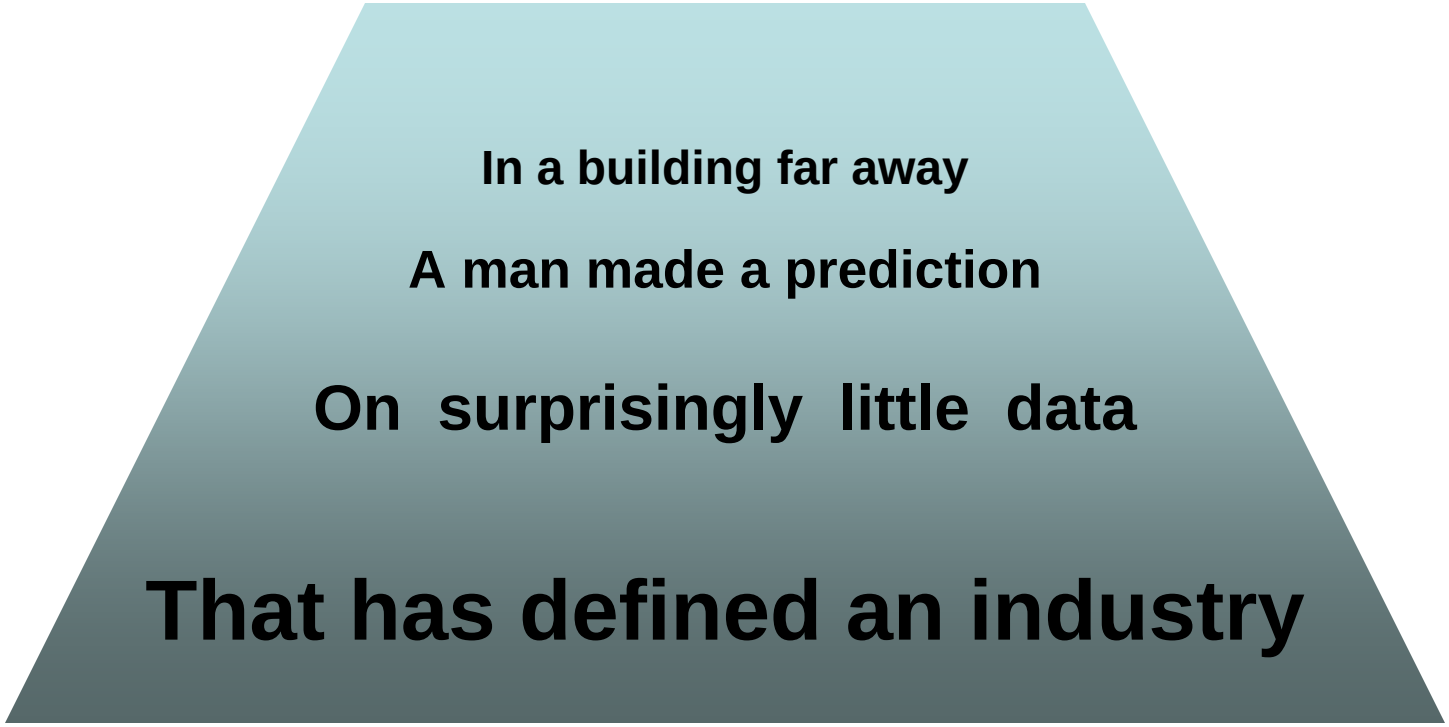
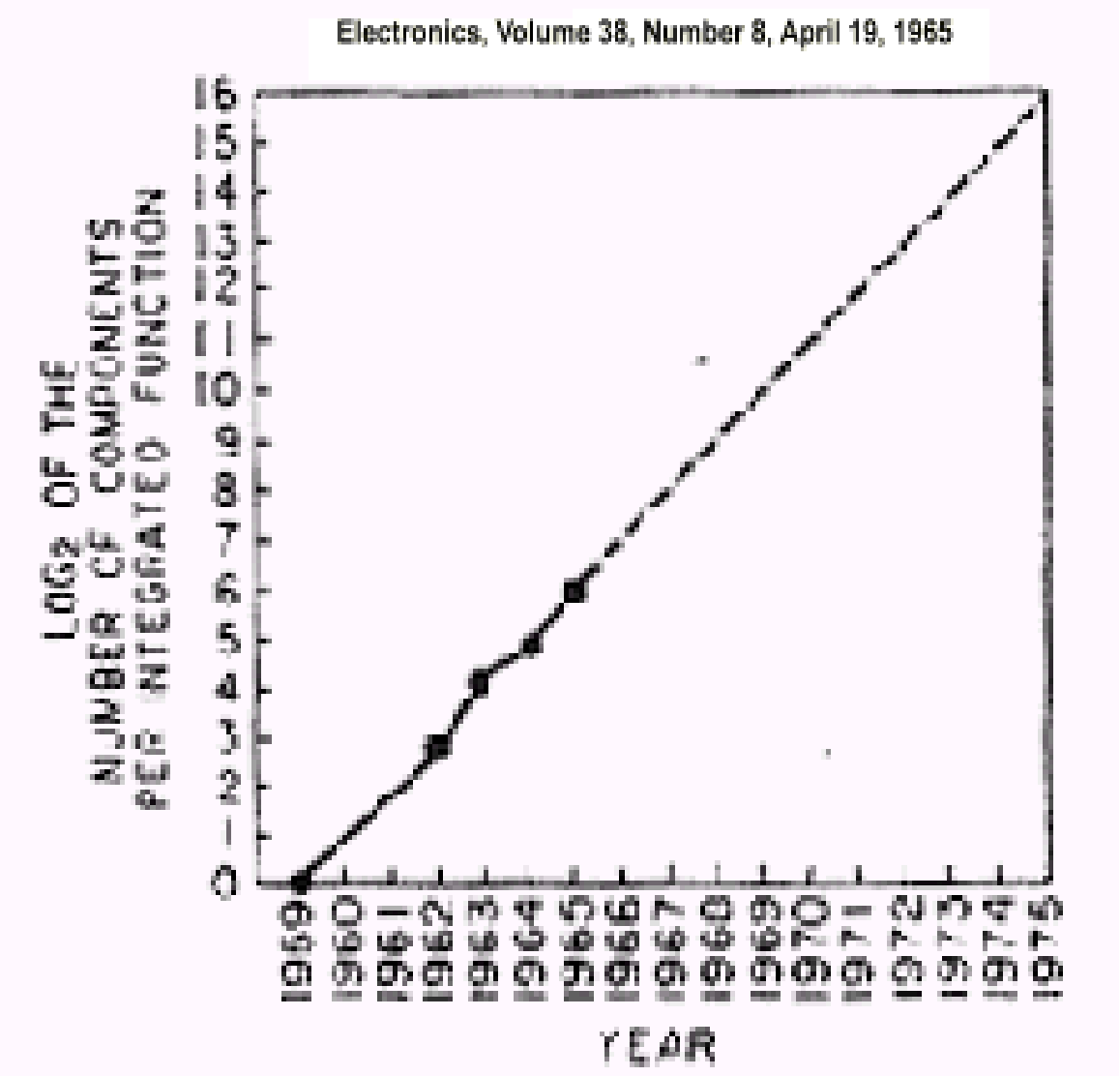
Scaling Power and the Future of CMOS

Mark Horowitz, EE/CS Stanford University

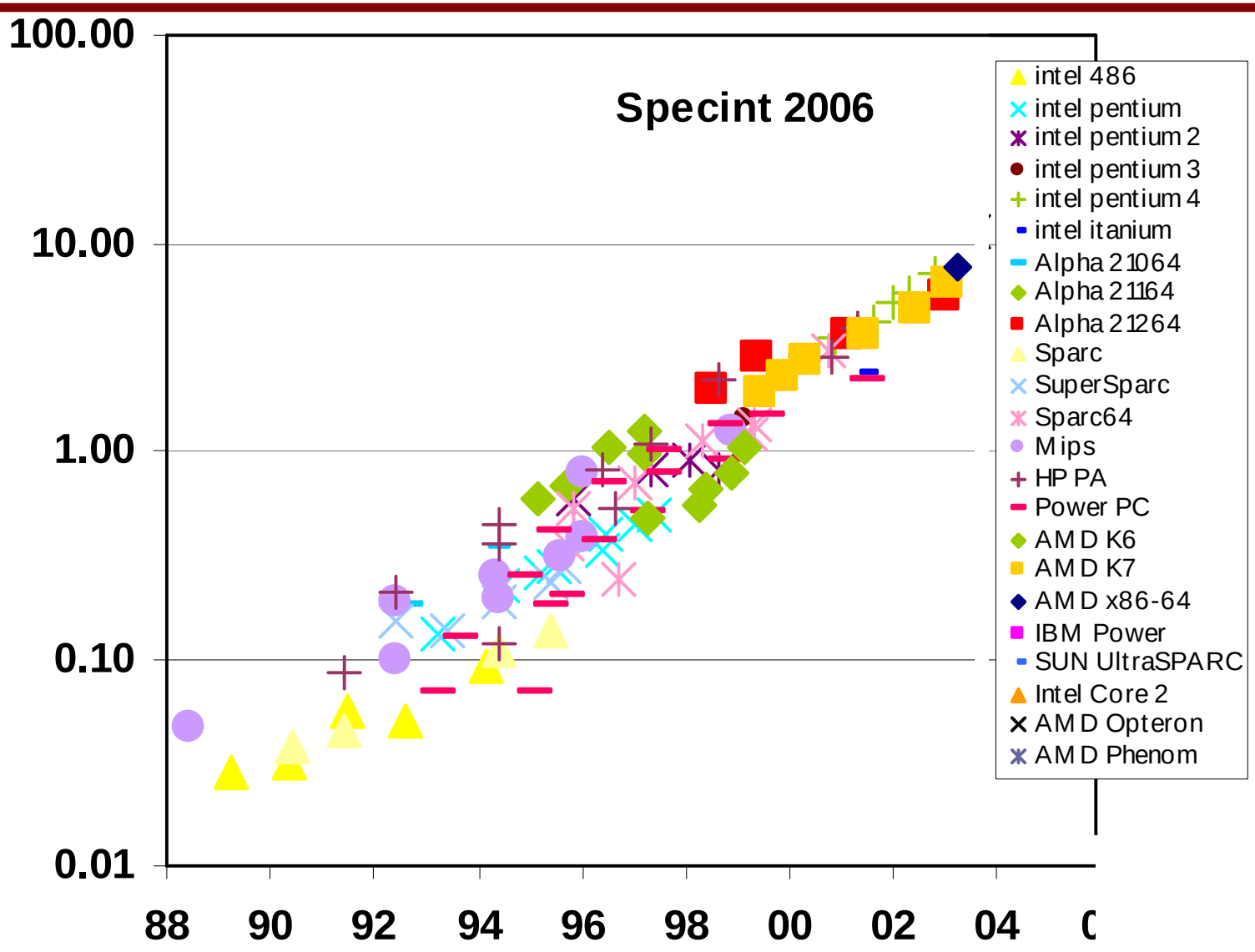
A Long Time Ago



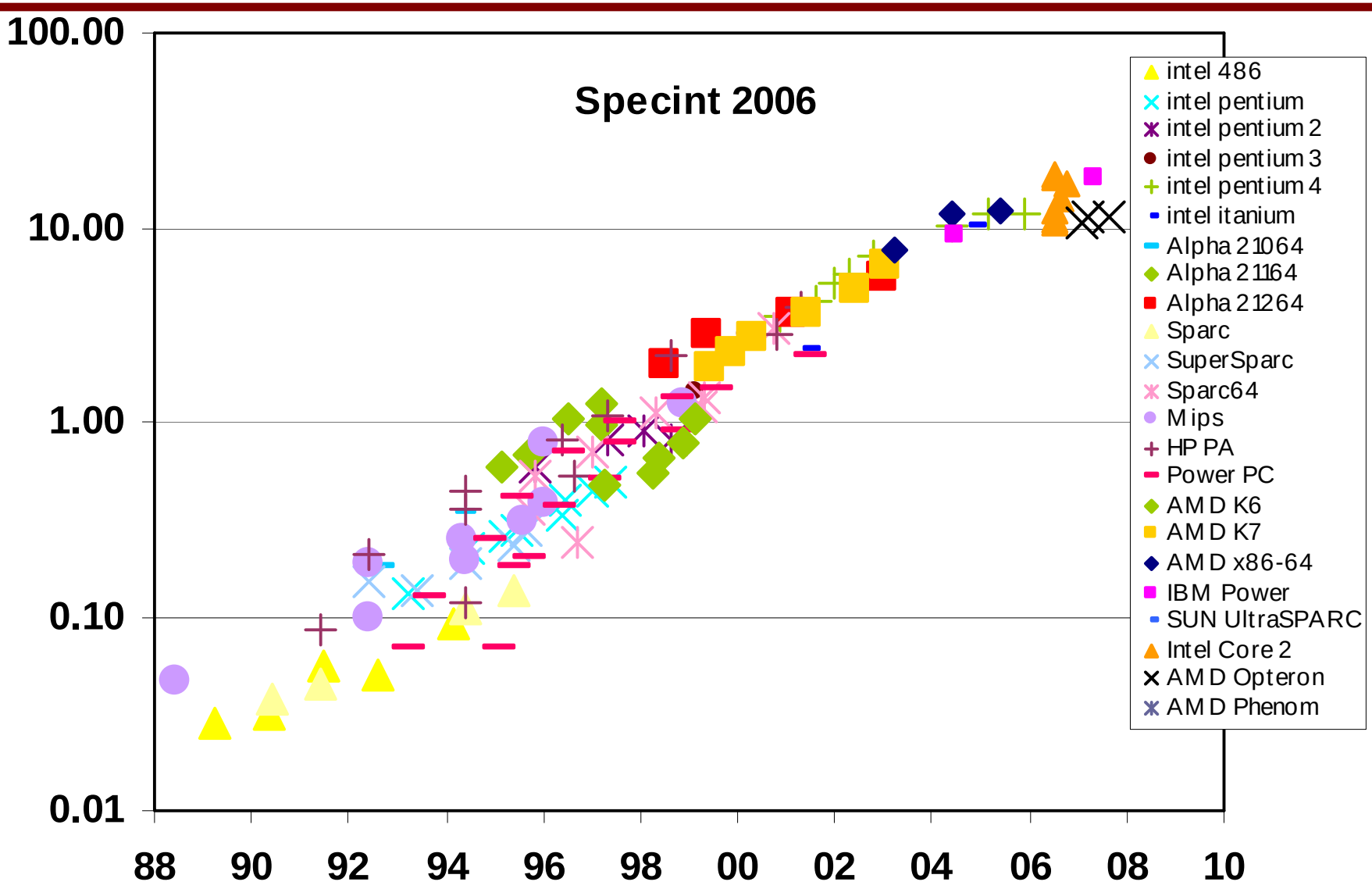
Moore's Law


Electronics, Volume 38, Number 8, April 19, 1965
LOG2 OF THE
NUMBER OF COMPONENTS
PER INTEGRATED FUNCTION
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
YEAR

CMOS Computer Performance



CMOS Computer Performance



Moore's Original Issues

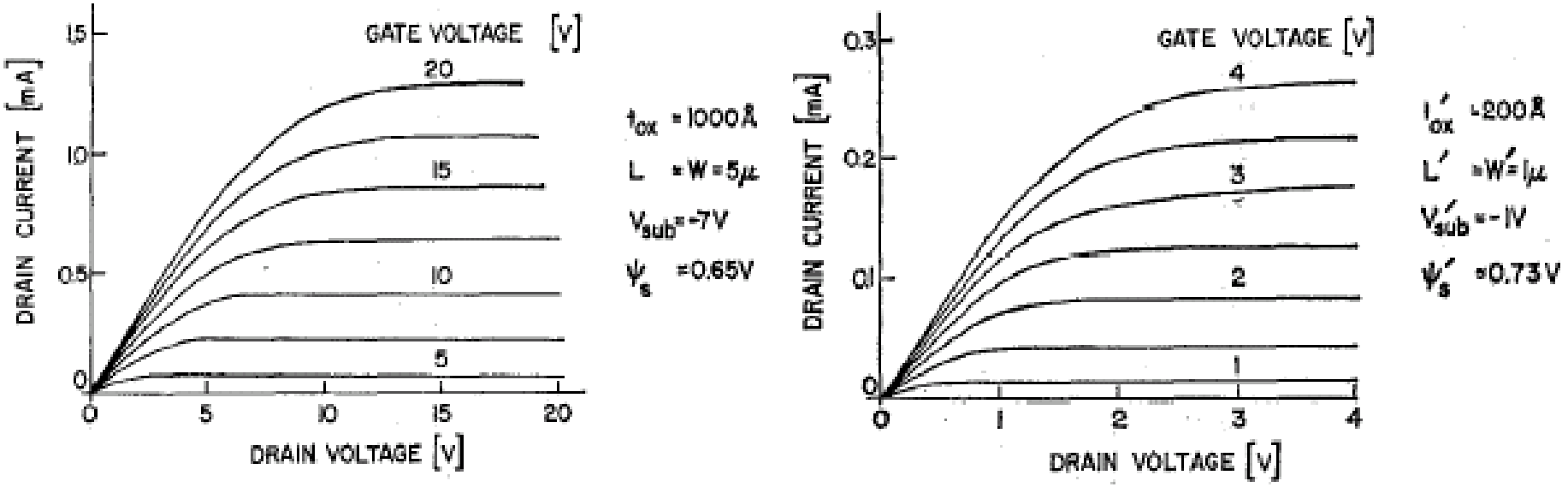
Design cost

Power dissipation

What to do with all the functionality possible

ftp://download.intel.com/research/silicon/moorespaper.pdf

Scaling MOS Devices

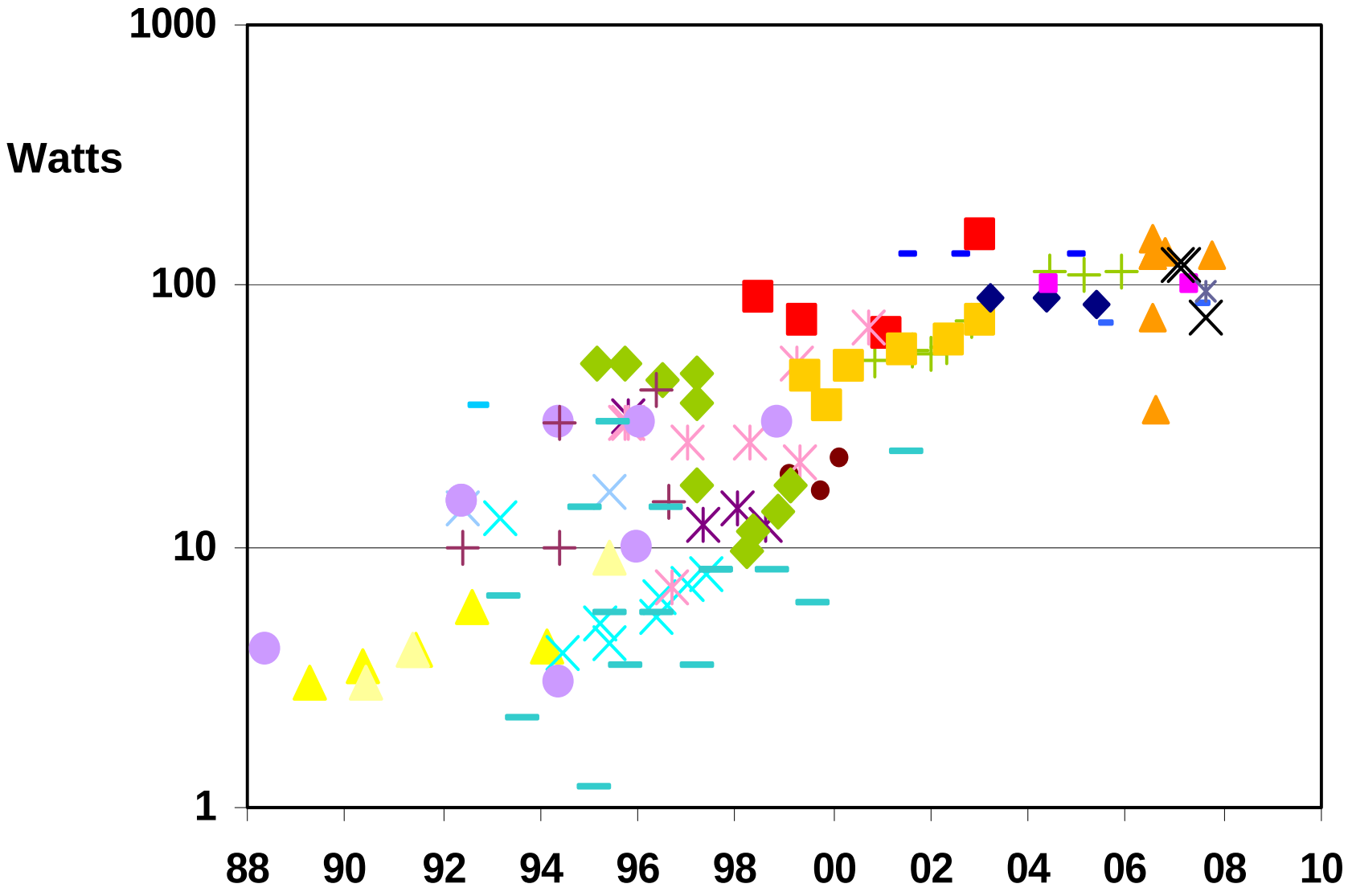


JSSC Oct 74, pg 256

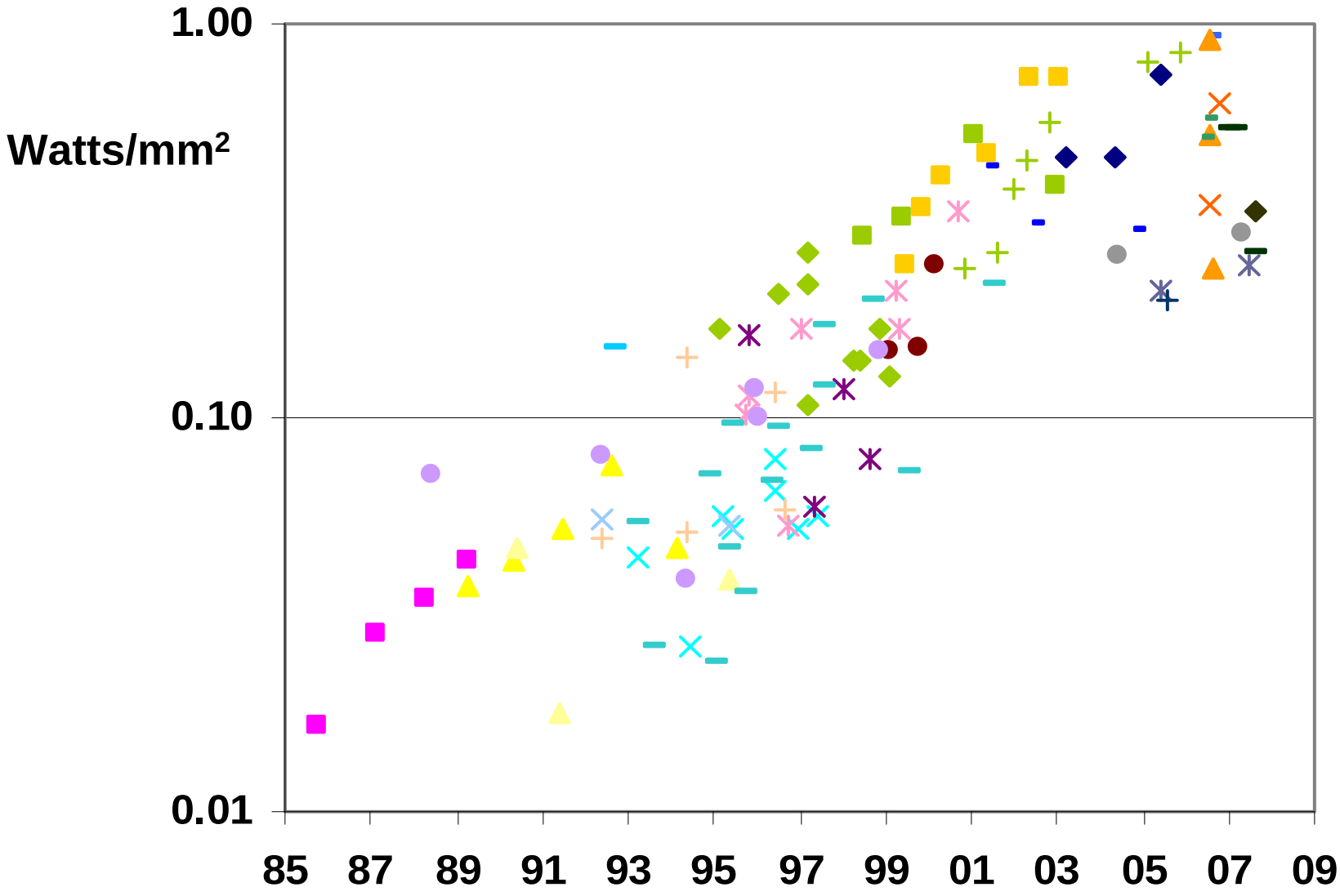
In this ideal scaling

- V scales to αV, L scales to αL
- So C scales to αC, i scales to αi (i/μ is stable)
- Delay = CV/I scales as α
- Energy = CV^2 scales as α^3

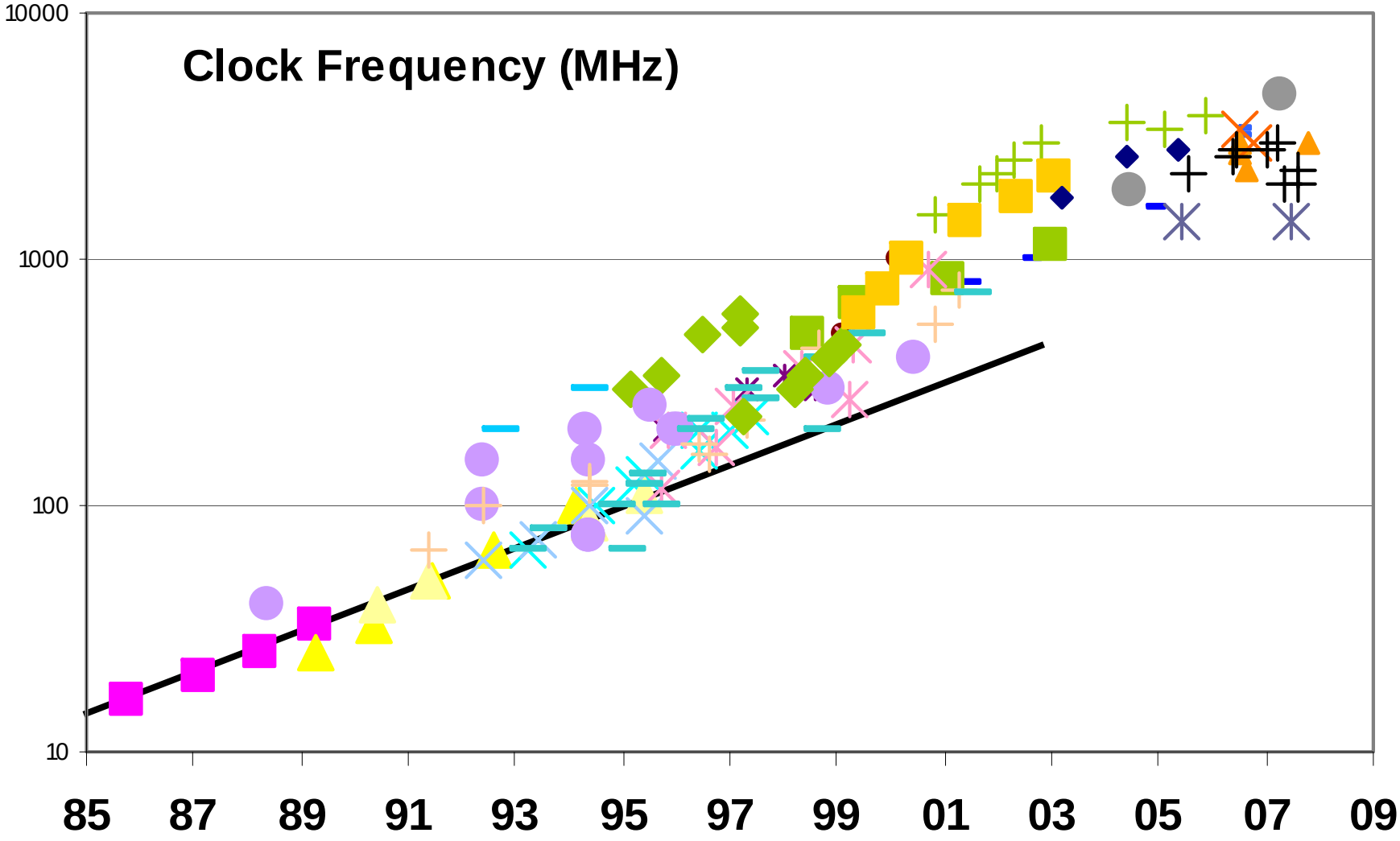
Processor Power



Power Density

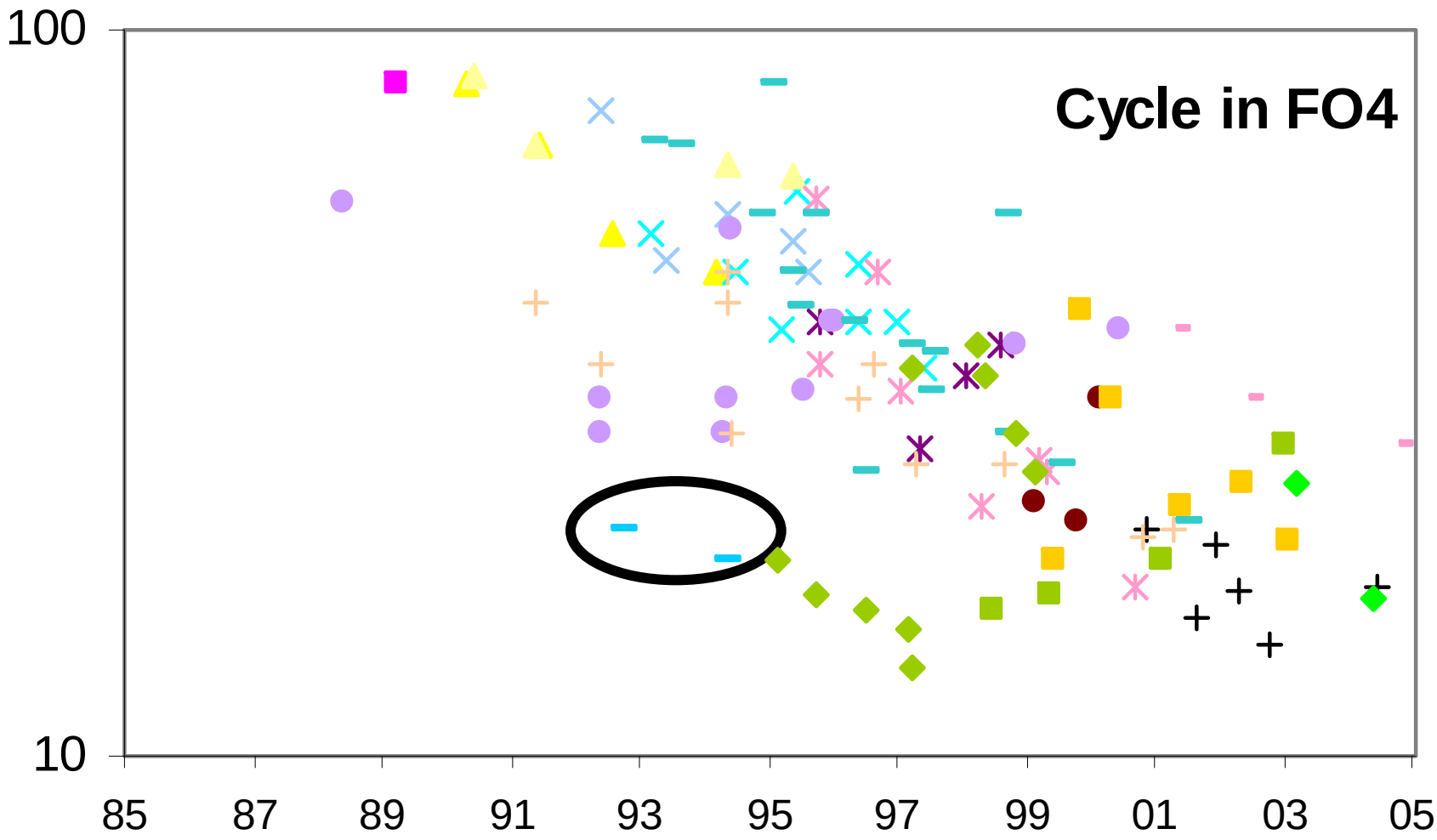


Why Power Increased



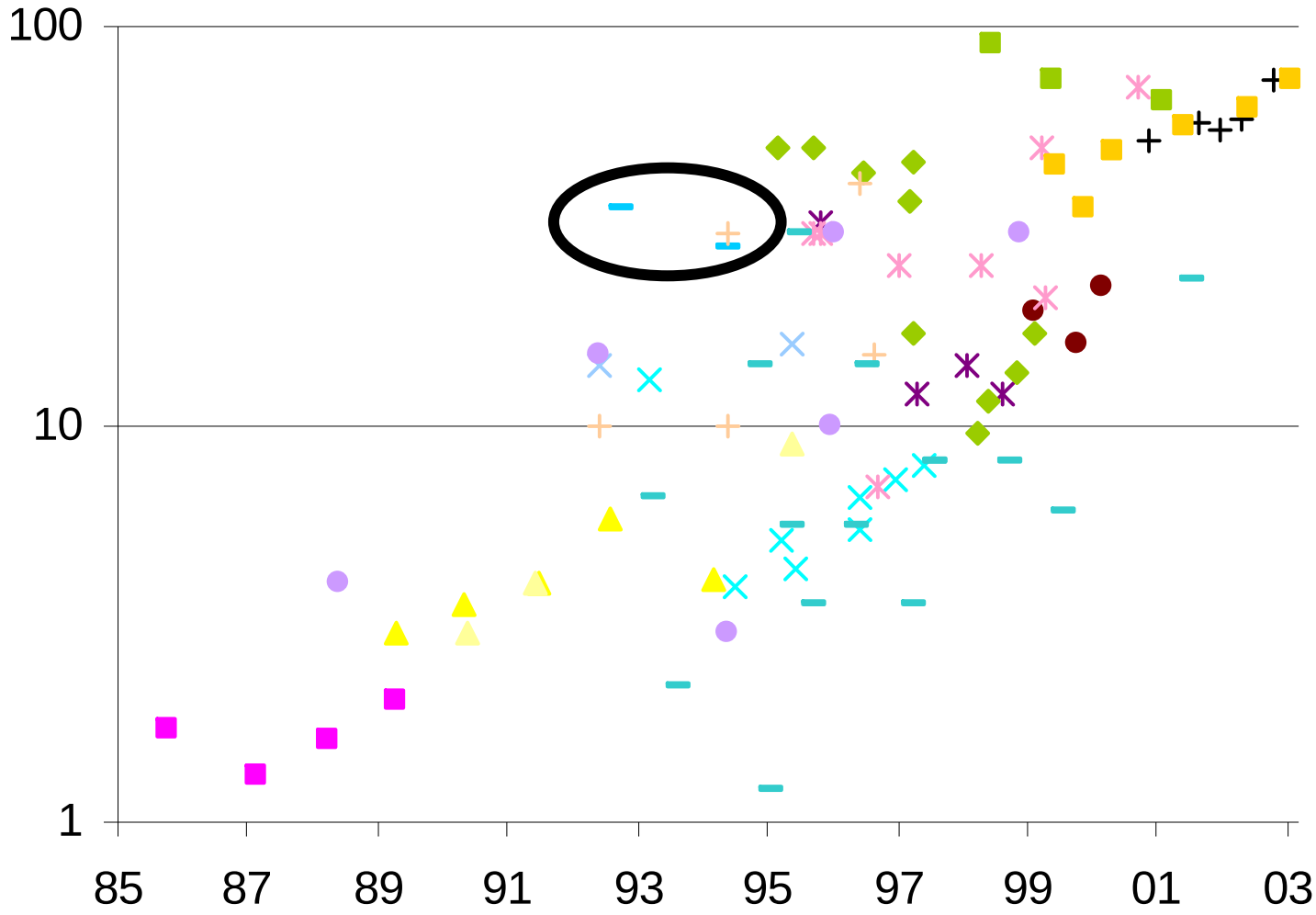
Good News

Die growth & super frequency scaling have stopped



Processor Power

They were high power too



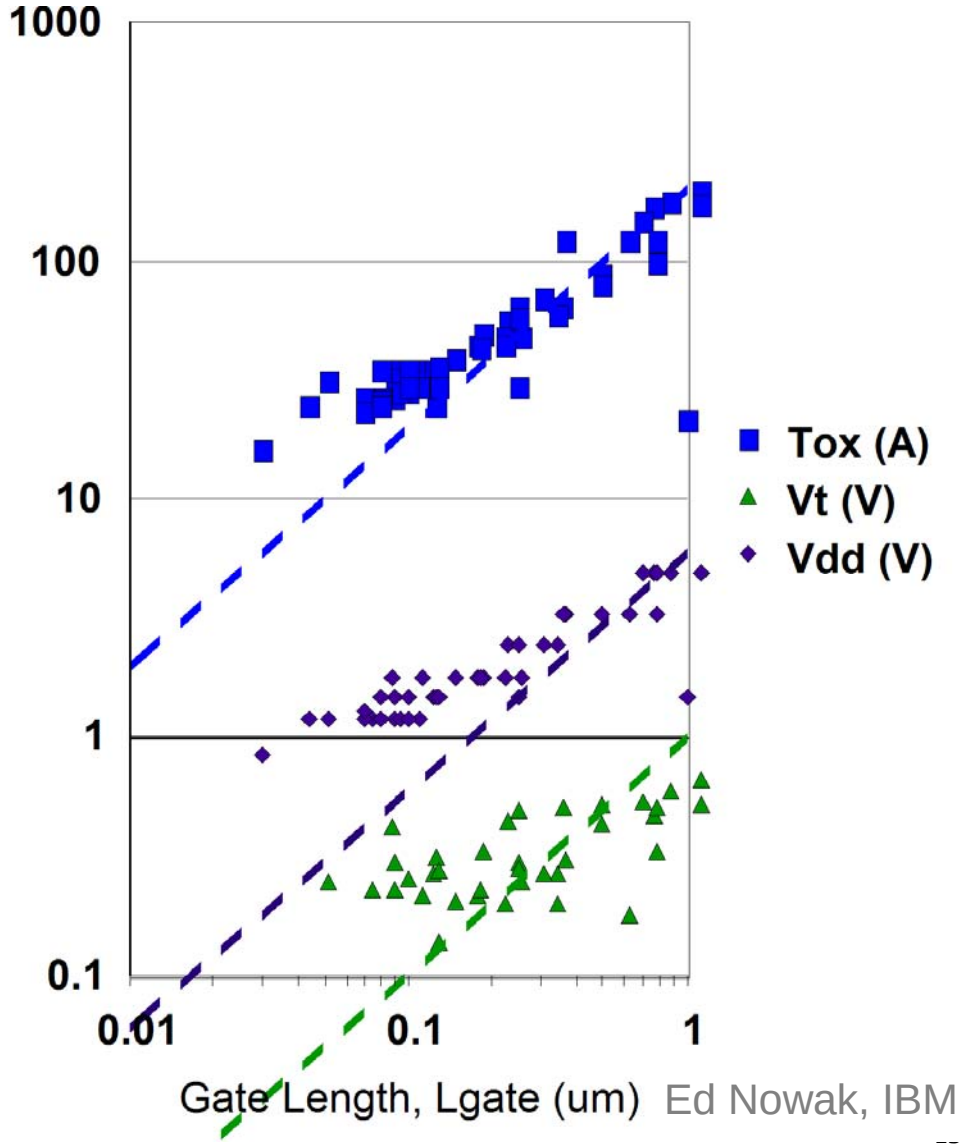
Bad News

Voltage scaling has stopped as well

- kT/q does not scale
- Vth scaling has power consequences

If Vdd does not scale

- Energy scales slowly



Technology Scaling Today

Device sizes are still scaling

- Cost/device is still scaling down
- This is what is driving scaling

Voltages are not scaling very fast

- Threshold voltages set by leakage
- Gate oxide thickness is set by leakage

  This means that the channel lengths are not scaling

  Current is increasing by stressing silicon

Now Vdd and Vth are set by optimization

Other Technologies

For computing, I am not optimistic

Current problems are set by Physics:

- Vdd set by kT/q

  Sets the on-off ratio
- Wire energy by $CVdd^2$

To get around these limitations

- Need to create something very different!

Problem with Different Technologies

Design processes have been optimized for silicon

- Working on making it better for over 30 years

Silicon has set:

- Notions of logic (binary signals), digital design styles
- Computing (distinct memory and logic)
- Relative size and speed of memory logic

No new technology will fit this mold well

- Changing the world is hard
- If you build it, generally they don't come

  Unless they absolutely have to

Maturing of Silicon

Silicon will not disappear

- It will still be a huge business

  Growth rate is slower, Eventually very slow scaling

Silicon will become like concrete and steel

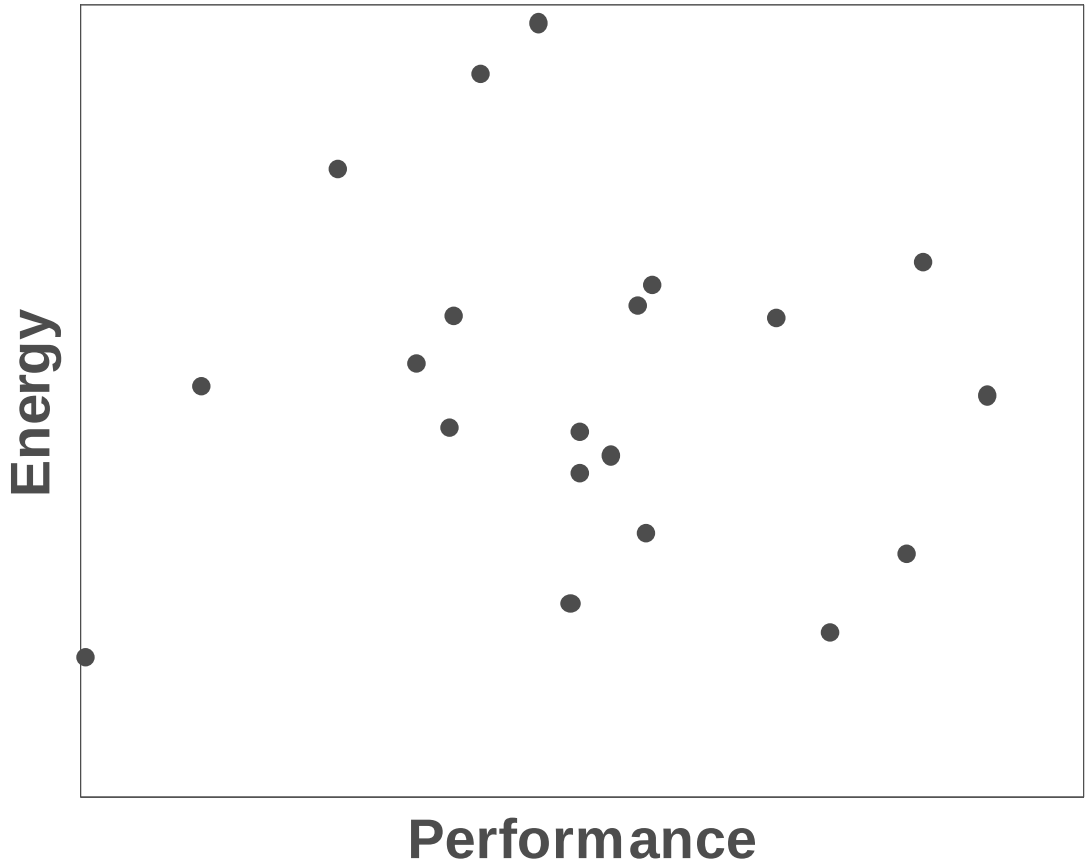
- Basis of a huge industry
- Critical to nearly everything
- But fairly stable and predictable

Will remain the dominate substrate for computing

- And performance be limited by power dissipation

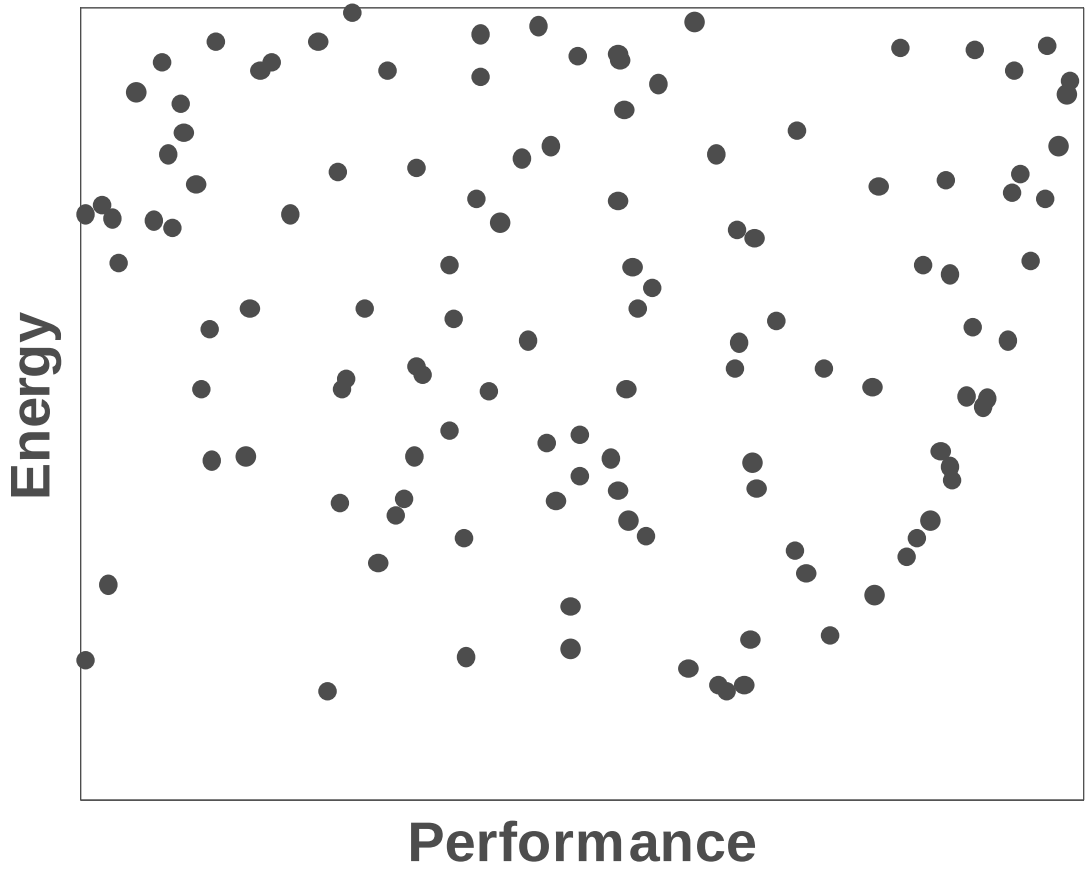
Optimizing Energy

Every design is a point on a 2-D plane



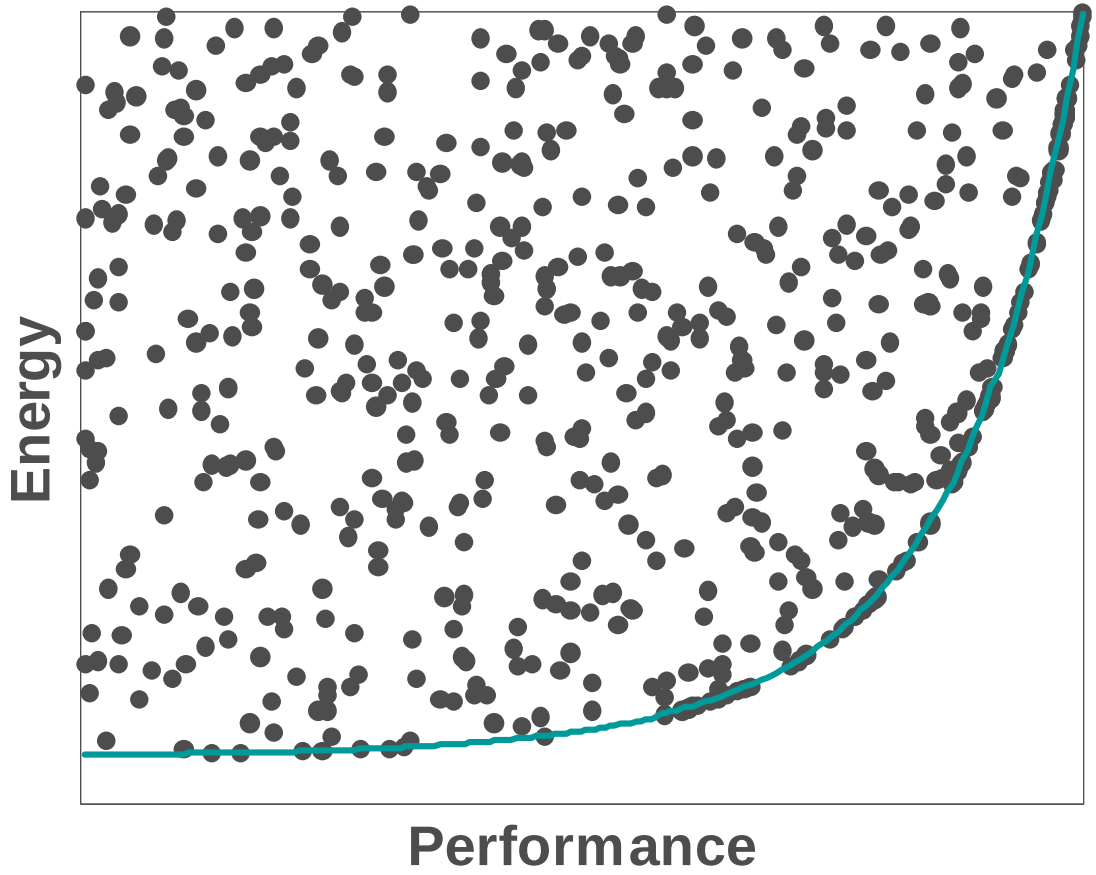
Optimizing Energy

Every design is a point on a 2-D plane



Optimizing Energy

Every design is a point on a 2-D plane



Years of Low Power Research …

Shown only one design technique to reduce power

- Reduce waste

Can waste

- Energy (clock gating, leakage control, etc)
- Performance

  Adding additional constraints to operation flow

If technology scaling has stalled

- Need to focus on reducing waste in our systems

Increase in efficiency in our designs will set performance

Future Systems

Some simple math

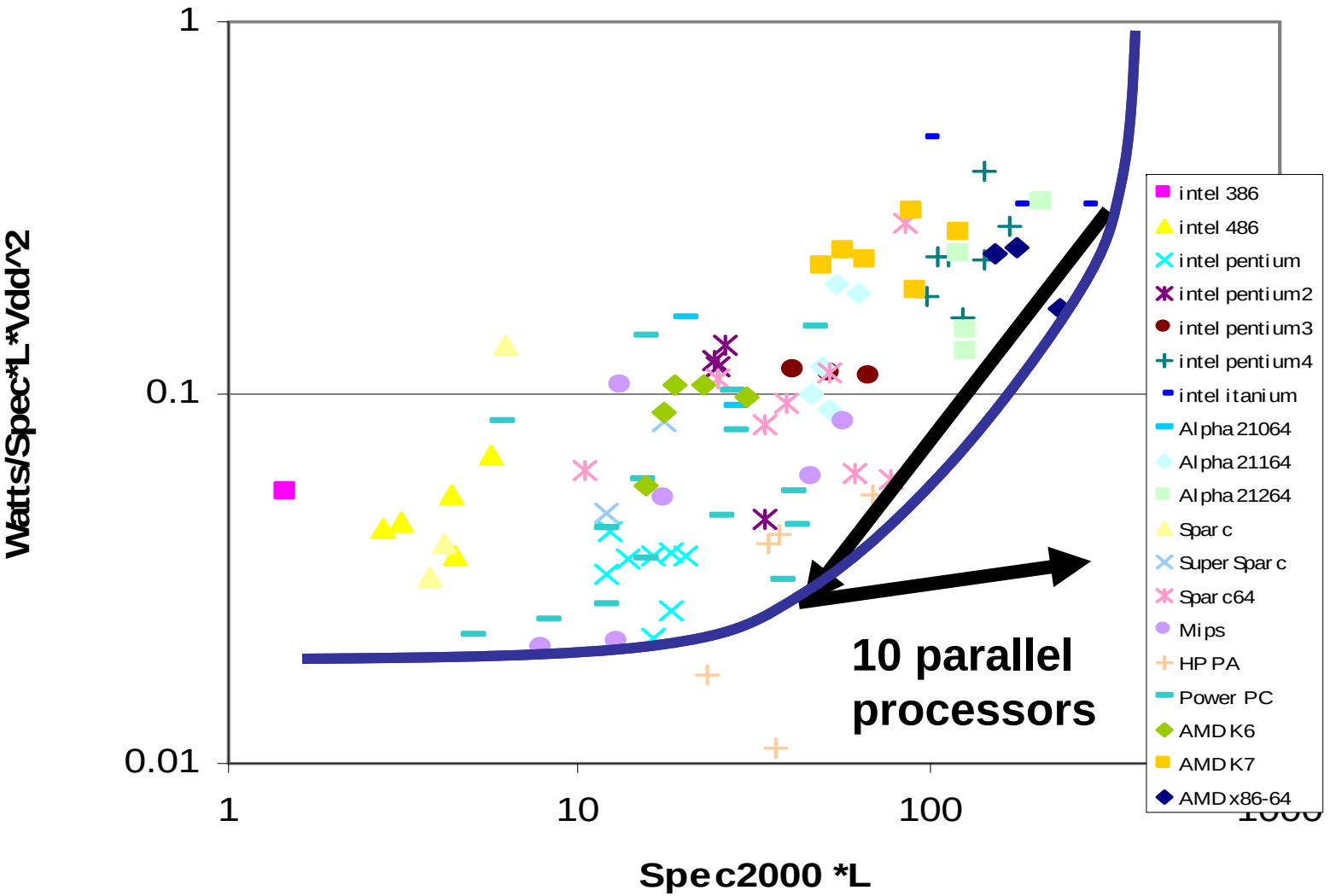
- Assume scaling continues
- Dies don’t shrink in size
- Average power/gate must decrease by 2x / gen

  Or need to build systems that increase in power

Since gates are shrinking in size

- Get 1.4x from capacitive reduction
- Where is the other factor of 1.4x ?

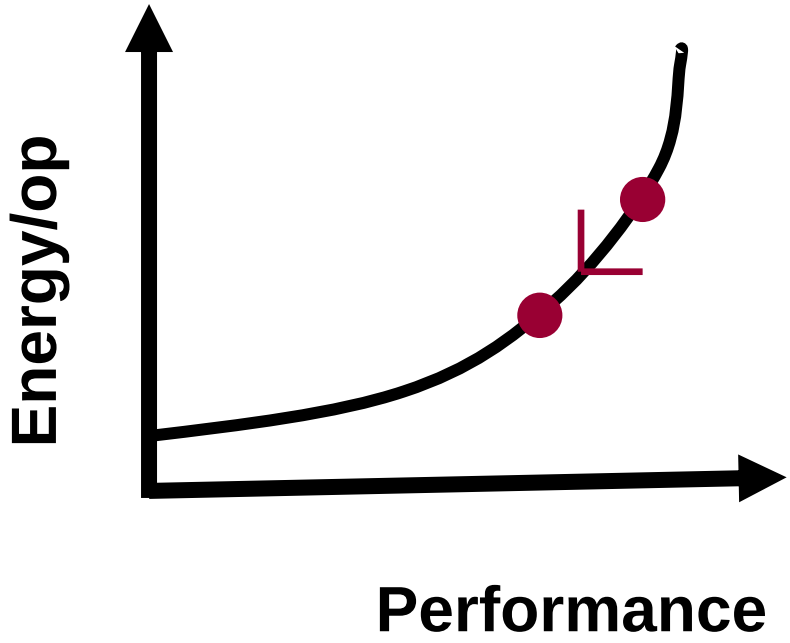
The Push for Parallelism



Exploit Parallelism / Scale Vdd

If you have parallelism

- Add more function units
  Fill up new die (2x)
- Lower energy/op
  ΔE/ΔP will decrease
  Vdd, sizes, etc will reduce
  Build simpler architectures



Works well when ΔE/ΔP is large

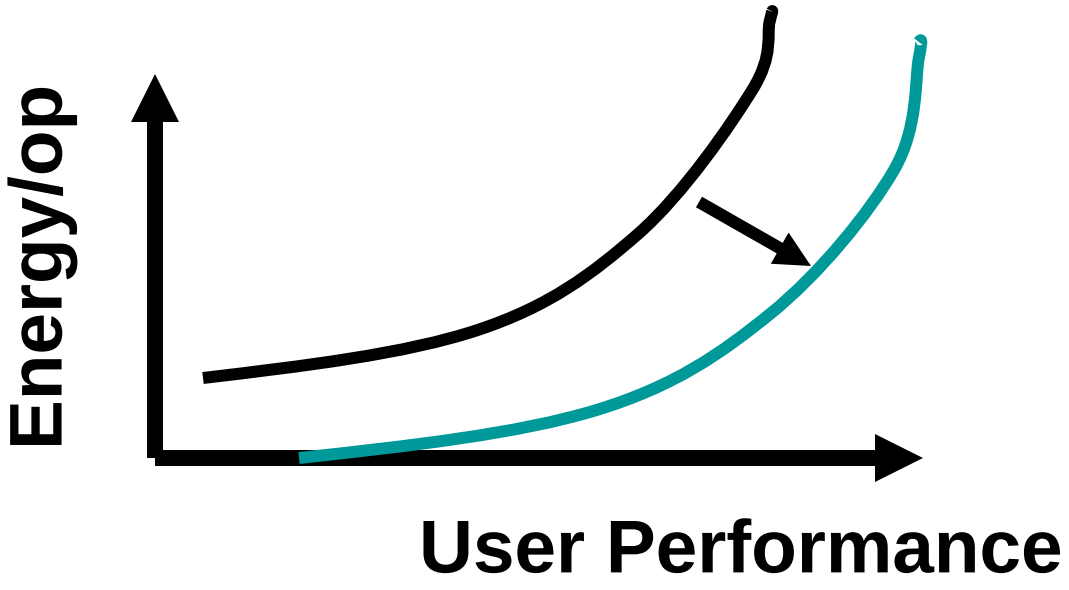
- But what happens when that runs out?

Problem Reformulation

Best way to save energy is to do less work

- Energy directly reduced by the reduction in work
- But required time for the function decreases as well

  Convert this into extra power gains
- Shifts the optimal curve down and to the right



Exploit Specialization

Optimize execution units for specific applications

- Reformulate the hardware to reduce needed work
- Can improve energy efficiency for a class of applications

DSP/Vector engines are more efficient than CPUs

- Exploit locality, reuse
- High compute density

ASICs are more efficient than DSP/Vector engines

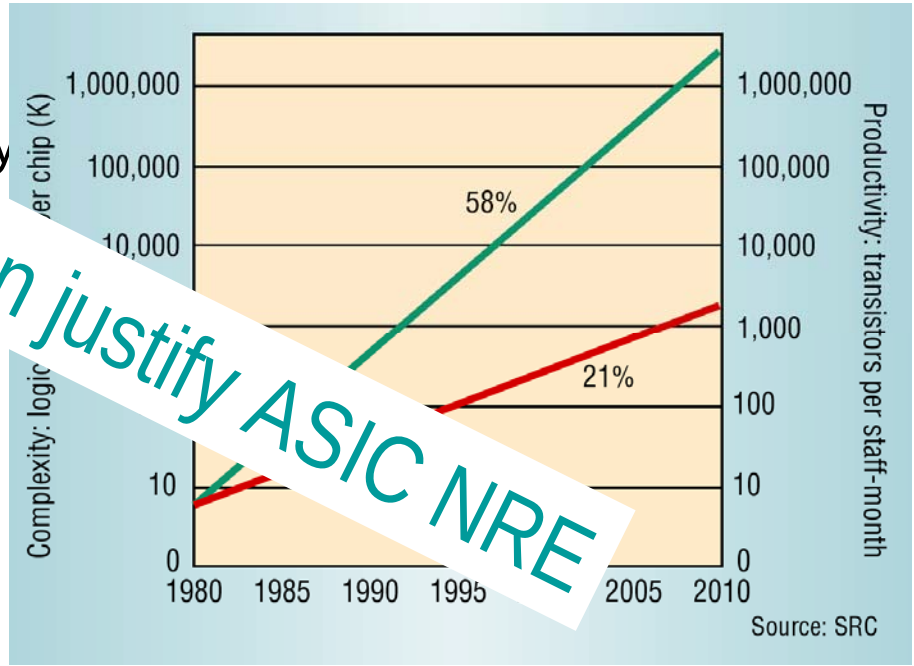
- If we want efficiency, we need more application optimization

ASIC/SOC Design Trends

Rising *non-recurring engineering* costs

- Increasi[illegible] complexity
- Growing verifi[illegible]lexity
- Challenging physical d[illegible]
- Rising mask costs

Few markets can justify ASIC NRE



What Will the Industry Do?

ASIC Future Depends on Your Religion

Believe in correct by construction?

Believe in a generic high-level design language?

Historically both have not worked

- I believe history is correct

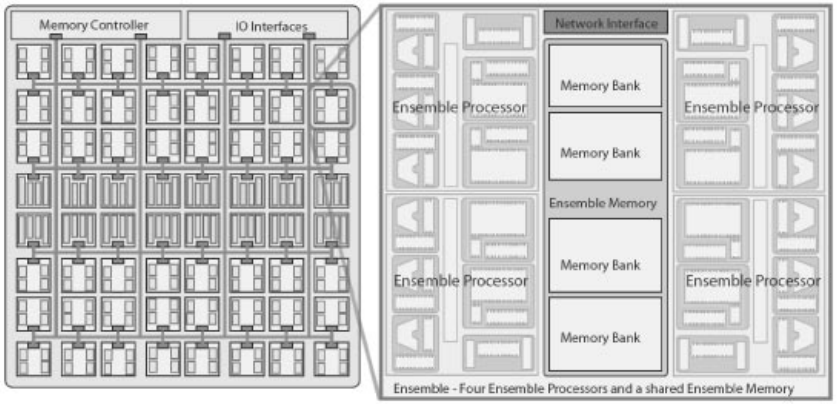
Allowing people to connect complex blocks

- Yields a complex validation problem, and a $20M+ design
- General SoC, SiP will never be cheap

Computing's Future:

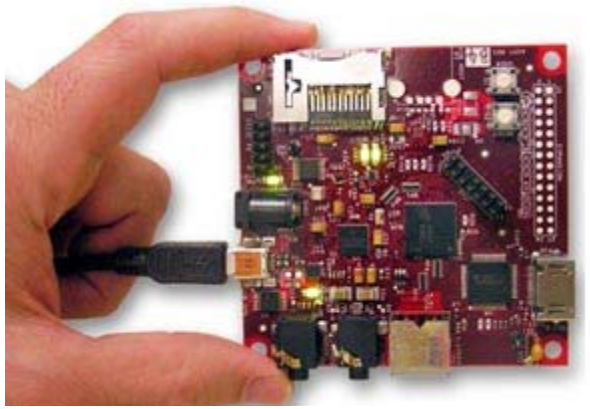
Create a new universal computing platform

- That is more efficient that today/tomorrow's CMP
- Bill Dally is working on this one



Leverage existing large volume processors for other applications

- GPUs moving into general processing
- OMAP being distributed as Unix system



Can We Do Better?

Chip design is expensive since chips are complex

But the building blocks are well known

- Many of the optimizations are well known too
- Designer often do many of the same steps
- Part of the reason for off-shoring

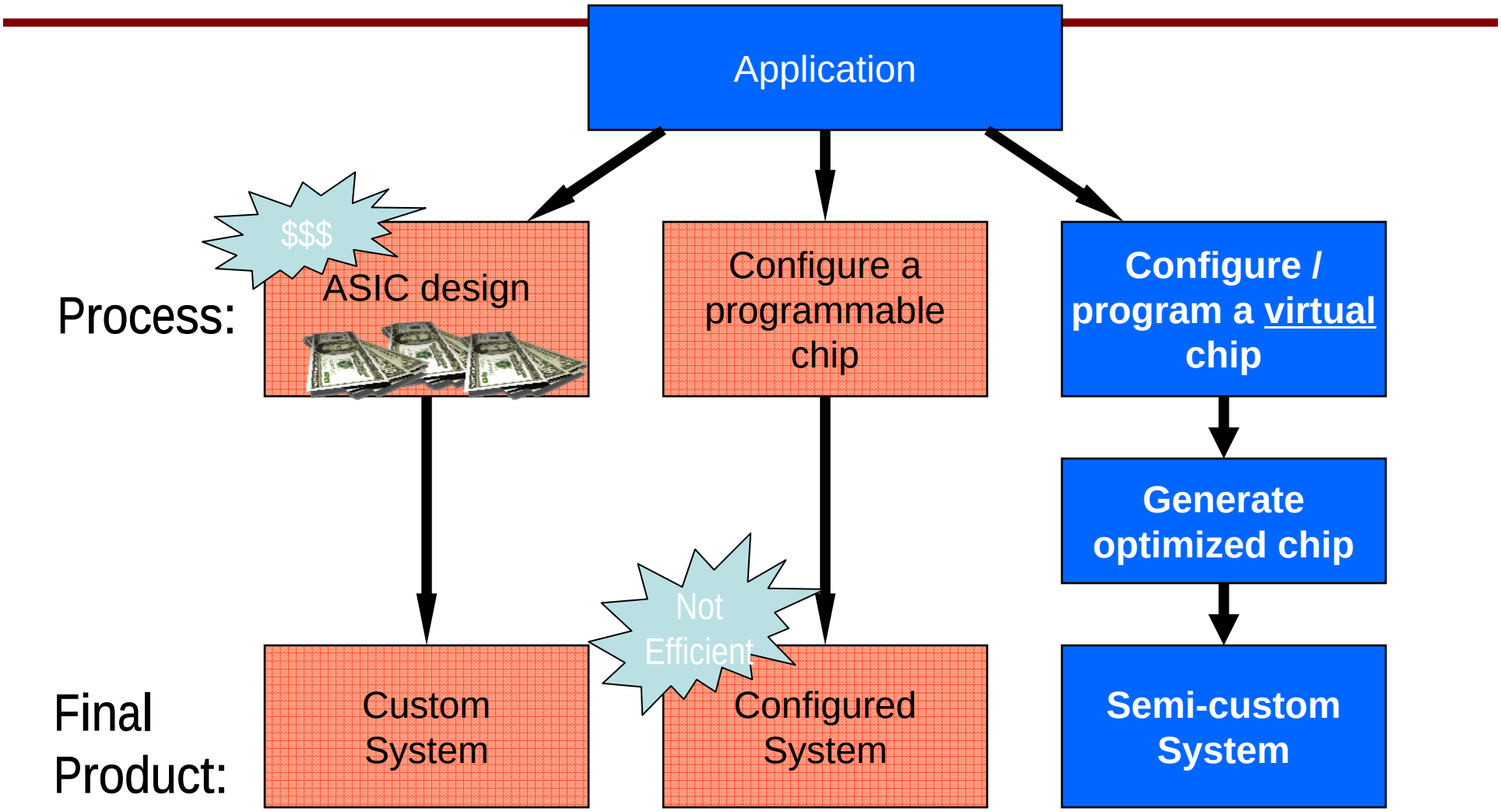
  Don’t need experience

Getting the system to work is hard

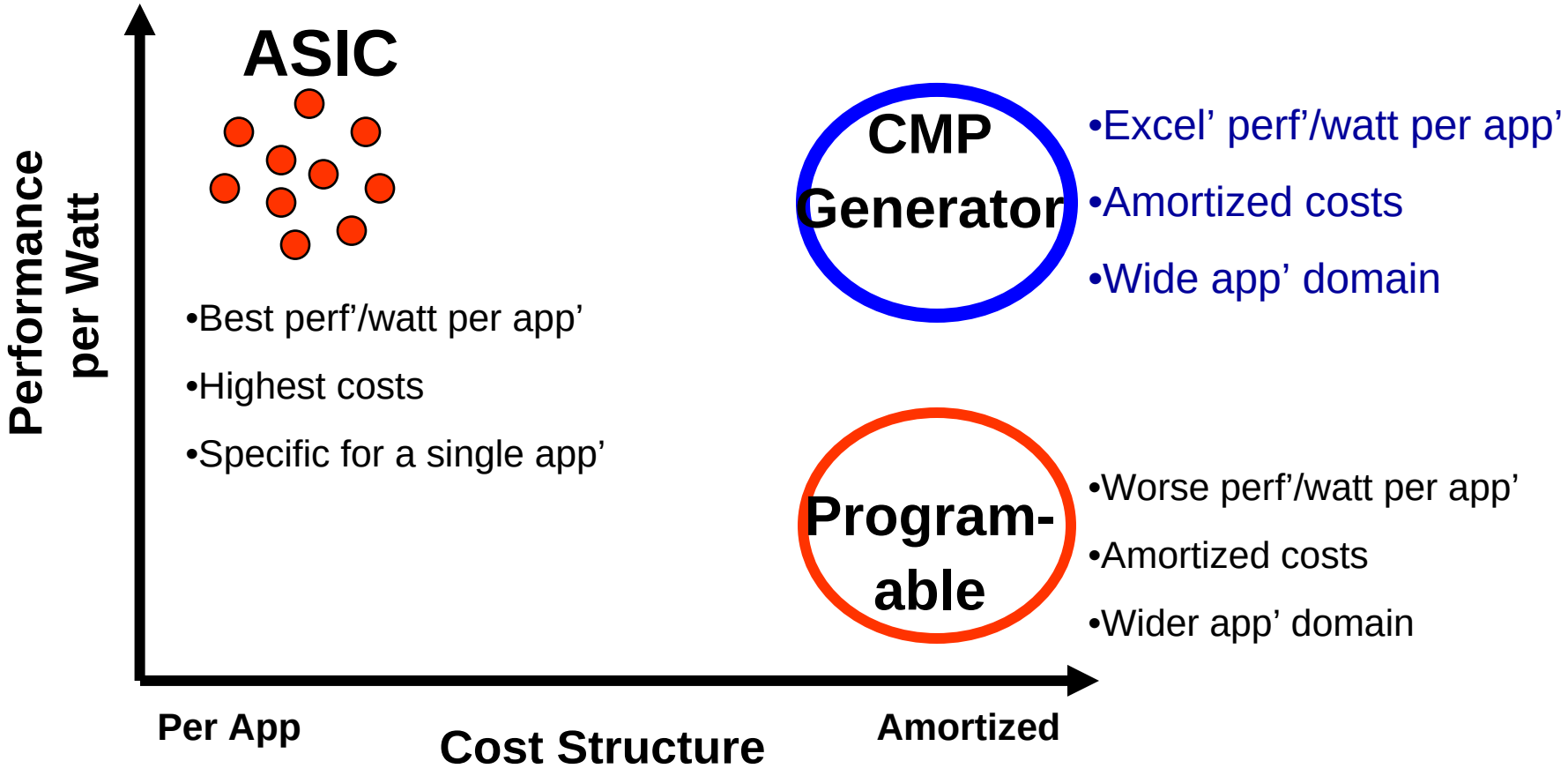
- There is a lot of turning the crank that is needed

Can we automate some of the crank turning?

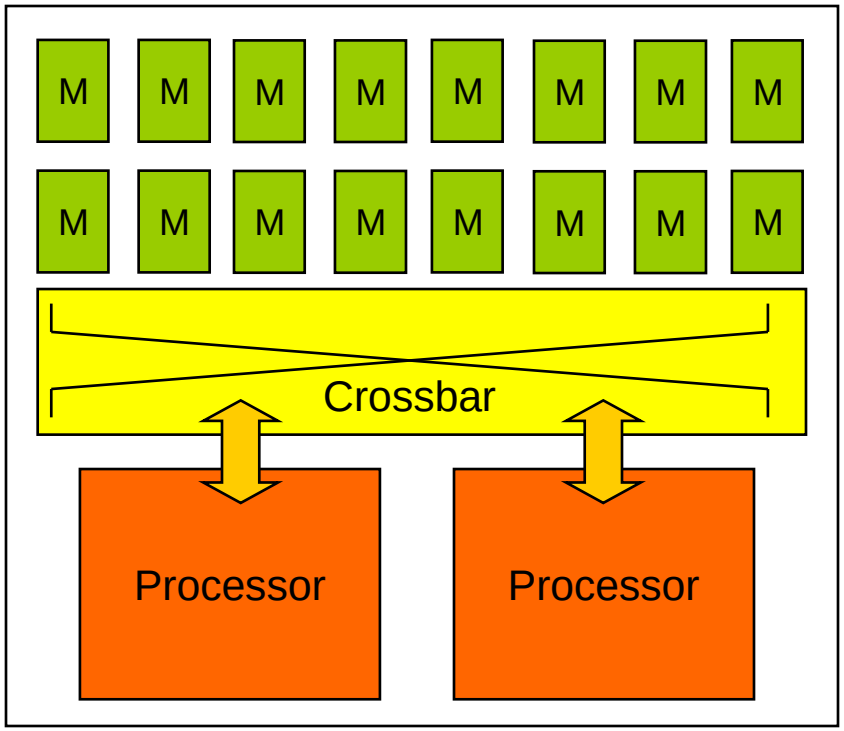
Chip Generator Idea



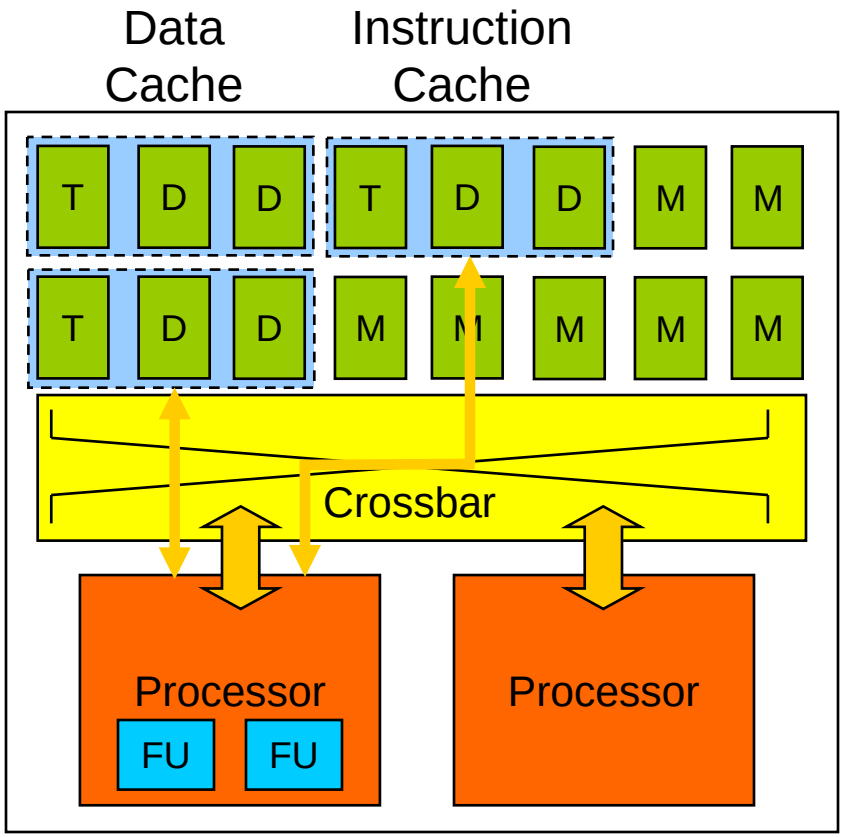
Another Way To Put It…



Smart Memories - A “Pretend” Generator



Smart Memories Architecture: Single Tile



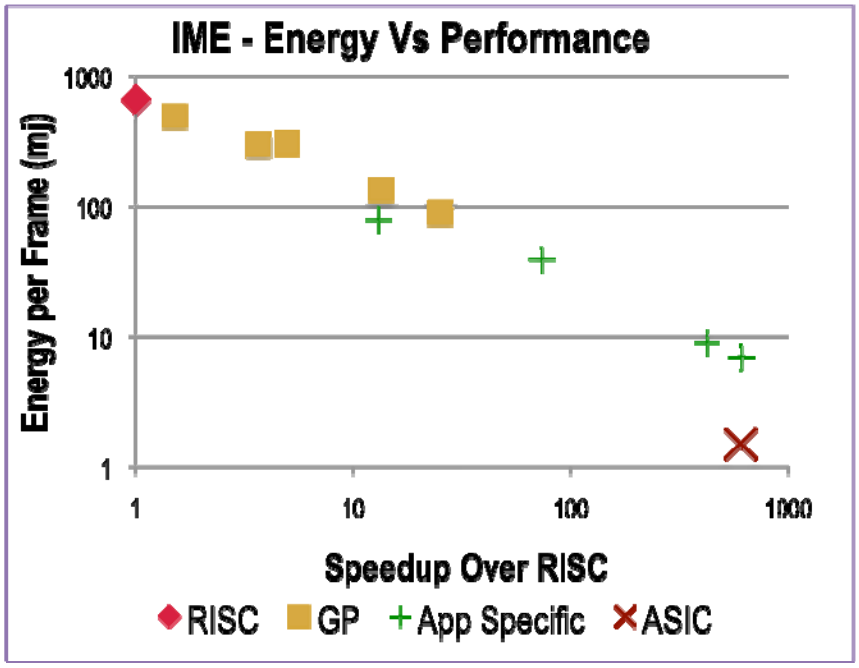
Chip Generator Derivative

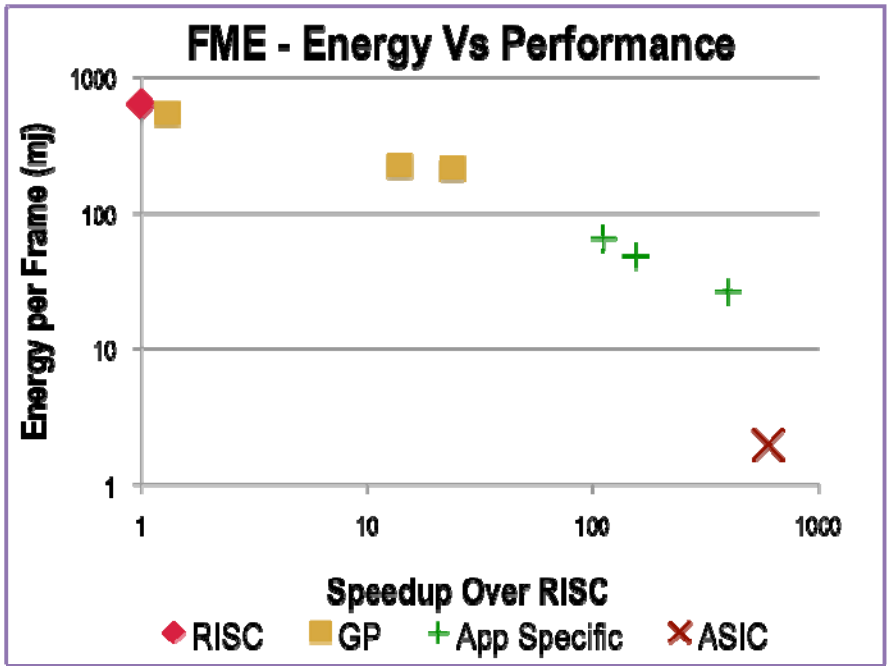
Looks Promising

Large energy / performance gains are possible:

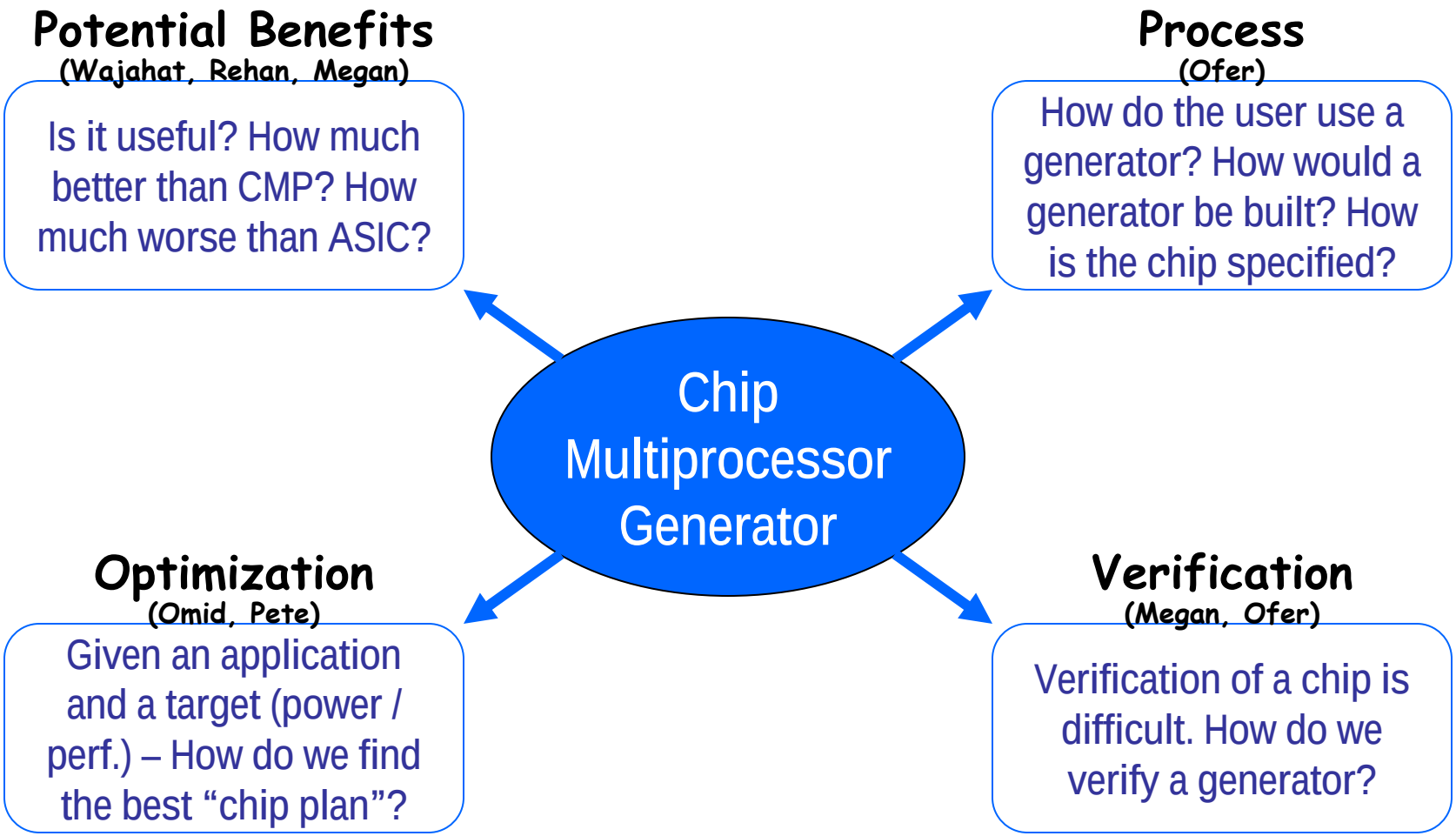
- Use H.264 as example application
- Use a GP-CMP chip generator

400-600X initial perf. gap





New And Exciting Challenges



Conclusions

The technology engine driving IT is slowing down

- Power efficiency is the real problem

Application optimization leads to efficiency

- But design is too expensive today to do this

Need to rethink design

- Build chip generators not chips

  These are virtual programmable chips

  Have tools generates the real chips that customers want