INTRODUCTION TO ASICs

1

Key concepts: The difference between full-custom and semicustom ASICs • The difference between standard-cell, gate-array, and programmable ASICs • ASIC design flow • Design economics • ASIC cell library

An **ASIC** (“a-sick”) is an **application-specific integrated circuit**

A **gate equivalent** is a NAND gate $F = \overline{A \bullet B}$ (IBM uses a NOR gate), or four transistors

History of integration: **small-scale integration** (**SSI**, ~10 gates per chip, 60’s), **medium-scale integration** (**MSI**, ~100–1000 gates per chip, 70’s), **large-scale integration** (**LSI**, ~1000–10,000 gates per chip, 80’s), **very large-scale integration** (**VLSI**, ~10,000–100,000 gates per chip, 90’s), **ultralarge scale integration** (**ULSI**, ~1M–10M gates per chip)

History of technology: **bipolar technology** and **transistor–transistor logic** (**TTL**) preceded **metal-oxide-silicon** (**MOS**) technology because it was difficult to make metal-gate n-channel MOS (**nMOS** or **NMOS**); the introduction of **complementary MOS** (**CMOS**, never cMOS) greatly reduced power

The **feature size** is the smallest shape you can make on a chip and is measured in λ or **lambda**

Origin of ASICs: the **standard parts**, initially used to design **microelectronic systems**, were gradually replaced with a combination of **glue logic**, **custom ICs**, **dynamic random-access memory** (**DRAM**) and **static RAM** (**SRAM**)

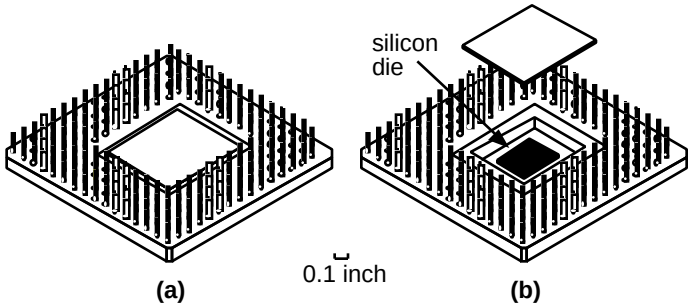
History of ASICs: The *IEEE Custom Integrated Circuits Conference* (CICC) and *IEEE International ASIC Conference* document the development of ASICs

Application-specific standard products (**ASSPs**) are a cross between standard parts and ASICs

1.1 Types of ASICs

ICs are made on a **wafer**. Circuits are built up with successive **mask layers**. The number of **masks** used to define the **interconnect** and other layers is different between **full-custom ICs** and **programmable ASICs**

A silicon chip or **integrated cicuit** (IC) is more properly called a die



1.1.1 Full-Custom ASICs

All mask layers are customized in a **full-custom ASIC**.

It only makes sense to design a full-custom IC if there are no libraries available.

Full-custom offers the highest performance and lowest part cost (smallest die size) with the disadvantages of increased design time, complexity, design expense, and highest risk.

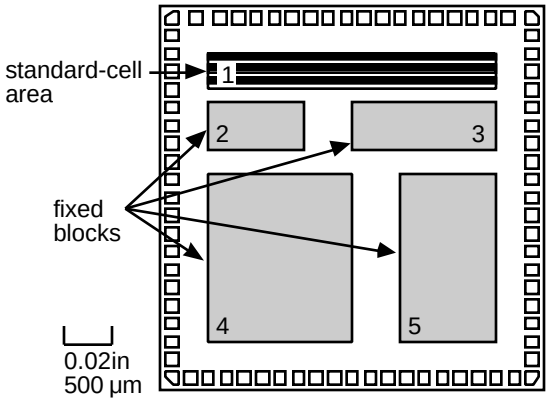
Microprocessors were exclusively full-custom, but designers are increasingly turning to semicustom ASIC techniques in this area too.

Other examples of full-custom ICs or ASICs are requirements for high-voltage (automobile), analog/digital (communications), or sensors and actuators.

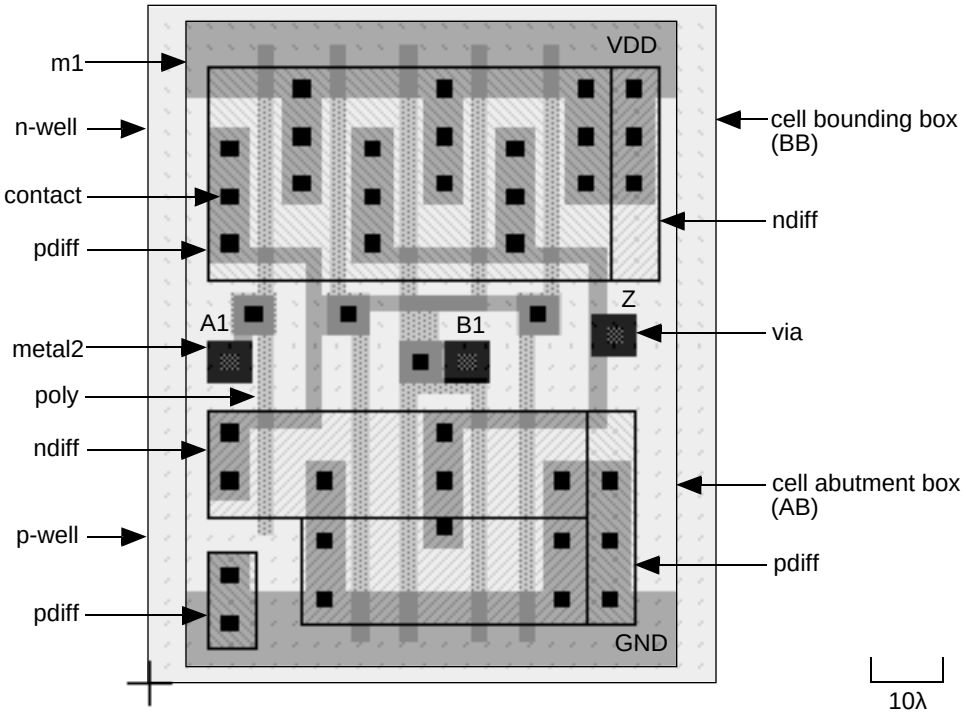
1.1.2 Standard-Cell–Based ASICs

A **cell-based ASIC** (**CBIC**—"sea-bick")

- Standard cells
- Possibly **megacells**, **megafunctions**, **full-custom blocks**, **system-level macros** (**SLMs**), **fixed blocks**, **cores**, or **Functional Standard Blocks** (**FSBs**)
- All mask layers are customized—transistors and interconnect
- Custom blocks can be embedded
- Manufacturing lead time is about eight weeks.



In **datapath** (**DP**) logic we may use a **datapath compiler** and a **datapath library**. Cells such as **arithmetic and logical units** (**ALUs**) are **pitch-matched**to each other to improve timing and density.

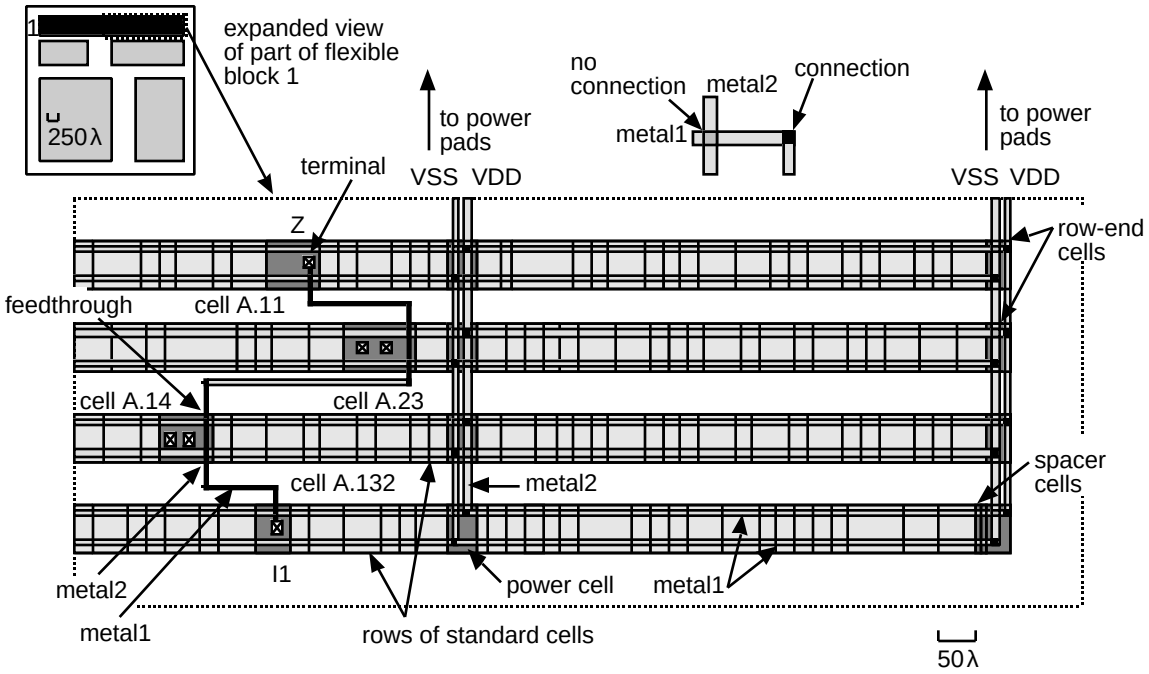


Looking down on the layout of a **standard cell** from a **standard-cell library**

1.1.3 Gate-Array–Based ASICs

A **gate array**, **masked gate array**, **MGA**, or **prediffused array** uses **macros** (**books**) to reduce **turnaround time** and comprises a **base array** made from a **base cell** or **primitive cell**. There are three types:

- Channeled gate arrays
- Channelless gate arrays
- Structured gate arrays



Routing a CBIC (cell-based IC)

• A “wall” of standard cells forms a **flexible block**

• **metal2** may be used in a **feedthrough cell** to cross over cell rows that use **metal1** for wiring

• Other wiring cells: **spacer cells**, **row-end cells**, and **power cells**

A note on the use of hyphens and dashes in the spelling (orthography) of compound nouns: Be careful to distinguish between a “high-school girl” (a girl of high-school age) and a “high school girl” (is she on drugs or perhaps very tall?).

We write “channeled gate array,” but “channeled gate-array architecture” because the *gate array* is *channeled;* it is not “channeled-gate array architecture” (which is an array of channeled-gates) or “channeled gate array architecture” (which is ambiguous).

We write gate-array–based ASICs (with a en-dash between array and based) to mean (gate array)-based ASICs.

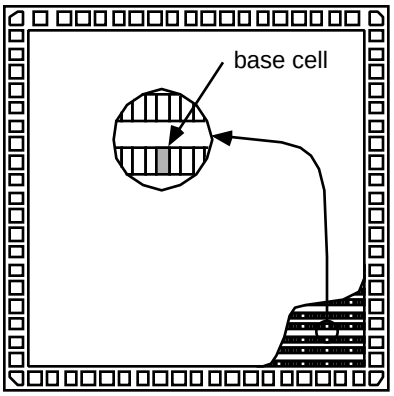
1.1.4 Channeled Gate Array

A **channeled gate array**

• Only the interconnect is customized

• The interconnect uses predefined spaces between rows of base cells

• Manufacturing lead time is between two days and two weeks

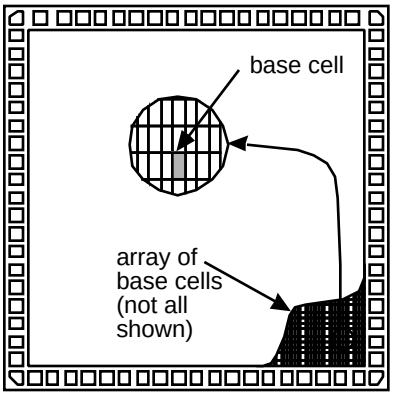


1.1.5 Channelless Gate Array

A **channelless gate array** (**channel-free gate array**, **sea-of-gates array**, or **SOG** array)

• Only some (the top few) mask layers are customized—the interconnect

• Manufacturing lead time is between two days and two weeks.



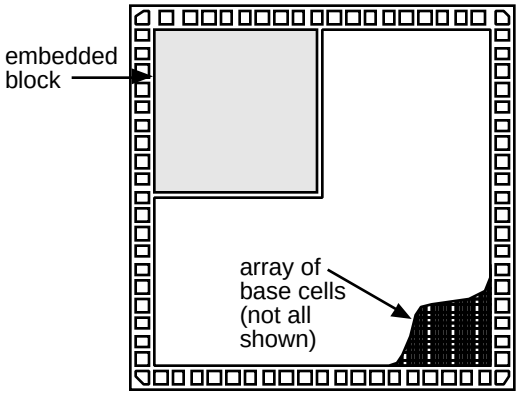
1.1.6 Structured Gate Array

An **embedded gate array** or **structured gate array** (**masterslice** or **masterimage**)

• Only the interconnect is customized

• Custom blocks (the same for each design) can be embedded

• Manufacturing lead time is between two days and two weeks.



1.1.7 Programmable Logic Devices

Examples and types of PLDs: **read-only memory** (**ROM**) • **programmable ROM** or **PROM** • **electrically programmable ROM**, or **EPROM** • An **erasable PLD** (EPLD) • **electrically erasable PROM**, or **EEPROM** • **UV-erasable PROM**, or **UVPROM** • **mask-programmable ROM** • A **mask-programmed PLD** usually uses bipolar technology

Logic arrays may be either a **Programmable Array Logic** (PAL®, a registered trademark of AMD) or a **programmable logic array** (PLA); both have an **AND plane** and an **OR plane**

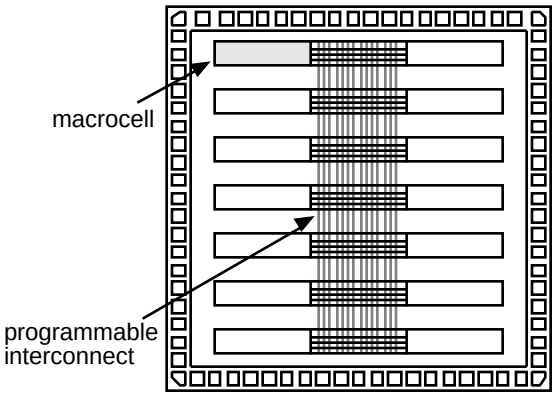
A **programmable logic device** (**PLD**)

• No customized mask layers or logic cells

• Fast design turnaround

• A single large block of programmable interconnect

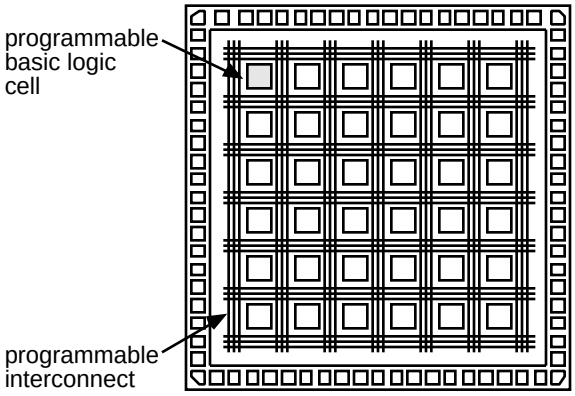
• A matrix of logic macrocells that usually consist of programmable array logic followed by a flip-flop or latch



1.1.8 Field-Programmable Gate Arrays

A **field-programmable gate array** (**FPGA**) or **complex PLD**

- None of the mask layers are customized
- A method for programming the basic logic cells and the interconnect
- The core is a regular array of programmable basic logic cells that can implement combinational as well as sequential logic (flip-flops)
- A matrix of programmable interconnect surrounds the basic logic cells
- Programmable I/O cells surround the core
- Design turnaround is a few hours



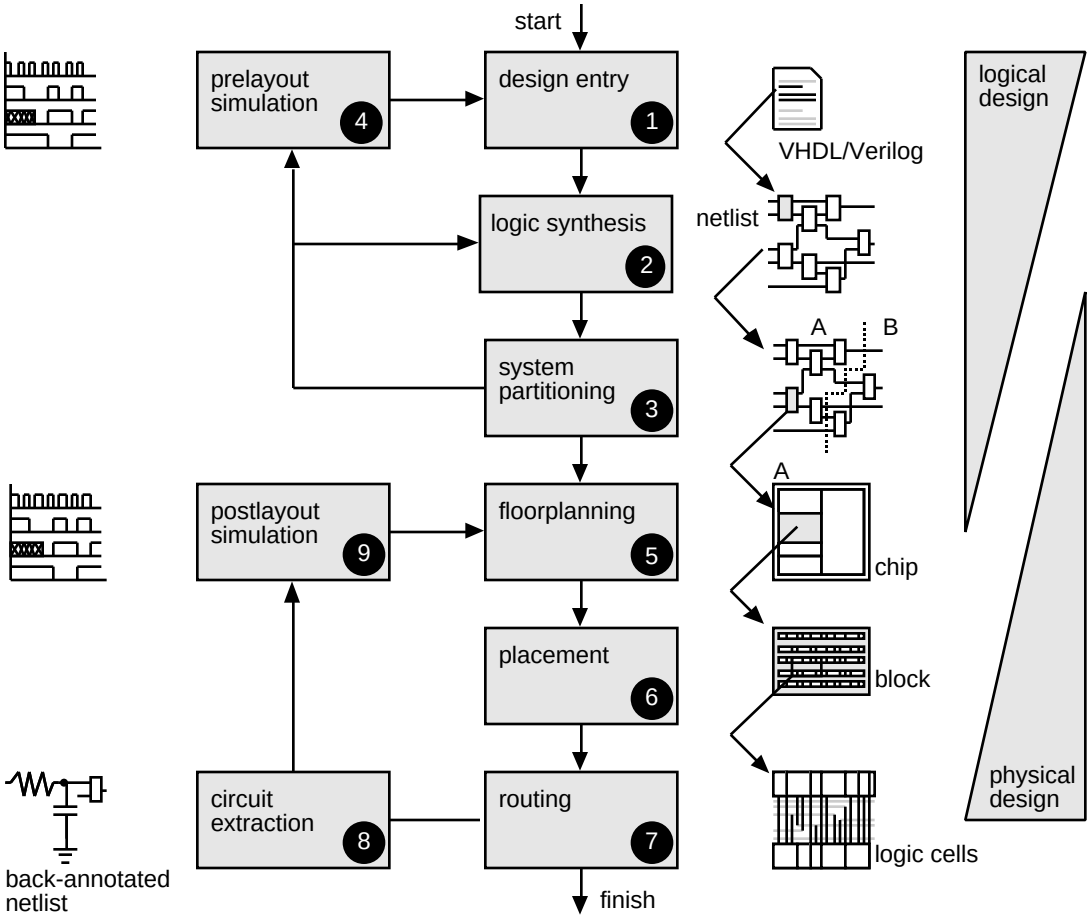
1.2 Design Flow

A **design flow** is a sequence of steps to design an ASIC

1. **Design entry**. Using a **hardware description language** (**HDL**) or schematic entry.
2. **Logic synthesis**. Produces a **netlist**—logic cells and their connections.
3. **System partitioning**. Divide a large system into ASIC-sized pieces.
4. **Prelayout simulation**. Check to see if the design functions correctly.
5. **Floorplanning**. Arrange the blocks of the netlist on the chip.
6. **Placement**. Decide the locations of cells in a block.
7. **Routing**. Make the connections between cells and blocks.
8. **Extraction**. Determine the resistance and capacitance of the interconnect.
9. **Postlayout simulation**. Check to see the design still works with the added loads of the interconnect.

1.3 Case Study

SPARCstation 1: Better performance at lower cost • Compact size, reduced power, and quiet operation • Reduced number of parts, easier assembly, and improved reliability



ASIC design flow. Steps 1–4 are **logical design**, and steps 5–9 are **physical design**

The ASICs in the Sun Microsystems SPARCstation 1

	SPARCstation 1 ASIC	Gates (k-gates)
1	SPARC integer unit (IU)	20
2	SPARC floating-point unit (FPU)	50
3	Cache controller	9
4	Memory-management unit (MMU)	5
5	Data buffer	3
6	Direct memory access (DMA) controller	9
7	Video controller/data buffer	4
8	RAM controller	1
9	Clock generator	1

The CAD tools used in the design of the Sun Microsystems SPARCstation 1		
Design level	**Function**	**Tool**
ASIC design	ASIC physical design	LSI Logic
	ASIC logic synthesis	Internal tools and UC Berkeley tools
	ASIC simulation	LSI Logic
Board design	Schematic capture	Valid Logic
	PCB layout	Valid Logic Allegro
	Timing verification	Quad Design Motive and internal tools
Mechanical design	Case and enclosure	Autocad
	Thermal analysis	Pacific Numerix
	Structural analysis	Cosmos
Management	Scheduling	Suntrac
	Documentation	Interleaf and FrameMaker

1.4 Economics of ASICs

We'll compare the most popular types of ASICs: an FPGA, an MGA, and a CBIC. The figures in the following sections are approximate and used to illustrate the different components of cost.

1.4.1 Comparison Between ASIC Technologies

Example of an ASIC **part cost**: A 0.5 µm, 20k-gate array might cost 0.01–0.02 cents/gate (for more than 10,000 parts) or $2–$4 per part, but an equivalent FPGA might be $20.

When does it make sense to use a more expensive part? This is what we shall examine next.

1.4.2 Product Cost

In a product cost there are **fixed costs** and **variable costs** (the number of products sold is the **sales volume**):

total product cost = fixed product cost + variable product cost × products sold

In a product made from parts the total cost for any part is

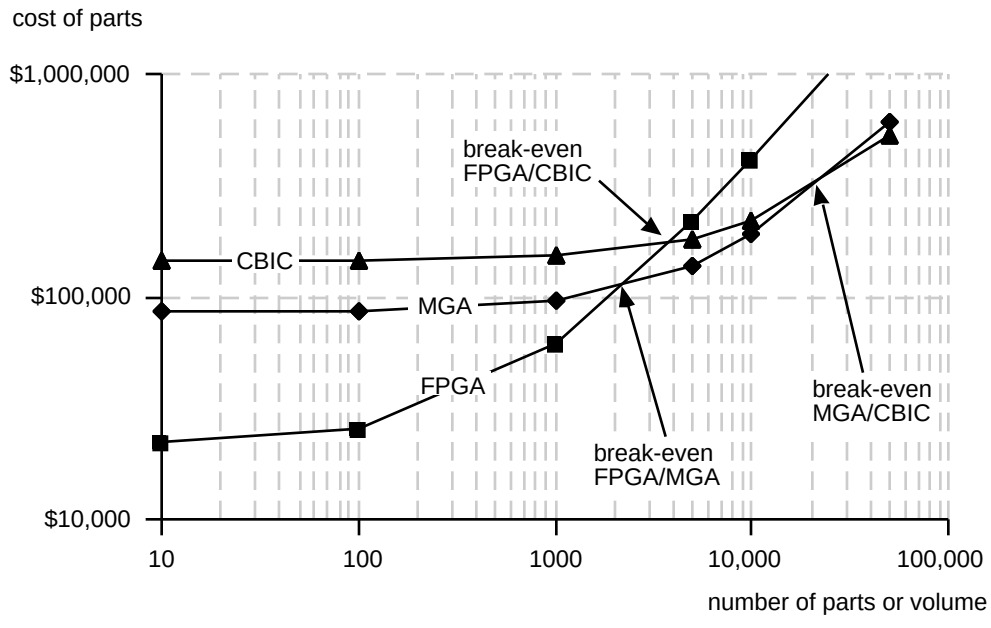
total part cost = fixed part cost + variable cost per part × volume of parts

For example, suppose we have the following (imaginary) costs:

- FPGA: $21,800 (fixed) $39 (variable)
- MGA: $86,000 (fixed) $10 (variable)
- CBIC $146,000 (fixed) $8 (variable)

Then we can calculate the following **break-even volumes**:

- FPGA/MGA ≈ 2000 parts
- FPGA/CBIC ≈ 4000 parts
- MGA/CBIC ≈ 20,000 parts



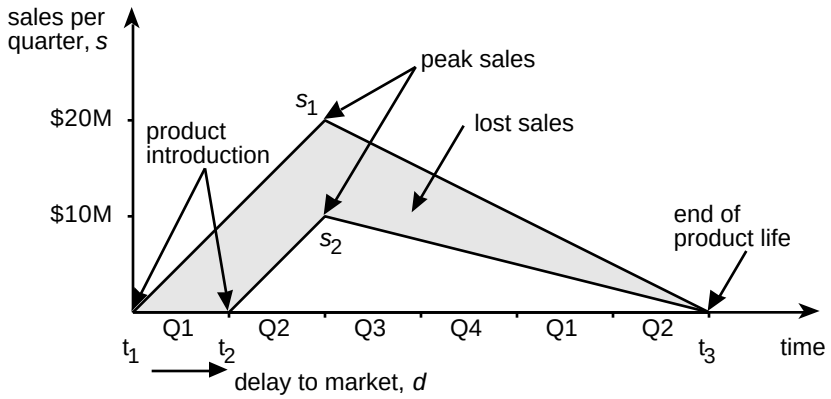
Break-even graph

1.4.3 ASIC Fixed Costs

Examples of fixed costs: **training cost** for a new **electronic design automation** (**EDA**) system • **hardware and software cost** • **productivity** • **production test** and **design for test** • **programming costs** for an FPGA • **nonrecurring-engineering** (**NRE**) • **test vectors** and **test-program development cost** • **pass** (**turn** or **spin**) • **profit model** represents the **profit flow** during the **product lifetime** • **product velocity** • **second source**

	FPGA		MGA		CBIC	
Training:	$800		$2,000		$2,000	
Days		2		5		5
Cost/day		$400		$400		$400
Hardware	$10,000		$10,000		$10,000	
Software	$1,000		$20,000		$40,000	
Design:	$8,000		$20,000		$20,000	
Size (gates)		10,000		10,000		10,000
Gates/day		500		200		200
Days		20		50		50
Cost/day		$400		$400		$400
Design for test:			$2,000		$2,000	
Days				5		5
Cost/day				$400		$400
NRE:			$30,000		$70,000	
Masks				$10,000		$50,000
Simulation				$10,000		$10,000
Test program				$10,000		$10,000
Second source:	$2,000		$2,000		$2,000	
Days		5		5		5
Cost/day		$400		$400		$400
Total fixed costs	$21,800		$86,000		$146,000	

Spreadsheet, “Fixed Costs”



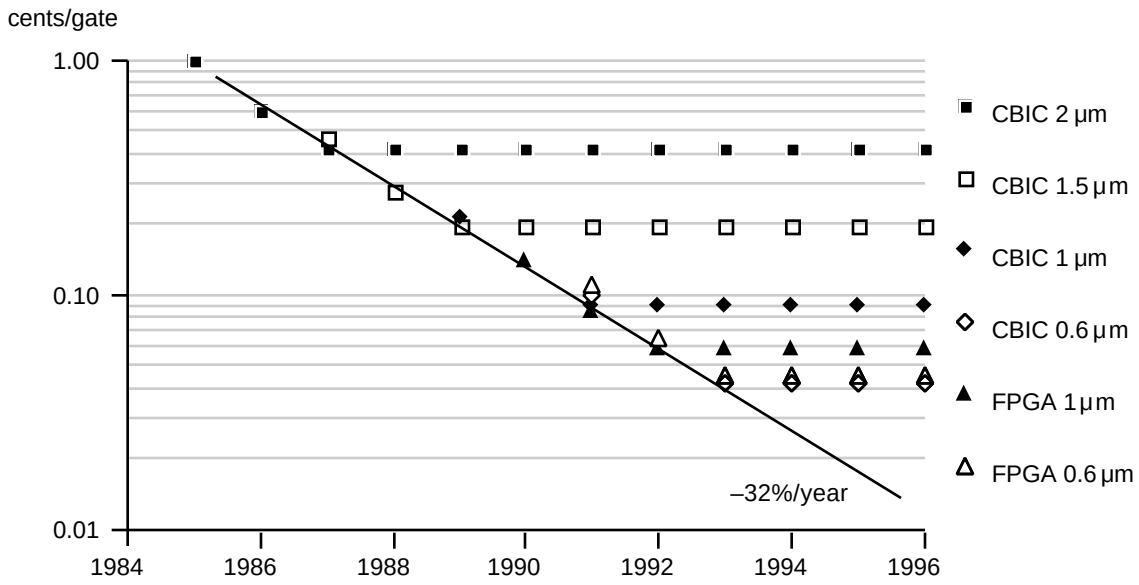
Profit model

1.4.4 ASIC Variable Costs

Factors affecting fixed costs: **wafer size • wafer cost • Moore's Law** (Gordon Moore of Intel) **• gate density • gate utilization • die size • die per wafer • defect density • yield • die cost • profit margin** (depends on **fab** or **fabless**) **• price per gate • part cost**

	FPGA	MGA	CBIC	Units
Wafer size	6	6	6	inches
Wafer cost	1,400	1,300	1,500	$
Design	10,000	10,000	10,000	gates
Density	10,000	20,000	25,000	gates/sq.cm
Utilization	60	85	100	%
Die size	1.67	0.59	0.40	sq.cm
Die/wafer	88	248	365	
Defect density	1.10	0.90	1.00	defects/sq.cm
Yield	65	72	80	%
Die cost	25	7	5	$
Profit margin	60	45	50	%
Price/gate	0.39	0.10	0.08	cents
Part cost	$39	$10	$8	

Spreadsheet, "Variable Costs"



Example price per gate figures

1.5 ASIC Cell Libraries

You can:

(1) use a **design kit** from the **ASIC vendor**

(2) buy an **ASIC-vendor library** from a **library vendor**

(3) you can build your own cell library

(1) is usually a **phantom library**—the cells are empty boxes, or **phantoms**, you **hand off** your design to the ASIC vendor and they perform **phantom instantiation** (Synopsys CBA)

(2) involves a **buy-or-build decision**. You need a **qualified cell library** (qualified by the **ASIC foundry**) If you own the masks (the **tooling**) you have a **customer-owned tooling** (**COT**, pronounced “see-oh-tee”) solution (which is becoming very popular)

(3) involves a complex **library development** process: **cell layout** • **behavioral model** • Verilog/VHDL model • **timing model** • test strategy • **characterization** • **circuit extraction** • **process control monitors** (**PCMs**) or **drop-ins** • **cell schematic** • cell icon • **layout versus schematic** (**LVS**) check • **cell icon** • **logic synthesis** • **retargeting** • **wire-load model** • **routing model** • **phantom**

1.6 Summary

Key concepts:

- We could define an ASIC as a design style that uses a cell library
- The difference between full-custom and semicustom ASICs
- The difference between standard-cell, gate-array, and programmable ASICs
- The ASIC design flow
- Design economics including part cost, NRE, and breakeven volume
- The contents and use of an ASIC cell library

Types of ASIC

ASIC type	Family member	Custom mask layers	Custom logic cells
Full-custom	Analog/digital	All	Some
Semicustom	Cell-based (CBIC)	All	None
	Masked gate array (MGA)	Some	None
Programmable	Field-programmable gate array (FPGA)	None	None
	Programmable logic device (PLD)	None	None

1.7 Problems

Suggested homework: 1.4, 1.5, 1.9 (from *ASICs... the book*)

1.8 Bibliography

EE Times (ISSN 0192-1541, `http://techweb.cmp.com/eet`), *EDN* (ISSN 0012-7515, `http://www.ednmag.com`), EDAC (Electronic Design Automation Companies) (`http://www.edac.org`), The Electrical Engineering page on the World Wide Web (E2W3) (`http://www.e2w3.com`), SEMATECH (Semiconductor Manufacturing Technology) (`http://www.sematech.org`), The MIT Semiconductor Subway (`http://www-mtl.mit.edu`), EDA companies at `http://www.yahoo.com` under `Business_and_Economy` in `Companies/Computers/Software/Graphics/CAD/IC_Design`, The MOS Implementation Service (MOSIS) (`http://www.isi.edu`), The Microelectronic Systems Newsletter at `http://www-ece.engr.utk.edu/ece`, NASA (`http://nppp.jpl.nasa.gov/dmg/jpl/loc/asic`)

1.9 References

Glasser, L. A., and D. W. Dobberpuhl. 1985. *The Design and Analysis of VLSI Circuits.* Reading, MA: Addison-Wesley, 473 p. ISBN 0-201-12580-3. TK7874.G573. Detailed analysis of circuits, but largely nMOS.

Mead, C. A., and L. A. Conway. 1980. *Introduction to VLSI Systems.* Reading, MA: Addison-Wesley, 396 p. ISBN 0-201-04358-0. TK7874.M37.

Weste, N. H. E., and K. Eshraghian. 1993. *Principles of CMOS VLSI Design: A Systems Perspective.* 2nd ed. Reading, MA: Addison-Wesley, 713 p. ISBN 0-201-53376-6. TK7874.W46. Concentrates on full-custom design.