

---

# **ECE 697J – Advanced Topics in Computer Networks**

Design Basics of Network Processors

10/14/03

# Network Processors

---

- Programmable packet processing engines
  - Programmability provides flexibility for new applications
  - Parallelism to achieve scalable processing power
- Network processors are embedded “systems-on-a-chip”
  - Why embedded?
  - What is a system-on-a-chip?
- System-on-a-chip
  - Processing: RISC core
  - Memory: embedded SRAM and (possibly) DRAM
  - I/O: network / switch fabric interfaces
- So, what’s hard about building network processors?

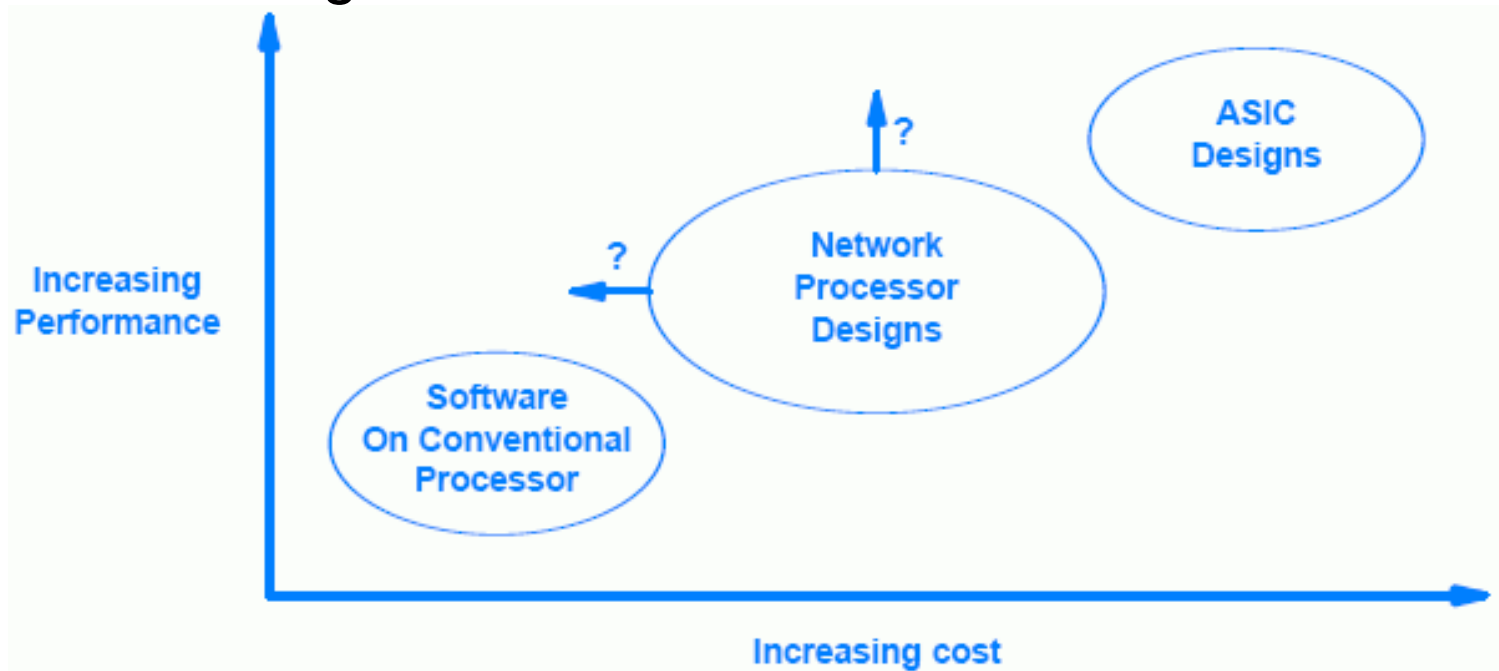
# Generality

---

- Network processors should be able to handle any protocol
  - Should not be specialized only for particular protocol (e.g., IPv4)
  - But we can assume NP processes network traffic
- Packet processing functions:
  - Error detection and correction
  - Traffic measurement and policing
  - Frame and protocol demultiplexing
  - Address lookup and packet forwarding
  - Segmentation, fragmentation, and reassembly
  - Packet classification
  - Traffic shaping
  - Timing and scheduling
  - Queuing
  - Encryption and authentication
- So, what's hard about building network processors?

# Economic Factors

- A few thoughts on cost:



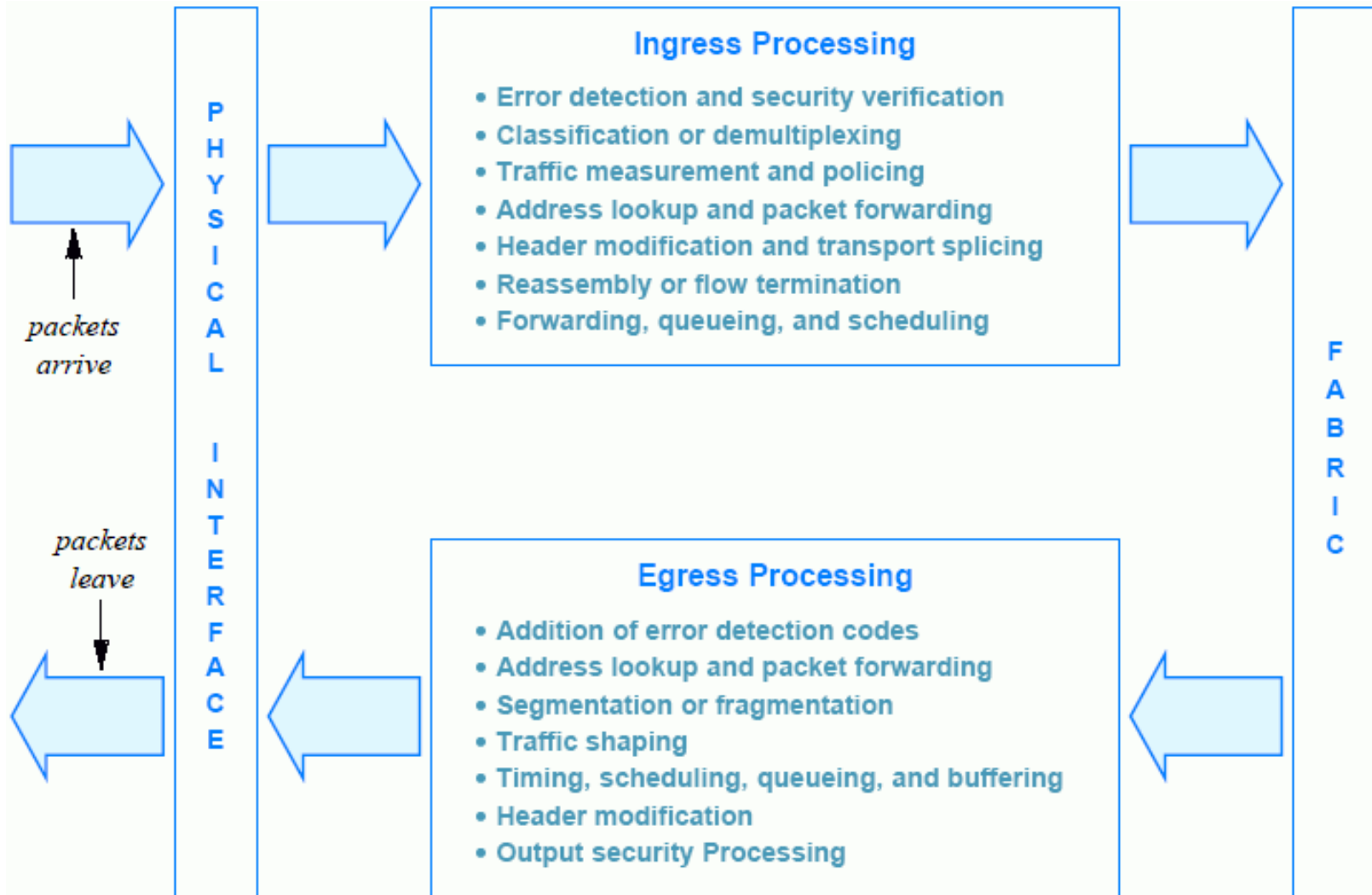
- ASICs are expensive to develop, but cheaper per-chip
- NPs are for quickly changing, moderate quantity market
- The cheaper the better

# Minimality

---

- “A network processor is not designed to process a specific protocol or part of a protocol. Instead, designers seek a minimal set of instructions that are sufficient to handle an arbitrary protocol processing task at high speed”
- Generality
  - Already achieved through general-purpose processors
- Performance
  - Achieved by supporting certain functions in hardware
- Minimality
  - Choose only those functions that common
- What functions should be supported in hardware?

# Typical Processing



# Ingress Processing

---

## Ingress Processing

- Error detection and security verification
- Classification or demultiplexing
- Traffic measurement and policing
- Address lookup and packet forwarding
- Header modification and transport splicing
- Reassembly or flow termination
- Forwarding, queueing, and scheduling

# Egress Processing

---

## Egress Processing

- Addition of error detection codes
- Address lookup and packet forwarding
- Segmentation or fragmentation
- Traffic shaping
- Timing, scheduling, queueing, and buffering
- Header modification
- Output security Processing



# NP Co-Processors

---

- What should be done on a co-processor?
  - Functions that are computationally intense
  - Functions that are similar/same across different protocols
  - Functions that can be implemented more efficiently in hardware
  - Functions that are used frequently
- Examples?
  - Error correction/detection: checksum, CRC
  - Hash computations
  - Table lookups
  - Cryptographic processing: encryption/decryption
  - Others?

# NP Software

---

- Software needs to be developed together with NP
- Challenges:
  - Needs to integrate all hardware components
  - Requires suitable abstractions for application developer
  - Software simulator/emulator
  - Support functions: traffic generation etc.
- Software and hardware are co-designed
- Software environment is current topic of research and current solutions are challenging to use

# Technology Trends

---

- Relevant technologies for network processors
  - Link speed
  - CMOS feature size (density)
  - Maximum chip size
  - Clock speed
  - Memory technologies
  - Application complexity
- Moore's Law: "Number of components on chip doubles every 18 months"

# Moore's Law 1965

---

The experts look ahead

## **Cramming more components onto integrated circuits**

**With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65,000 components on a single silicon chip**

By Gordon E. Moore

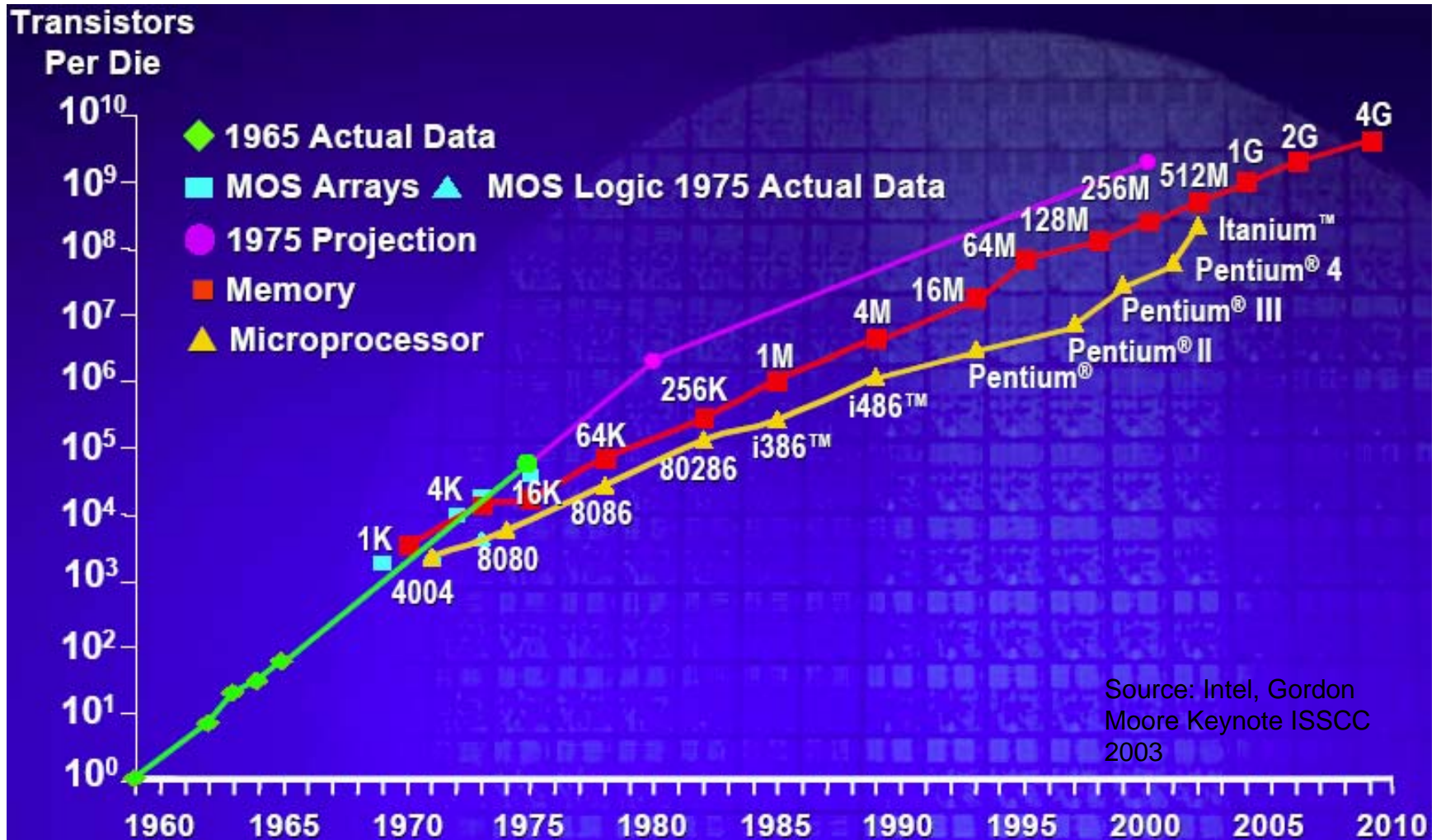
Director, Research and Development Laboratories, Fairchild Semiconductor  
division of Fairchild Camera and Instrument Corp.

# Humor in Moore's Paper

---



# Moore's Law Data

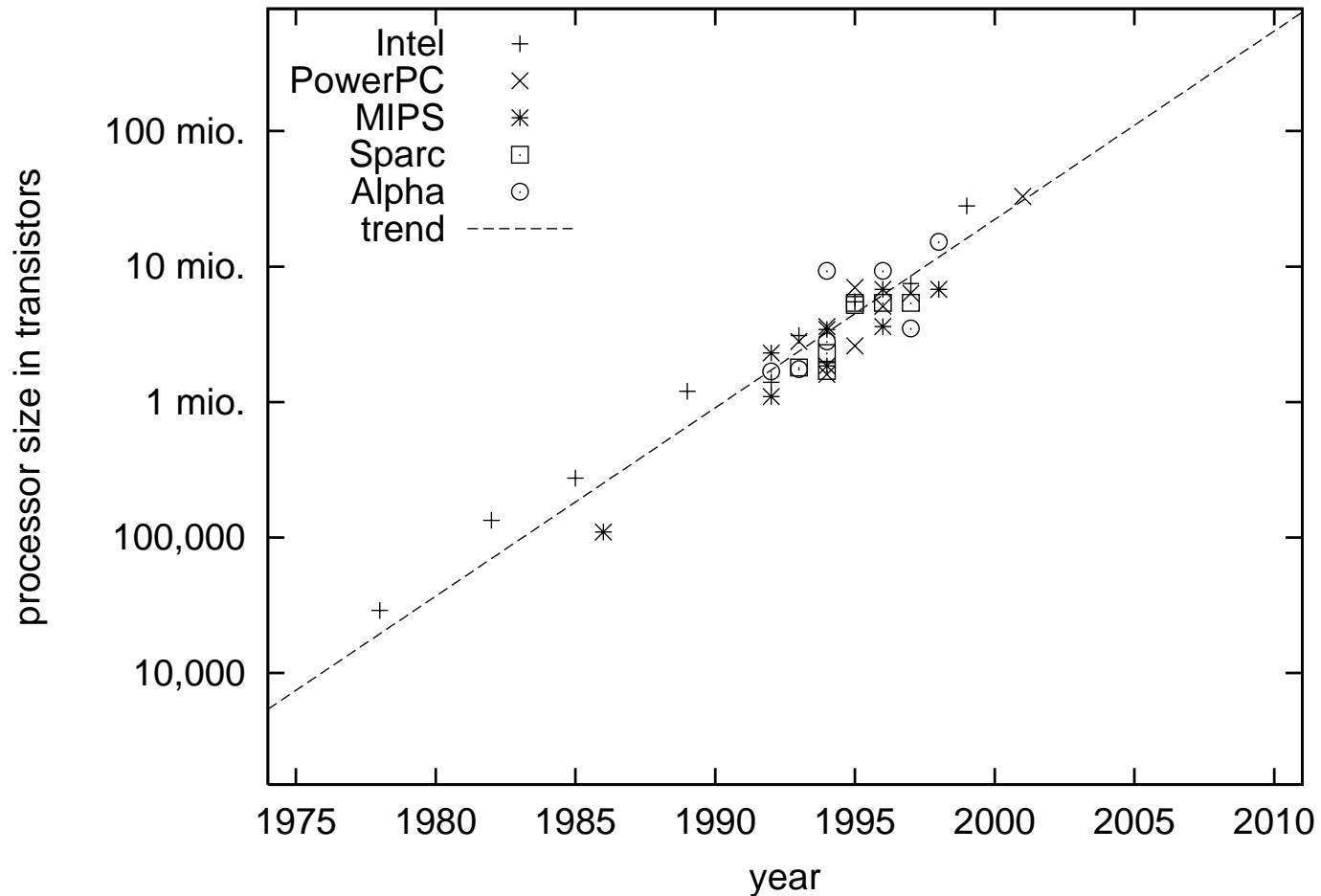


# Long-Term Trends

---

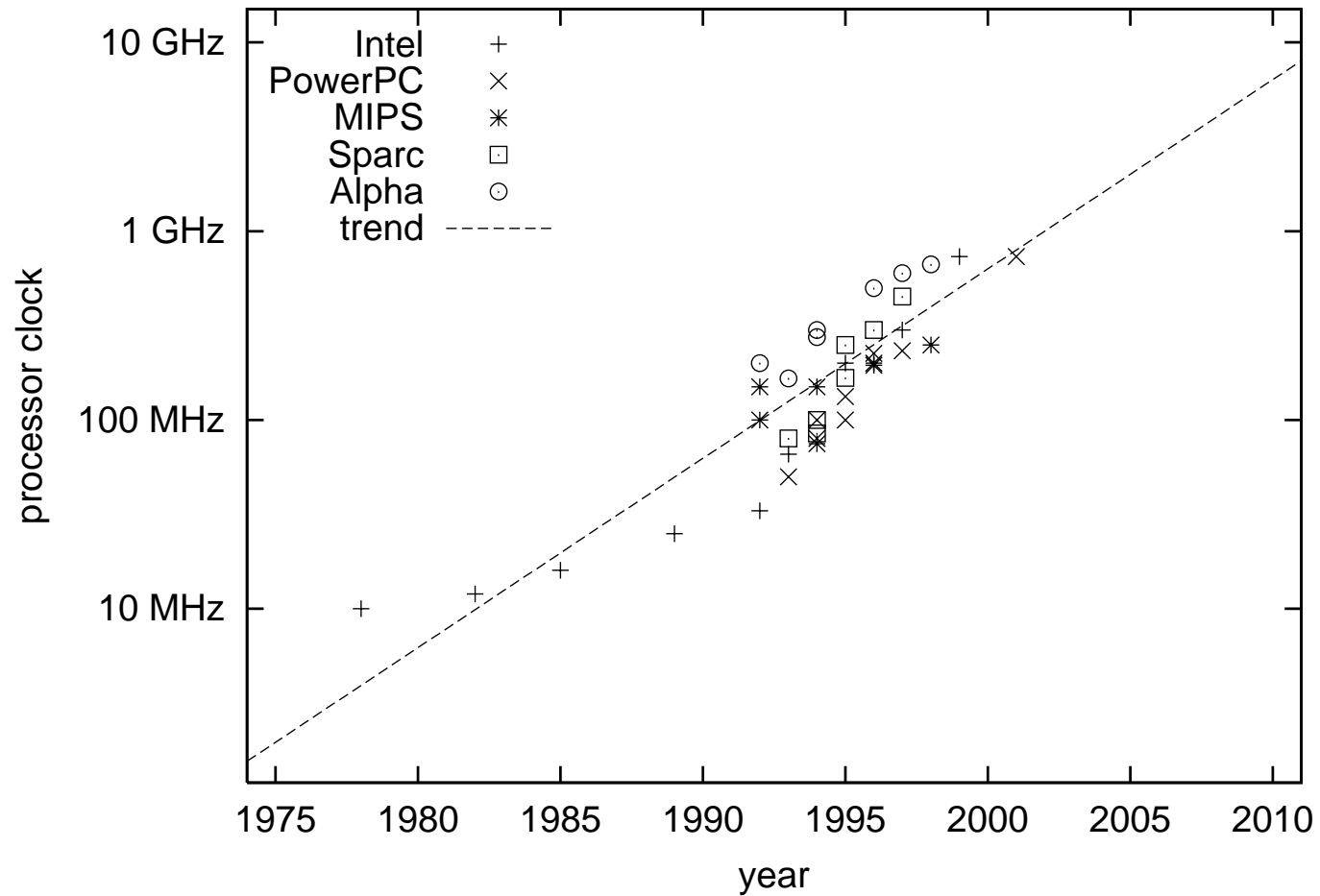
- Moore's Law is pretty "stable"
  - Of course it's not a "law"
  - Self-fulfilling prophecy
  - Extremely beneficial for industry to do long-term planning
- Will probably continue until end of decade
  - Semiconductor Industry Association's roadmap
- Let's look at individual metrics

# Workstation Processor Size

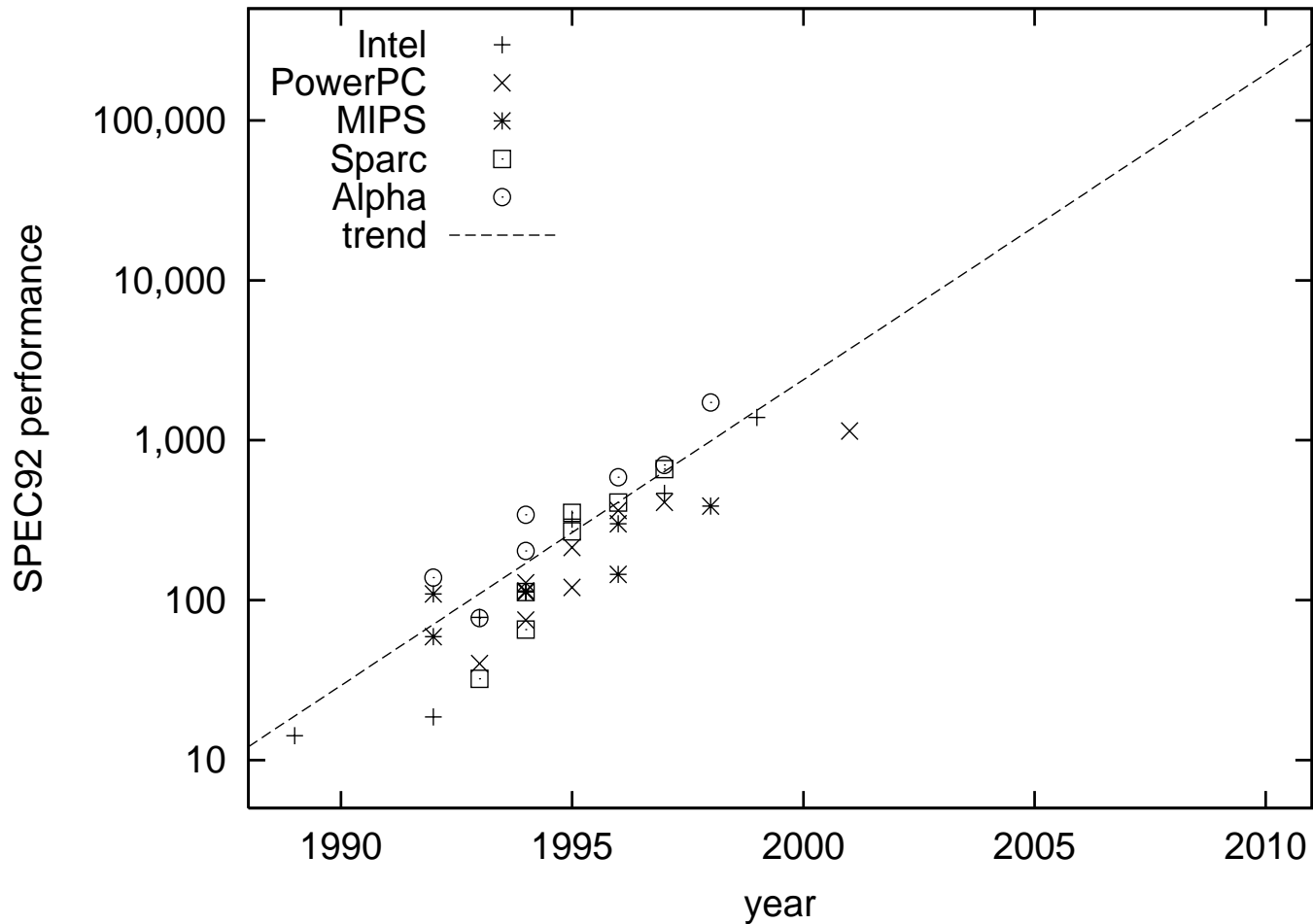




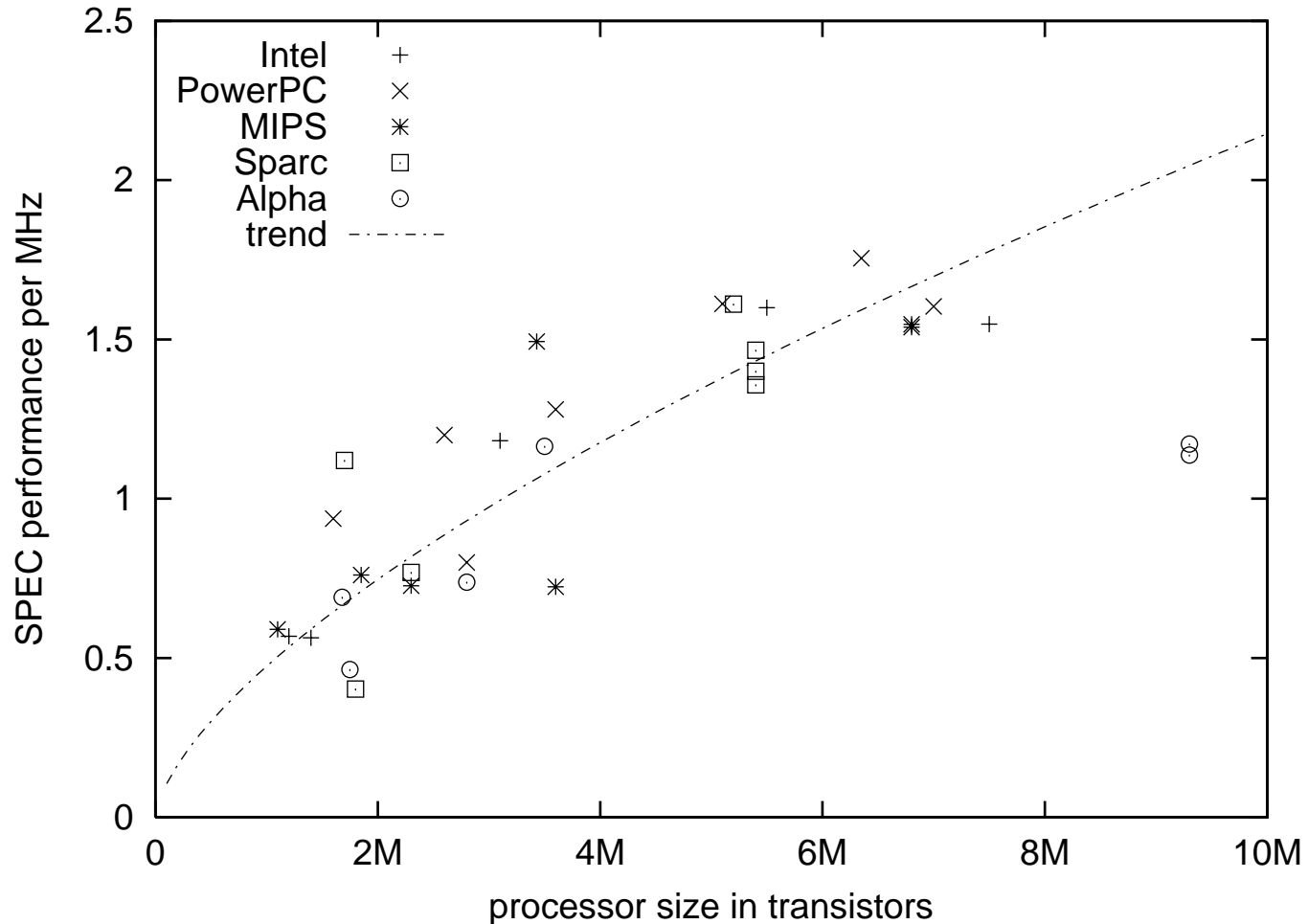
# Processor Clock Rate



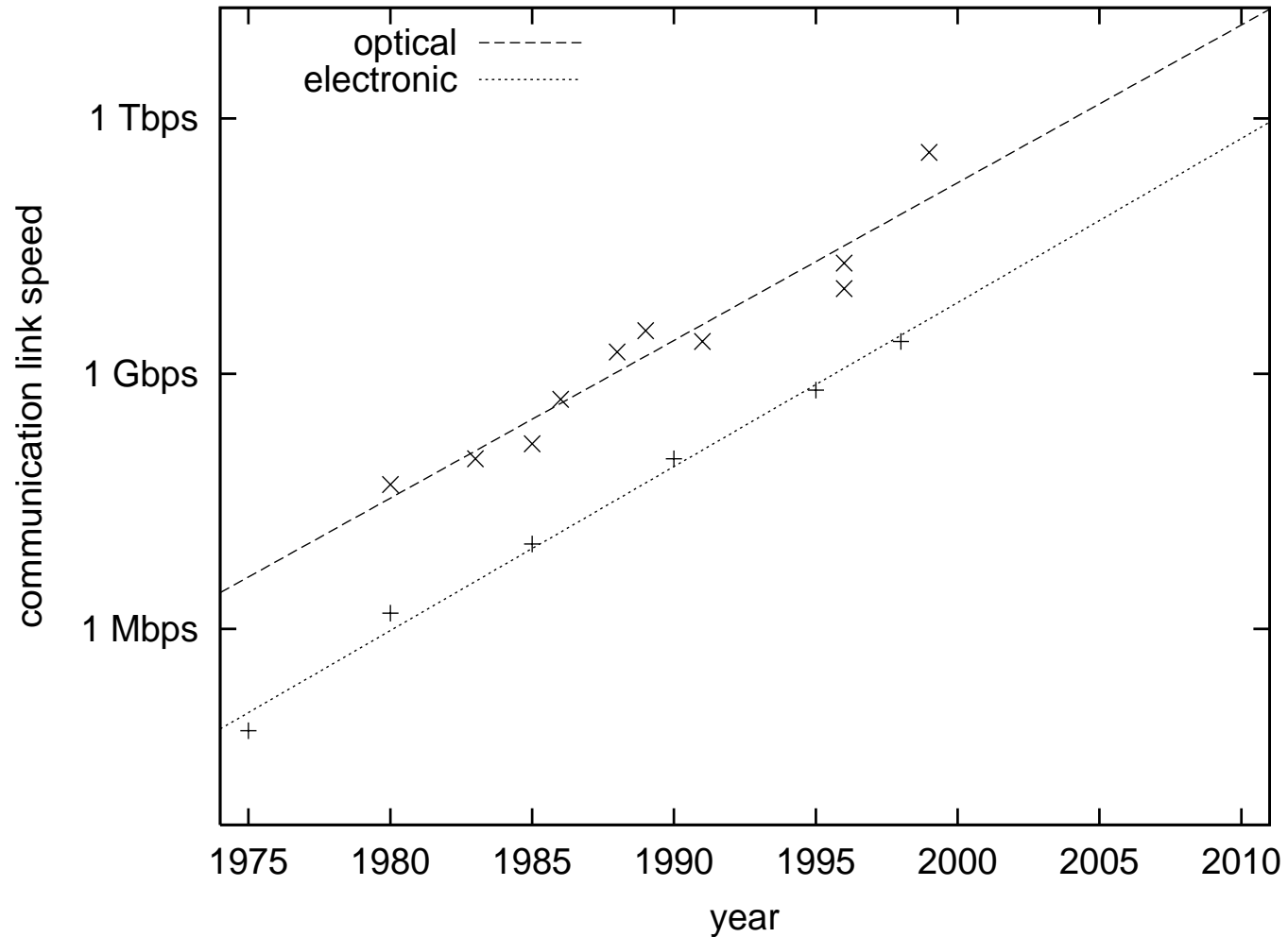
# SPEC Performance



# Performance vs. Size



# Link Speed



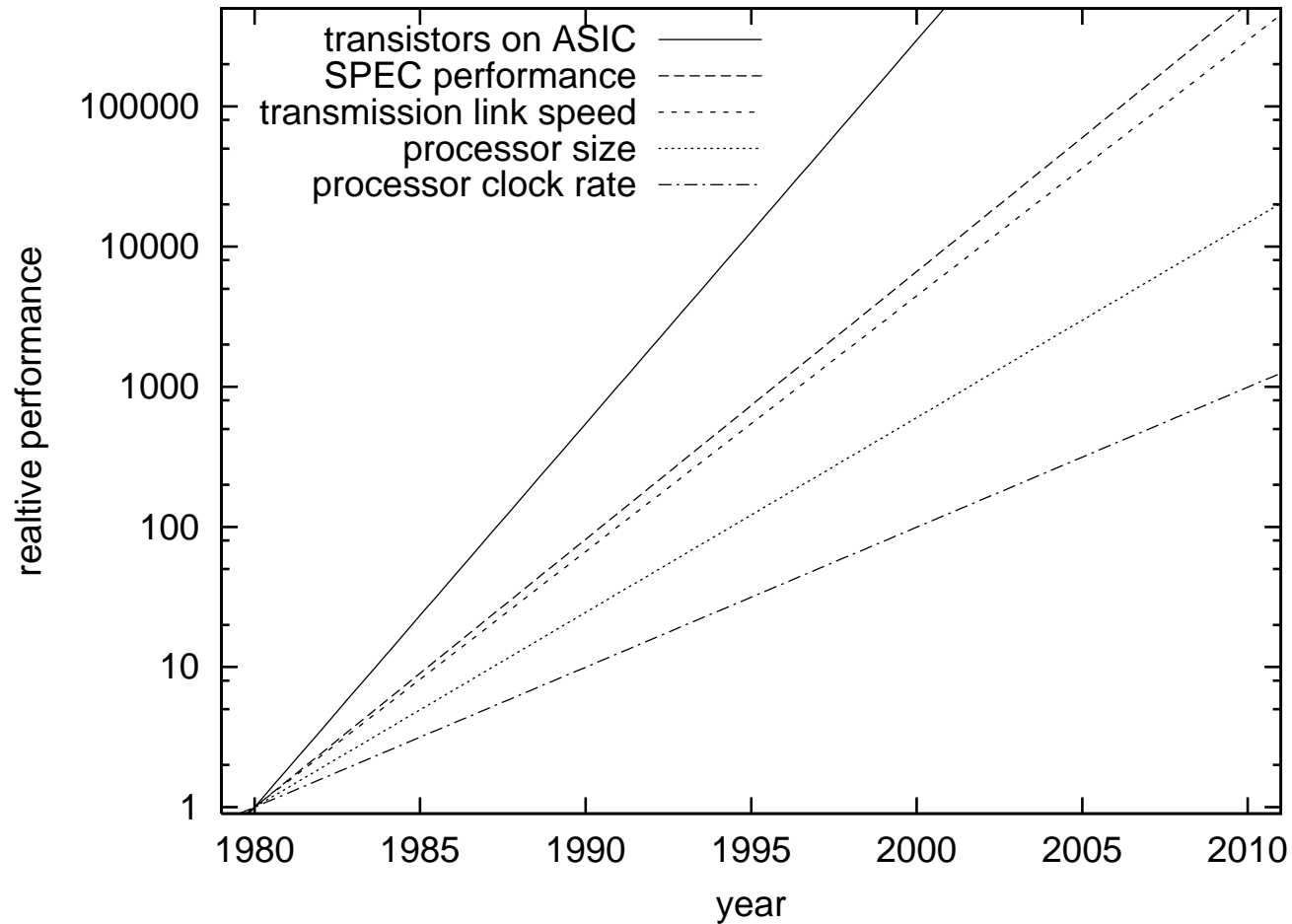
# Comparison of Trends

$$\text{performance in year } x = a_{t_0} \cdot e^{b \cdot (x - t_0)}$$

Table 2.1: Growth Parameters of Key Technologies. The values for  $a$  are normalized to the year 2000. Sizes are given in million transistors (Mtx) and million gates (Mgates) (1 gate  $\approx$  4 transistors).

Technology		$a$	$b$	Time to double
Communication	electronic links	6.8 Gbps	0.44	18 months
	optical links	175 Gbps	0.42	19 months
Processor	SPEC performance	2400	0.44	18 months
	clock	630 MHz	0.23	36 months
	size	22 Mtx	0.32	26 months
ASIC	size	55 Mgates	0.63	8 months

# Comparison of Trends

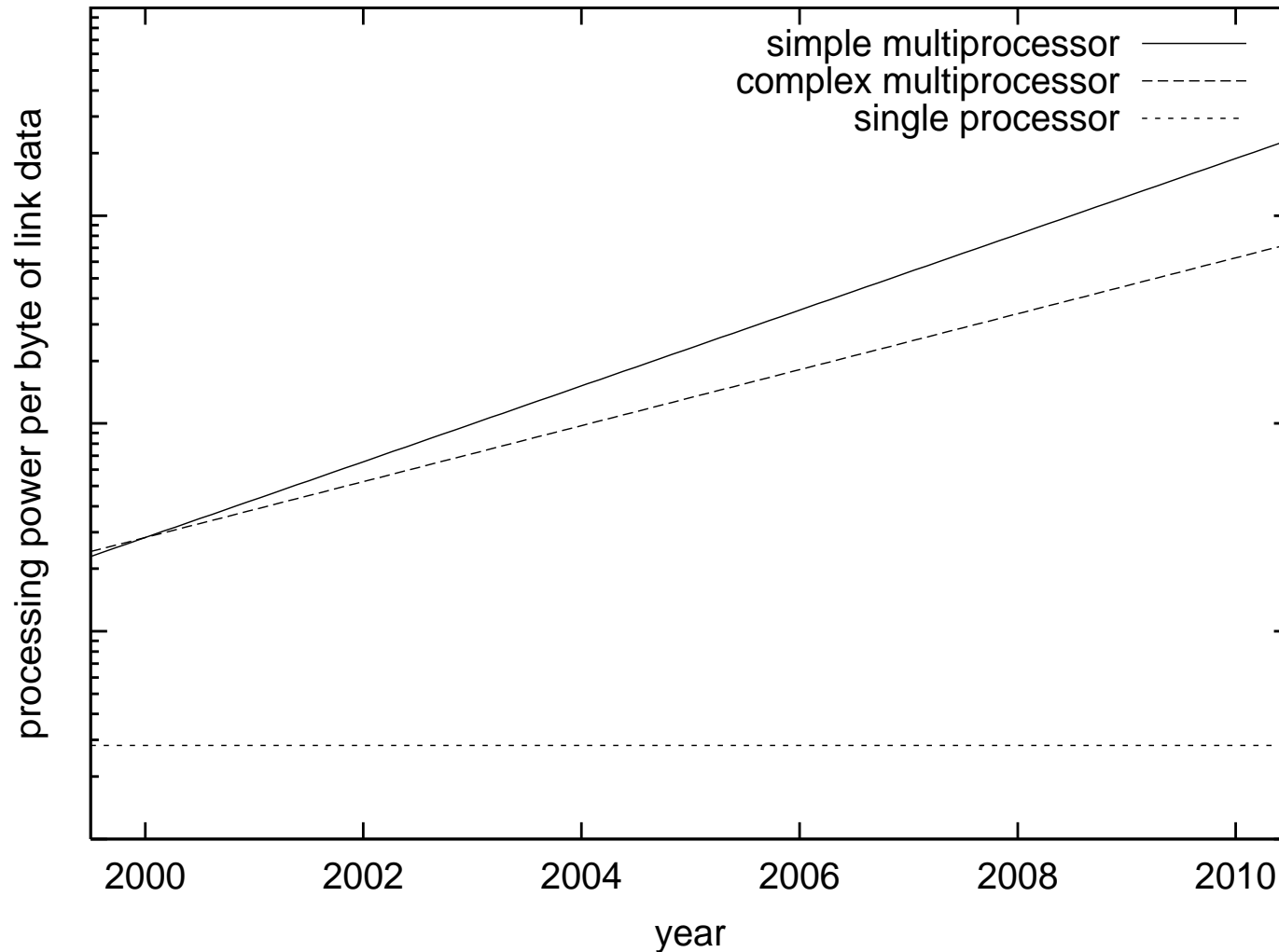


# Impact for NPs

---

- Possible conclusions for architectures:
  - Arch 1: single CPU
  - Arch 2: CMP with high-performance processors
  - Arch 3: CMP with low-performance processors
- Performance criteria:
  - How much processing for each packet
  - Measured in SPEC per byte of link data

# Performance Trends



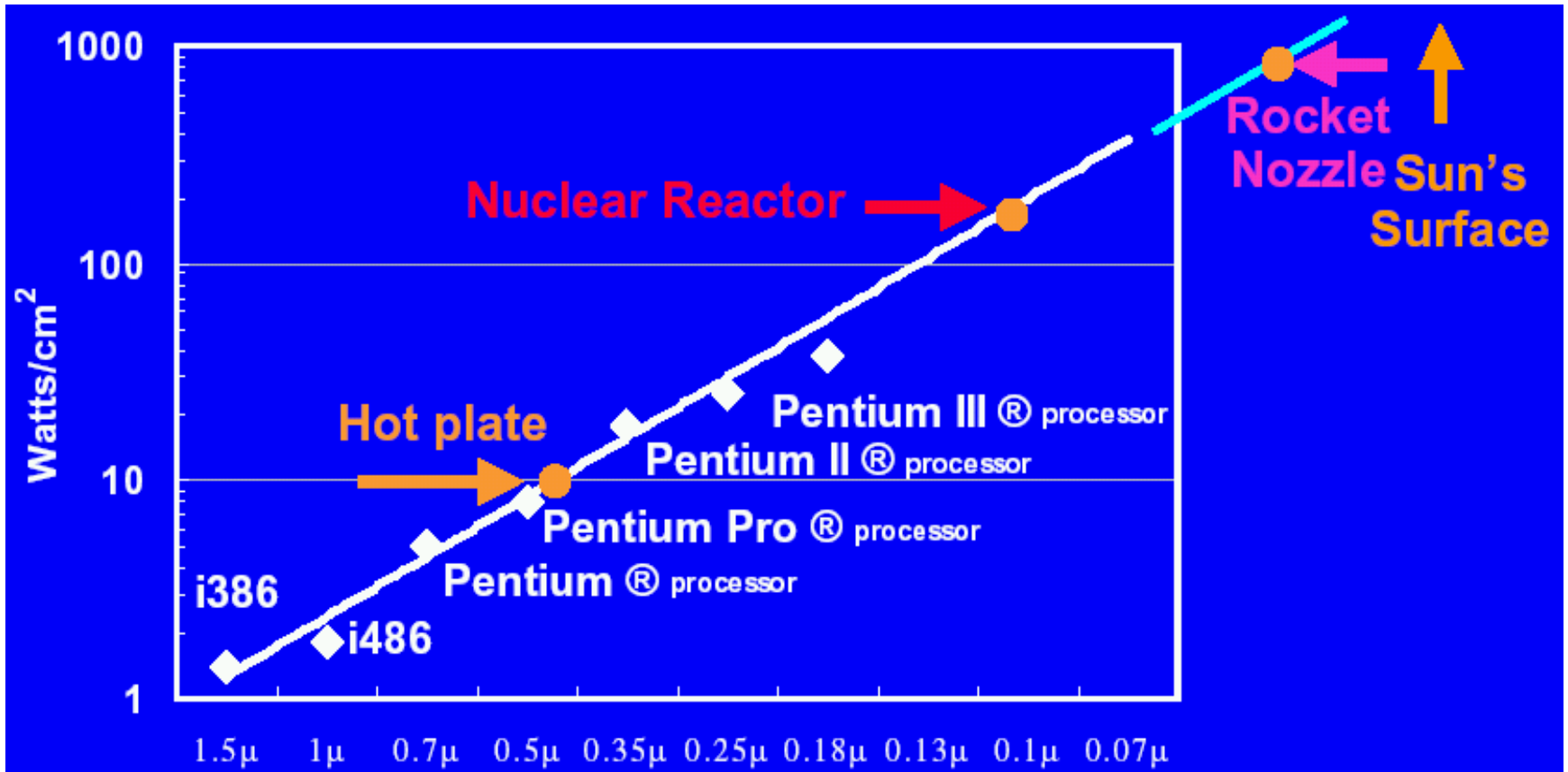


# Limitations

---

- What are the limits to these trends?
  - Bottleneck in centralized components
  - Memory gap
  - Power consumption
  - Power density

# Power Density



by Fred Pollack

# Next Class

---

- Next Class: more NPArchitecture
  - Read Chapter 13 & 14
- Next week: commercial NPs
- Everybody gets to present one:
  - Multi-Chip Pipeline by Agere
  - Augmented RISC Processor by Alchemy
  - Embedded Processor Plus Coprocessor by AMCC
  - Pipeline of Homogeneous Processors by Cisco
  - Configurable Instruction Set Processor by Cognigine
  - Pipeline of Heterogeneous Processors by EZchip
  - Extensive and Diverse Processors by IBM
  - Flexible RISC Plus Coprocessor by Motorola