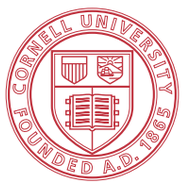
The Future of FPGAs

Rajit Manohar
Computer Systems Lab
Cornell University, Ithaca, NY 14853
http://vlsi.cornell.edu/

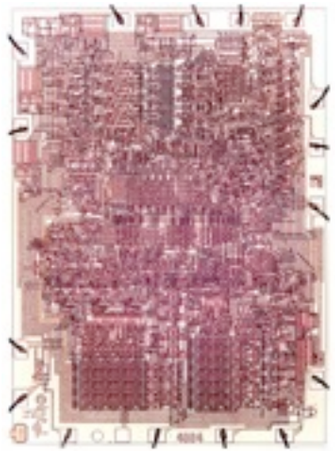
Chief Scientist
Achronix Semiconductor Corp.
http://www.achronix.com/



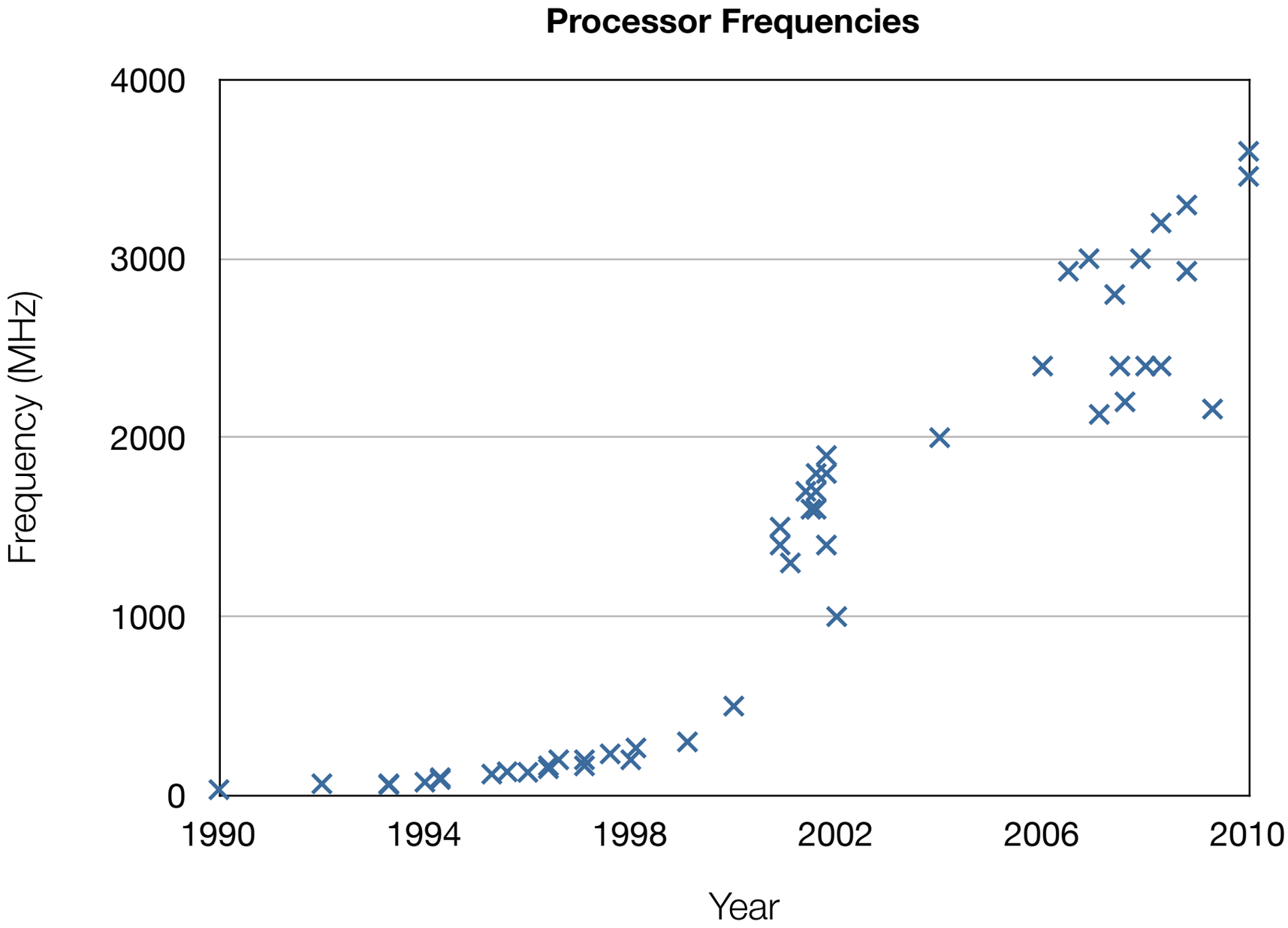
Cornell University

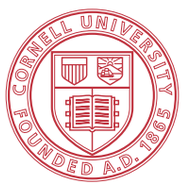
System complexity in VLSI

Year	Name	Transistors
1982	80286	134,100
1985	80386	275,000
1993	Pentium	3.1 million
1995	Pentium Pro	6 million
1998	Pentium III	9.5 million
2000	Pentium IV	42 million
2002	McKinley	243 million
2005	Montecito	1.7 billion
2020	????	~50 billion



CPU frequency scaling



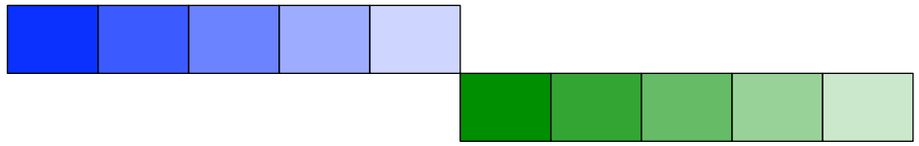


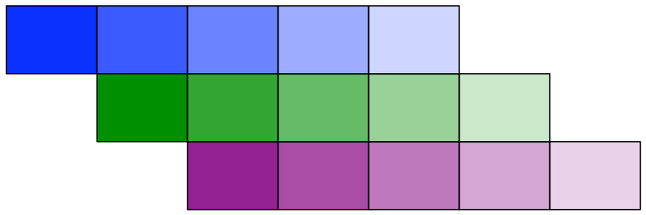
Cornell University

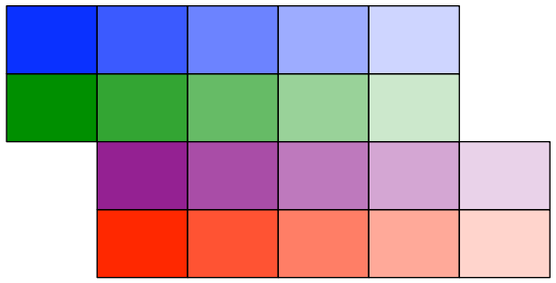
Source: Intel, IBM

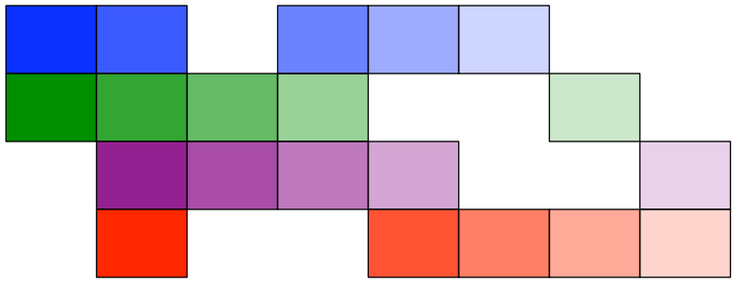
What did CPUs do with the transistors?

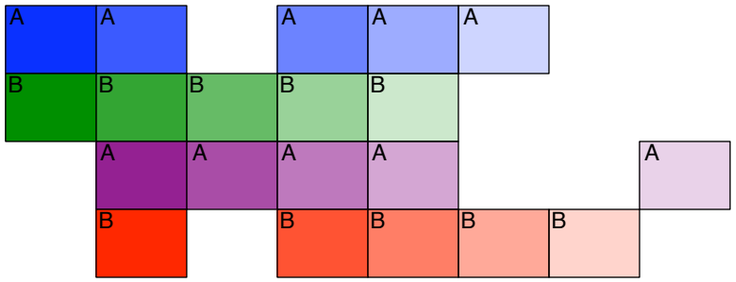
- Improved throughput while *preserving a sequential programming model*
  - ❖ Multi-cycle
  - ❖ Pipelined
  - ❖ Superscalar
  - ❖ Out-of-order
  - ❖ Multi-threaded





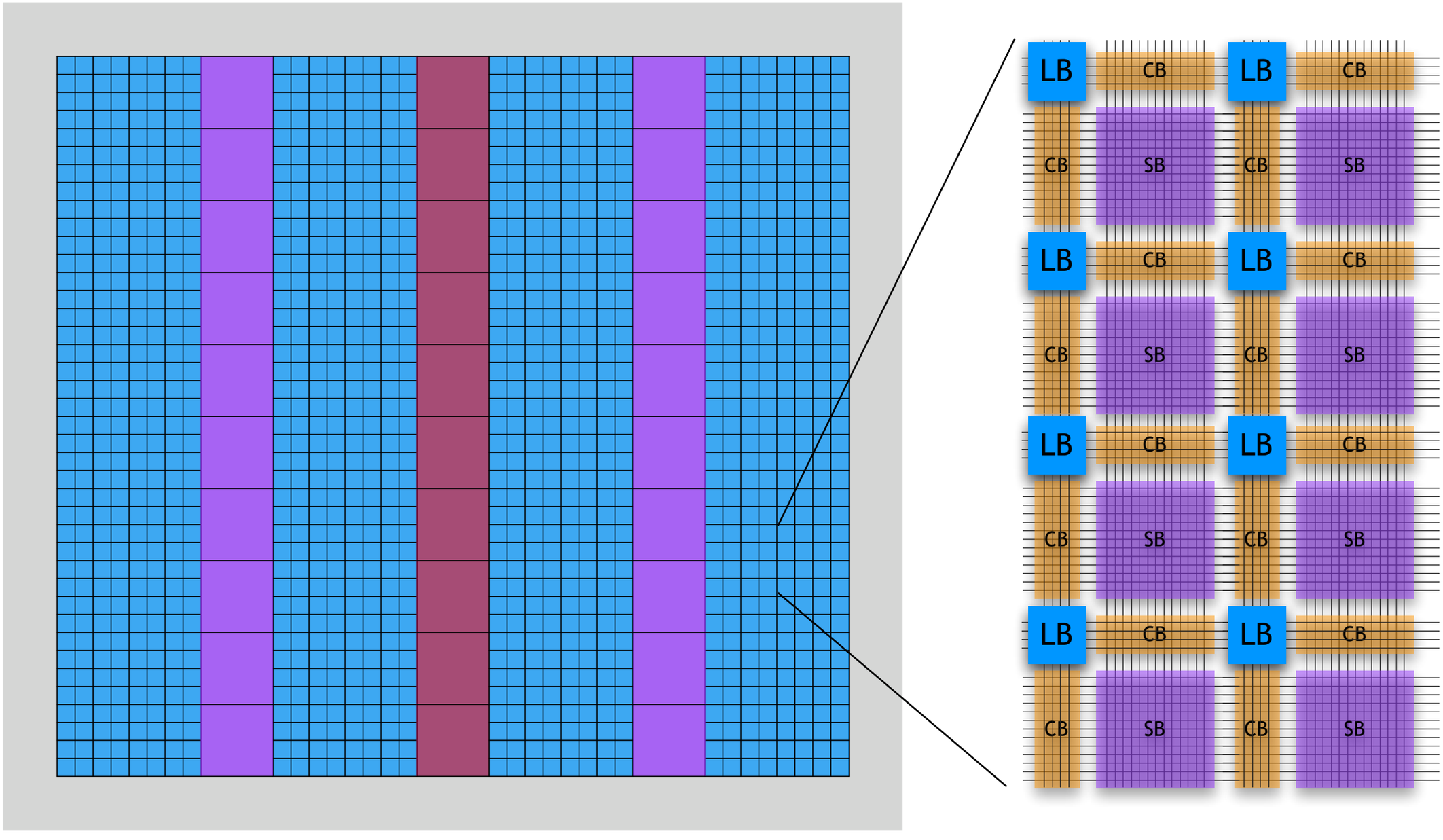




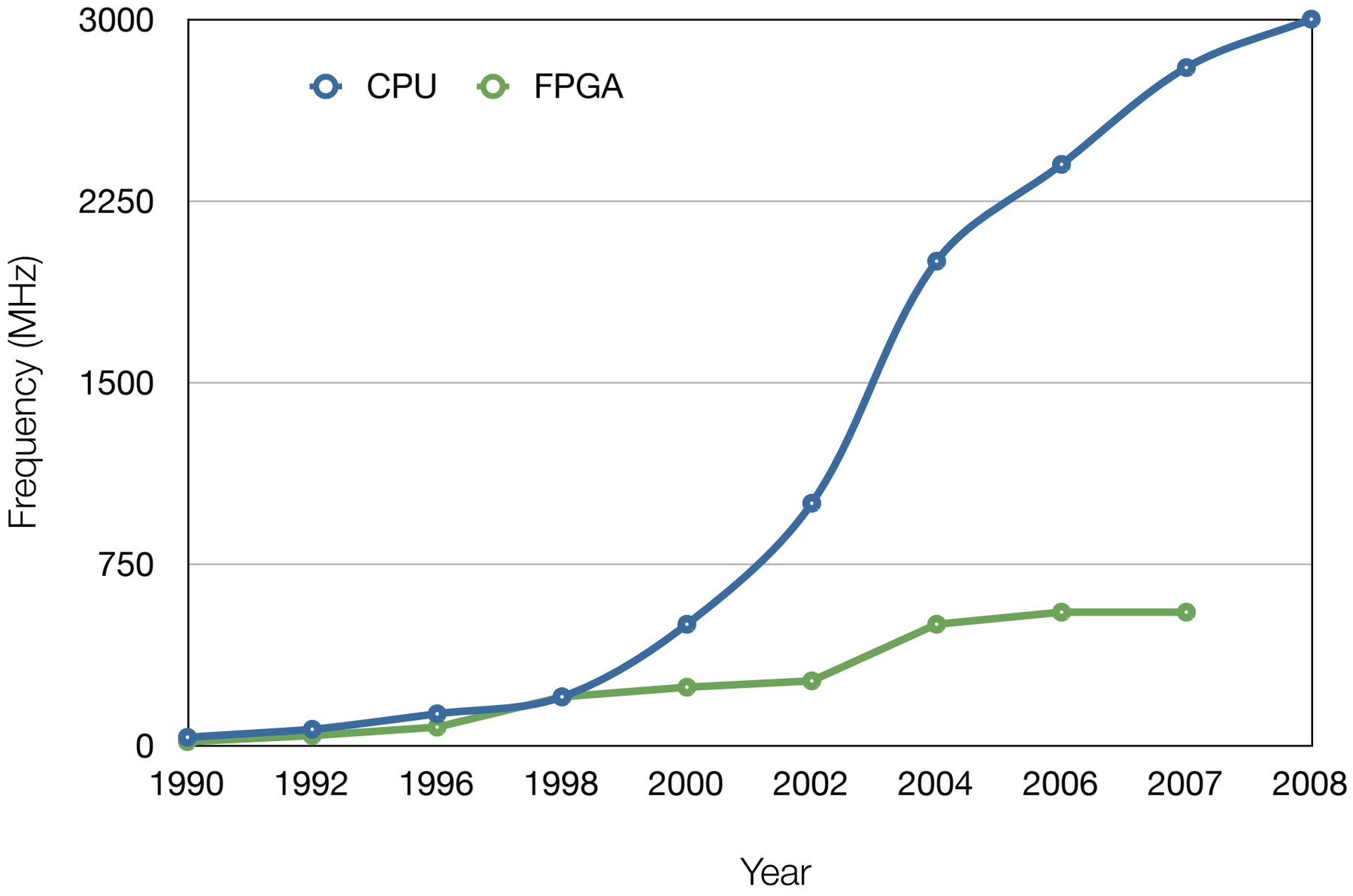


Today's supercomputers use commodity microprocessors as building-blocks

FPGA architectures



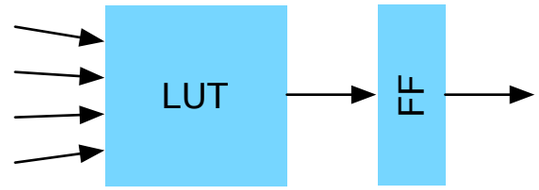
FPGA frequency scaling

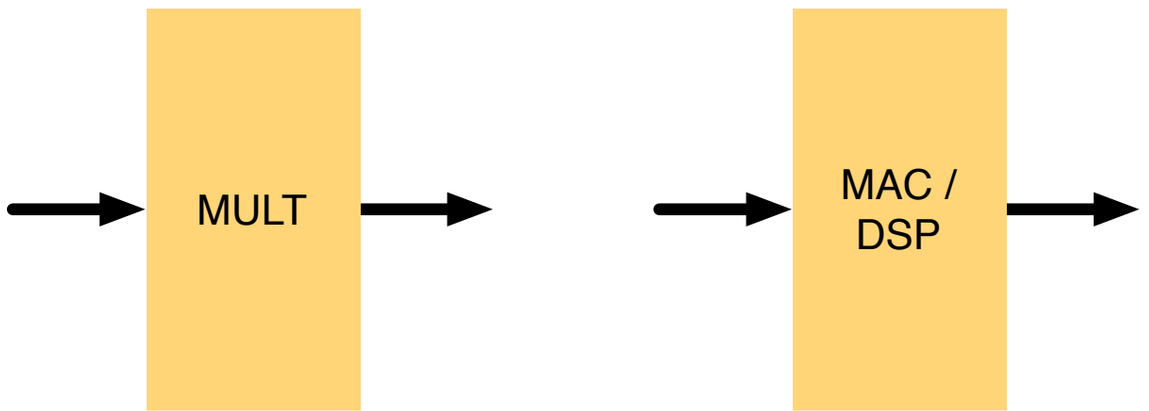


Source: Intel, IBM, Xilinx, Altera datasheets

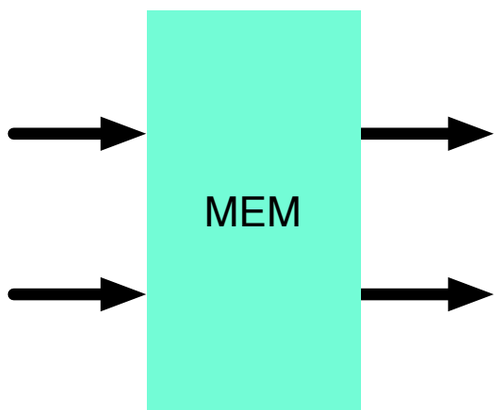
What did FPGAs do with the transistors?

- Basic lookup-table (LUT) / flip-flop (FF) / carry-chain
- Embedded multiplier
- Embedded memory
- Embedded processor
- DSP slices
- Hardened I/Os
- Larger LUT configurations





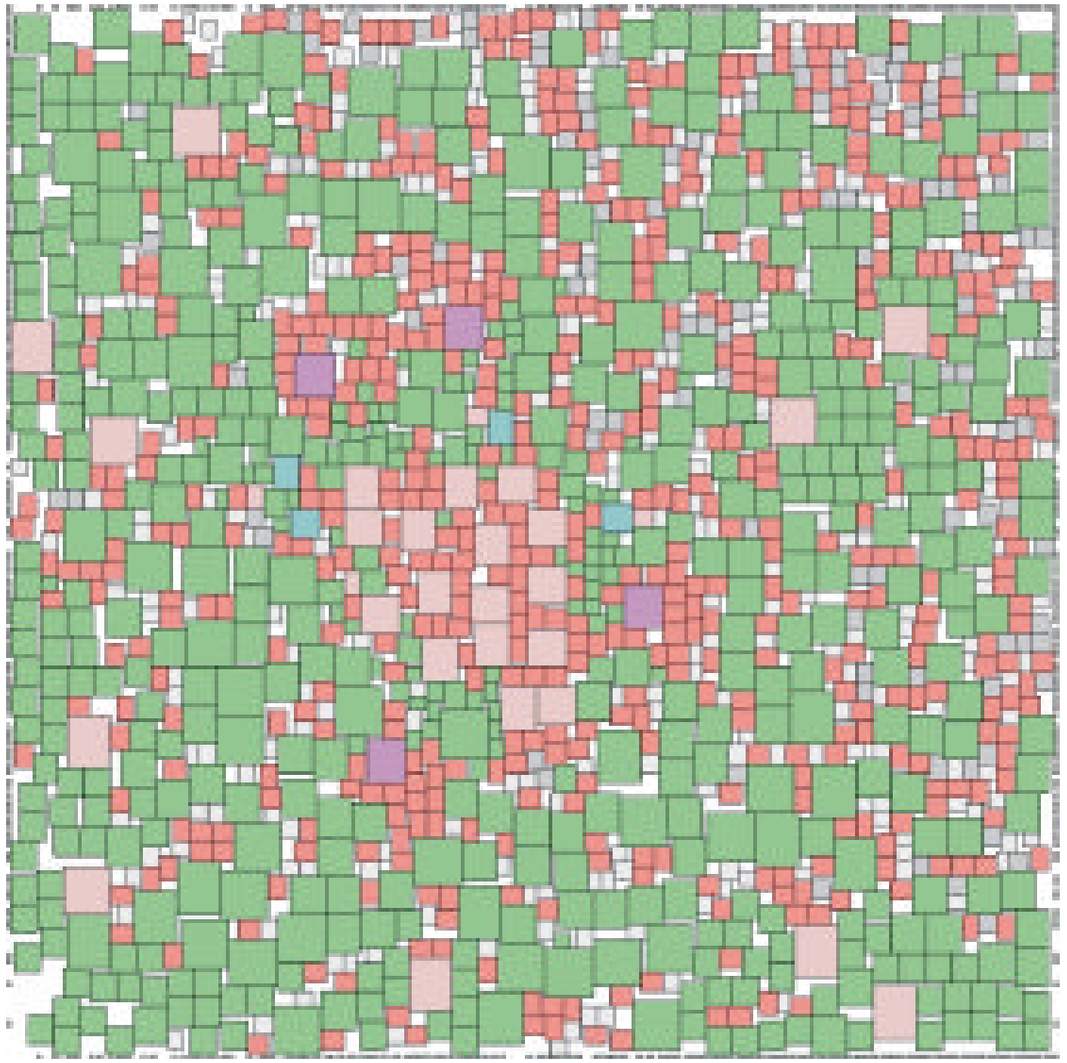
- ... more logic
  ... reduce logic depth



Seems simple compared to CPUs...

Routing versus logic

Type	Color
Buffer	Green
Config SRAM	Red
Mux	Pink
Switch	White/Grey
LUT	Purple
Flop	Blue



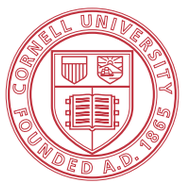
... so why not keep adding sophisticated logic blocks?

FPGAs are supposed to be general

- Hard macro versus more lookup-tables
  - ❖ How often is the macro used v/s loss of general logic functionality

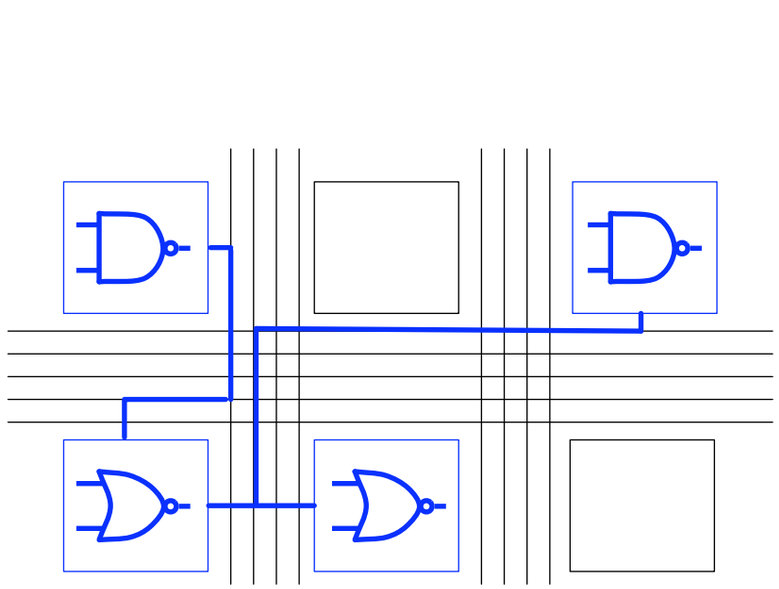
- It's about the tools
  - ❖ Can they determine how to *use* the macro block?
    ... automatically?
  - ❖ How useful is the function for a wide range of applications?
  - ❖ Will designers modify their RTL to accommodate it?

- Many attempts at this...
  - ❖ The curse of success: there is a *large* installed base of legacy RTL

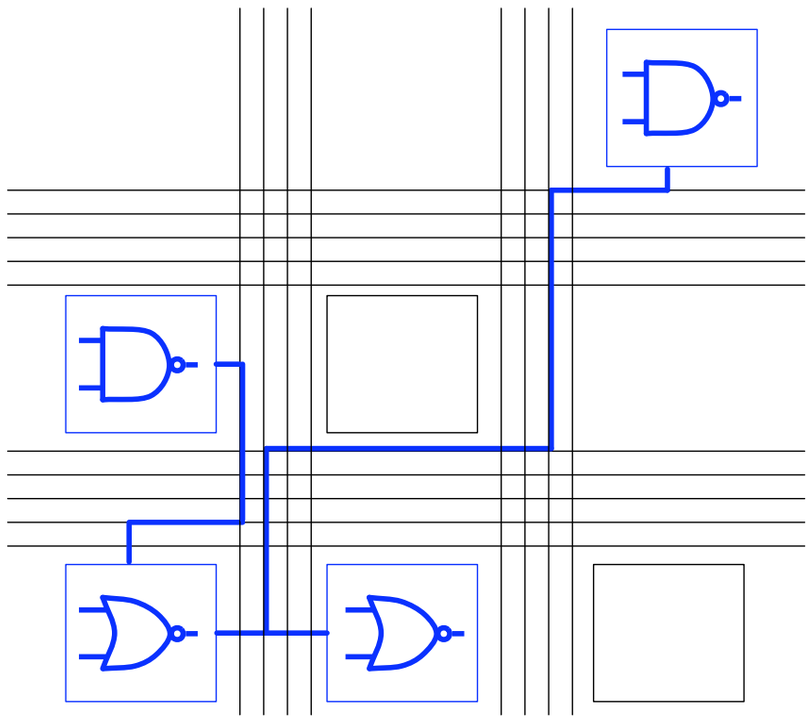


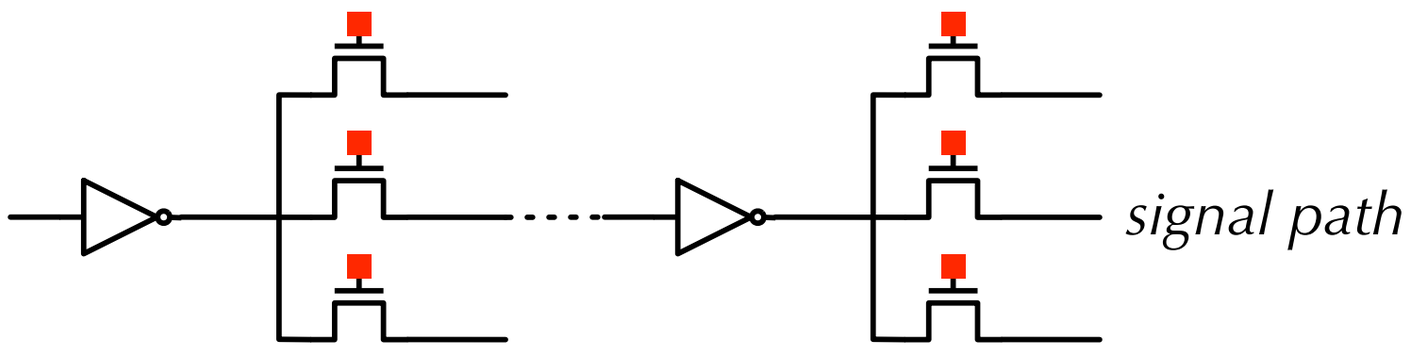
FPGA performance loss

- Imperfect placement and routing can create major performance issues
- Architecture looks “tiled” and regular
  ... but electrically, the architecture is *not* “tiled”



v/s

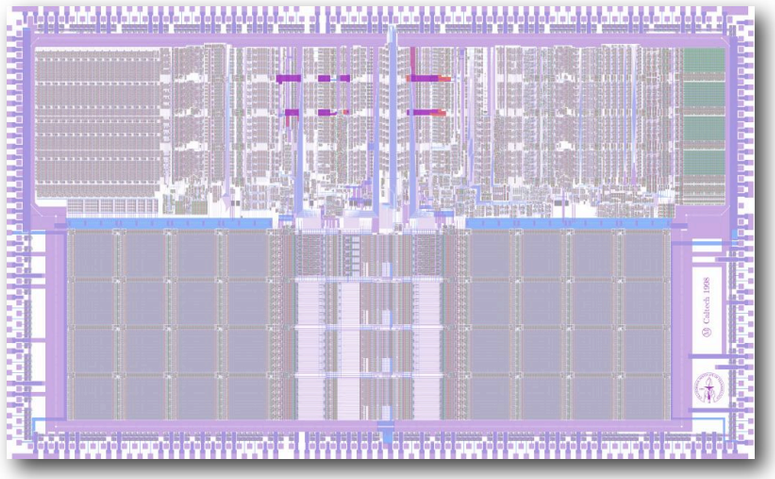




Increasing FPGA performance

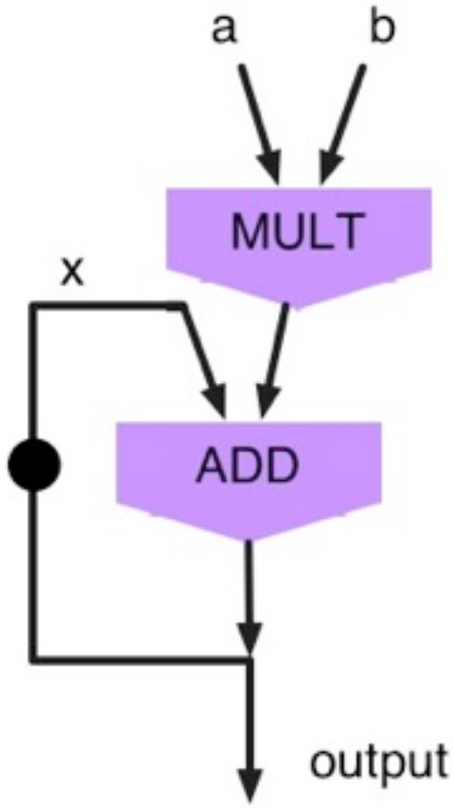
- Don’t map gates and wires, but *functionality*

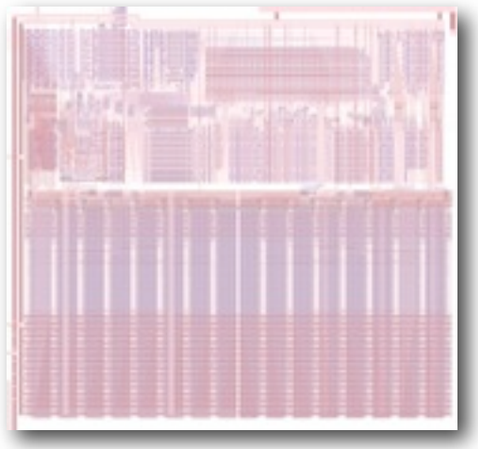
- Lessons from high-performance *asynchronous* logic
  - ❖ Standard circuit styles or “templates”
  - ❖ Data-driven computation: static dataflow



High-performance MIPS processor (1998)

- Use pipelined circuits for the asynchronous FPGA

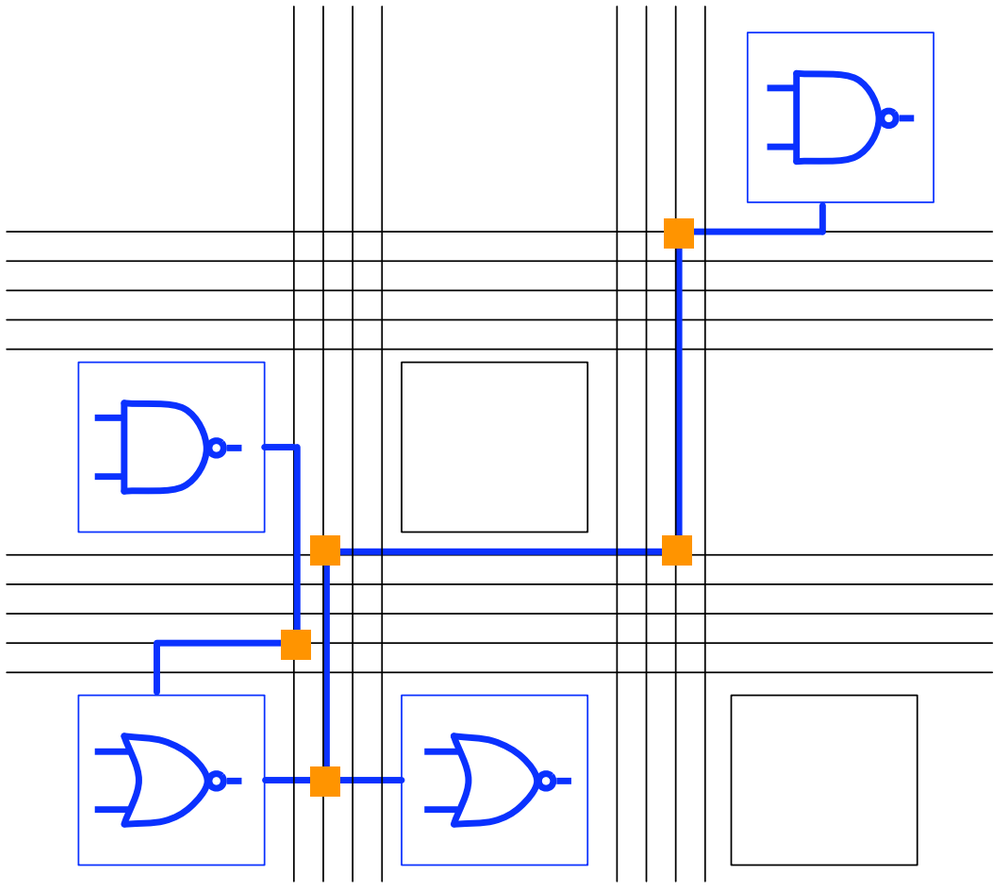


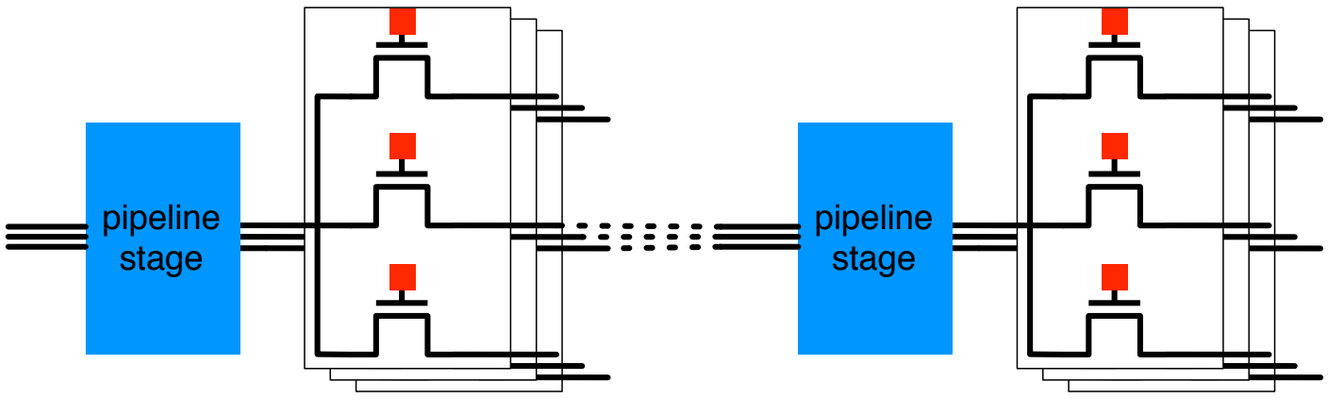


Ultra low power sensor network processor (2004)

An asynchronous FPGA architecture

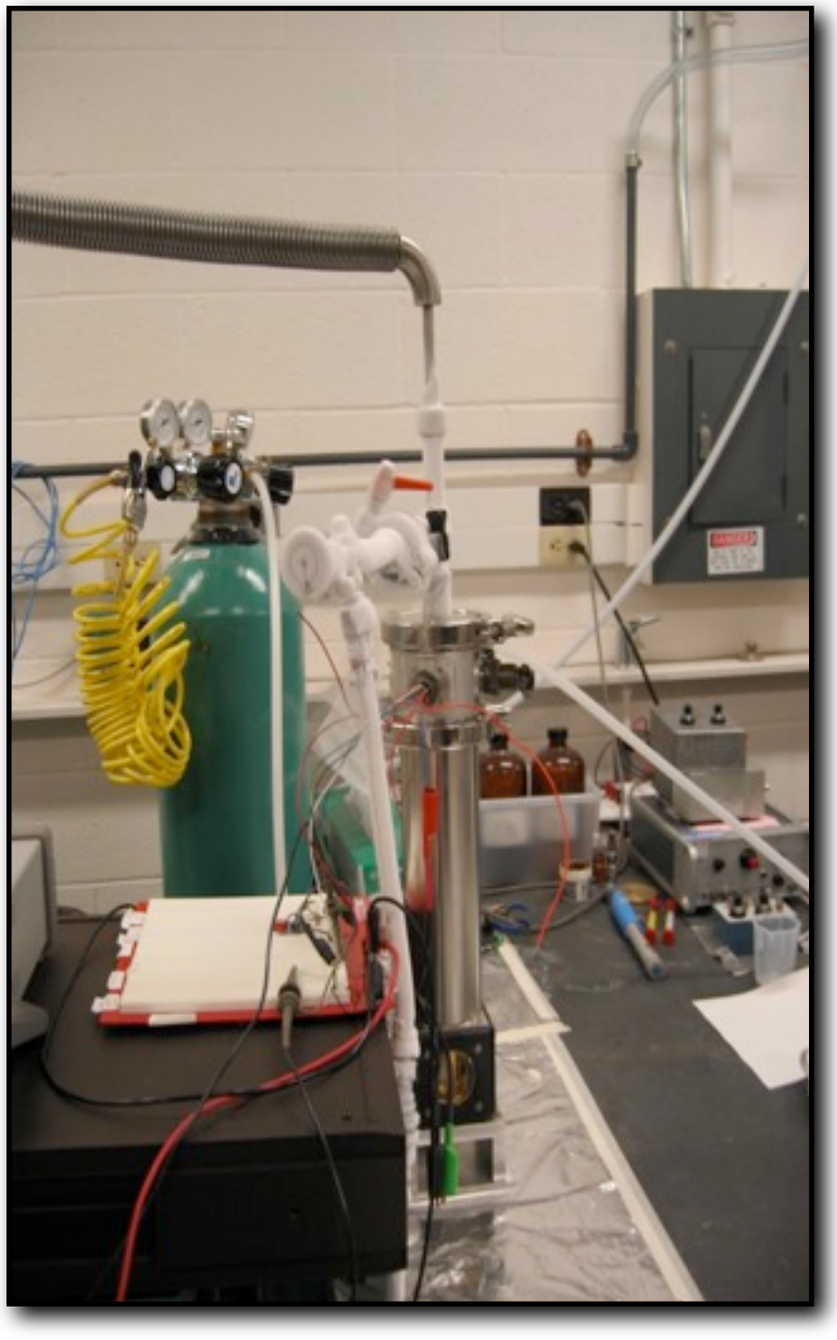
- Implement a dataflow FPGA with asynchronous logic

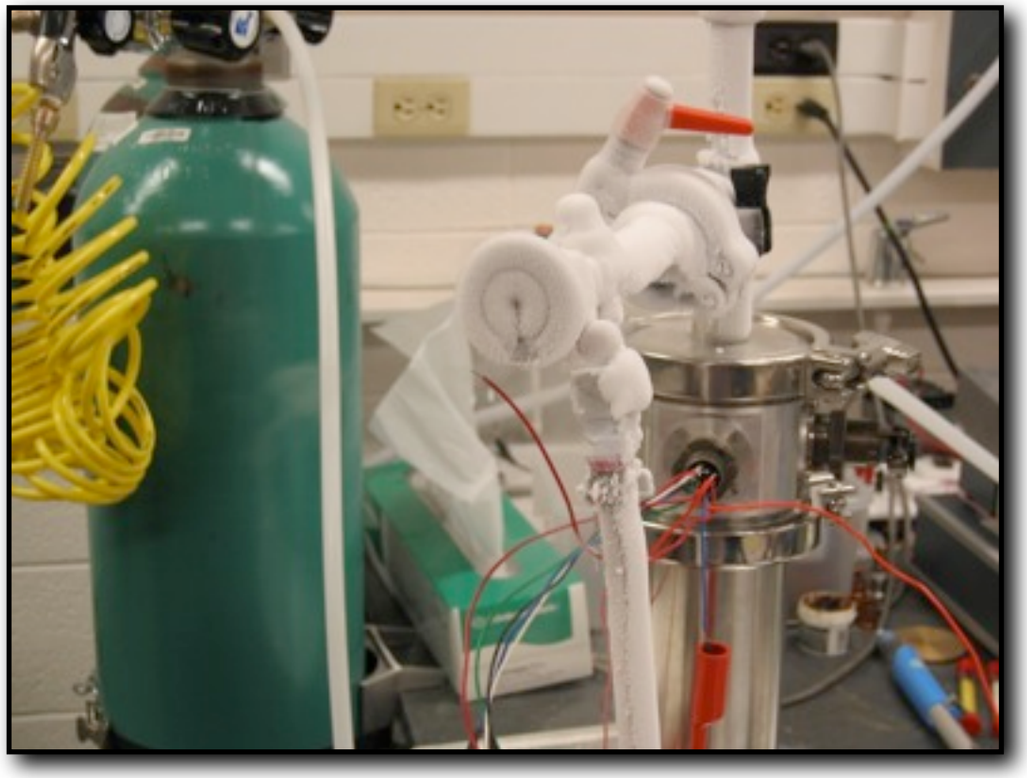


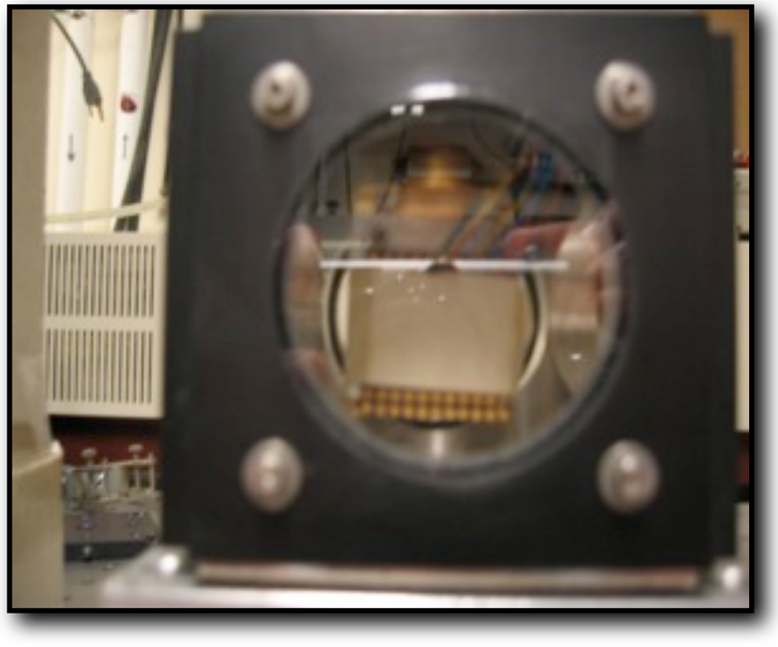


Pipelined dataflow

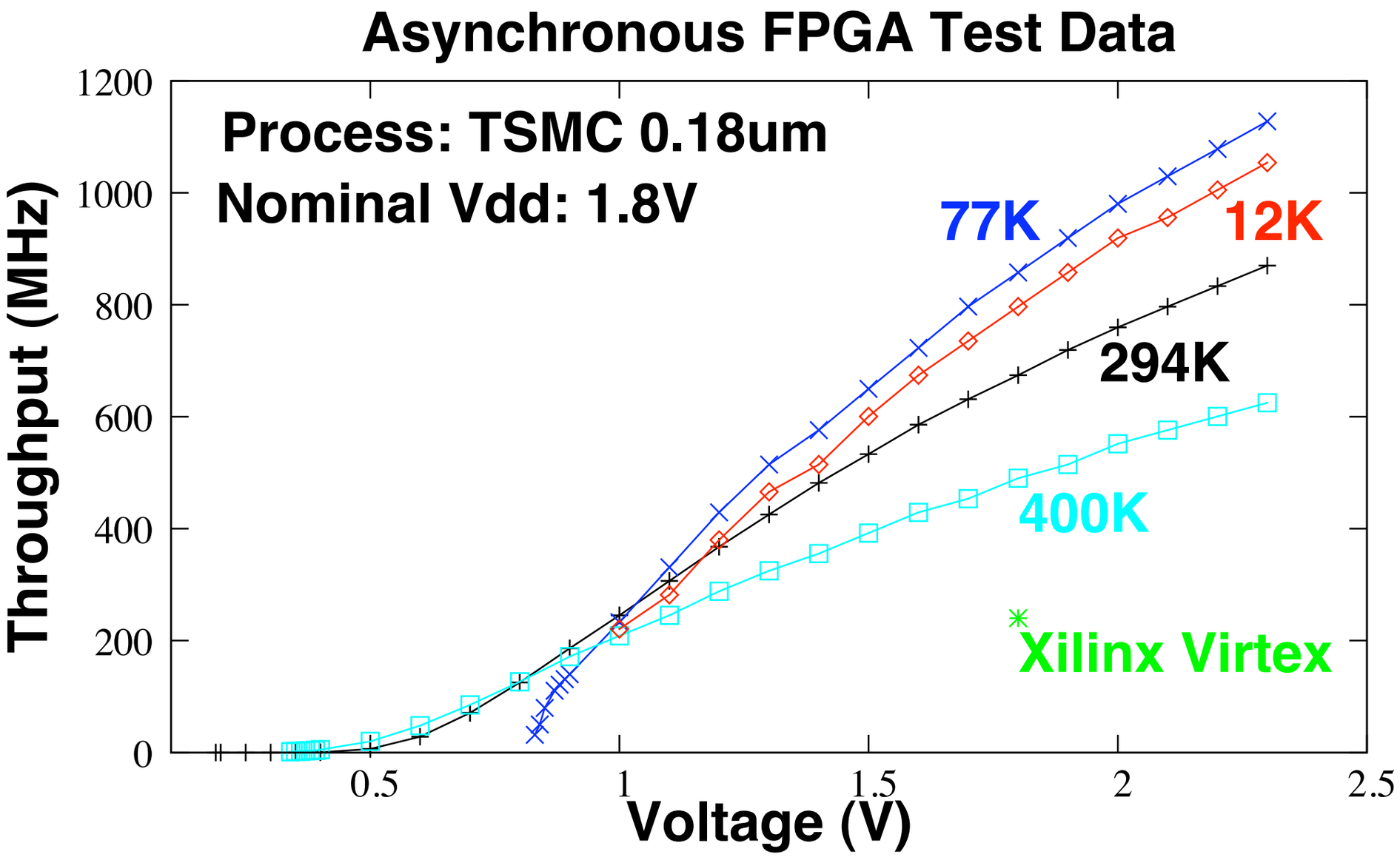
Fun with liquid Helium



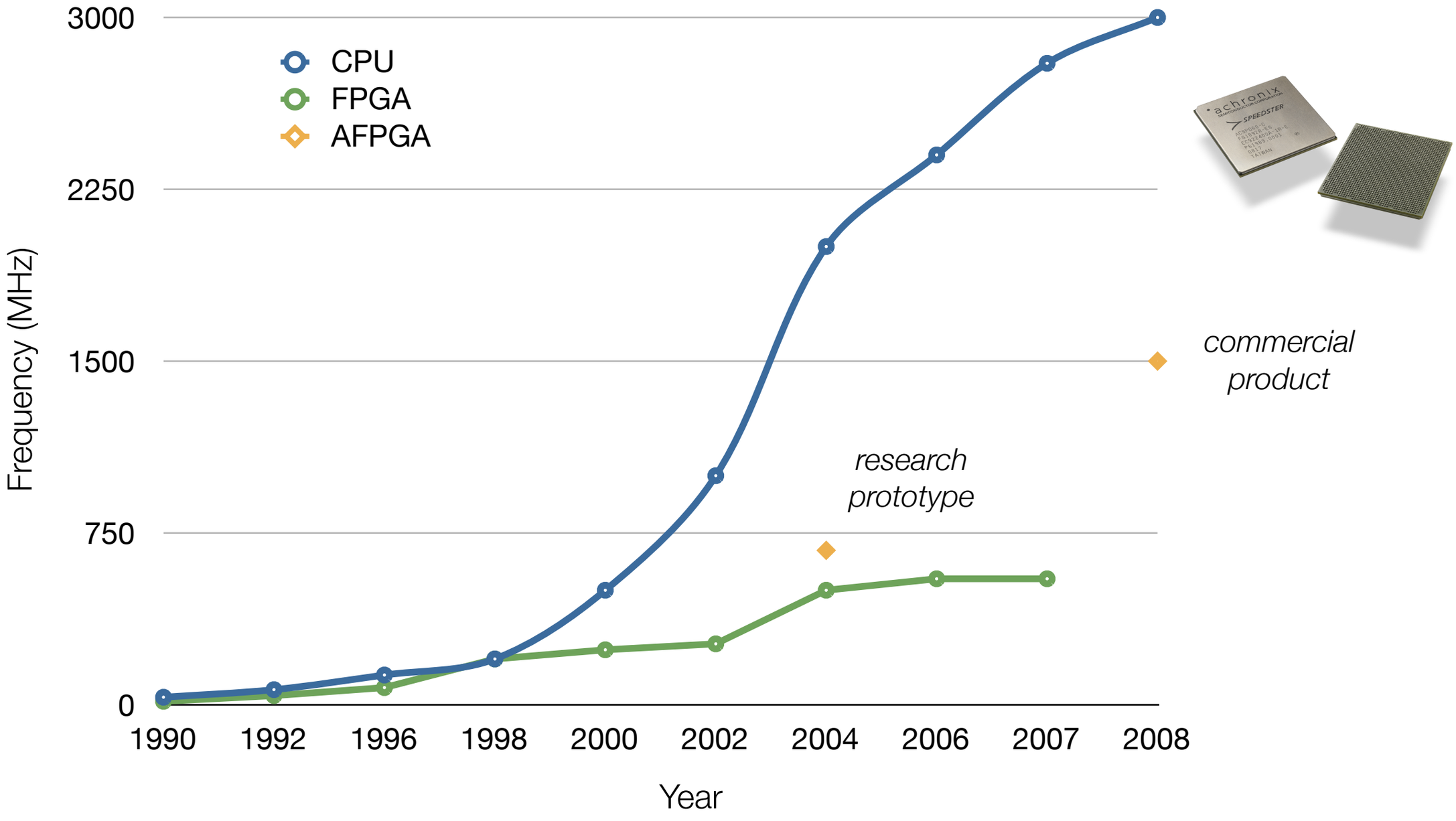




Results

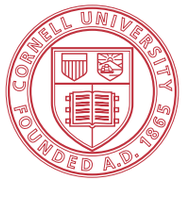

Asynchronous FPGA Test Data
Process: TSMC 0.18um
Nominal Vdd: 1.8V
77K
12K
294K
400K
Xilinx Virtex
Throughput (MHz)
Voltage (V)
0
200
400
600
800
1000
1200
0.5
1
1.5
2
2.5

FPGA frequency scaling



FPGA trends

- FPGAs can always use more transistors!
  - ❖ Area overhead v/s an ASIC is high
  - ❖ More transistors = larger designs possible
  - ❖ Leakage is the major issue
- Hard macros
  - ❖ Mainly for I/Os (e.g. memory controllers)
    - ▸ Avoid the tools issue!
  - ❖ “Core generator” that targets hard macros (e.g. DSP blocks)
    - ▸ Are there other “common” core generators / macro blocks?
- Another use for transistors: pipelined architecture
  - ❖ Eliminate global signals
  - ❖ Improve throughput
- Complex architectures seem unlikely



Summary

- FPGAs have a bright future!

- Differentiation into
  - ❖ Large LUT-count FPGAs
  - ❖ Low cost FPGAs (low LUT-count, low performance)
  - ❖ Medium LUT-count but high-performance FPGAs

- FPGAs can use all the transistors they can get