Lecture 7
Overview of Design Flow

Xuan ‘Silvia’ Zhang

Washington University in St. Louis

http://classes.engineering.wustl.edu/ese566/

Application Specific Integrated Circuit (ASIC) Type

- Full-custom
  - transistors are hand-drawn
  - best performance (although almost extinct)
  - still using to optimized standard cell
- Gate Array (for small volumes)
  - use sea of gates (mask-programmable gate arrays)
  - FPGA (reconfigurable)
- Standard Cell
  - only use standard cell from the library
  - dominate design style for ASIC (what we are going to use)

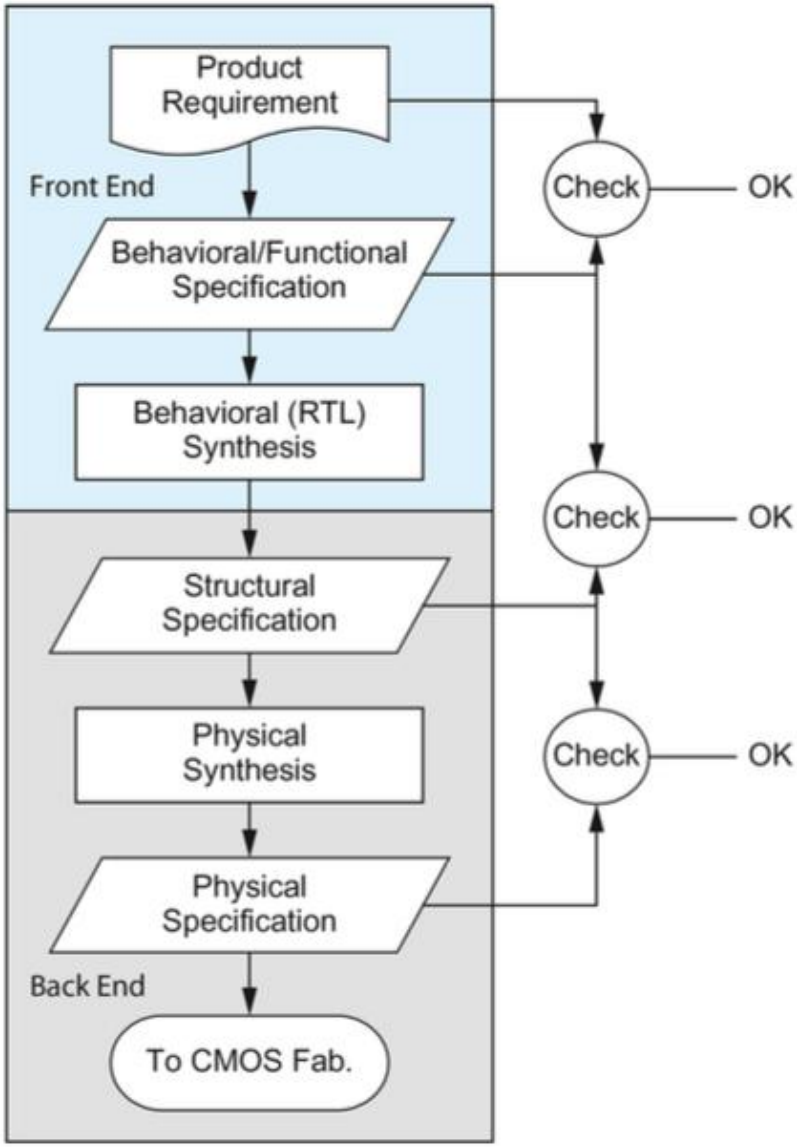
Outline

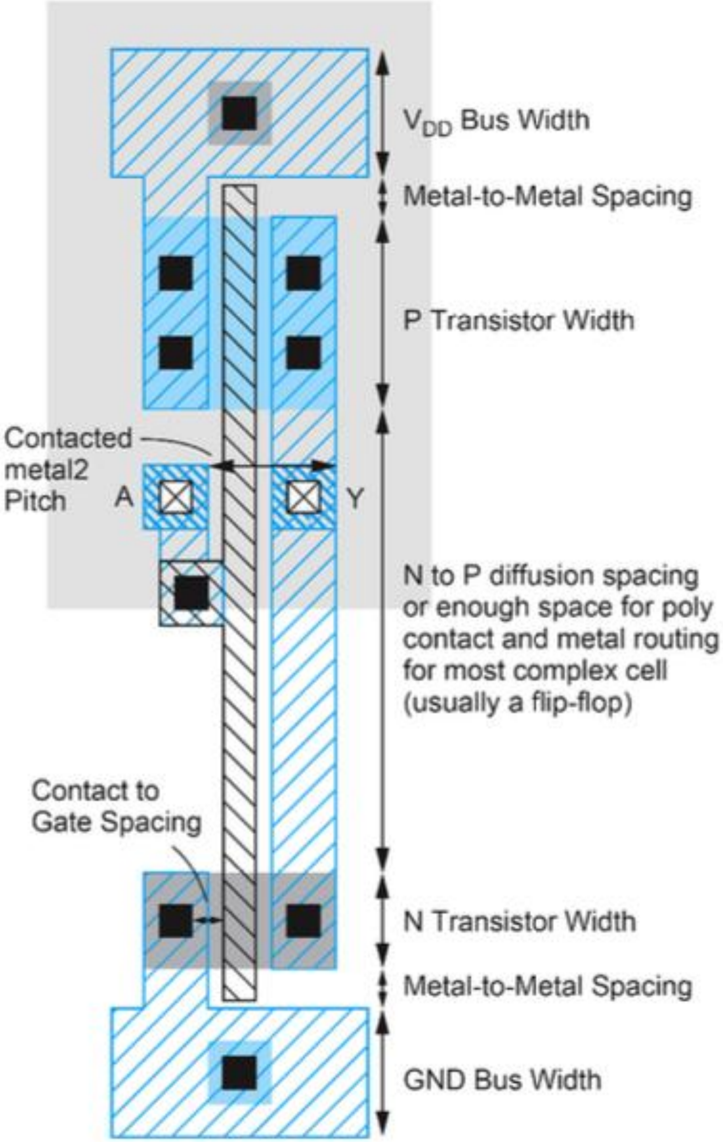
Standard-Cell-Based Design

SoC-Platform-Based Design

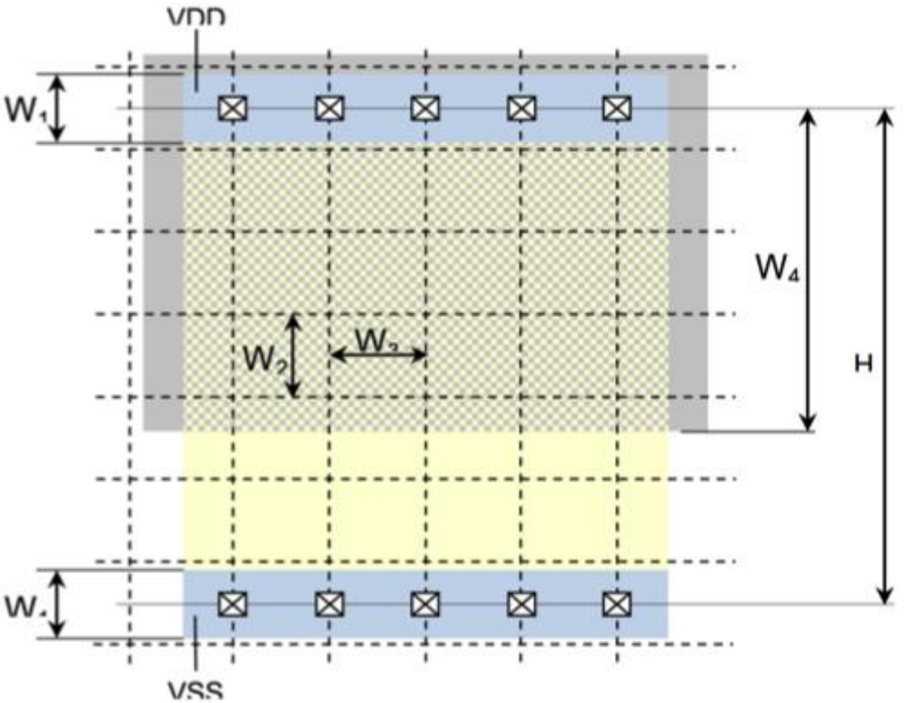
Methodology Comparison

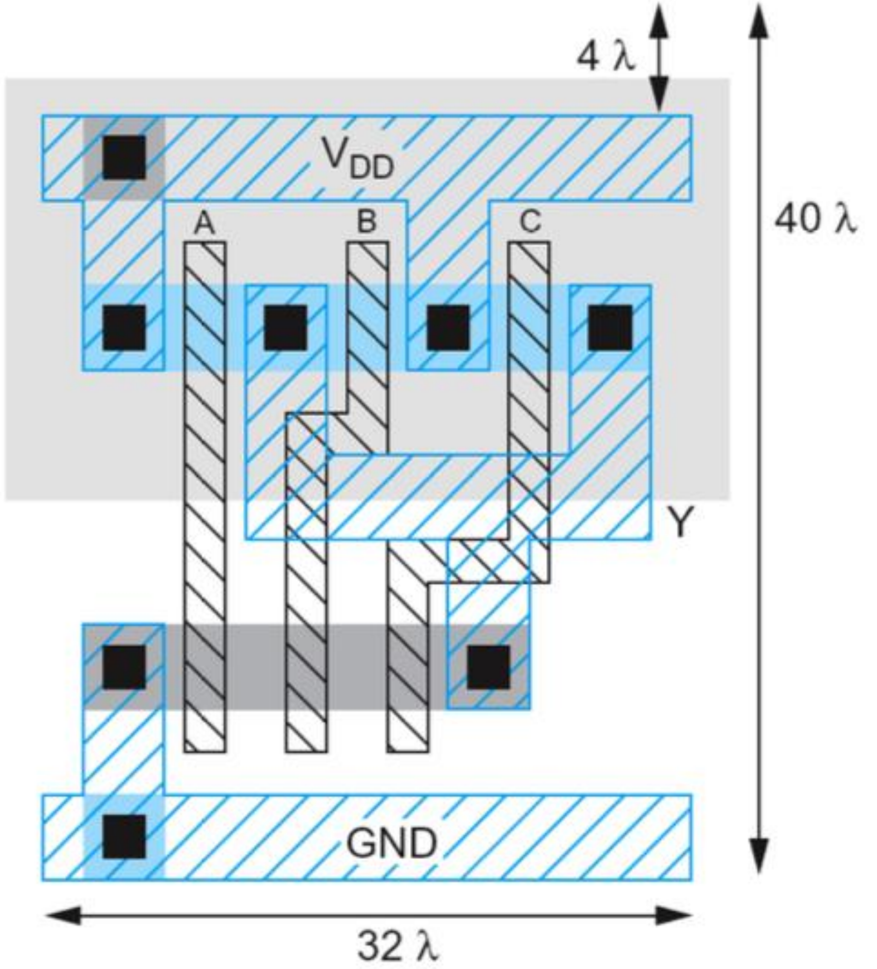
Standard-Cell-Based Design





Example Standard Cell

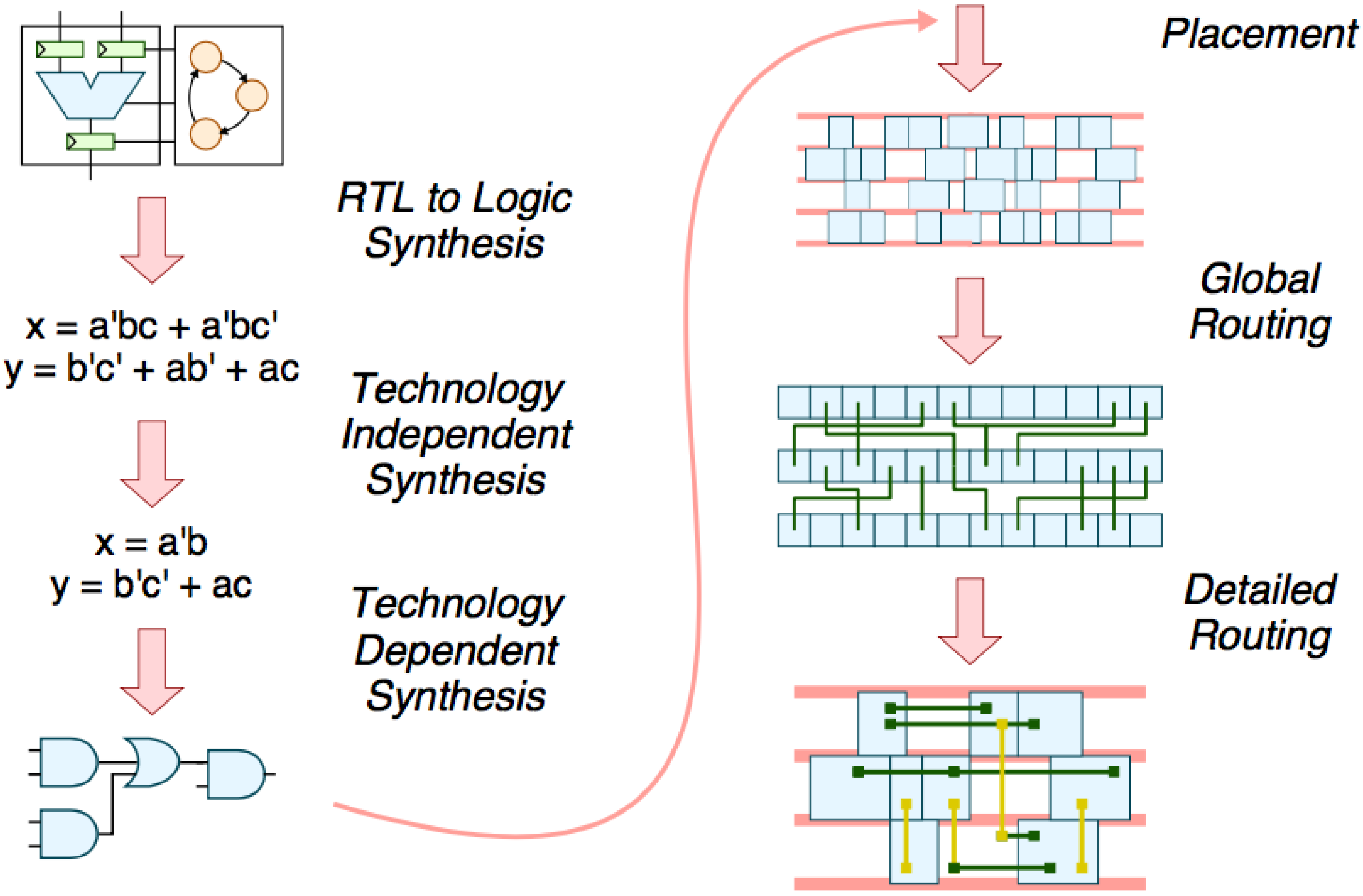




Standard-Cell-Based Flow CAD Algorithms

Synthesis Algorithms

Physical Design Automation



RTL to Logic Synthesis

x = a'bc + a'bc'
y = b'c' + ab' + ac

Technology Independent Synthesis

x = a'b
y = b'c' + ac

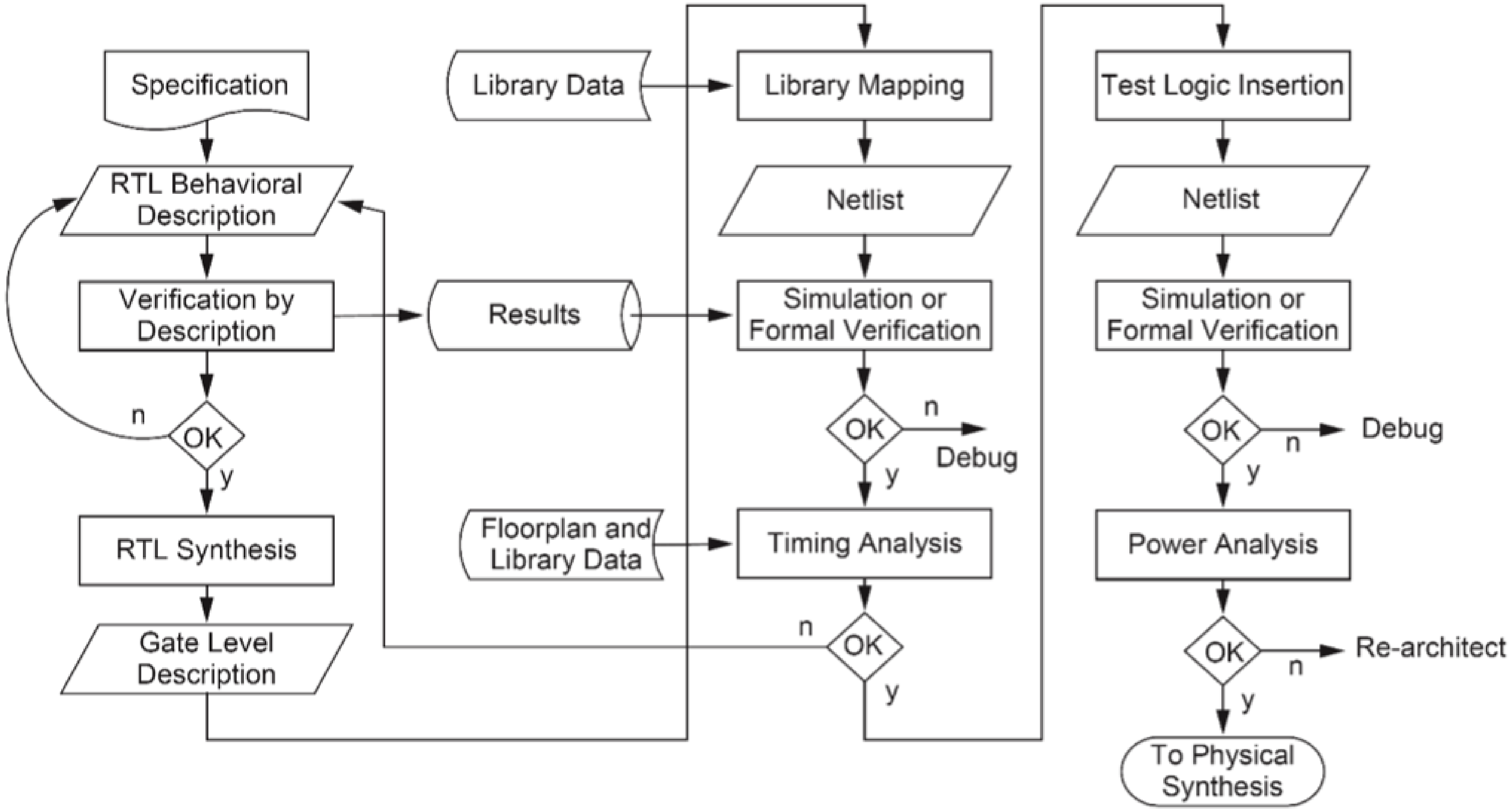
Technology Dependent Synthesis

Placement

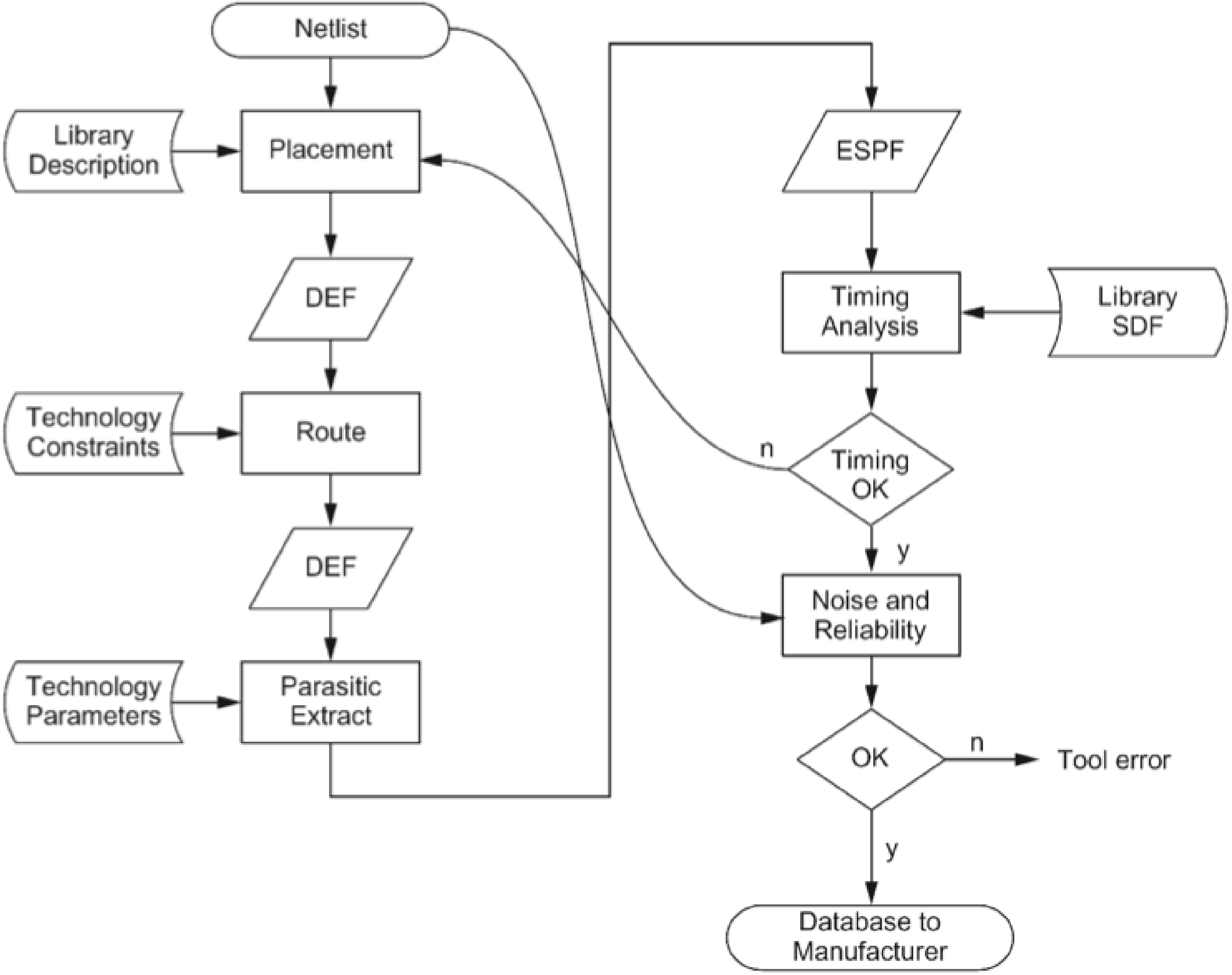
Global Routing

Detailed Routing

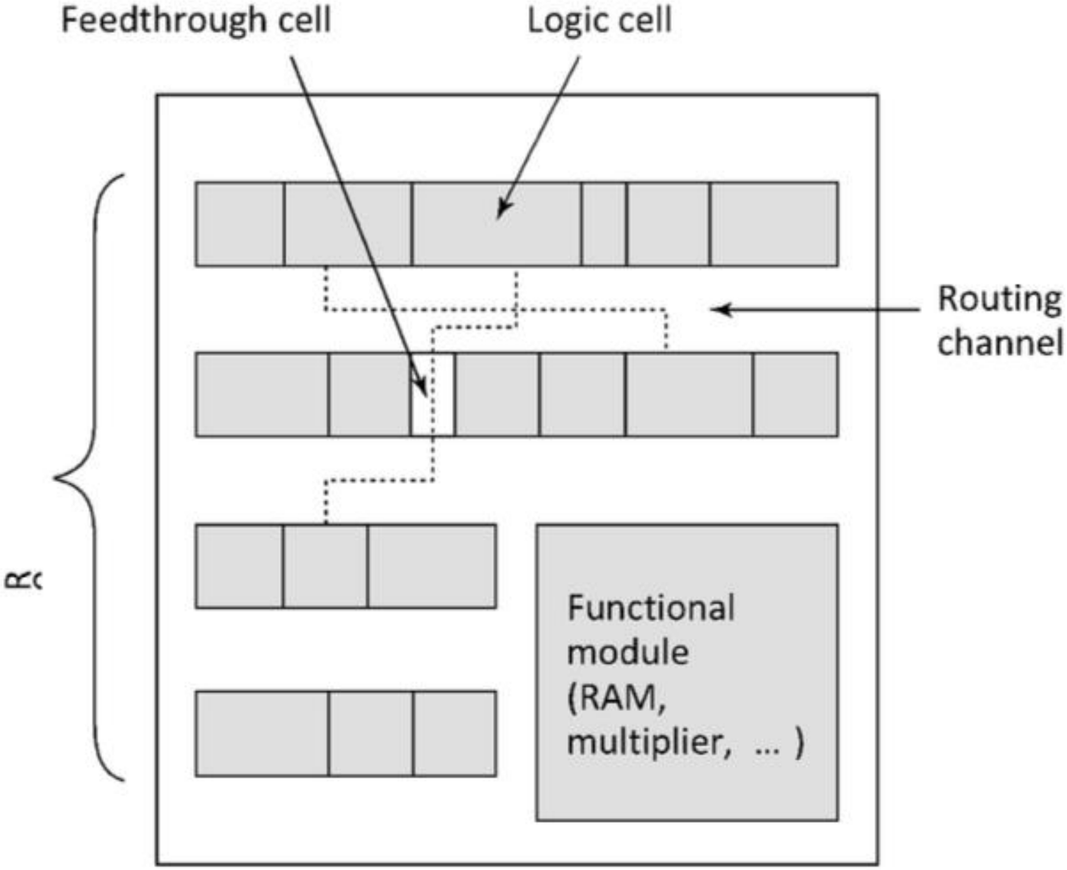
Front-End Flow

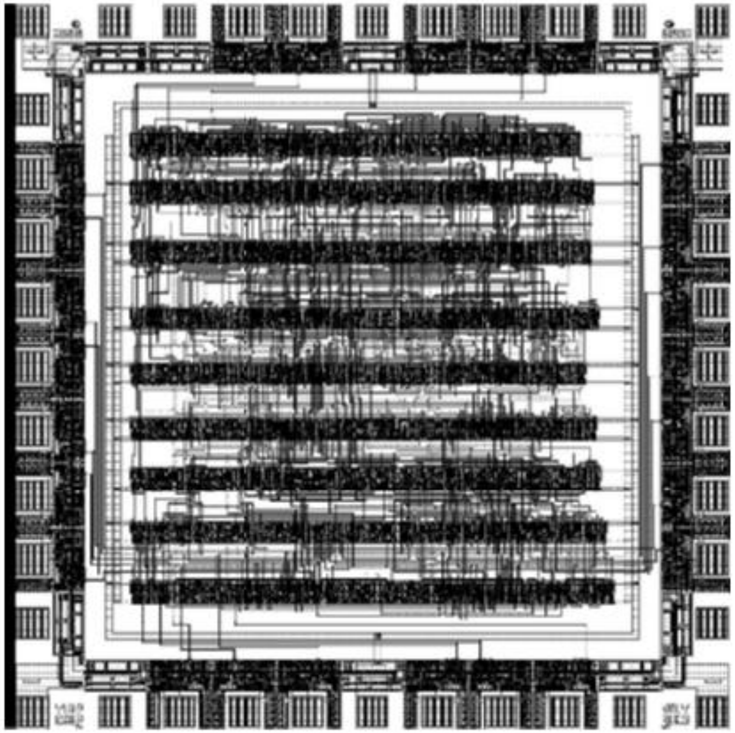


Back-End Flow



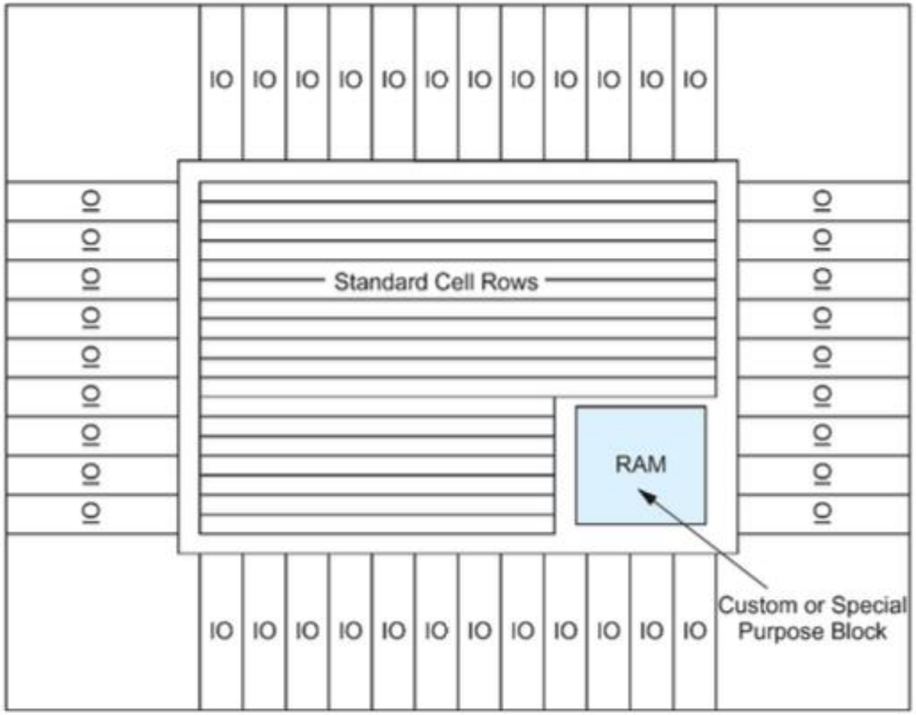
Older Standard-Cell ASICs

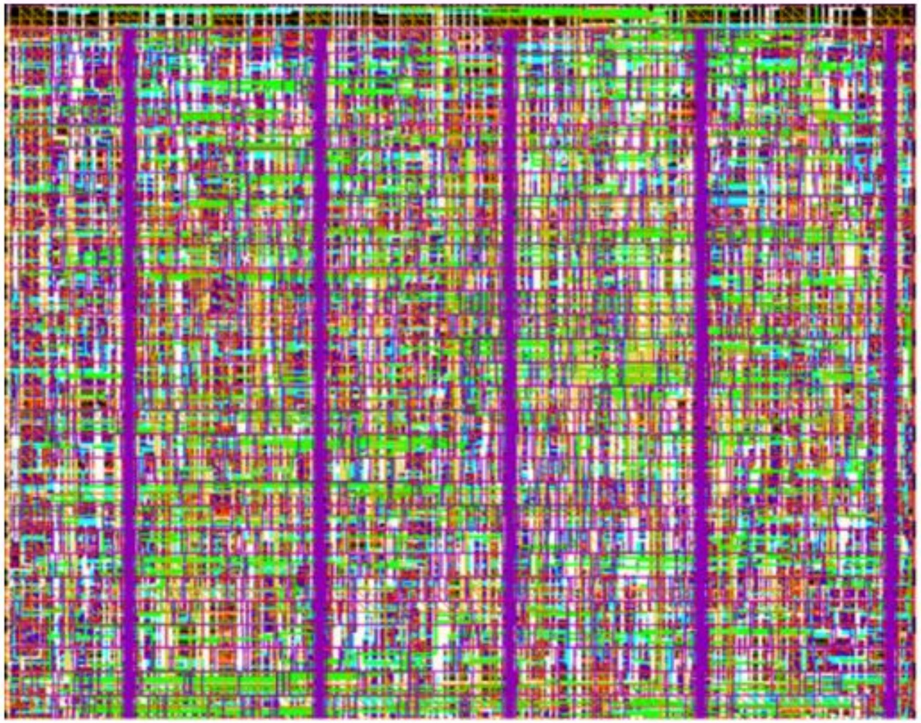




Limited metal layers require dedicated routing channels and feedthrough cells

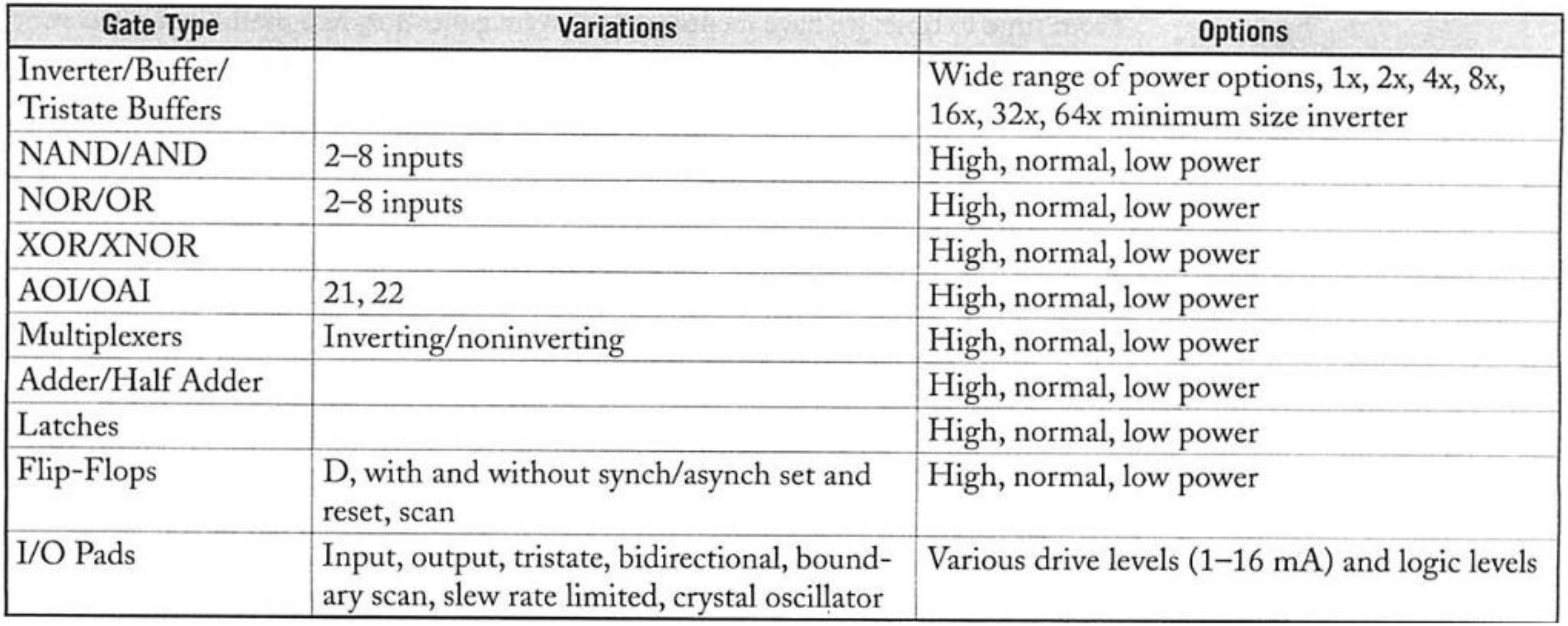
Modern Standard-Cell ASICs





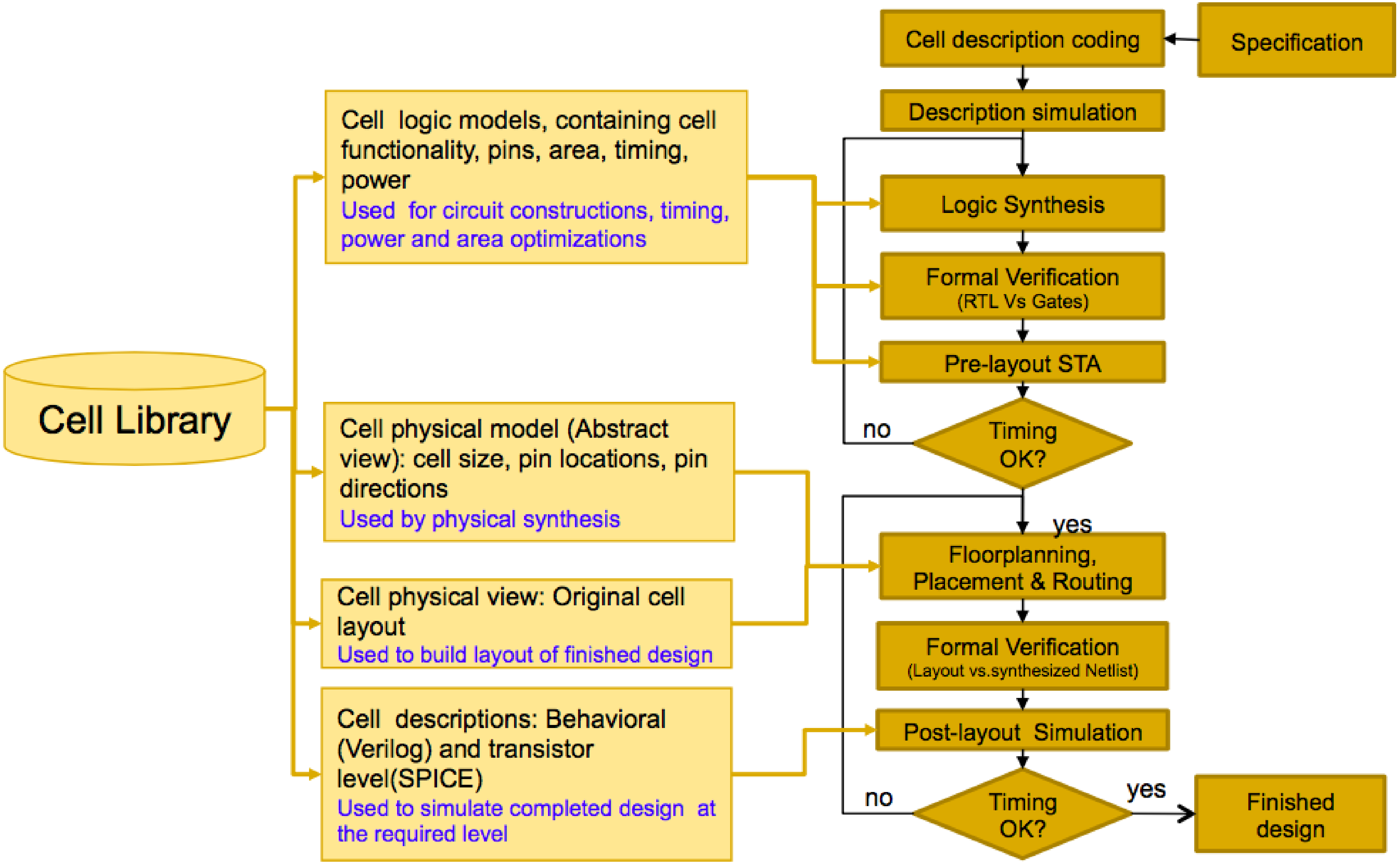
Increasing number of metal layers allow cells to be hidden under interconnect

Standard-Cell Libraries



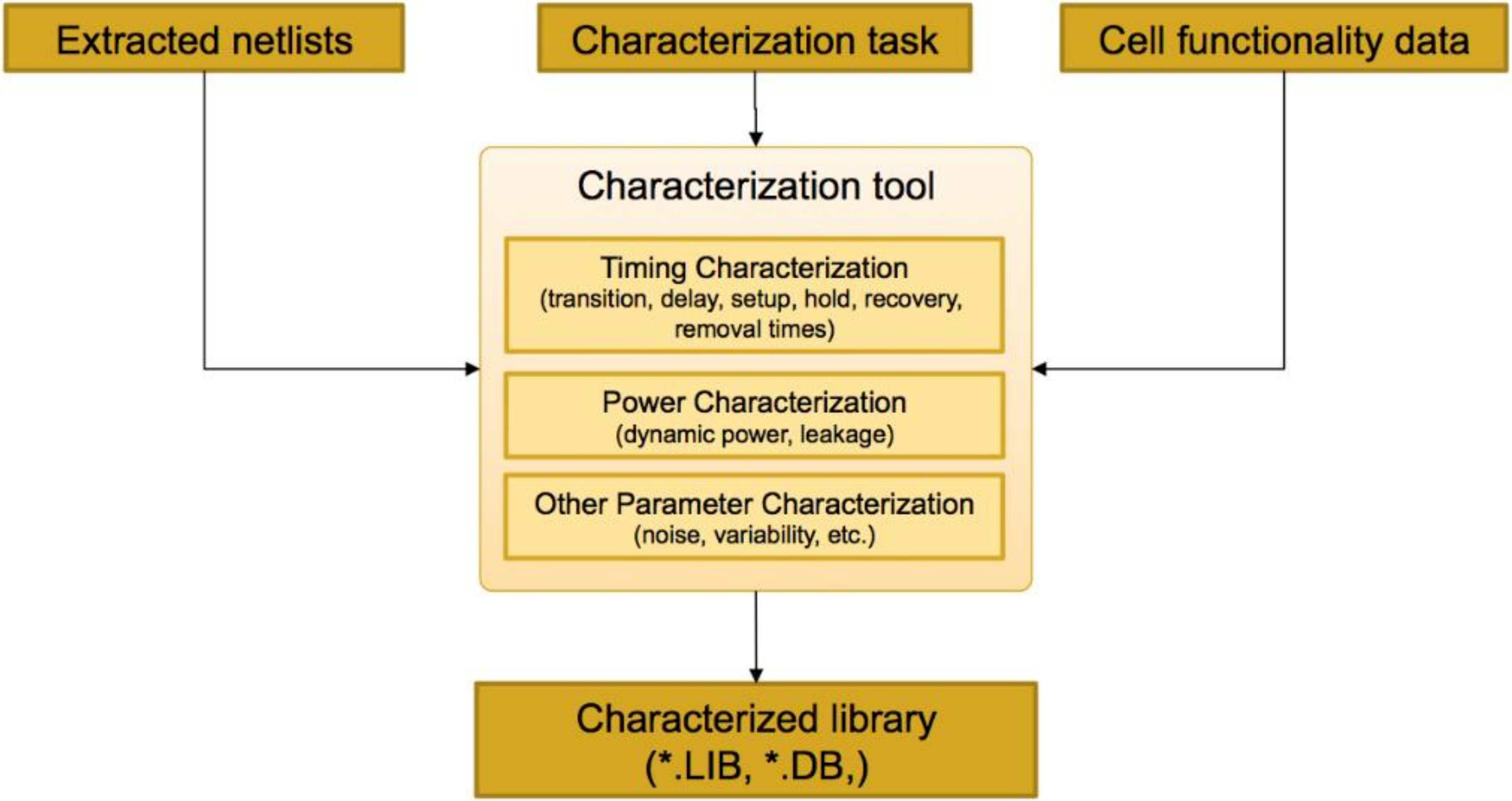
Gate Type	Variations	Options
Inverter/Buffer/ Tristate Buffers		Wide range of power options, 1x, 2x, 4x, 8x, 16x, 32x, 64x minimum size inverter
NAND/AND	2–8 inputs	High, normal, low power
NOR/OR	2–8 inputs	High, normal, low power
XOR/XNOR		High, normal, low power
AOI/OAI	21, 22	High, normal, low power
Multiplexers	Inverting/noninverting	High, normal, low power
Adder/Half Adder		High, normal, low power
Latches		High, normal, low power
Flip-Flops	D, with and without synch/asynch set and reset, scan	High, normal, low power
I/O Pads	Input, output, tristate, bidirectional, boundary scan, slew rate limited, crystal oscillator	Various drive levels (1–16 mA) and logic levels

Standard-Cell Libraries



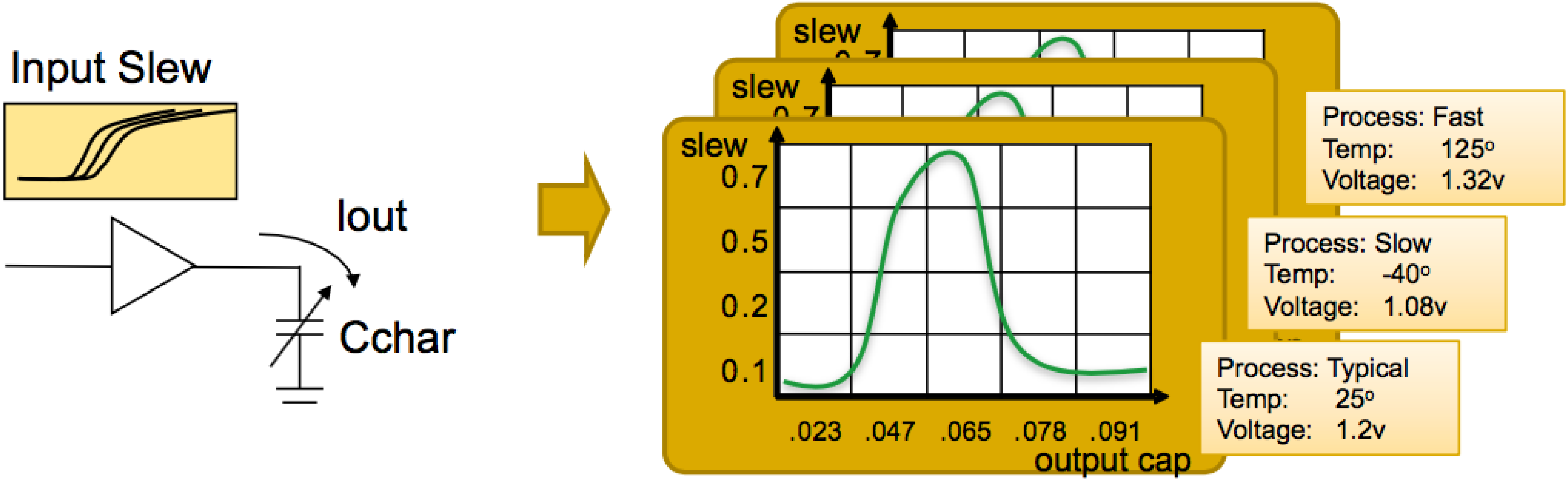
Cell Library
Cell logic models, containing cell functionality, pins, area, timing, power
Used for circuit constructions, timing, power and area optimizations
Cell physical model (Abstract view): cell size, pin locations, pin directions
Used by physical synthesis
Cell physical view: Original cell layout
Used to build layout of finished design
Cell descriptions: Behavioral (Verilog) and transistor level(SPICE)
Used to simulate completed design at the required level
Specification
Cell description coding
Description simulation
Logic Synthesis
Formal Verification
(RTL Vs Gates)
Pre-layout STA
Timing OK?
no
yes
Floorplanning, Placement & Routing
Formal Verification
(Layout vs.synthesized Netlist)
Post-layout Simulation
Timing OK?
no
yes
Finished design

Standard-Cell Library Characterization



Standard-Cell Electrical Characterization

- Characterization computes cell parameters
  - e.g. delay, output current
  - depend on input variables, i.e. output load, input skew, etc.
- Characterization is performed under conditions
  - combination of process, voltage, temperature (PVT)



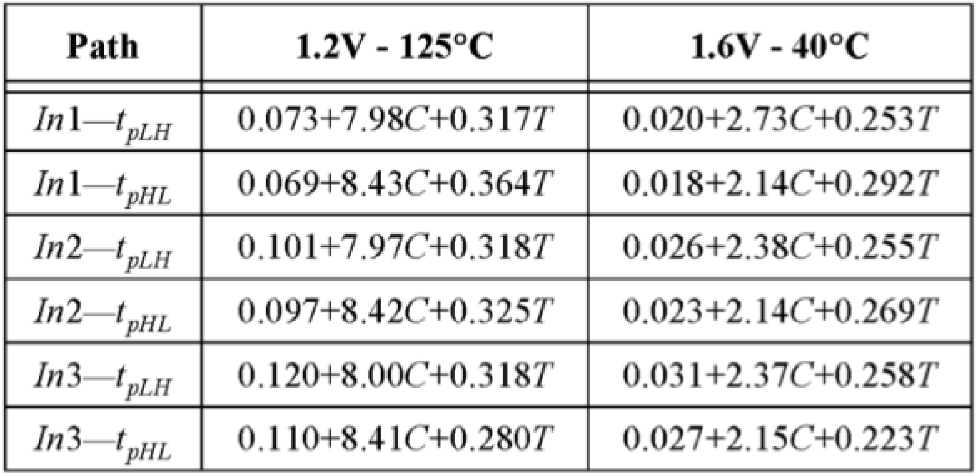
Process: Fast
Temp: 125°
Voltage: 1.32v

Process: Slow
Temp: -40°
Voltage: 1.08v

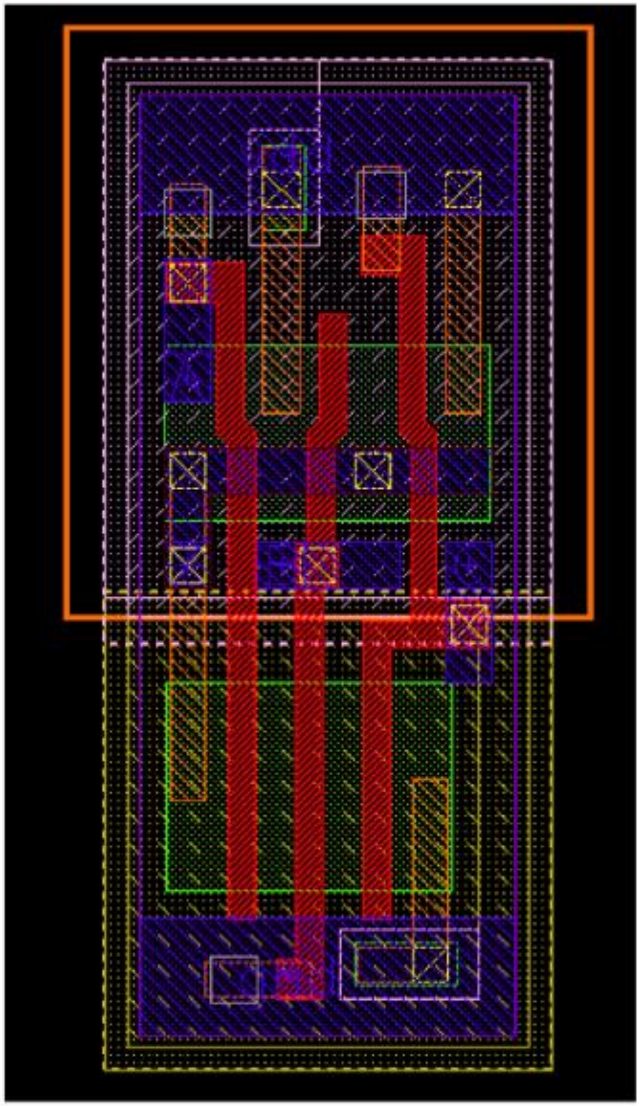
Process: Typical
Temp: 25°
Voltage: 1.2v

ST Microelectronics: 3-Input NAND Gate

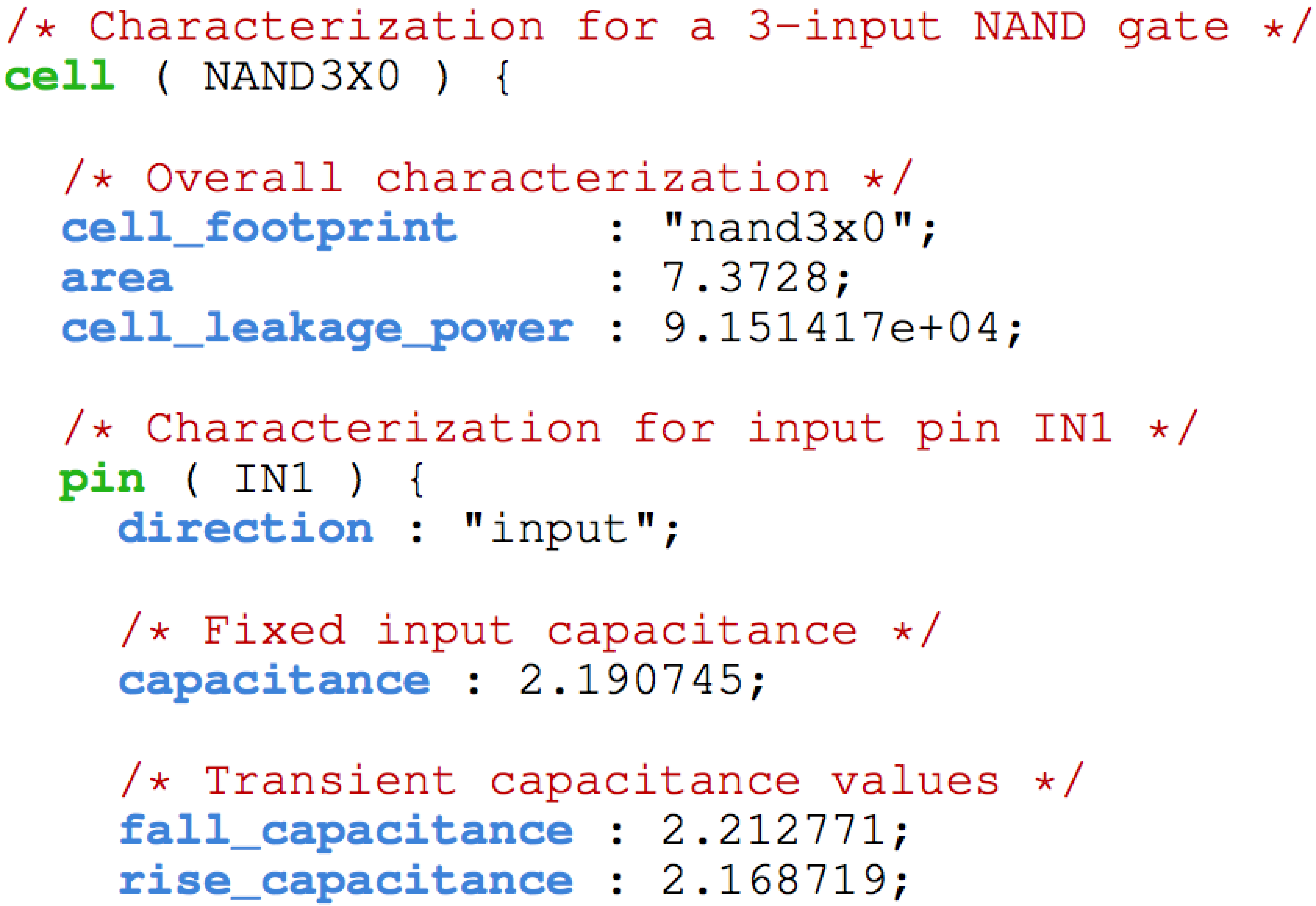
- C = load capacitance
- T = input rise/fall time



Path	1.2V - 125°C	1.6V - 40°C
$In1—t_{pLH}$	$0.073+7.98C+0.317T$	$0.020+2.73C+0.253T$
$In1—t_{pHL}$	$0.069+8.43C+0.364T$	$0.018+2.14C+0.292T$
$In2—t_{pLH}$	$0.101+7.97C+0.318T$	$0.026+2.38C+0.255T$
$In2—t_{pHL}$	$0.097+8.42C+0.325T$	$0.023+2.14C+0.269T$
$In3—t_{pLH}$	$0.120+8.00C+0.318T$	$0.031+2.37C+0.258T$
$In3—t_{pHL}$	$0.110+8.41C+0.280T$	$0.027+2.15C+0.223T$



Standard-Cell Electrical Characterization (.lib)



```
/* Characterization for a 3-input NAND gate */
cell ( NAND3X0 ) {

  /* Overall characterization */
  cell_footprint     : "nand3x0";
  area               : 7.3728;
  cell_leakage_power : 9.151417e+04;

  /* Characterization for input pin IN1 */
  pin ( IN1 ) {
    direction : "input";

    /* Fixed input capacitance */
    capacitance : 2.190745;

    /* Transient capacitance values */
    fall_capacitance : 2.212771;
    rise_capacitance : 2.168719;
```

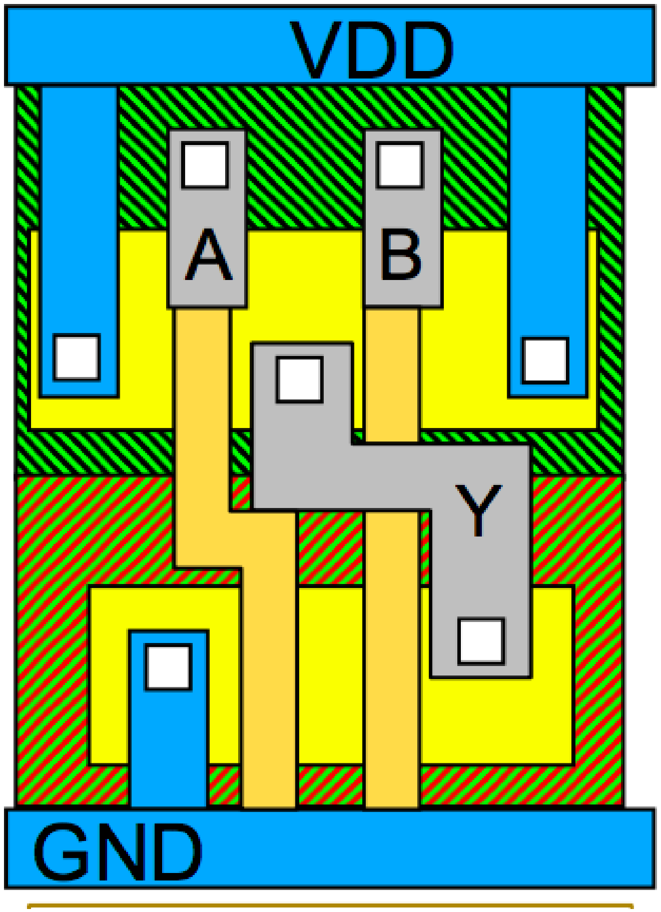
Standard-Cell Electrical Characterization (.lib)

```
cell ( NAND3X0 ) {
  pin ( IN1 ) {

     /* Short-circuit and internal switching
        power when IN2 and IN3 are zero */
     internal_power () {
       when : "!IN2&!IN3";

       /* 1D lookup tables to calculate power as
          function of input slew */
       rise_power ( "power_inputs_1" ) {
         index_1( " 0.0160000,  0.0320000,  0.0640000" );
         values(  "-1.2575404, -1.2594251, -1.2887053" );
       }
       fall_power ( "power_inputs_1" ) {
         index_1( " 0.0160000,  0.0320000,  0.0640000" );
         values(  " 1.9840914,  1.9791286,  2.0696119" );
       }
     }
```

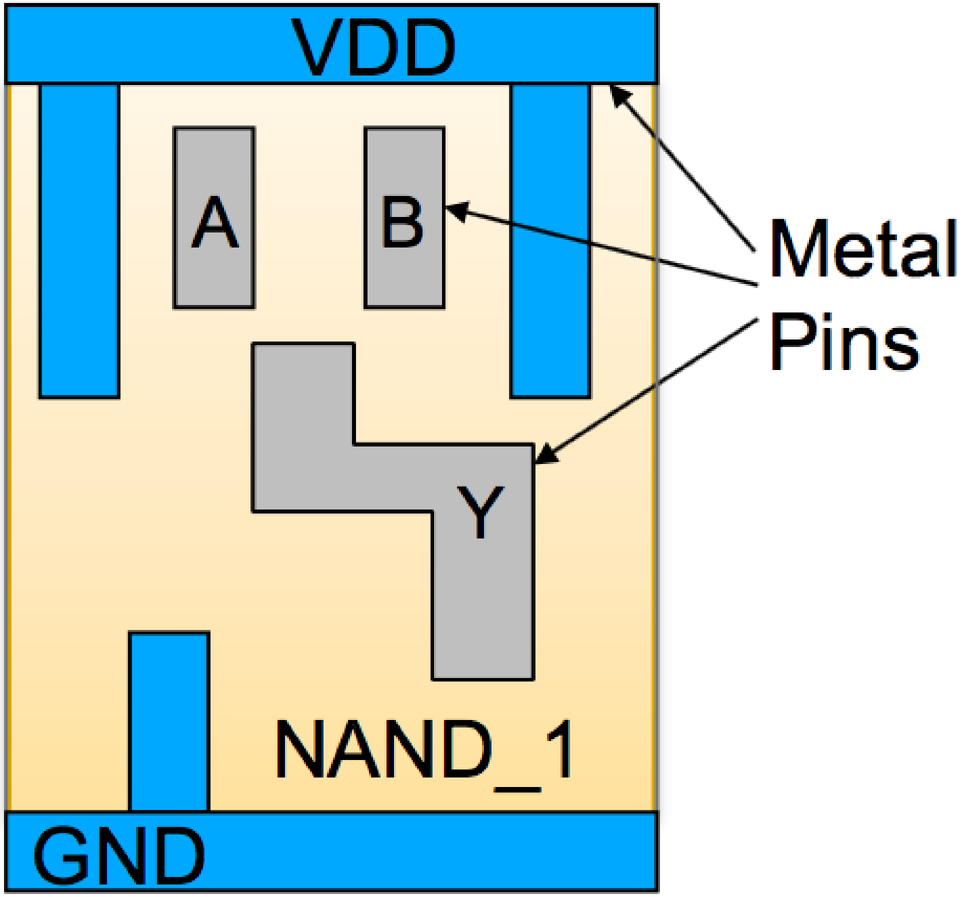
Standard-Cell Electrical Characterization (.lib)

```
cell ( NAND3X0 ) {
  pin ( QN ) {
    direction : "output";

    /* Boolean logic eq for QN as function of inputs */
    function : "(IN3*IN2*IN1)'";

    timing () {
      related_pin : "IN1";
      cell_rise ( "del_1_7_7" ) {

        index_1( "0.016, 0.032, 0.064" ); /* Input slew */
        index_2( "0.1,   3.75,  7.5"   ); /* Load cap */

        /* Lookup table to calculate delay as non-linear
           function of input signal slew and load cap */
        values( "0.0178632, 0.0275957, 0.0374970", \
                "0.0215562, 0.0316225, 0.0414275", \
                "0.0261721, 0.0387623, 0.0496870" )
      }
```

Standard-Cell Physical Characterization



Layout View



Metal Pins

Abstract Physical View

SAED 90nm Library: NAND Gate Data Sheet

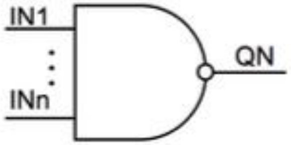
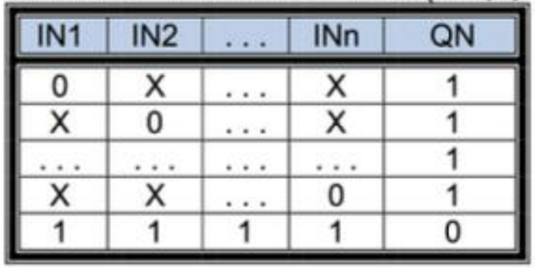
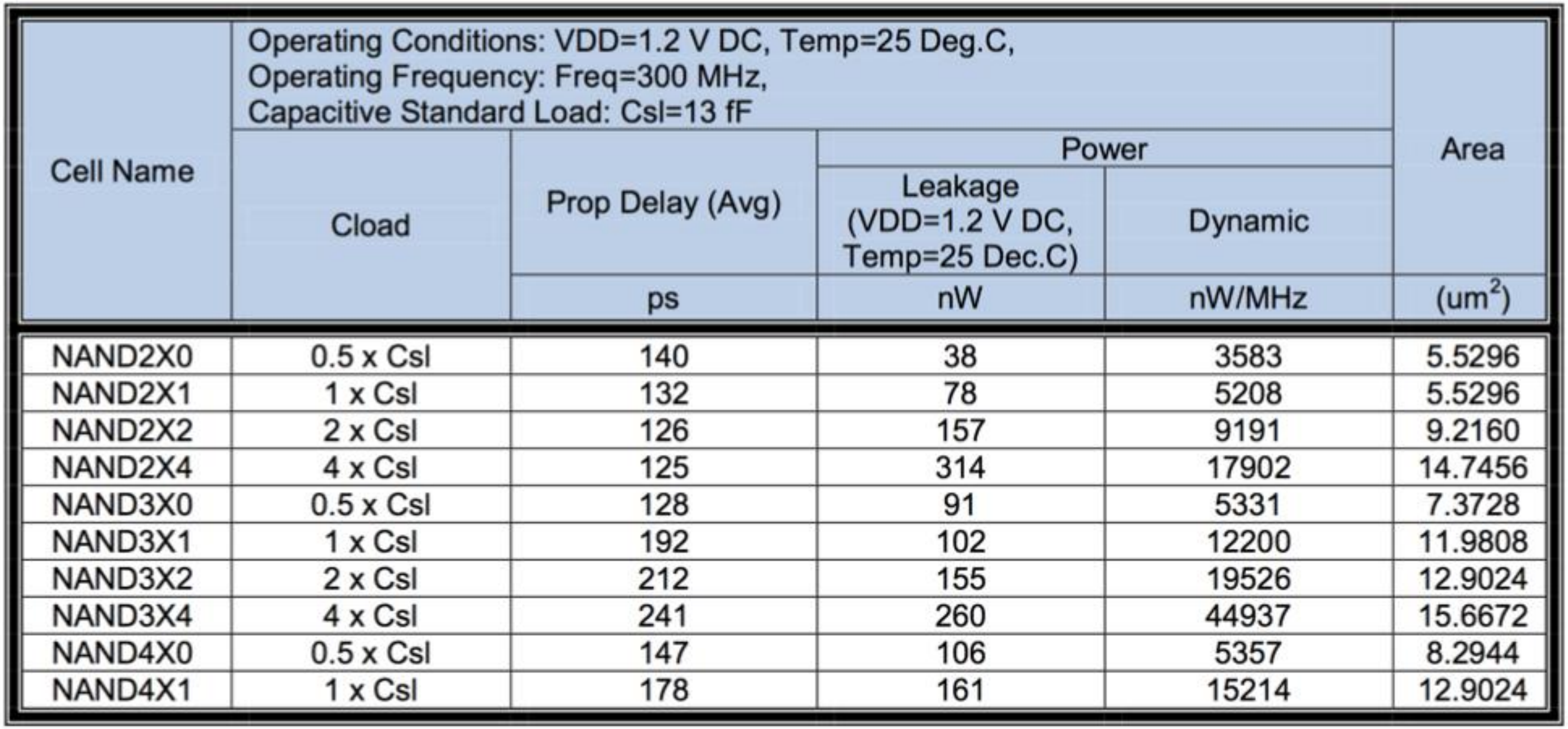


Figure 10.6. Logic Symbol of NAND

Table 10.11. NAND Truth Table (n=2,3,4)



IN1	IN2	. . .	INn	QN
0	X	. . .	X	1
X	0	. . .	X	1
. . .	. . .	. . .	. . .	1
X	X	. . .	0	1
1	1	1	1	0



Cell Name	Operating Conditions: VDD=1.2 V DC, Temp=25 Deg.C, Operating Frequency: Freq=300 MHz, Capacitive Standard Load: Csl=13 fF				Area
	Cload	Prop Delay (Avg)	Power		
			Leakage (VDD=1.2 V DC, Temp=25 Dec.C)	Dynamic	
		ps	nW	nW/MHz	(um^2)
NAND2X0	0.5 x Csl	140	38	3583	5.5296
NAND2X1	1 x Csl	132	78	5208	5.5296
NAND2X2	2 x Csl	126	157	9191	9.2160
NAND2X4	4 x Csl	125	314	17902	14.7456
NAND3X0	0.5 x Csl	128	91	5331	7.3728
NAND3X1	1 x Csl	192	102	12200	11.9808
NAND3X2	2 x Csl	212	155	19526	12.9024
NAND3X4	4 x Csl	241	260	44937	15.6672
NAND4X0	0.5 x Csl	147	106	5357	8.2944
NAND4X1	1 x Csl	178	161	15214	12.9024

Outline

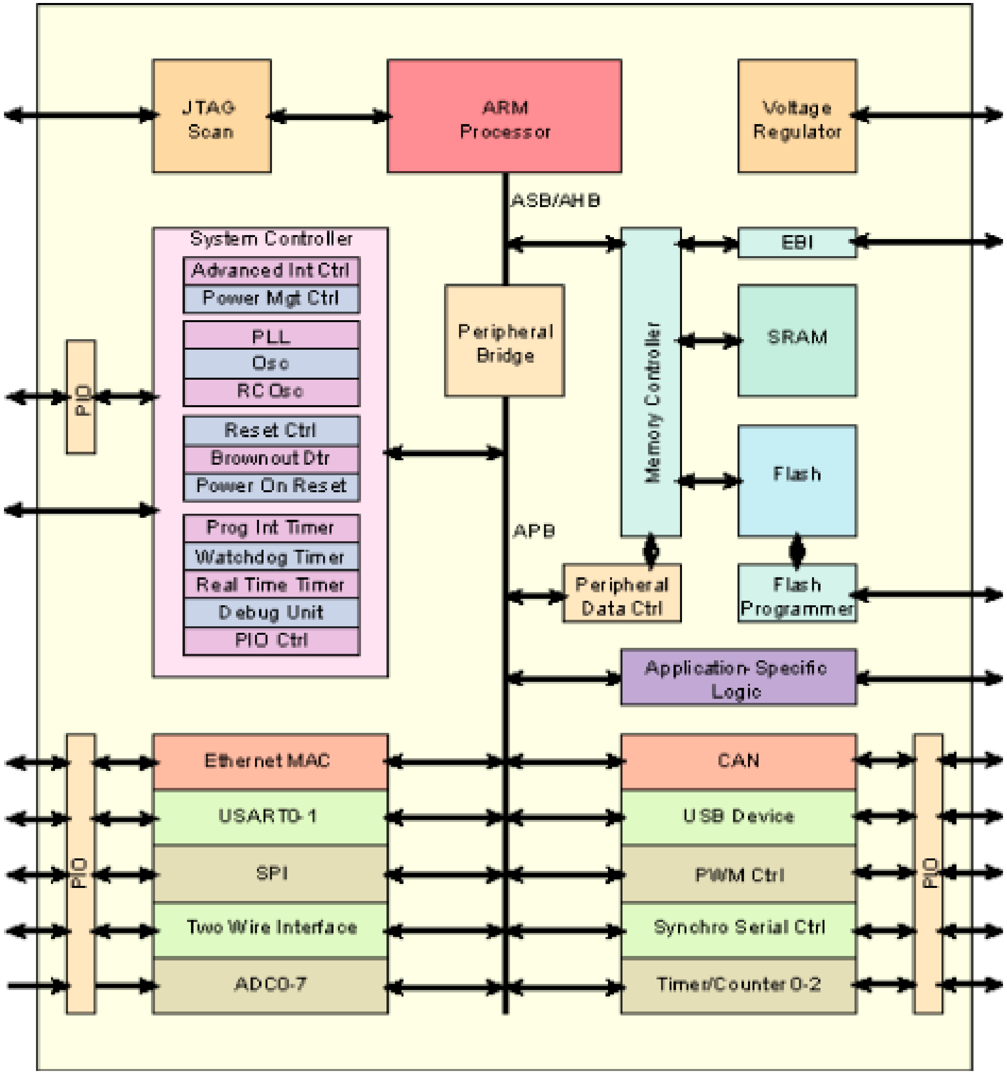
Standard-Cell-Based Design

SoC-Platform-Based Design

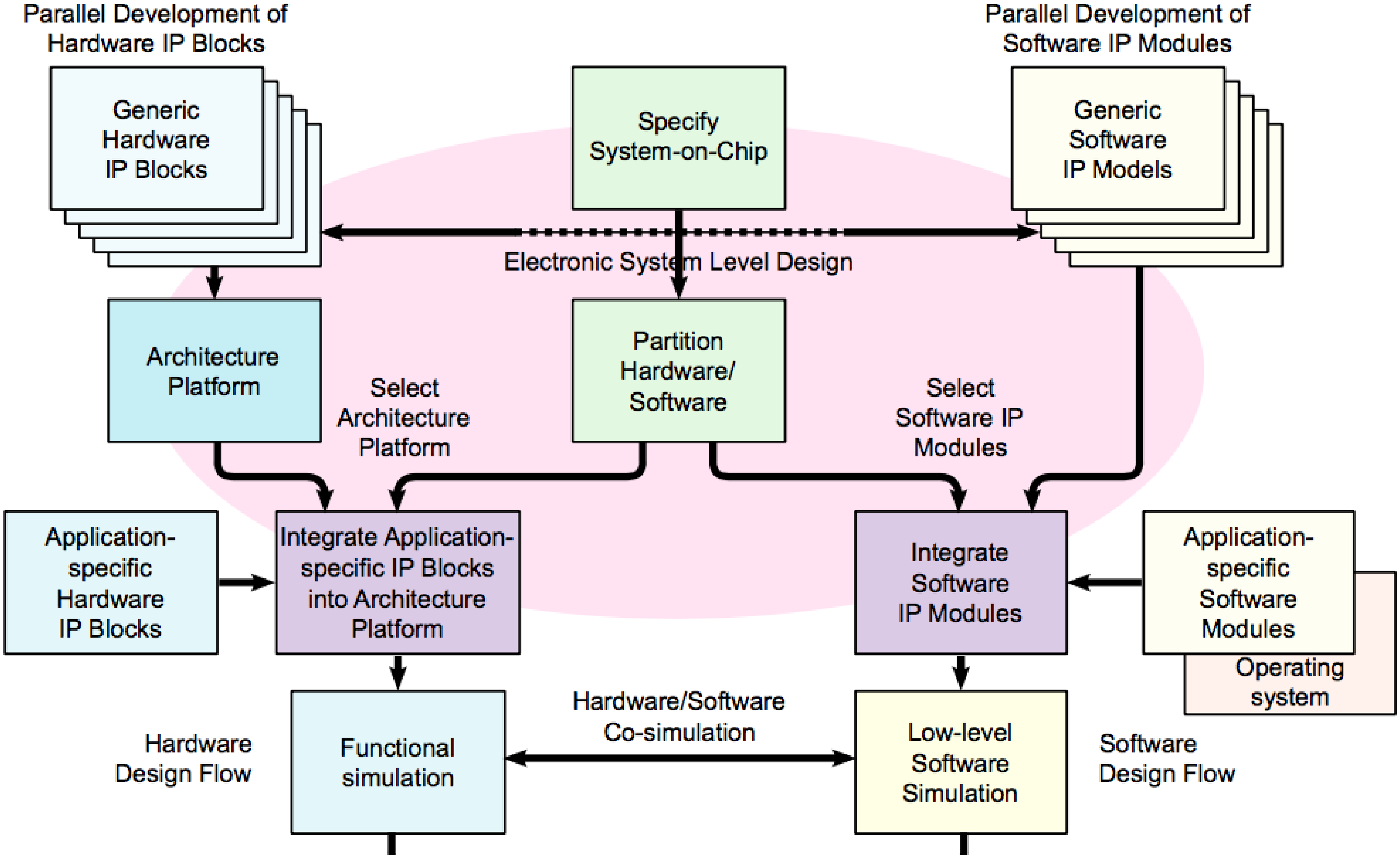
Methodology Comparison

System-on-Chip (SoC) Platform-Based Design

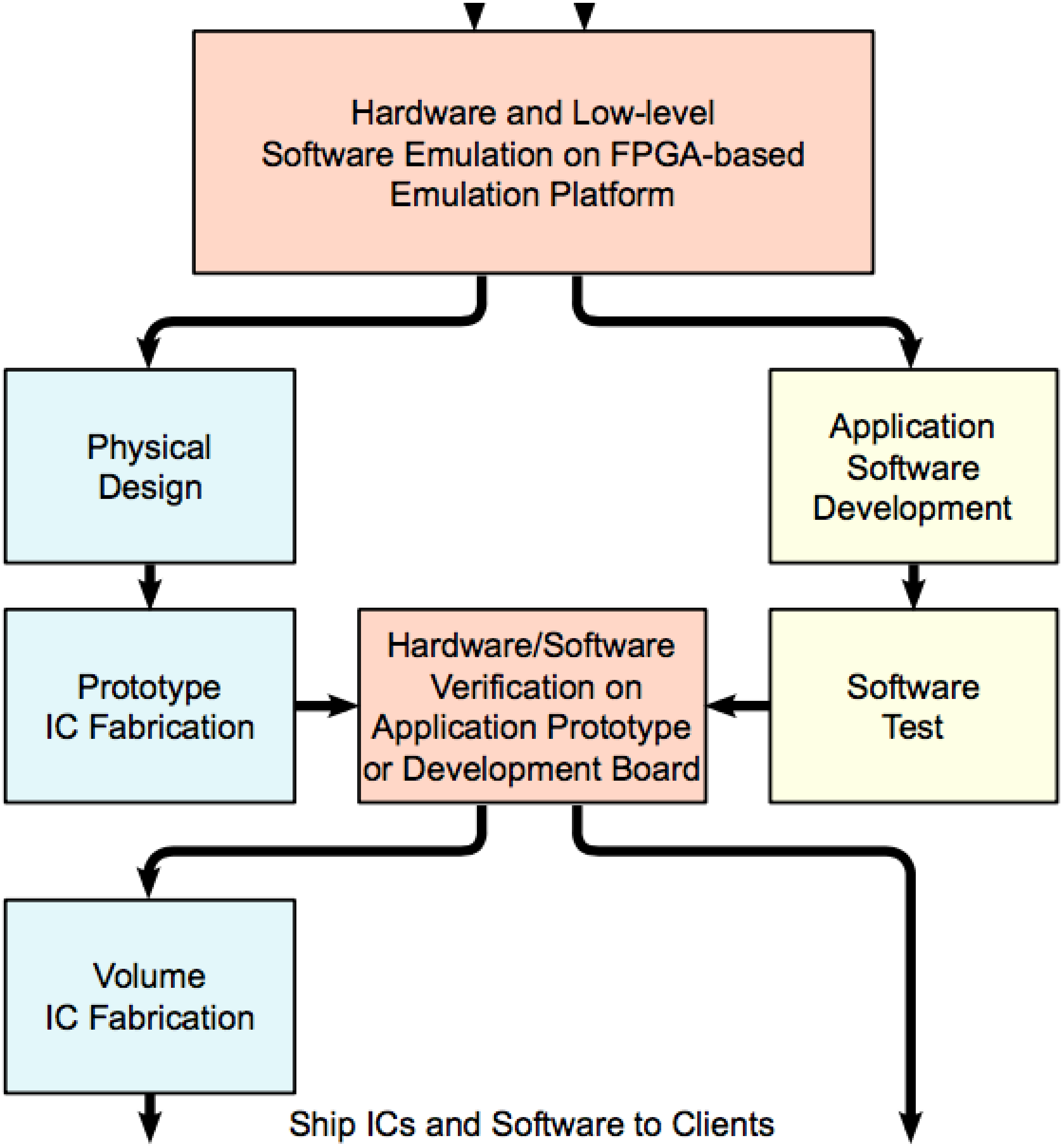
- Integration
  - standard cell block
  - custom analog
  - processor
  - memory
- Standardized Bus
  - AMBA, Sonics, …
- IP Business Model
  - hard or soft IP from 3rd party provider
  - e.g. ARM



SoC Hardware/Software Co-Design

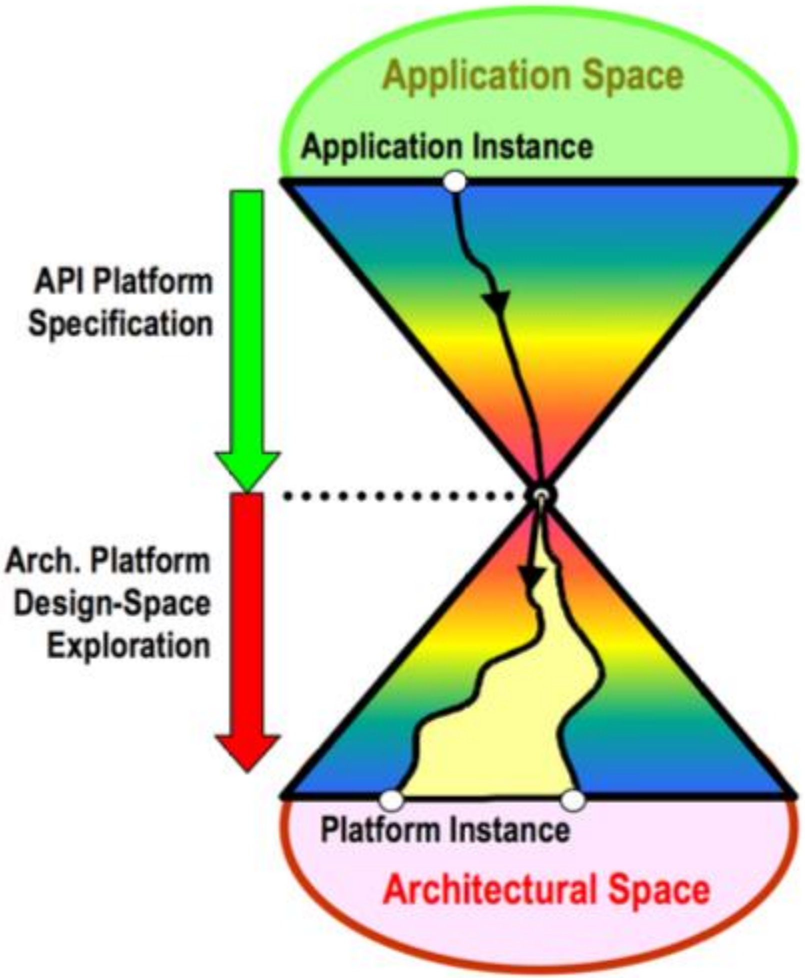

Parallel Development of Hardware IP Blocks
Generic Hardware IP Blocks
Specify System-on-Chip
Parallel Development of Software IP Modules
Generic Software IP Models
Electronic System Level Design
Architecture Platform
Partition Hardware/ Software
Select Architecture Platform
Select Software IP Modules
Application-specific Hardware IP Blocks
Integrate Application-specific IP Blocks into Architecture Platform
Integrate Software IP Modules
Application-specific Software Modules
Operating system
Hardware Design Flow
Functional simulation
Hardware/Software Co-simulation
Low-level Software Simulation
Software Design Flow

SoC Hardware/Software Co-Design


Hardware and Low-level Software Emulation on FPGA-based Emulation Platform
Physical Design
Application Software Development
Prototype IC Fabrication
Hardware/Software Verification on Application Prototype or Development Board
Software Test
Volume IC Fabrication
Ship ICs and Software to Clients

SoC Platform-Based Design

- Platform allows restriction on design space
  - limit implementation choice
  - provide well-defined abstraction of the underlying technology for app developer
- New platform
  - at architecture and micro-architecture boundary
- Representation of communication
  - key to the platform design approach



Outline

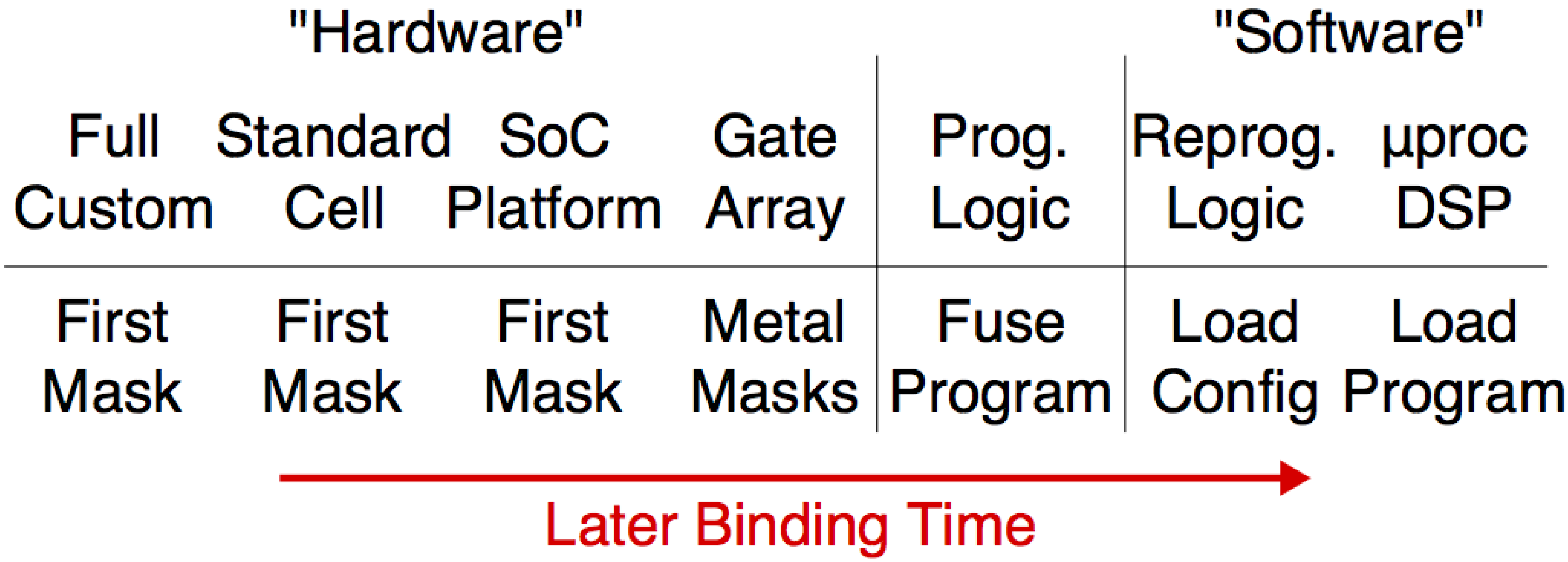
Standard-Cell-Based Design

SoC-Platform-Based Design

Methodology Comparison

Operation Binding Time

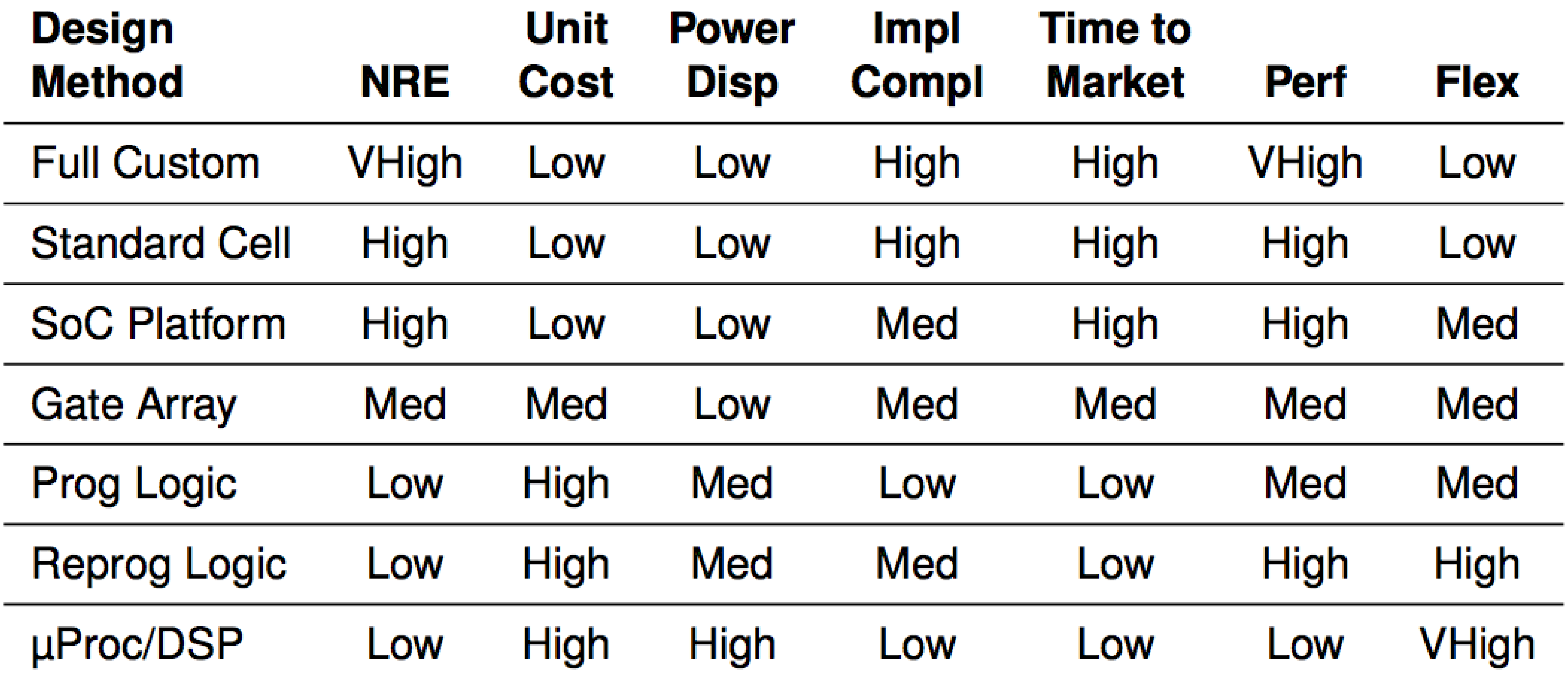
- Earlier the operation is bound, the less area, delay, and energy required for implementation
- Later the operation is bound, the more flexible the device



"Hardware"					"Software"	
Full Custom	Standard Cell	SoC Platform	Gate Array	Prog. Logic	Reprog. Logic	µproc DSP
First Mask	First Mask	First Mask	Metal Masks	Fuse Program	Load Config	Load Program

Later Binding Time

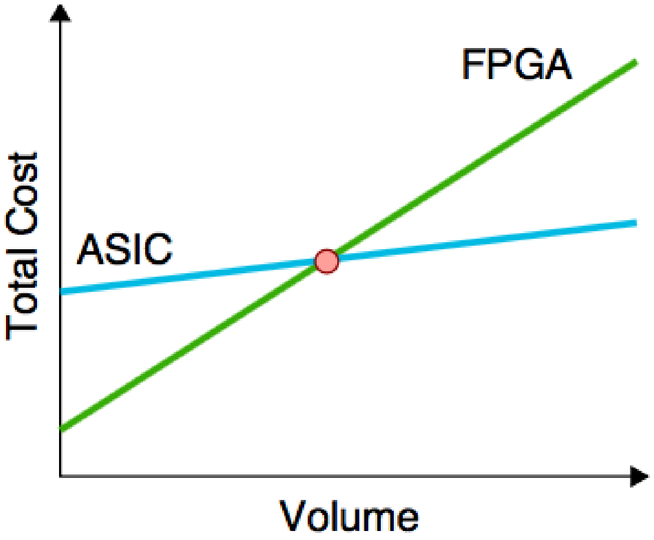
Comparison of Specific Design Methodologies

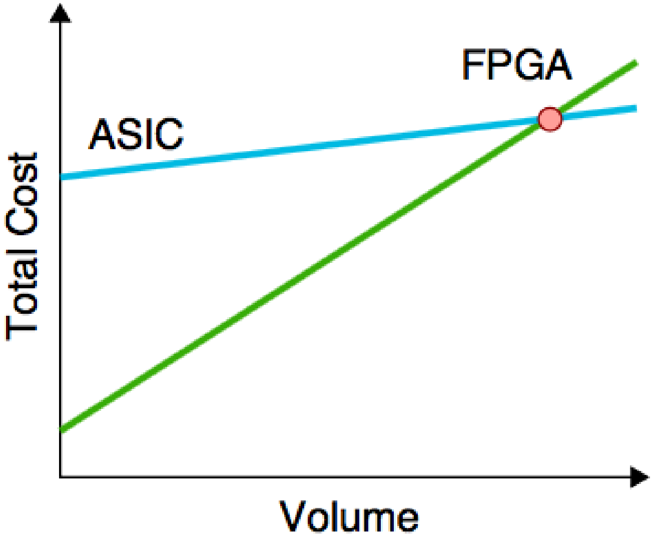


Design Method	NRE	Unit Cost	Power Disp	Impl Compl	Time to Market	Perf	Flex
Full Custom	VHigh	Low	Low	High	High	VHigh	Low
Standard Cell	High	Low	Low	High	High	High	Low
SoC Platform	High	Low	Low	Med	High	High	Med
Gate Array	Med	Med	Low	Med	Med	Med	Med
Prog Logic	Low	High	Med	Low	Low	Med	Med
Reprog Logic	Low	High	Med	Med	Low	High	High
µProc/DSP	Low	High	High	Low	Low	Low	VHigh

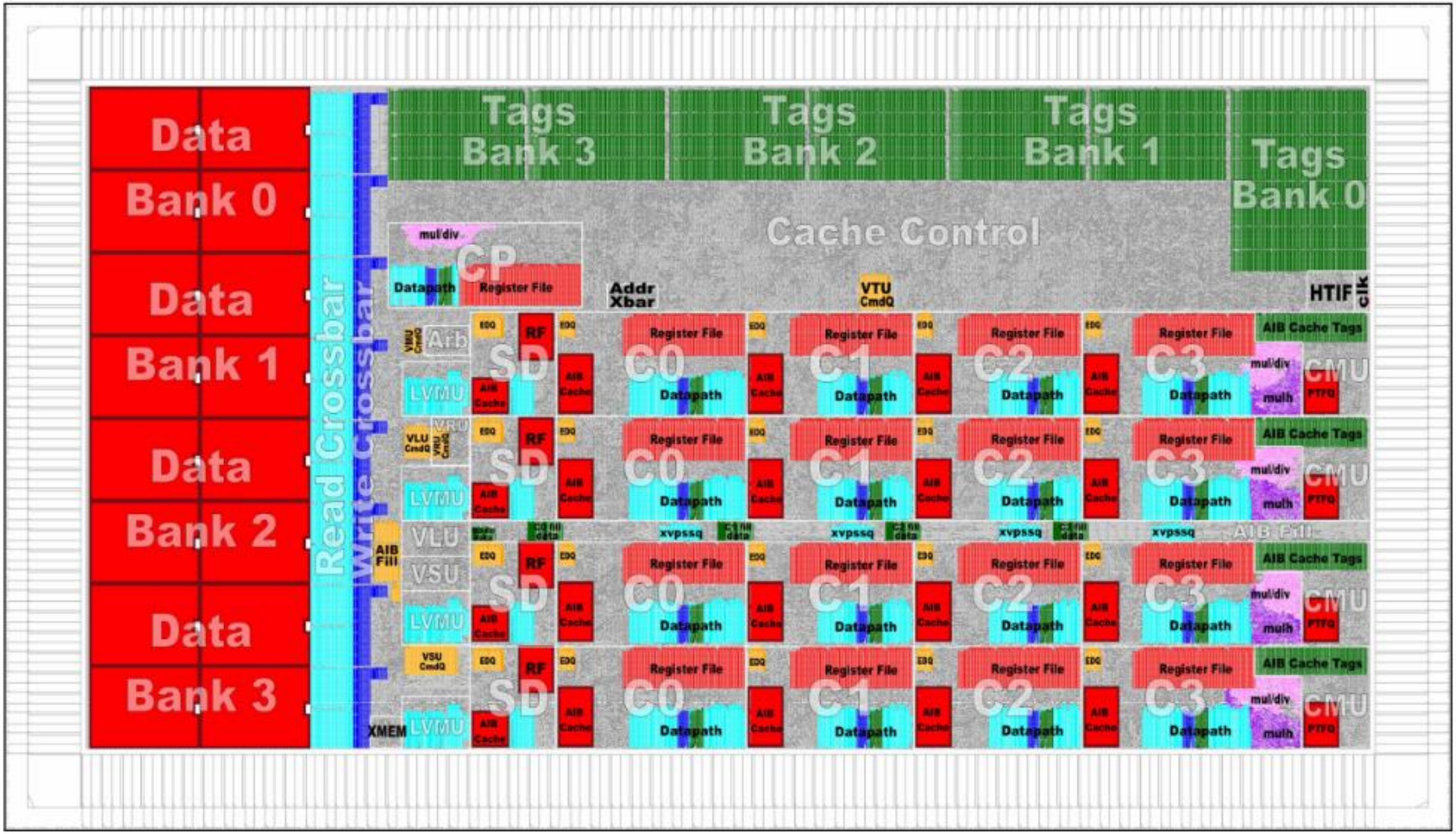
ASIC vs. FPGA

- Traditional Argument
  - ASIC: high NRE ($2M for 0.35um), low marginal cost, best efficiency
  - FPGA: low NRE, high marginal cost, lower efficiency
  - crossover point: ~10,000
- Current Trends
  - ASIC: increasing NRE ($40M for 90nm) due to design, verification, and mask costs, etc.
  - FPGA: better able to track Moore’s law integrating fixed function blocks
  - crossover point: ~100,000

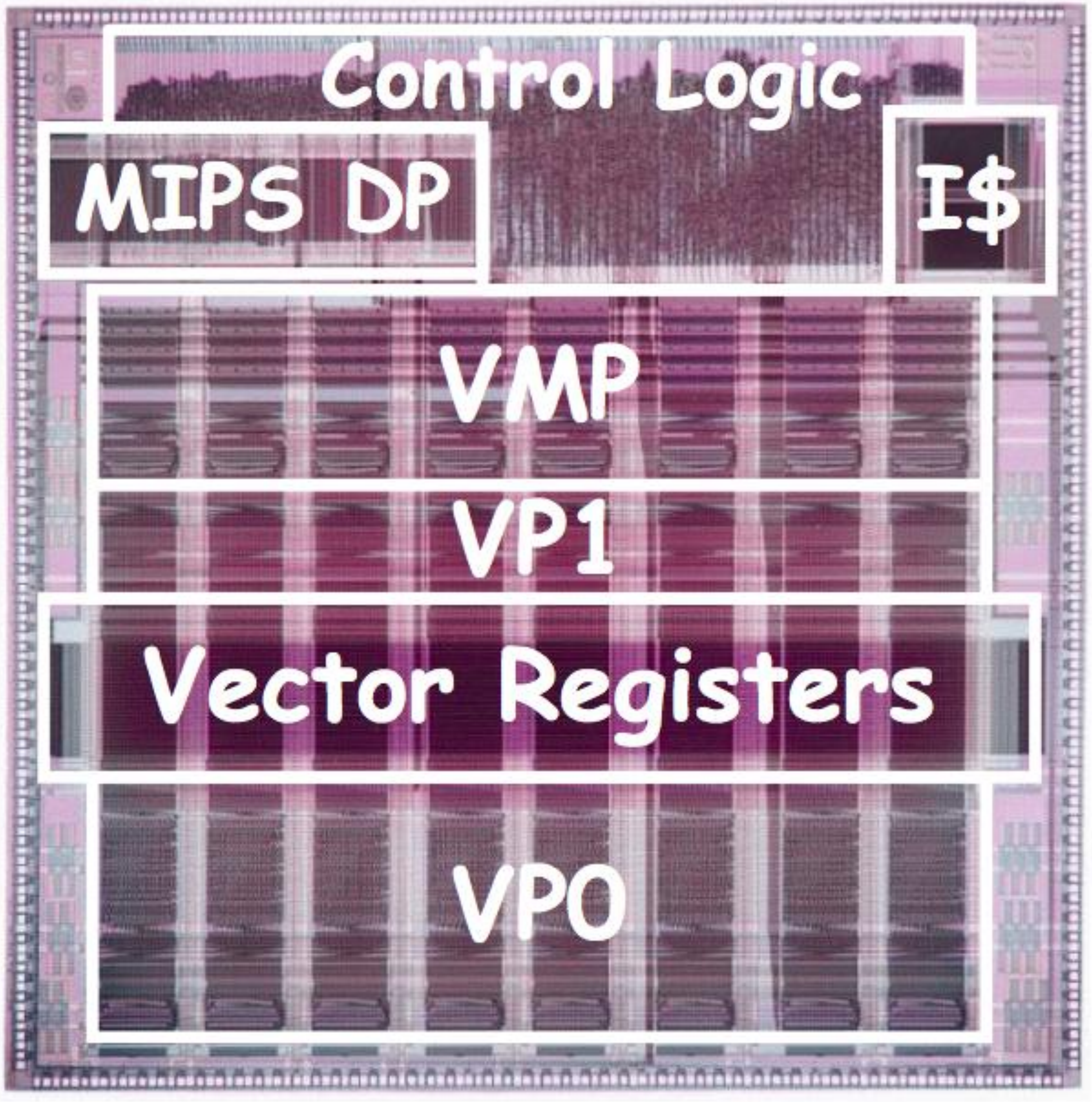




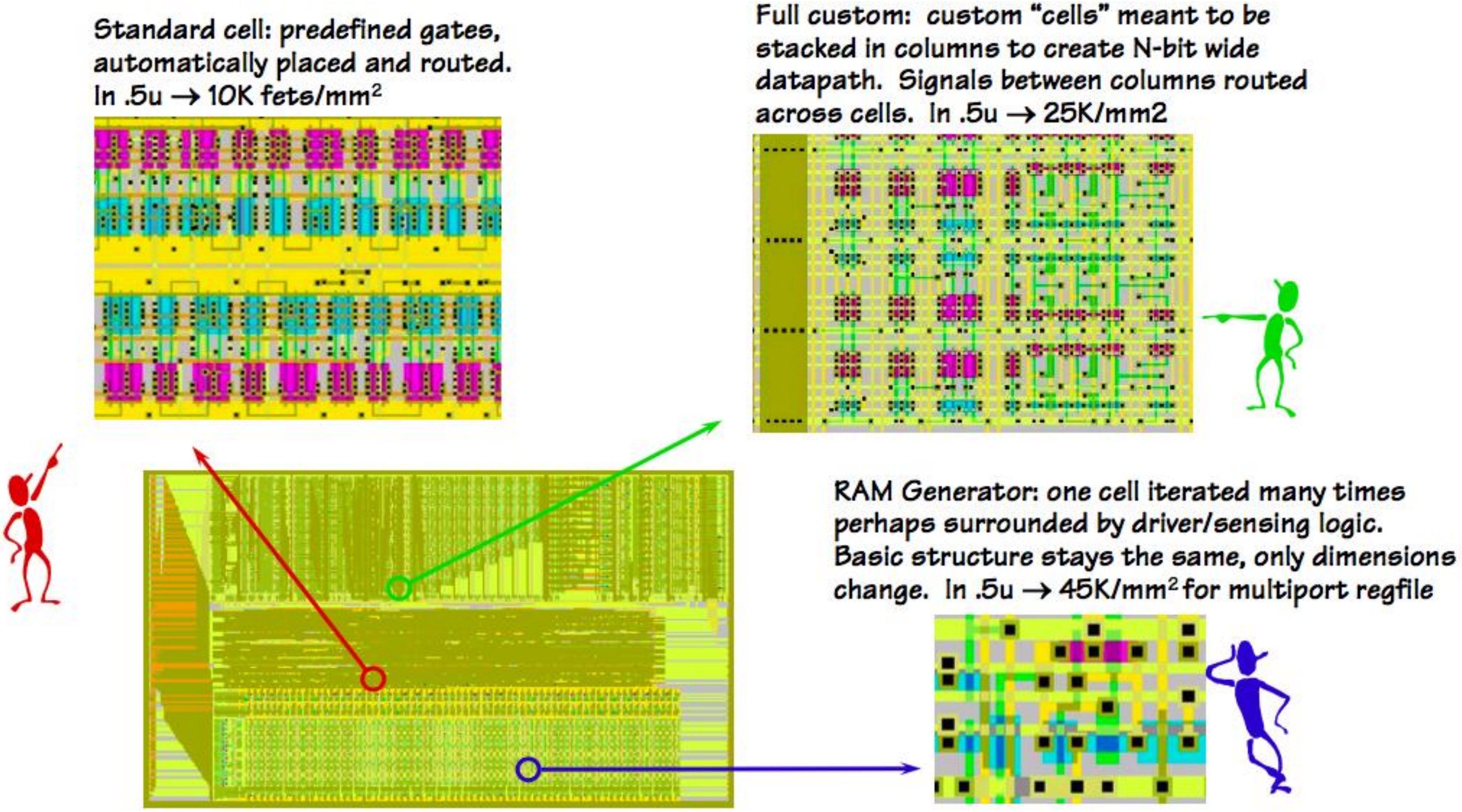
Scale: ASIC with Pre-Placement & SRAMs



T0: Full Custom with Standard Cells



ASIC and Full Custom with Standard Cell

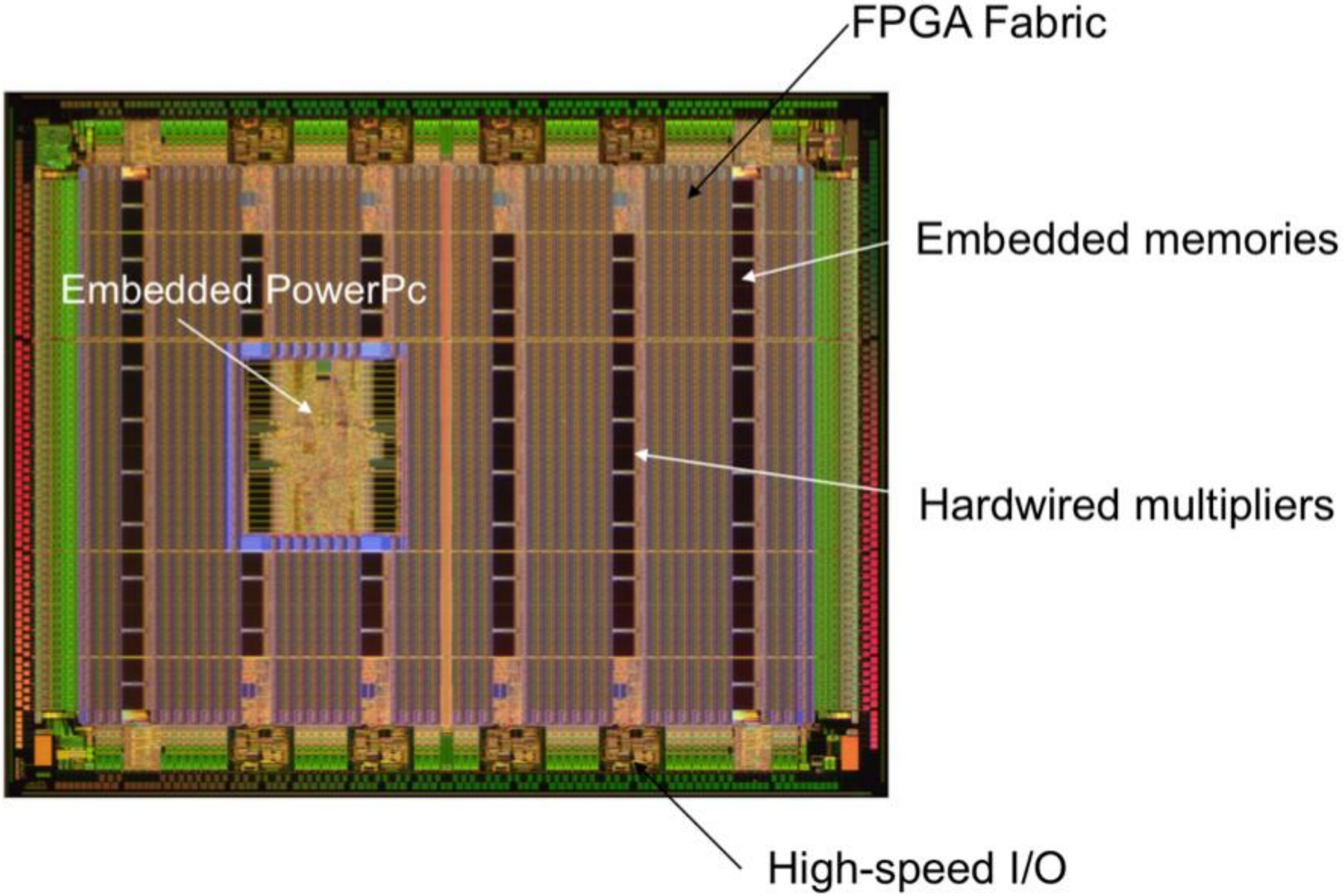


Standard cell: predefined gates, automatically placed and routed. In .5u → 10K fets/mm^2

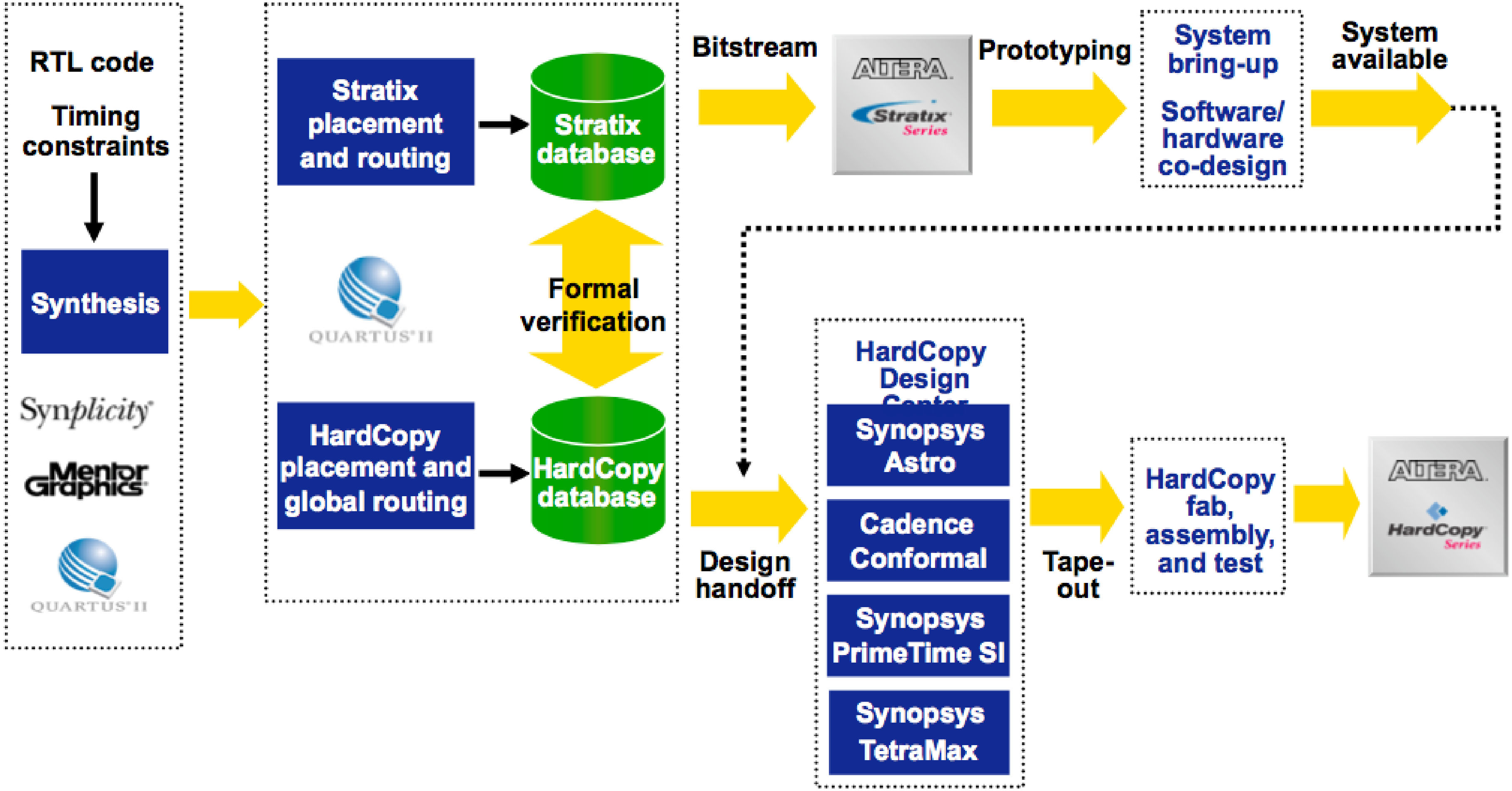
Full custom: custom "cells" meant to be stacked in columns to create N-bit wide datapath. Signals between columns routed across cells. In .5u → 25K/mm2

RAM Generator: one cell iterated many times perhaps surrounded by driver/sensing logic. Basic structure stays the same, only dimensions change. In .5u → 45K/mm^2 for multiport regfile

Xilinx Vertex-II Pro: FPGA with Hard Processor



Altera HardCopy: FPGA to Gate-Array-Like Tapeout



RTL code

Timing constraints

Synthesis

Synplicity

Mentor Graphics

QUARTUS II

Stratix placement and routing

Stratix database

Formal verification

HardCopy placement and global routing

HardCopy database

QUARTUS II

Bitstream

ALTERA Stratix Series

Prototyping

System bring-up

Software/ hardware co-design

System available

Design handoff

HardCopy Design Center

Synopsys Astro

Cadence Conformal

Synopsys PrimeTime SI

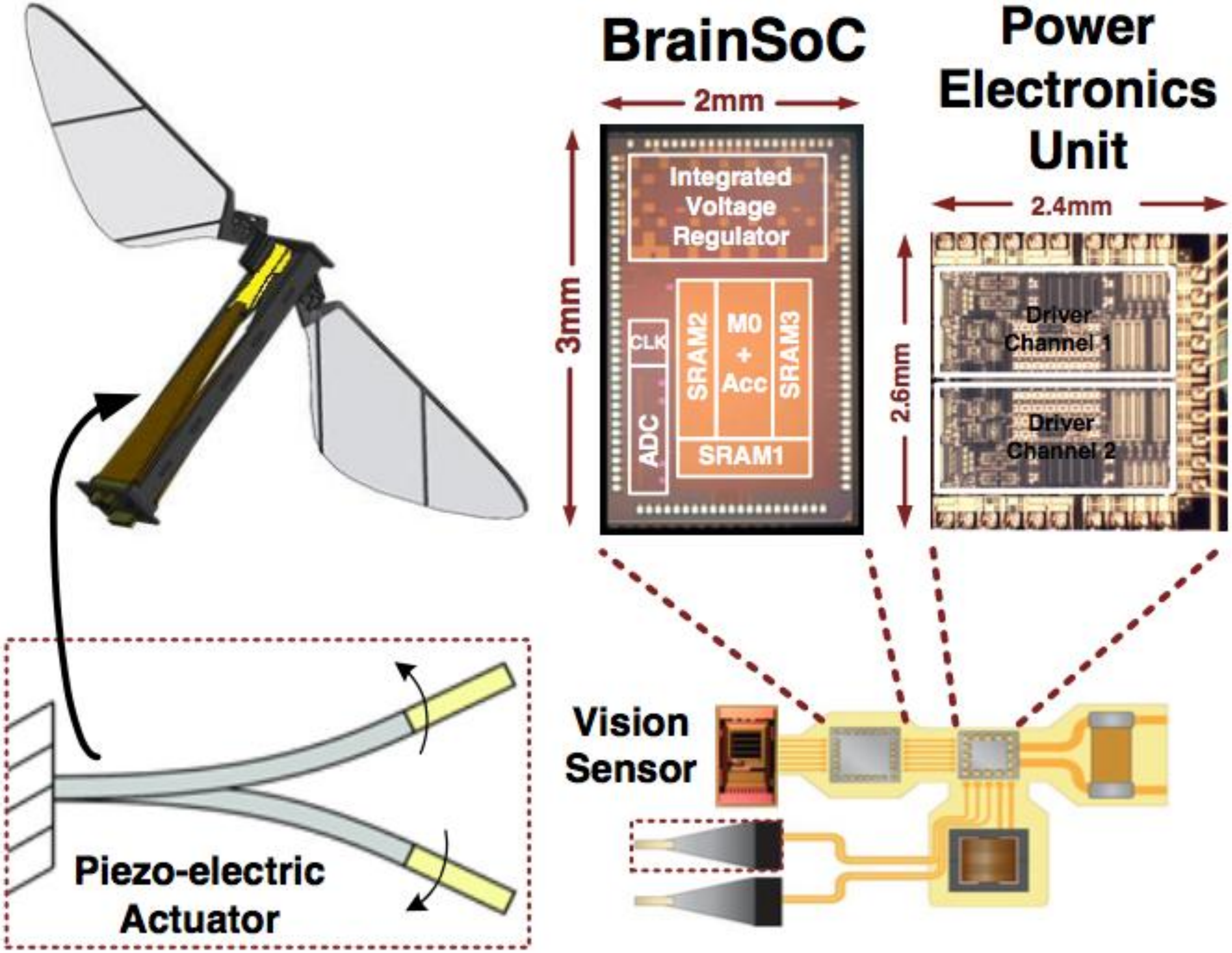
Synopsys TetraMax

Tape-out

HardCopy fab, assembly, and test

ALTERA HardCopy Series

Multi-Chip System: BrainSoC and PEU in RoboBee



Design Principle in Automated Methodologies

- Modularity
  - to enable mixing different custom and automated methods
- Hierarchy
  - to efficiently handle automatically transforming large design
- Encapsulation
  - significant higher emphasis on encapsulation in all domains (behavioral, structural, physical) at all level of abstraction (architecture, RTL, and gate-level)
- Regularity
  - to create automated tools to generate structures like datapaths and memories
- Extensibility
  - automated methodologies enable more highly parameterized and flexible implementation improving extensibility

Questions?

Comments?

Discussion?

Acknowledgement

Cornell University, ECE 5745