Administrivia

- Thursday lectures now from 10:00
- Submit both lab2 tasks for review, if possible

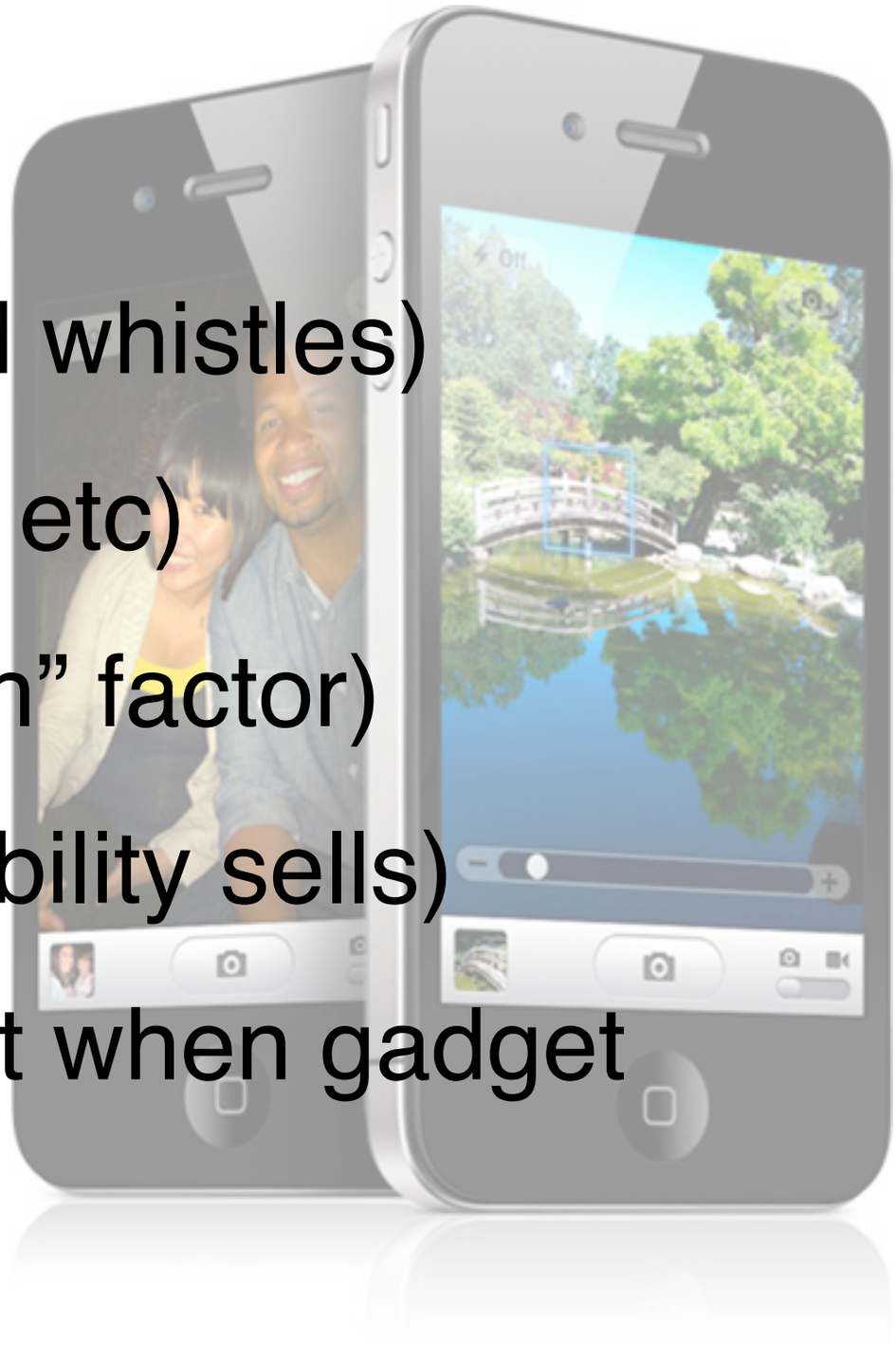
Technology platforms

DAT093

larssv@chalmers.se

What the end user wants

- Low cost (o/w no sale)
- Functionality (bells and whistles)
- Flexibility (applications etc)
- Performance (the “flash” factor)
- Size, battery life (portability sells)
- Reliability (user hates it when gadget fails)



How designer provides it

- Low cost: omit needless stuff, work fast
- Functionality: handle complex designs
- Flexibility: define behavior with software
- Performance: handle high clock rate etc
- Size, battery life: minimize power demands
- Reliability: no mistakes, robust designs, thorough verification

TALL ORDER!

Challenges

- Tradeoffs:
  - performance vs low power
  - low cost vs functionality
  - low cost vs reliability
  - performance vs flexibility …
- Best design approach is case dependent!
  - Technology platform has major influence

Technology platform?

narrow sense

- The physical implementation technology (e.g. silicon), *plus*
  - any partial hardware designs to be re-used (intellectual property blocks), and
  - any available software/firmware libraries (also IP!), and
  - the tools to put it all together

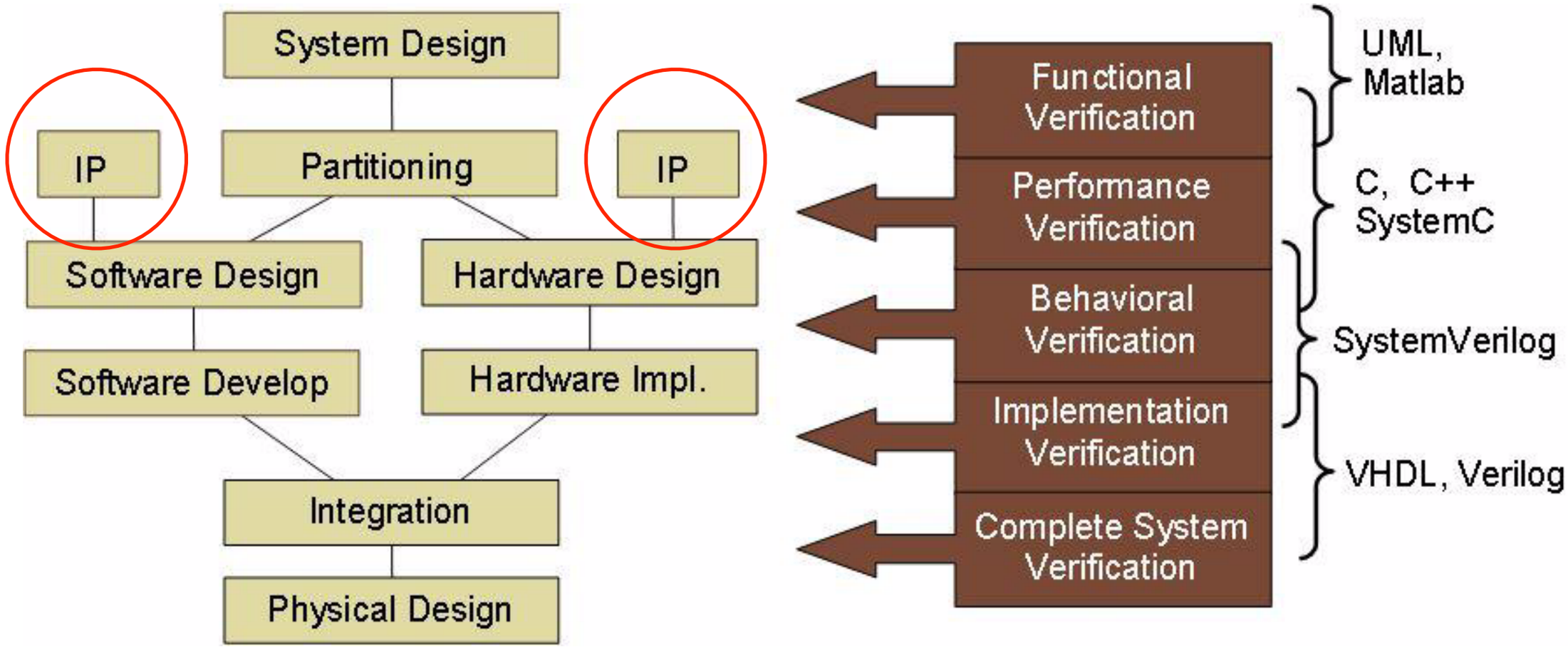
broad sense

Design re-use

- Intellectual-property (IP) cores / blocks
  - Subcomponents
    - in-house
    - external
  - HDL description: “soft macro” – technology independent (sort of…)
  - Physical implementation: “hard macro”
- Soft/firmware libraries

SAVE DEVELOPMENT AND VERIFICATION EFFORT

Design/verification tool chain



Some technology alternatives

- Application Specific Standard Products
  - Off-the-shelf, high volume
  - Processors, communication devices, etc
- Application Specific Integrated Circuit
  - Purpose-designed
- Field-programmable logic devices (FPGAs, CPLDs, etc)
- “Structured ASICs”

Evolving field!
Designer must keep up!

What drives platform development?

1. Costs

- Simple model: fixed vs. variable costs
- Fixed costs: per development project
  - “Non-Recurrent Engineering” (NRE)
    - Independent of # of parts built
- Variable costs: per unit built
  - Associated with production, testing, and other per-part activities
- Which one is more important? Why?

2. Silicon-CMOS trends

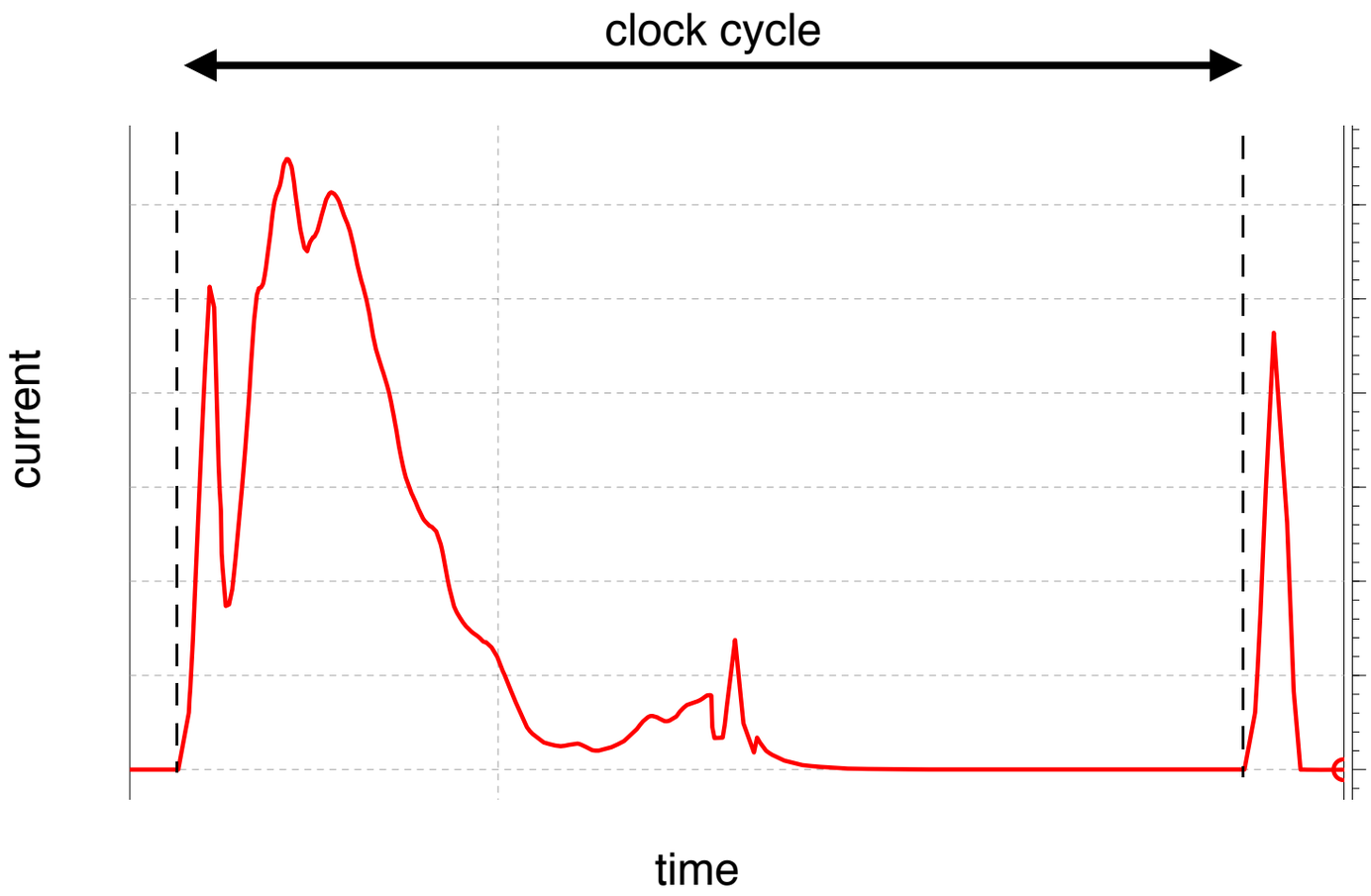
- CMOS power, speed both depend on voltage and capacitances
  - On-chip capacitances are small and getting smaller with scaling
  - Voltage is low and shrinking
- Performance, density, and power per function improve with each silicon process generation
  - Total power is another matter…

"Everything" is CMOS!

Will assume in rest of lecture

Power supplied

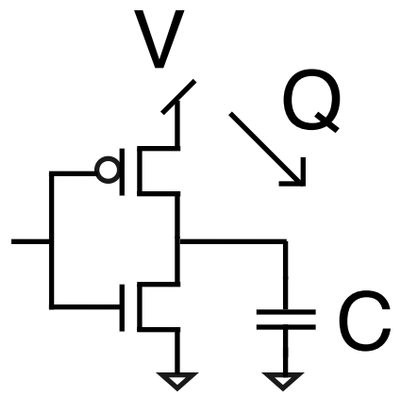
- Supply current, power varies with time



- <u>Switching power</u> is main contributor

CMOS switching power

- Inverter models general CMOS circuit
- Consider charging of C from 0 to V
- Injected charge: $Q = C \cdot V$

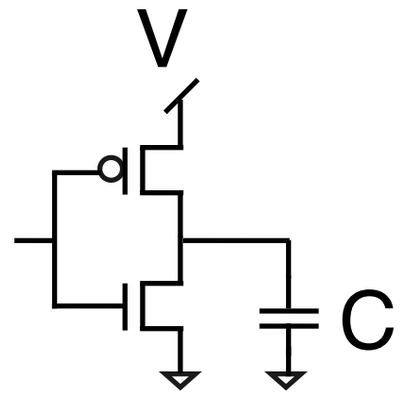


- Injected energy: $E_{inj} = Q \cdot V = CV^2$
- Dissipated energy: $Q \cdot V_{avg} = Q \cdot V/2 = CV^2/2$
- Stored energy (on C): $CV^2 - CV^2/2 = CV^2/2$

Switching power, cont.

- Stored energy dissipated during discharge
  - $E_{inj} = CV^2$ completely dissipated over full cycle
- If cycle frequency is f, then

$$P = f \cdot CV^2 = fCV^2$$



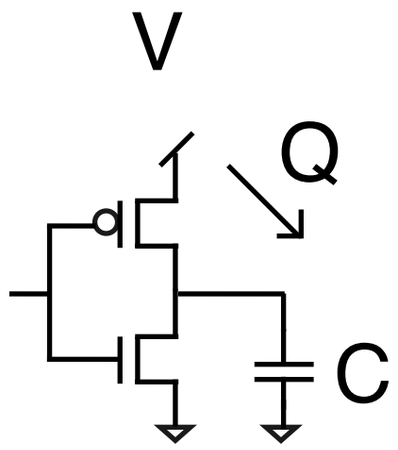
- Generalizes to multi-gate system:

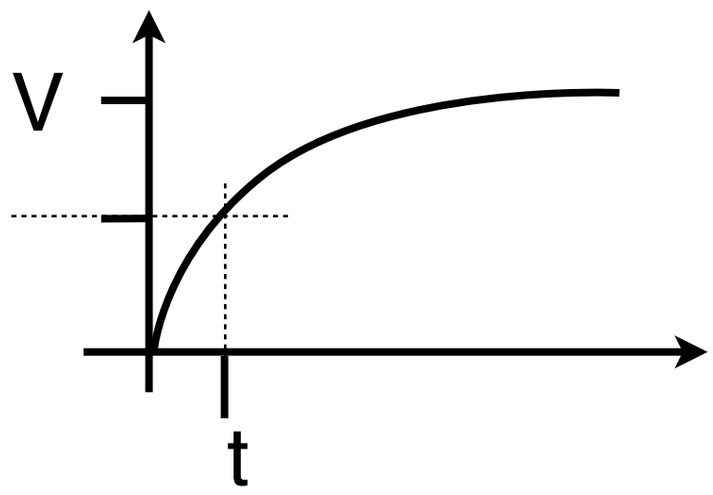
$$P_{Tot} = V^2 \cdot \sum_i (f_i \cdot C_i)$$

- Small C and small V give small P_{Tot}

CMOS performance

- $t \sim Q / I = CV / I$
- $I \sim V^a$, $1 < a < 2$
  - Depends on gate voltage, device size
- $t \sim C / V^{(a-1)}$
- Small C and large V give small t, so high speed





Circuit considerations

- Small C and small V give small P
- Small C and large V give small t, so high speed

… so …

- Reduce C to improve speed and power
- Choose V for speed/power tradeoff
  - … within reasonable range…
  - Typically system-level decision (one or few voltage levels)

Silicon scaling trends, summary

- Moore/Dennard scaling reduces capacitance C!
- Decreasing power for similar circuits
  - … so can afford to increase frequency and/or number of circuits
- Increasing number of devices for same cost
  - … so can afford to increase complexity
- Supports the other exponential trends (lecture 1)

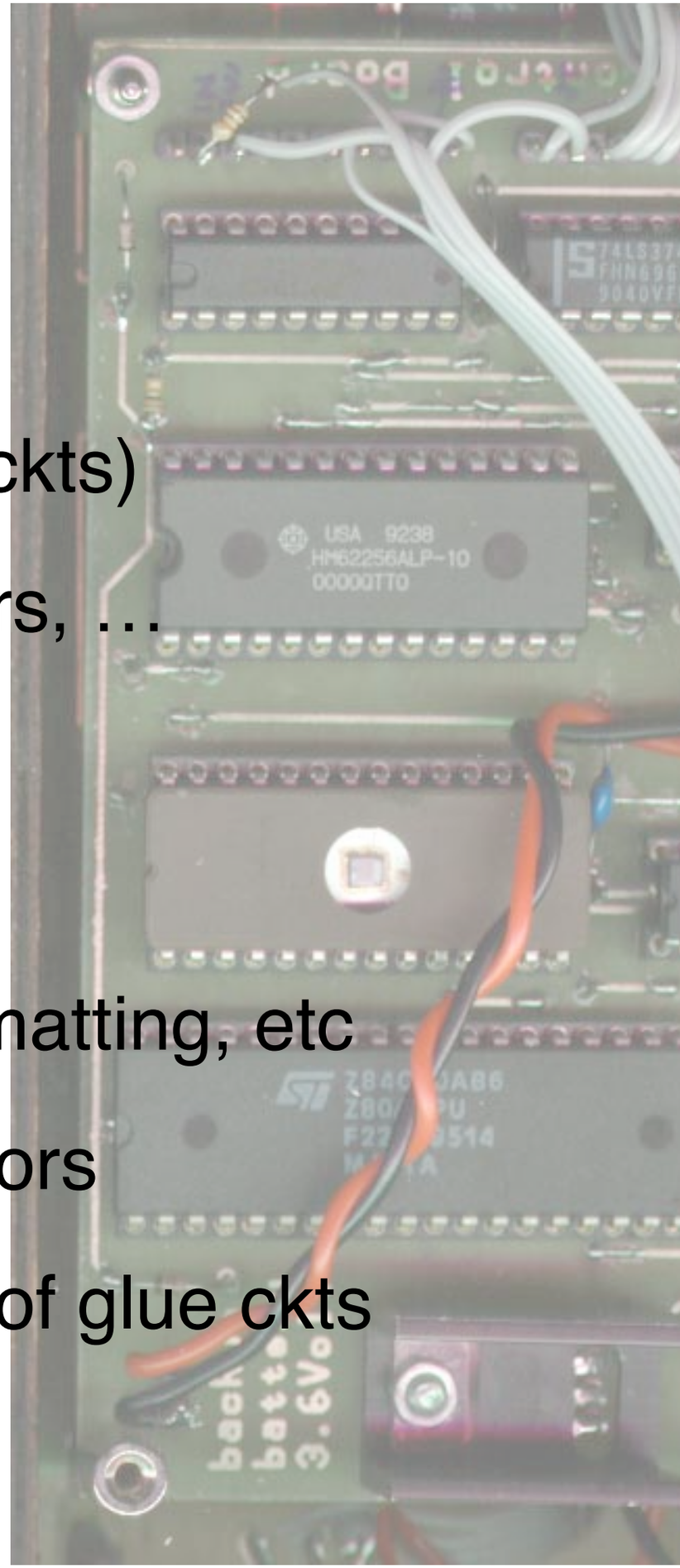
Trend is your friend… but:

- New technology platforms
  - have higher risk
    - initially have higher cost!
    - are less complete
- "Old" platform may be good choice for low complexity, modest performance

History, evolution

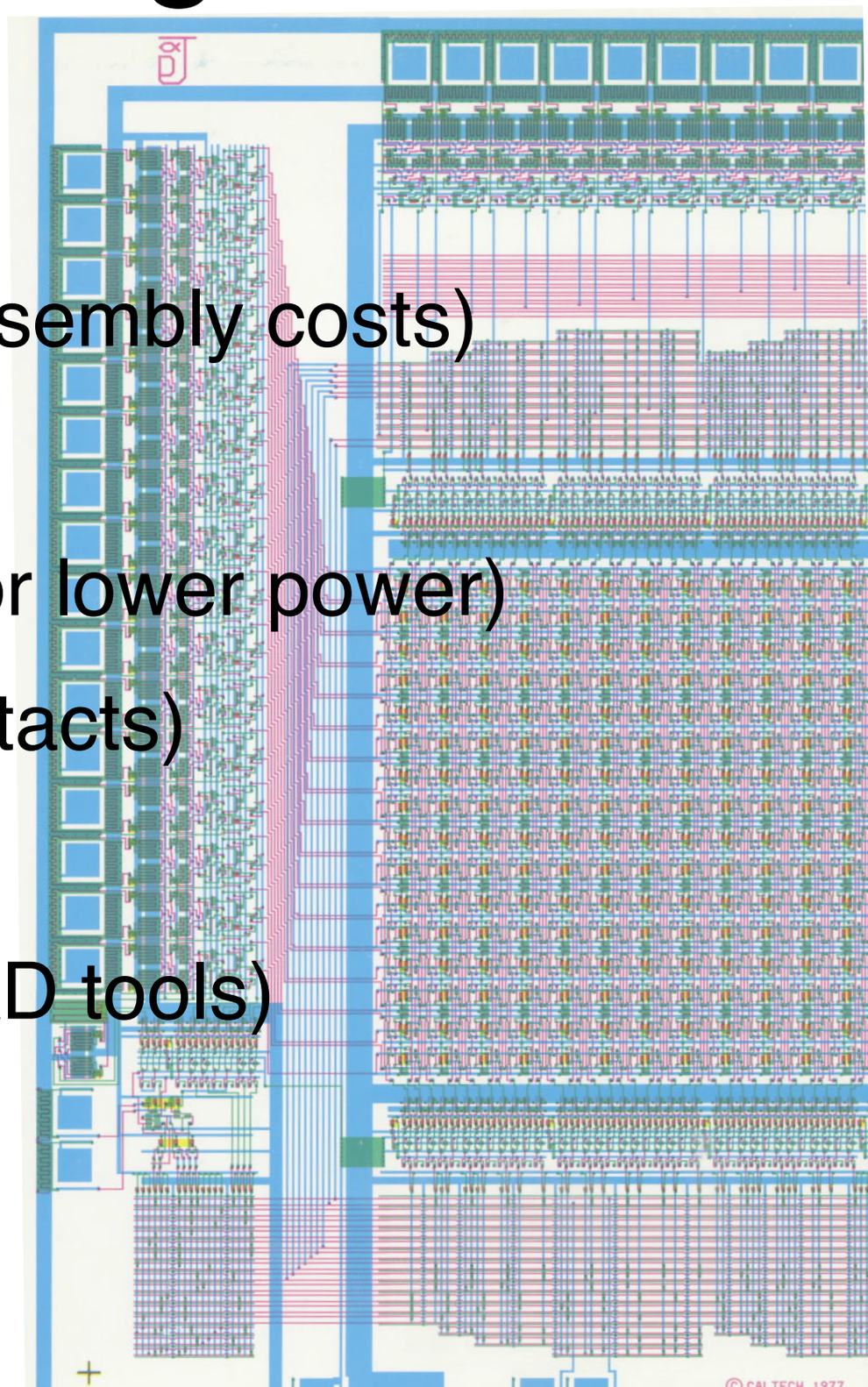
1970s: PCBs

- Main LSI (large-scale integrated ckts)
  - processor, timers, I/O controllers, …
    - ~10K transistors
- “Glue logic”
  - global control signals, data formatting, etc
    - few gates, dozens of transistors
- System: handful of LSIs, dozens of glue ckts



Early 1980s: glue logic ICs

- Benefits:
  - Fewer parts (thus lower assembly costs)
  - Smaller size
  - Higher performance (and/or lower power)
  - Better reliability (fewer contacts)
- … but also drawbacks:
  - Higher NRE costs (fab, CAD tools)
  - Longer time-to-market



Mid 1980s: Programmable logic

- Custom-IC glue logic good for high volumes
  - NRE cost is amortized over many parts
- For small volumes, programmable logic may be beneficial
- Lower NRE cost than ASICs
  - Higher unit costs
- Worse power/performance than ASICs
  - Higher density than SSI/MSI logic

FPGA CPLD

Late 1980s: Sea-of-gates

- Prefabricate chips w/ large numbers of similar transistors, gates
- Just add metal interconnect to define functionality
- Lower turn-around-time than for ASICs
- No reprogrammability
- Area utilization?

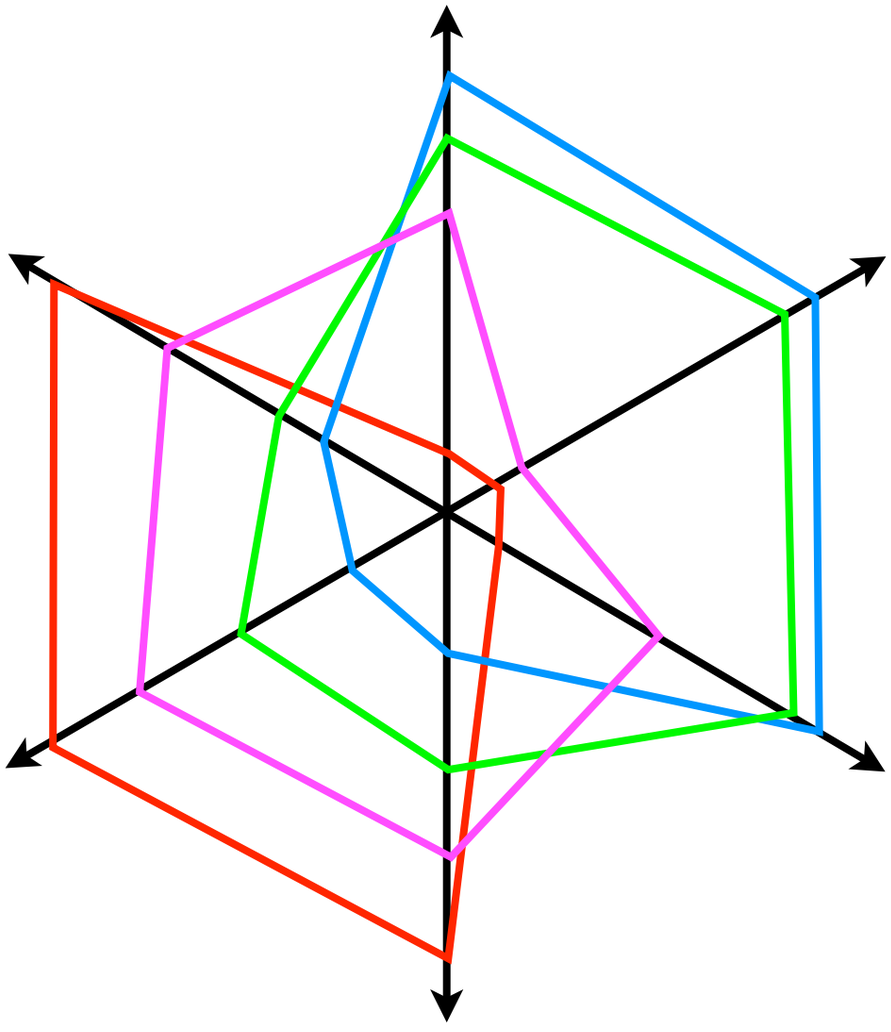
[Okabe et al, IEEE JSSC, Oct 1989]

2000s: “Structured” or “Platform” ASICs

- IP cores plus millions of gates/LUTs on one chip
- Pre-fabricate everything but the top interconnect layers (cf. sea-of-gates)
  - Use top metals to determine final functionality
- If IPR cores “fit” the application, almost ASIC performance with much lower NRE costs!
  - … but if “fit” is bad, then maybe high overhead

[Ho et al, IEEE TVS 2013]

Comparison (bigger is better)



Software on processor

Field upgrade

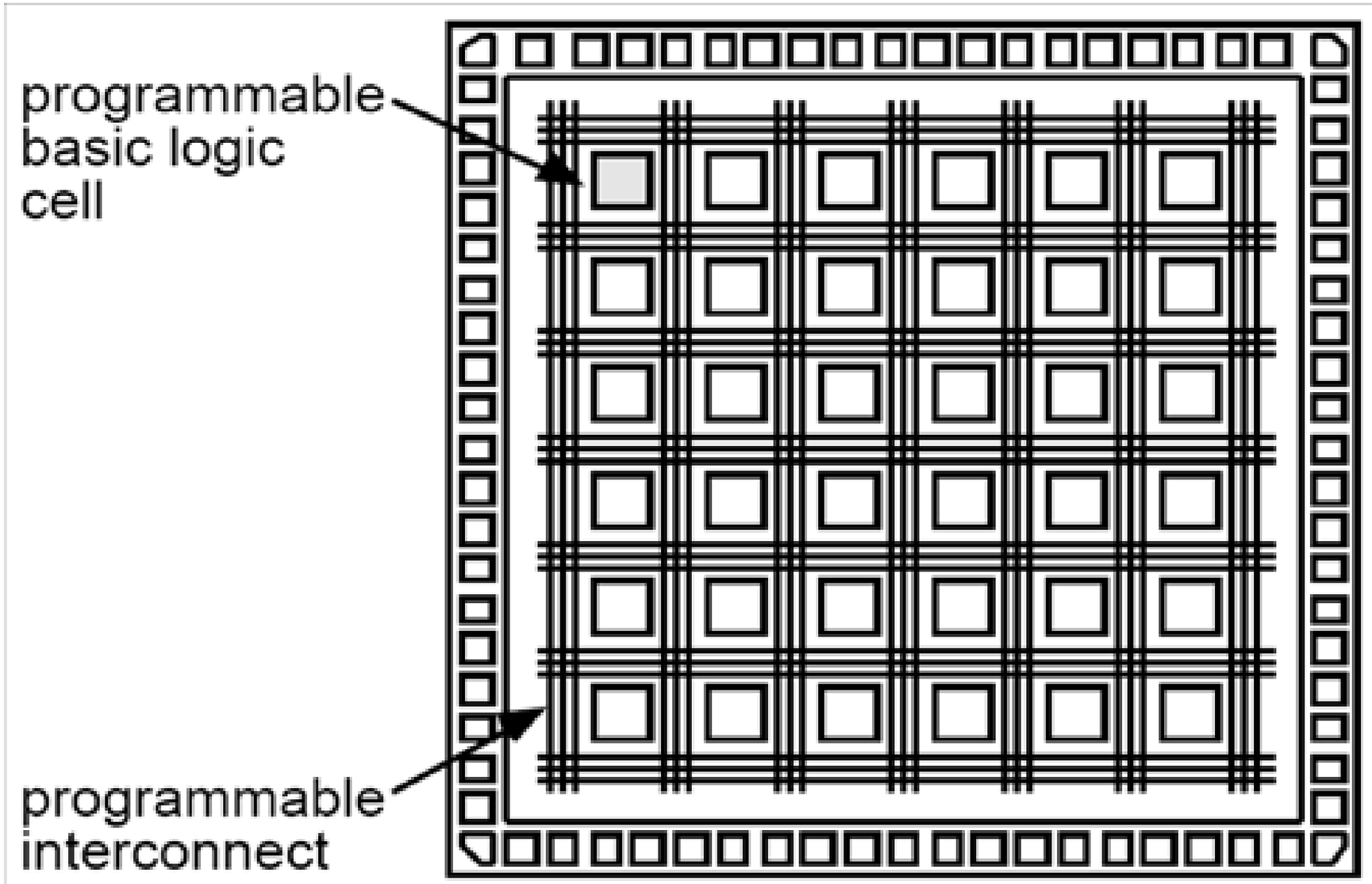
FPGA

Structured ASIC

ASIC

FPGA technology offerings

Original FPGA structure



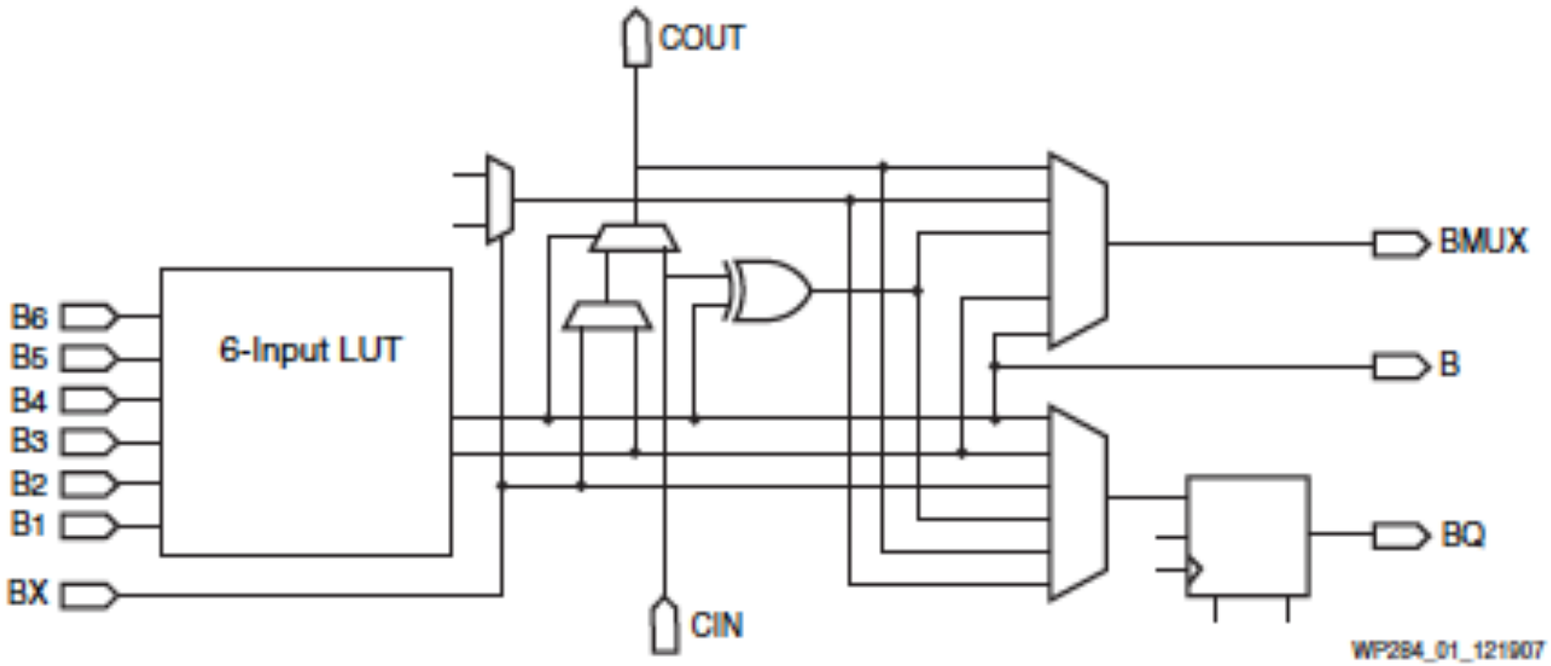
Look-Up Table (LUT)

a	b	x	y	z
0	0	1	0	0
0	1	0	1	1
1	0	0	1	1
1	1	0	1	0

Memory contents

- $M \cdot 2^N$ bits of memory can implement any M-valued logic function of N inputs

Logic Cell = LUT + some fixed logic



- Example from Xilinx Virtex-5 documentation
  - Lookup table, carry logic, muxes, FF

[Percey: Advantages of the Virtex-5 FPGA 6-input LUT architecture. Xilinx, 2007.]

Variations on FPGA theme

- Programmable cells + interconnect
- … plus memories
- … plus adders, multipliers, etc
- … plus I/O modules, PLLs, etc
- … plus entire microprocessors
- Combinations target certain market segments

FPGA configuration

- SRAM
  - Loads configuration at startup time
- Antifuse
  - program once, never erase
- EPROM
  - Erasable
- EEPROM + Flash
  - Electrically erasable

Variety

Device	Logic Cells[1]	Configurable Logic Blocks (CLBs)			DSP48A1 Slices[3]	Block RAM Blocks		CMTs[5]	Memory Controller Blocks (Max)[6]	Endpoint Blocks for PCI Express	Maximum GTP Transceivers	Total I/O Banks	Max User I/O
		Slices[2]	Flip-Flops	Max Distributed RAM (Kb)		18 Kb[4]	Max (Kb)						
XC6SLX4	3,840	600	4,800	75	8	12	216	2	0	0	0	4	132
XC6SLX9	9,152	1,430	11,440	90	16	32	576	2	2	0	0	4	200
XC6SLX16	14,579	2,278	18,224	136	32	32	576	2	2	0	0	4	232
XC6SLX25	24,051	3,758	30,064	229	38	52	936	2	2	0	0	4	266
XC6SLX45	43,661	6,822	54,576	401	58	116	2,088	4	2	0	0	4	358
XC6SLX75	74,637	11,662	93,296	692	132	172	3,096	6	4	0	0	6	408
XC6SLX100	101,261	15,822	126,576	976	180	268	4,824	6	4	0	0	6	480
XC6SLX150	147,443	23,038	184,304	1,355	180	268	4,824	6	4	0	0	6	576
XC6SLX25T	24,051	3,758	30,064	229	38	52	936	2	2	1	2	4	250
XC6SLX45T	43,661	6,822	54,576	401	58	116	2,088	4	2	1	4	4	296
XC6SLX75T	74,637	11,662	93,296	692	132	172	3,096	6	4	1	8	6	348
XC6SLX100T	101,261	15,822	126,576	976	180	268	4,824	6	4	1	8	6	498
XC6SLX150T	147,443	23,038	184,304	1,355	180	268	4,824	6	4	1	8	6	540

[source: xilinx.com]

- Large product families per generation
- CAD support essential

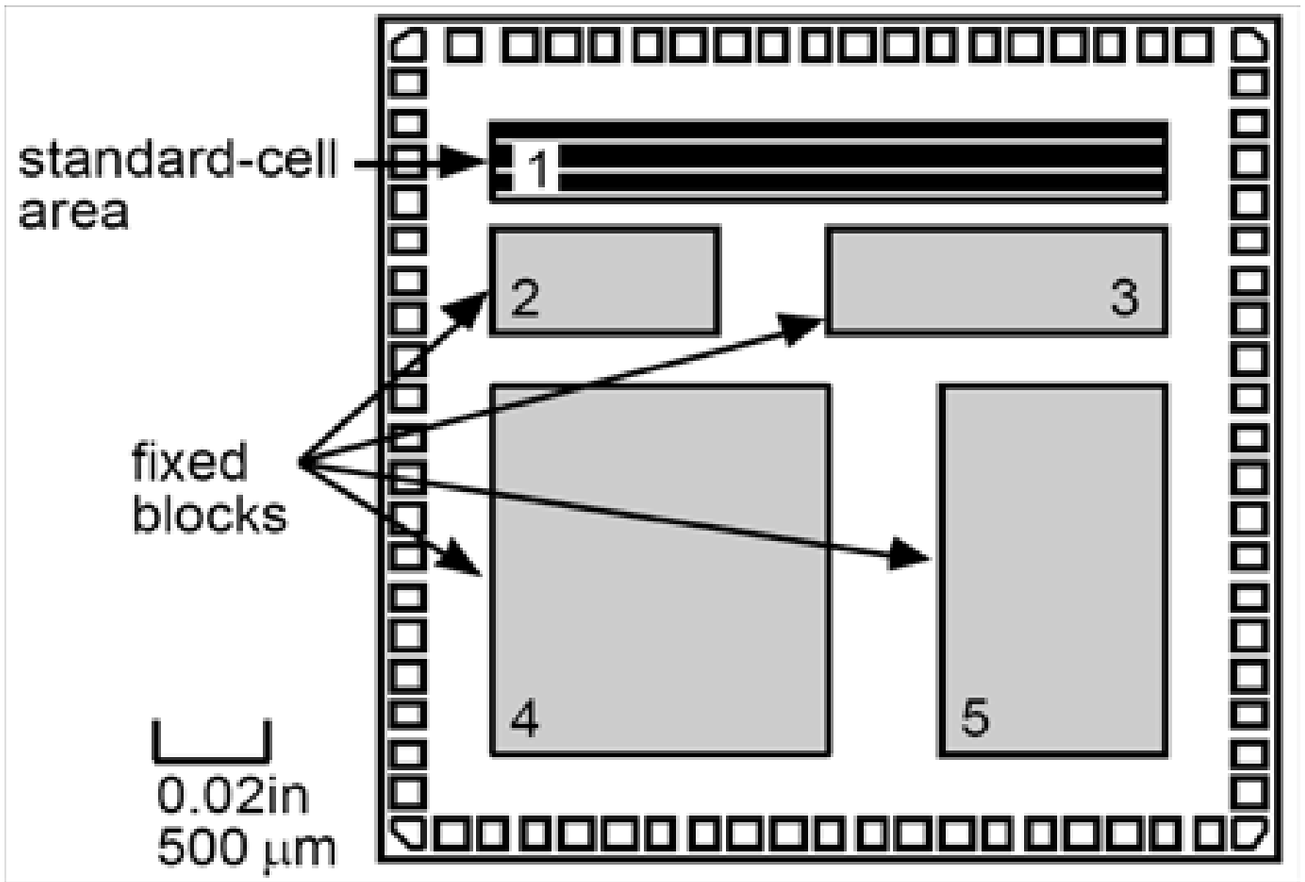
Typical FPGA CAD

- IDEs provided by FPGA vendors
  - Generic functionality described with HDLs (such as VHDL)
    - “Wizards” may be offered for configuration of IP modules
  - Edit / simulate / synth / download / test: minutes or hours

ASIC technology offerings

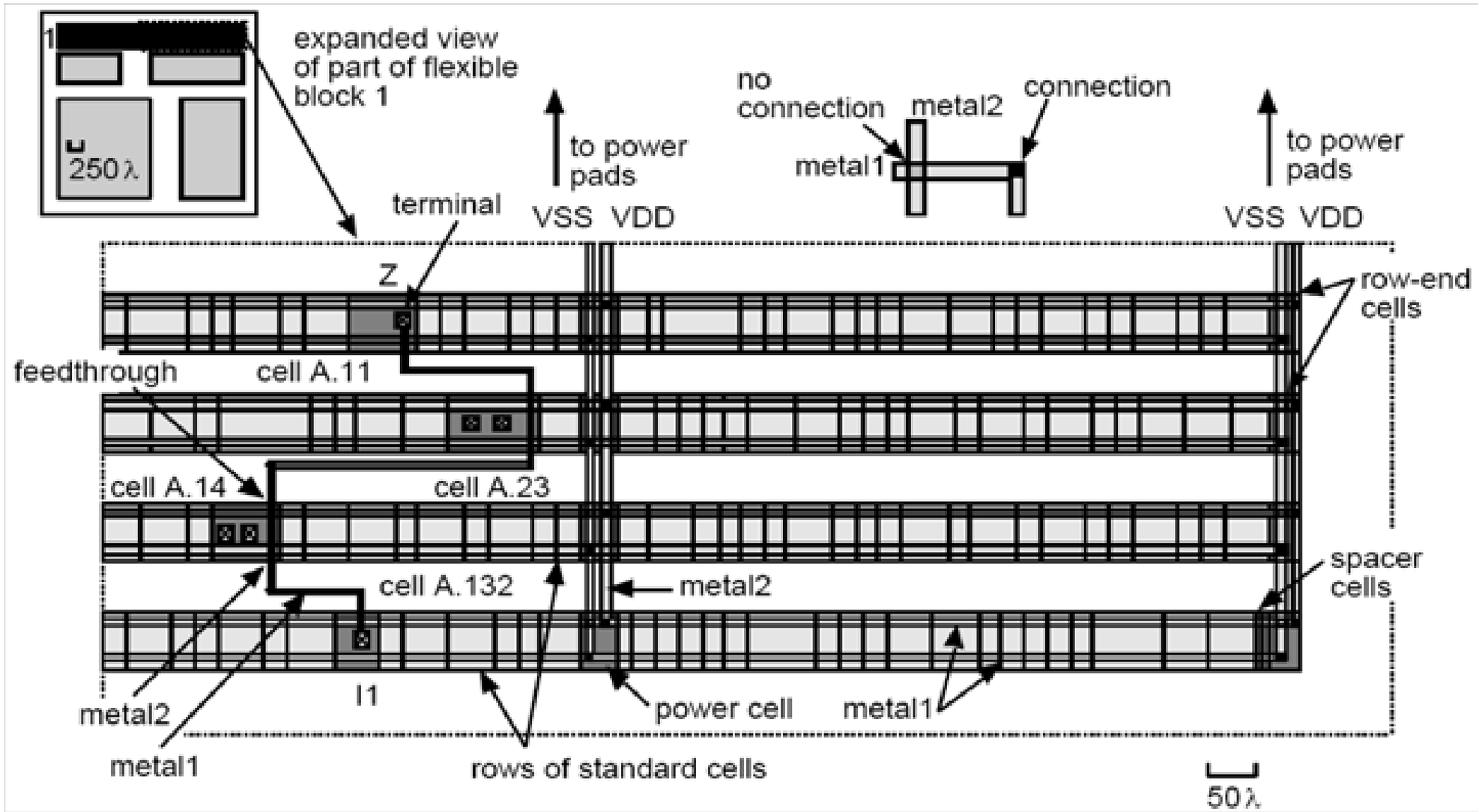
Cell-based ASIC

- Library of standard cells
- All mask layers are unique for this ASIC
- Custom blocks can be included



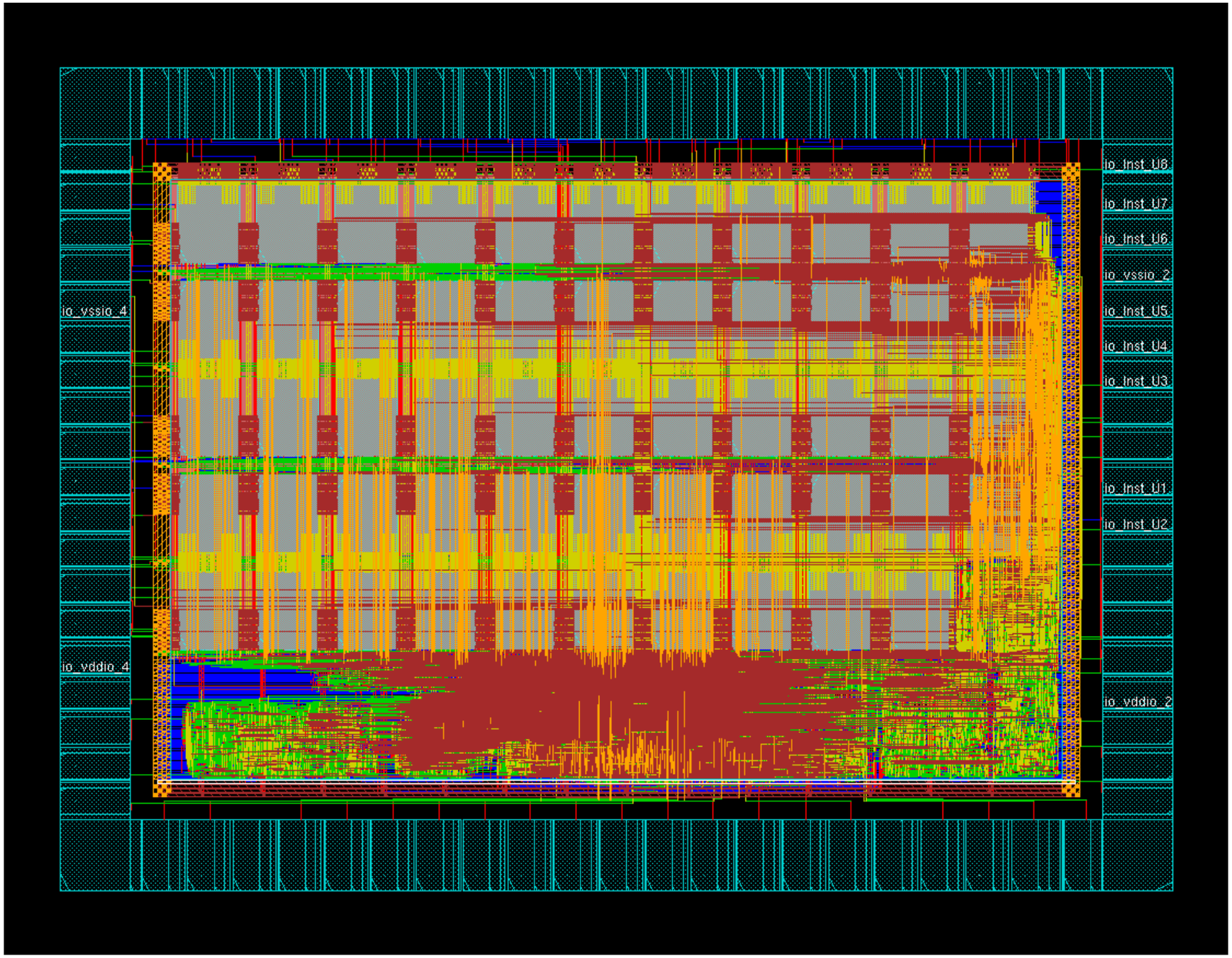
source: M J S Smith

Standard-cell technology



- Predefined cells are placed and connected

Standard-cell processor

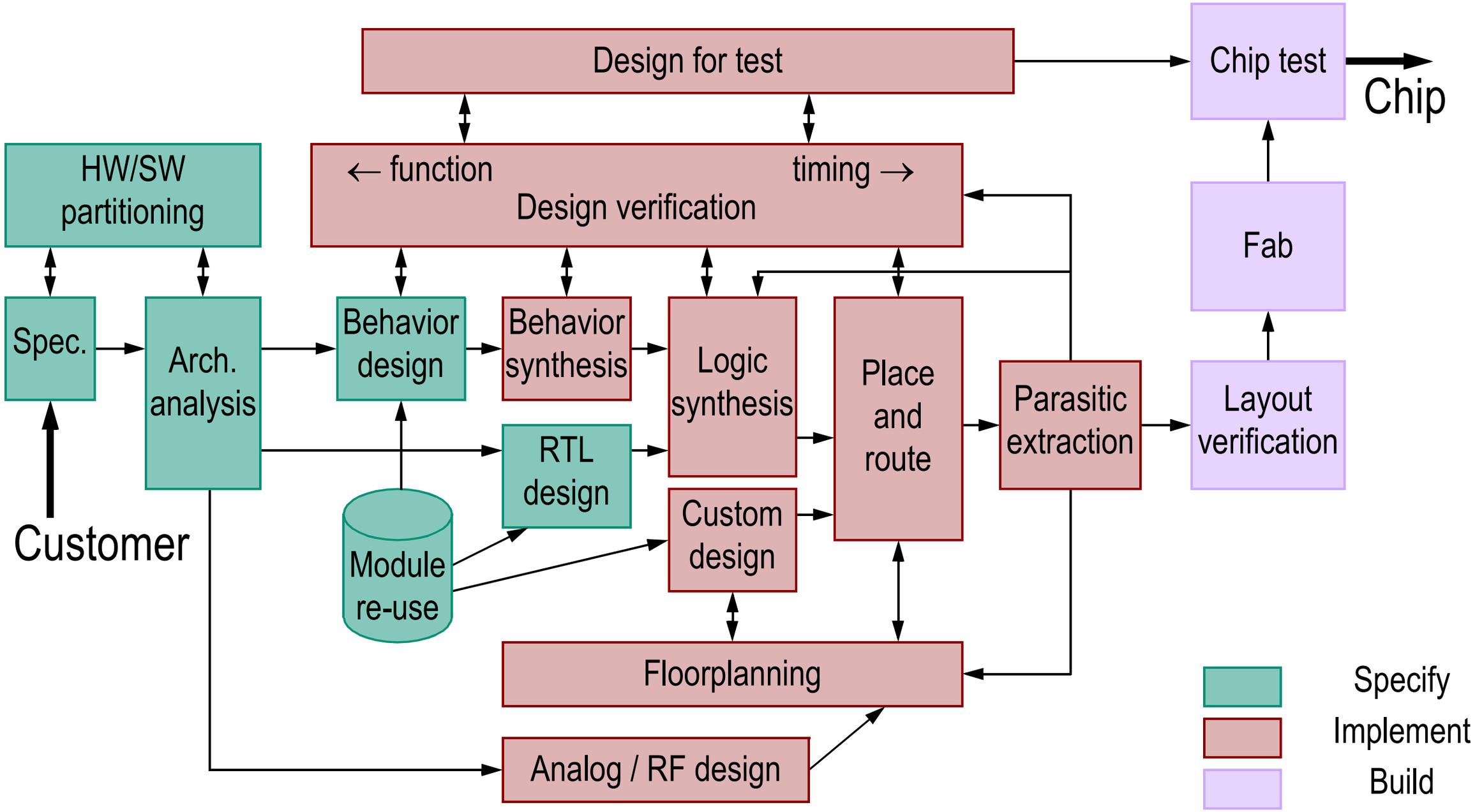


placement + routing

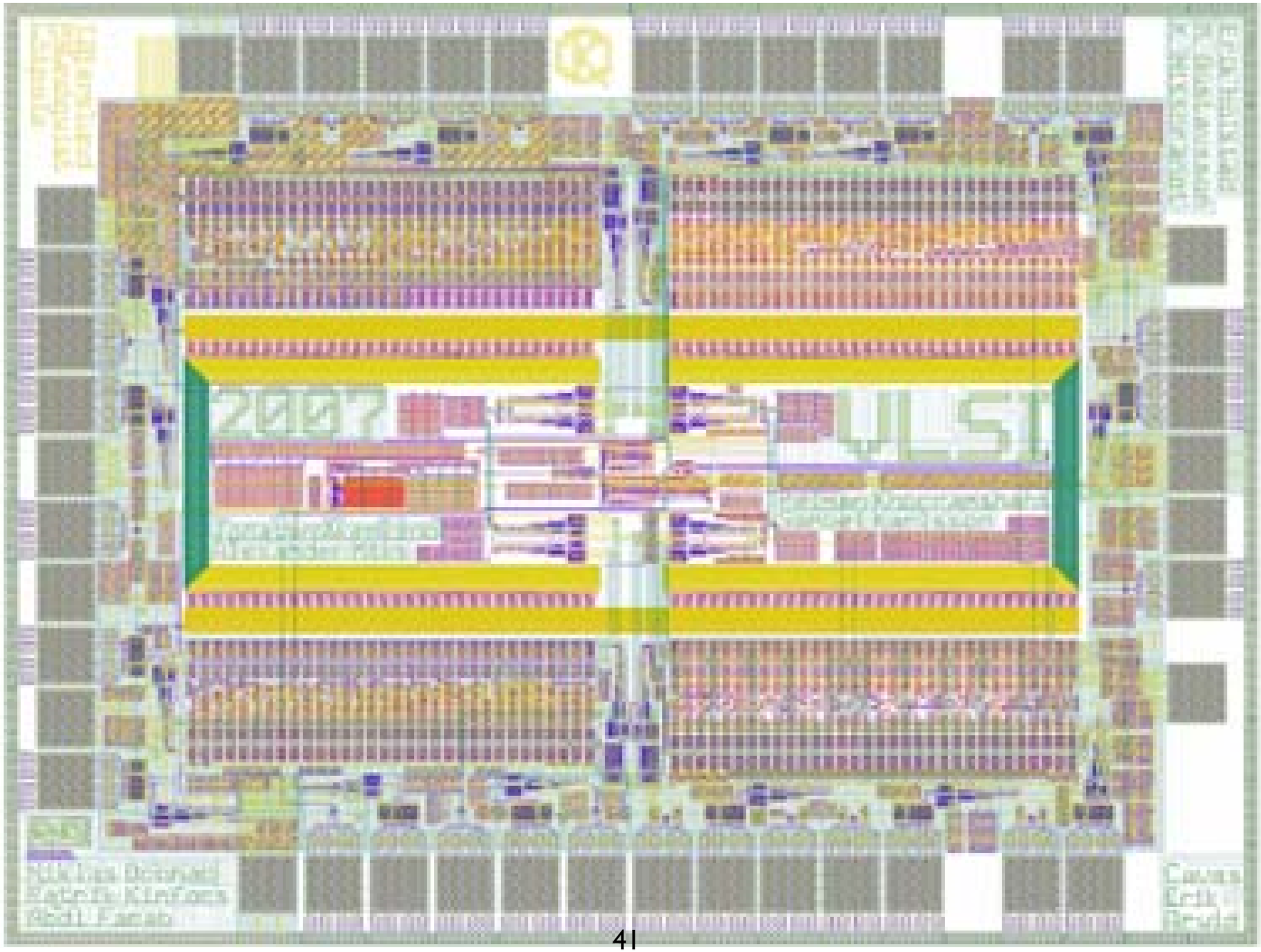
Full-custom ASIC

- All layers are custom designed
- No block or cell reuse
- Pure form rarely used for digital
  - Common for analog/RF parts
  - Extreme power/performance digital
  - Critical digital blocks

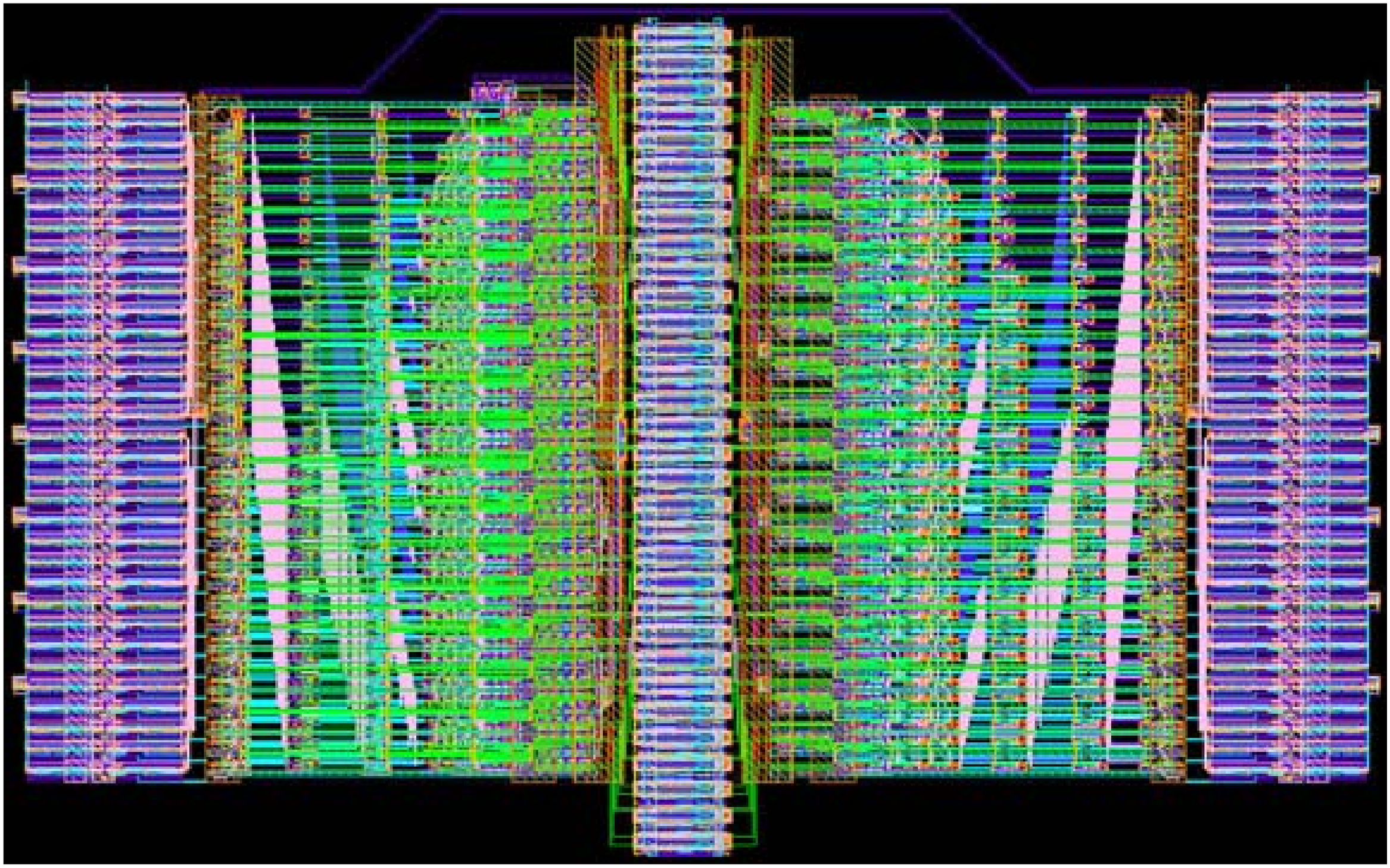
ASIC design flow in context



Full-custom student chip (2007)



Two 130-nm 32-bit adders



FPGAs vs. or and ASICs?

- First prototype?
  - Software on processor
  - Reconfigurable hardware (FPGA)
- First version of product?
  - FPGA may still make sense (low TTM, low total cost)
- If volume justifies it and/or performance requires it, build ASIC version

FPGAs strike back?

- FPGA vendors want to keep customer business when volumes increase!
  - Offer “same” device w/o reconfiguration at lower unit price
    - Guaranteed identical functionality
    - No extra NRE for ASIC tapeout

Micro case studies

Case Study I

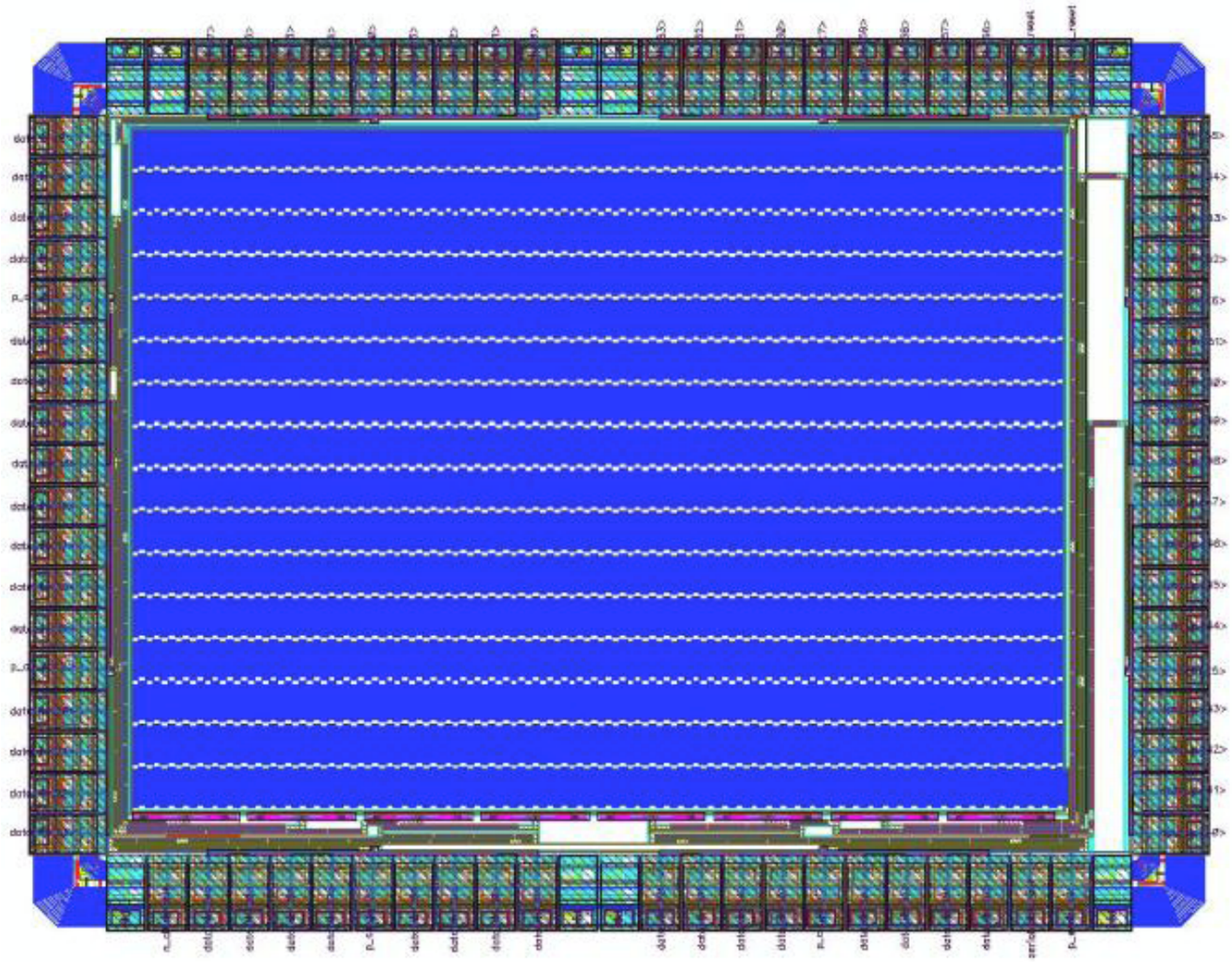
[Erik Ryman, PhD project]

- System designed for satellite deployment
- Compute cross-correlations between every pair of 64 input signals
- ~4 GS / s
- 1-bit resolution

Hand-tuned full-custom ASIC

Cross-correlator ASIC

Presented at the 2011 Custom Integrated Circuits Conference, Sep 20, 2011



- 3 mm² in 65nm, 3M transistors
- ~1 MY design time, SEK 250K for 25 dice

Case Study II

[Bayan Nasri, MSc project, 2010]

- Novel bone-anchored hearing aid
  - Miniaturize prototype PCB for clinical trials
    - Audio signal processing
    - Power not very critical
    - Volume: dozens

FPGA + analog

Summary

- Bewildering array of technology alternatives
- Each optimized for a certain combination of requirements

1. Low cost
2. Functionality
3. Flexibility
4. Performance
5. Size, battery life
6. Reliability

New developments

- More alternatives will appear
  - Address some corner of the design space
- Driven by market requirements (first slides)
- Enabled by
  - Technology evolution (“Moore’s Law”)
  - Ever more capable CAD tools
- Designers need to keep abreast!
- Last-week lecture will address future trends