprocessors

Design for high performance/high density: handcrafted full custom design
Design for fast time-to-market: design automation techniques

Digital Circuit Implementation Approaches

- Custom
  - Cell-based
    - Standard Cells
  - Array-based
    - Pre-diffused (Gate Arrays)
  - Semicustom
    - Macro Cells
    - Pre-wired (FPGA's)
The Custom Approach – early days

Intel 4004 microprocessor
The first monolithic processor
(740 KHz, 2300 transistors, 10um)

- high cost
- long time-to-market
It is OK when
- custom blocks can be reused
- cost can be amortized over a large volume (e.g., uP, memories)
- cost is not primary design criterion (e.g., supercomputers)

Key take-away:
When performance or design density are critical, handcrafting circuit topology and physical design are the best option
Transition to Automation and Regular Structures

Evolution of full custom design
- Replication of the **same** custom-designed block multiple times (e.g., memories)
- Composition of **different** custom-designed blocks with a regular composition pattern

In both cases **regularity enables deployment of automation**

© Digital Integrated Circuits 2nd Edition

Courtesy Intel Design Methodologies
Pentium 4 processor (2000)

- Almost all parts were designed **automatically**
  - composing custom blocks together in a regular way (**semicustom design with a library of cells**)
- Performance critical modules (PLL, clock buffers) were still designed **manually**
- Basic automation levels even for full-custom design:
  - layout editors, Design Rule Checkers (DRCs).
Full-custom design

- Complete control over transistor and interconnect dimensions (within design rule constraints)
- Design rules:
  - Minimum spacing between metal lines (varies per layer)
  - Line width
  - Transistor channel length
- Circuit Designers create application-specific building blocks
  - Technology Provider (foundry) provides SPICE/HSPICE transistor models, parasitic extraction tools
  - Models are used to drive transistor sizing/layout design
- Continual verification of design as it becomes more defined
- PRO: Produces Optimized Design (**density**, **power**, **performance**)
- CON: Time-consuming; error-prone, highest NRE
Layout editors

Magic Layout Editor
(UC Berkeley)
Stick Diagrams for Symbolic Layout Design

Stick diagram of an inverter

- Dimensionless layout sketches
- Only topology is important
- Separates transistor schematic from layout design phases
- Designer does not have to worry about design rules
- Final layout generated by a “compaction” program
- Outcome of compaction is often hard to predict (e.g., less dense than expected)
Cell-based semicustom design (CBD)

Predefined and custom-designed cells are instantiated multiple times and interconnected in regular pattern to yield a given logic function

**ADVANTAGES**
- cuts down on design time and costs
- reduces implementation effort by REUSING a library of cells for different designs
  - cells need to be designed and verified once for a given technology node
  - cell reuse amortizes the cost for their full-custom design

**DRAWBACKS**
- Reduced integration density and performance
- No design fine-tuning (i.e., transistor-level) allowed

CBD approaches are categorized based on the granularity of library elements
Standard cells

Standardizes the design entry-level at the logic gate

Based on a library of standard pre-designed, pre-verified cells

• basic logic functions (NOR, NOT, NAND, XOR,..)
• complex functions (basic MUX, decoders, adder, comparator,..)
• storage elements (DFFs, SR latches, ...)
• special cells (e.g., brute-force synchronizers; tie-high; tie-low)
• logic cell variants to cover a wide range of fan-in/fan-out conditions (e.g., 4:1 mux vs 2:1 mux; different transistor sizing)
• characterization for other parameters:
  - supply voltage, threshold voltage, temperature, process
• process corner cases: best, nominal, worst

Foundries or even fabless companies (in partnership with foundries) provide libraries of standard cells for semicustom design with tens or hundreds of cells
Standard cell layout methodology

Strong restrictions on the layout allow high levels of automation (e.g. automatic layout generation)

Row of standard cells (all cells must have same height)

Routing channel requirements are reduced by presence of more interconnect layers

Routing channel

Intermixing with other layout design approaches. For those modules which do not adapt to the logic cell paradigm (e.g., highly regular, more stringent performance requirements)
Standard Cell — Early Example

• Large area overhead for the interconnects
  ✓ Feedthrough cells
  ✓ large routing channels

• Adding more metal layers
  → less requirements on routing channels

[Brodersen92]
Standard Cell – The New Generation

Design in a 7 metal layers technology

Cell-structure hidden under interconnect layers
• Density: 90%
• small area overhead for interconnects
Standard cell structure

Routing channel

signals

PMOS transistors close to the Vdd rail

Intra-cell wiring

NMOS transistors close to the ground rail

Intra-cell wiring

No Routing channels

Mirrored Cell

Cell mirroring enables sharing of power and ground rails
Inverter standard cell layout

- Power rail
- p-mos diffusions
- N-well
- n-mos diffusions
- Ground rail
**Design rules**

- The feature size \( f \) is the minimum spacing between drain and source (min. poly width)
- Design rules expressed in terms of \( \lambda = f/2 \)
- A **wiring track** is the space required for a wire
  - E.g., \( 4\lambda \) width, \( 4\lambda \) spacing from neighbor = \( 8\lambda \) is called «the pitch»
  - The rule applies to transistors as well
Cell height

Cell height 12 metal tracks
(metal track is the M1 pitch)

Tall cells (11 or 12 metal tracks) support more complex routing, larger driving strength transistors and are typically tuned for performance, but may exhibit higher leakage power.

Short cells (7 or 8 metal tracks) are optimized for area efficiency, but generally designed with smaller, lower driving strength transistors, so are less appropriate for high-speed designs.

Standard height cells (9 or 10 tracks) are an intermediate trade-off
3-input NAND cell (from ST Microelectronics):
C = Load capacitance
T = input rise/fall time

<table>
<thead>
<tr>
<th>Path</th>
<th>1.2V - 125°C</th>
<th>1.6V - 40°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1→pLH</td>
<td>0.073+7.98(C+0.317T)</td>
<td>0.020+2.73(C+0.253T)</td>
</tr>
<tr>
<td>L1→pTTL</td>
<td>0.069+8.43(C+0.364T)</td>
<td>0.018+2.14(C+0.292T)</td>
</tr>
<tr>
<td>L2→pLH</td>
<td>0.101+7.97(C+0.318T)</td>
<td>0.026+2.38(C+0.255T)</td>
</tr>
<tr>
<td>L2→pTTL</td>
<td>0.097+8.42(C+0.325T)</td>
<td>0.023+2.14(C+0.269T)</td>
</tr>
<tr>
<td>L3→pLH</td>
<td>0.120+8.00(C+0.318T)</td>
<td>0.031+2.37(C+0.258T)</td>
</tr>
<tr>
<td>L3→pTTL</td>
<td>0.110+8.41(C+0.280T)</td>
<td>0.027+2.15(C+0.223T)</td>
</tr>
</tbody>
</table>

- 5 cell versions
  - C from 0.18 to 0.72 pF
  - area from 16.4 to 32.8 \(\text{um}^2\)

- **Not just performance, but also energy given in datasheet**

**Library cells documentation is critical, although time-intensive**
Technology library – Architectural effects

Power-Length/Speed trade-off for NoC link design

- Short and/or slow-clocked links don’t pose any problem
- Long and/or high-speed links force routing tools to infer a large number of buffering gates, increasing power

From:

© Digital Integrated Circuits 2nd Design Methodologies
A single technology library no longer exists for standard cell design

- An aggressively low-power library (LP-HVT) infers buffers with lower size and speed, resulting in much tighter constraints on operation frequency or length (i.e., link feasibility)
- The spread increases as technology scales down
- We need to pick the right library for specific design constraints
A standard cell library is complemented by an **I/O cell library**

- I/O circuits are analog in nature, and analog delays are not easy to predict/model
  - IC designers are faced with interfacing to a growing diversity of standards and parts
    - Memory, I/O, graphics, networking
    - Standards: DDR, SDRAM, PCI, USB, ...
    - Different signaling methods: LVDS, CML, ...
    - Circuits for latchup, ESD, isolation, ...
    - Ground and power pins (many)
Peripheral pads for bond wires allow the I/O cell circuitry to be placed in alignment with the pads leading to simple logical, electrical and physical structures.

Creating a grid array of pads for flip chip mounting allows for easier alignment in the packaging, but may cause the routing to and from the I/O circuitry to become very complex.
Designing a standard cell library is time consuming, although amortized among a large number of designs.

Today it is common practice to have several cell versions:
- number of inputs
- transistor sizing for different capacitive loads \textit{(driving strength)}
- pull-up/pull-down ratios
- characterization for PVT operating conditions: $V_{th}$, $V_{dd}$, temperature, process corner cases

Non-trivial choice of the mix of logic cells
• small library with most cells having limited fan-ins?
• large library with many versions of the same cell?
• conservative large driving capabilities lead to power/area overhead

Technology libraries are broadly differentiated based on the target design goal (low-power vs. high-performance)

Synthesis tools choose the correct cell version in the library based on speed/area/power constraints on the design.
Mixed-Library Design

During logic synthesis, it is possible to link different technology libraries at the same time to span the performance-power trade-off for the design at hand.

4x4 2D mesh NoC
65nm Library variants:
- Low-Vth (fast)
- High-Vth (low-power)
- Mixed-Vth (multiple Vths)
  - aiming for max performance
  - aiming for a power-perf. trade-off
- Clock gating always enabled except for Low-Vth, to avoid performance penalties

Handle with care:

When you link more libraries, you are increasing mask complexity and fabrication cost, since manufacturing steps for transistors are different.
- Gates on the critical path should come from the fastest library (Low-Vth).
- Gates on non-critical paths should come from the low-power library (High-Vth).
There is almost an order of magnitude difference in the power/performance ratios achievable by LVth and HVth libraries.
## Power-Performance Trade-off

<table>
<thead>
<tr>
<th>Library variant</th>
<th>HVth</th>
<th>MVthA</th>
<th>MVthB</th>
<th>LVth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency target</td>
<td>Max</td>
<td>300 MHz</td>
<td>Max</td>
<td>Max.</td>
</tr>
<tr>
<td>Clock gating</td>
<td>Enabled</td>
<td>Enabled</td>
<td>Enabled</td>
<td>Disabled</td>
</tr>
<tr>
<td>Frequency (MHz)</td>
<td>142</td>
<td>300</td>
<td>714</td>
<td>952</td>
</tr>
<tr>
<td>Bandwidth (GB/s)</td>
<td>27</td>
<td>57</td>
<td>137</td>
<td>183</td>
</tr>
<tr>
<td>Power (mW)</td>
<td>11</td>
<td>25</td>
<td>88</td>
<td>145</td>
</tr>
</tbody>
</table>

Mixed Vth is attractive:
- approaches LVth performance at a lower power
- Approaches HVth performance at almost the same power efficiency (GB/s over mW)