Modern Microprocessor Development Perspective

Prof. Vojin G. Oklobdzija, Fellow IEEE
IEEE CAS and SSC Distinguished Lecturer

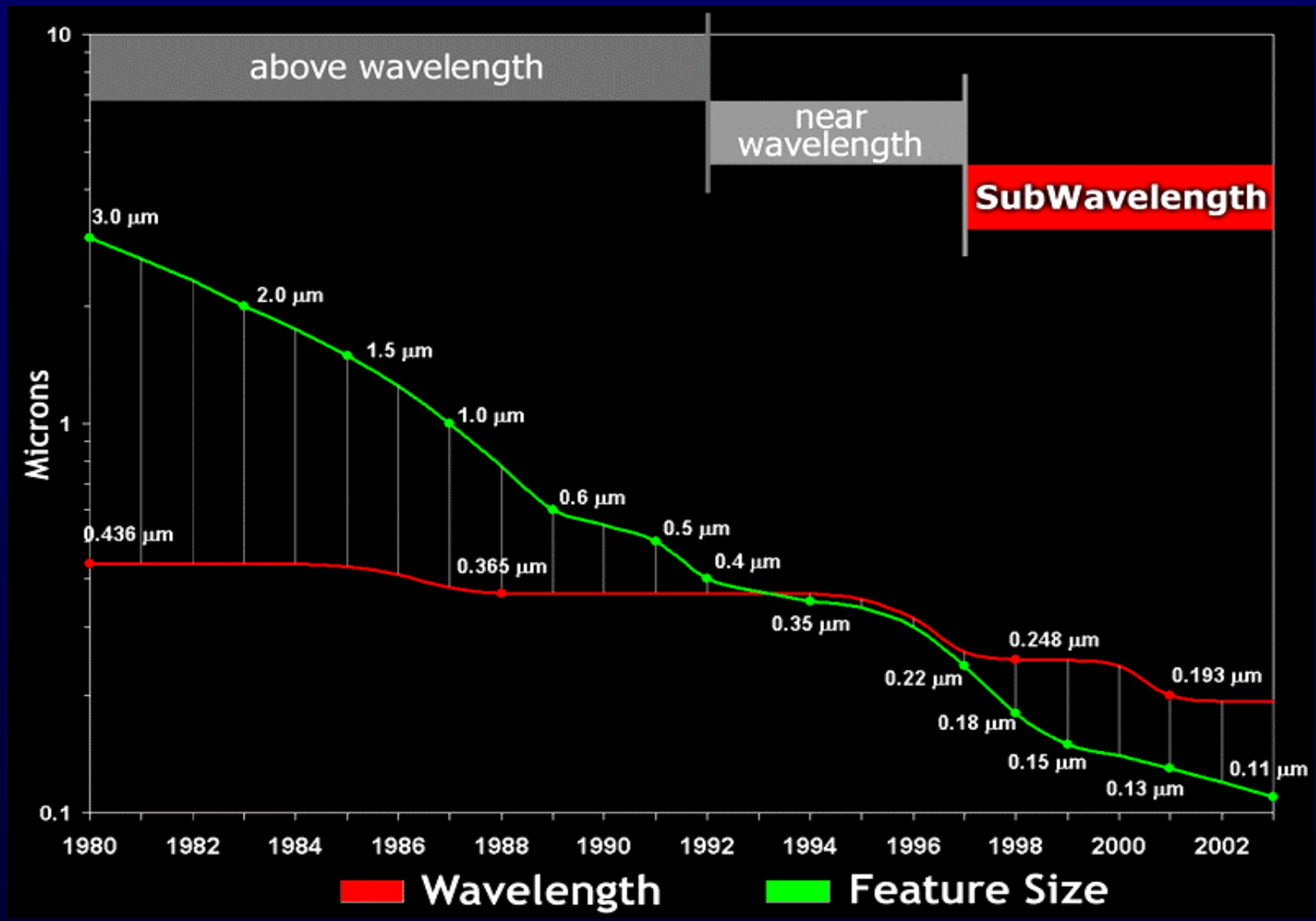
University of California
Davis, USA

This presentation is available at: http://www.ece.ucdavis.edu/acsel *under* Presentations

Outline of the Talk

- Historic Perspective
- Challenges
- Definitions
- Going beyond one instruction per cycle
- Issues in super-scalar machines
- New directions
- Future

TECHNOLOGY IN THE INTERNET ERA: Lithography

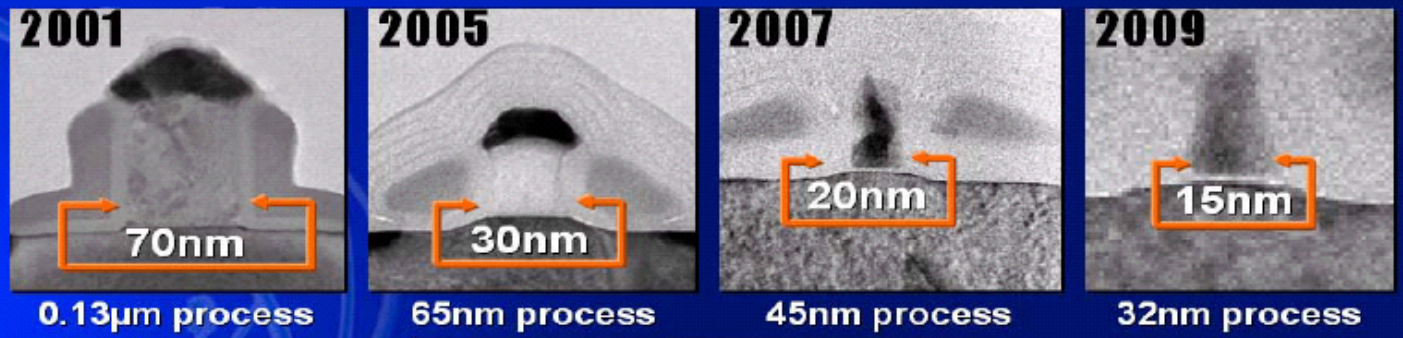


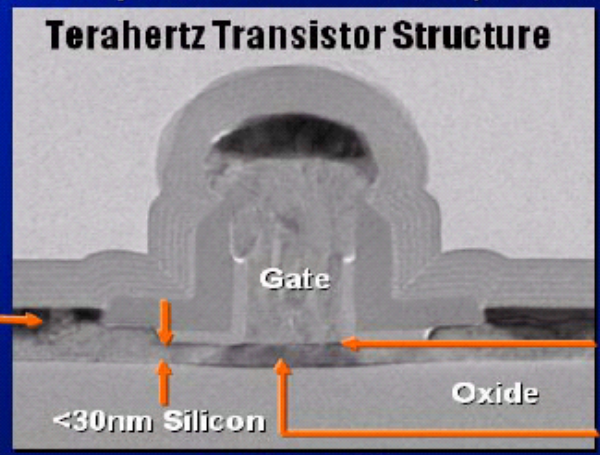
From Dennis Buss, Texas Instruments, ICECS, Malta 2001 presentation

Process Technology Trends

Intel: To the Terahertz Transistor

Transistor Leadership Continues



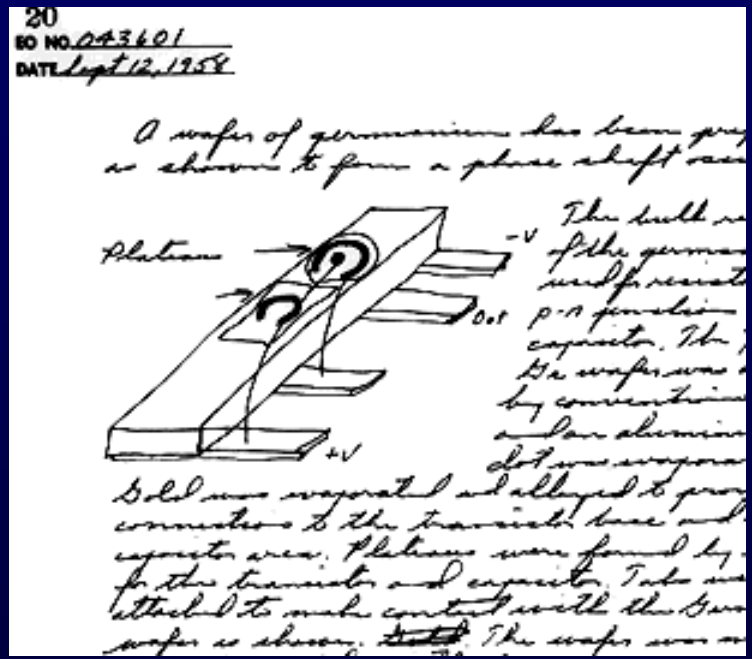


Source: Intel

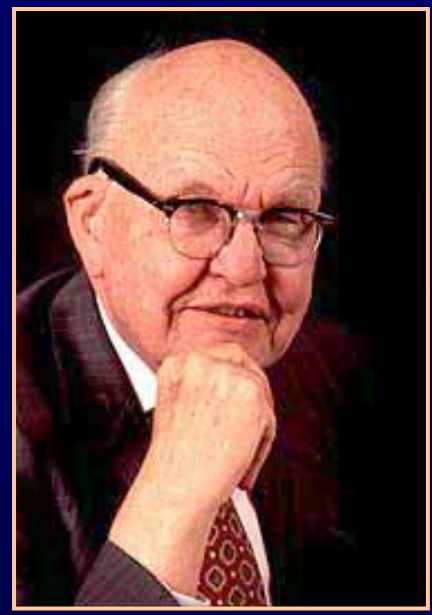
www.intel.com/labs

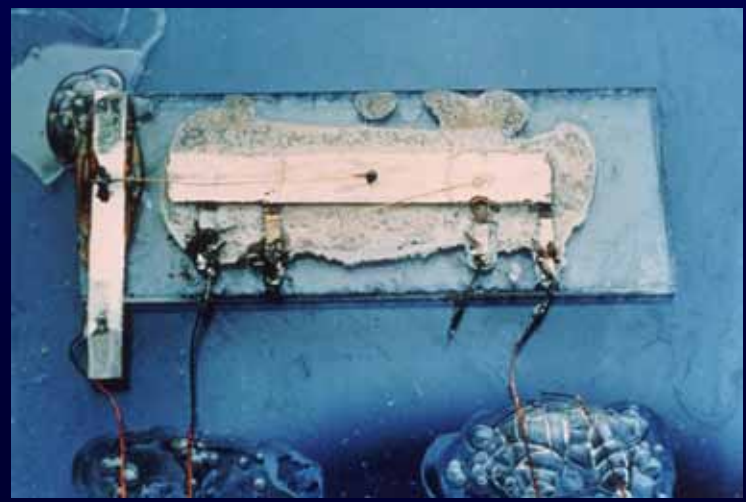
Intel Labs

INTEGRATED CIRCUIT - 1958


20
EO NO. 043601
DATE Sept 12, 1958

A wafer of germanium has been
is shown to form a phase shift
Platen
-V
0.1
+V
The built
of the germa
and forms
p-n junctio
capacitor. The
The wafers were
by conventio
silver aluminu
dot
Gold was vaporized and alloyed to
connection to the transistor base
capacitor area. Platens were formed by
for the transistor and capacitor. Tabs
attached to make contact with the Ge
wafer is shown. The wafer was





- US Patent # 3,138,743
  filed Feb. 6, 1959

From Dennis Buss, Texas Instruments, ICECS, Malta 2001 presentation

Moore's Law Continues

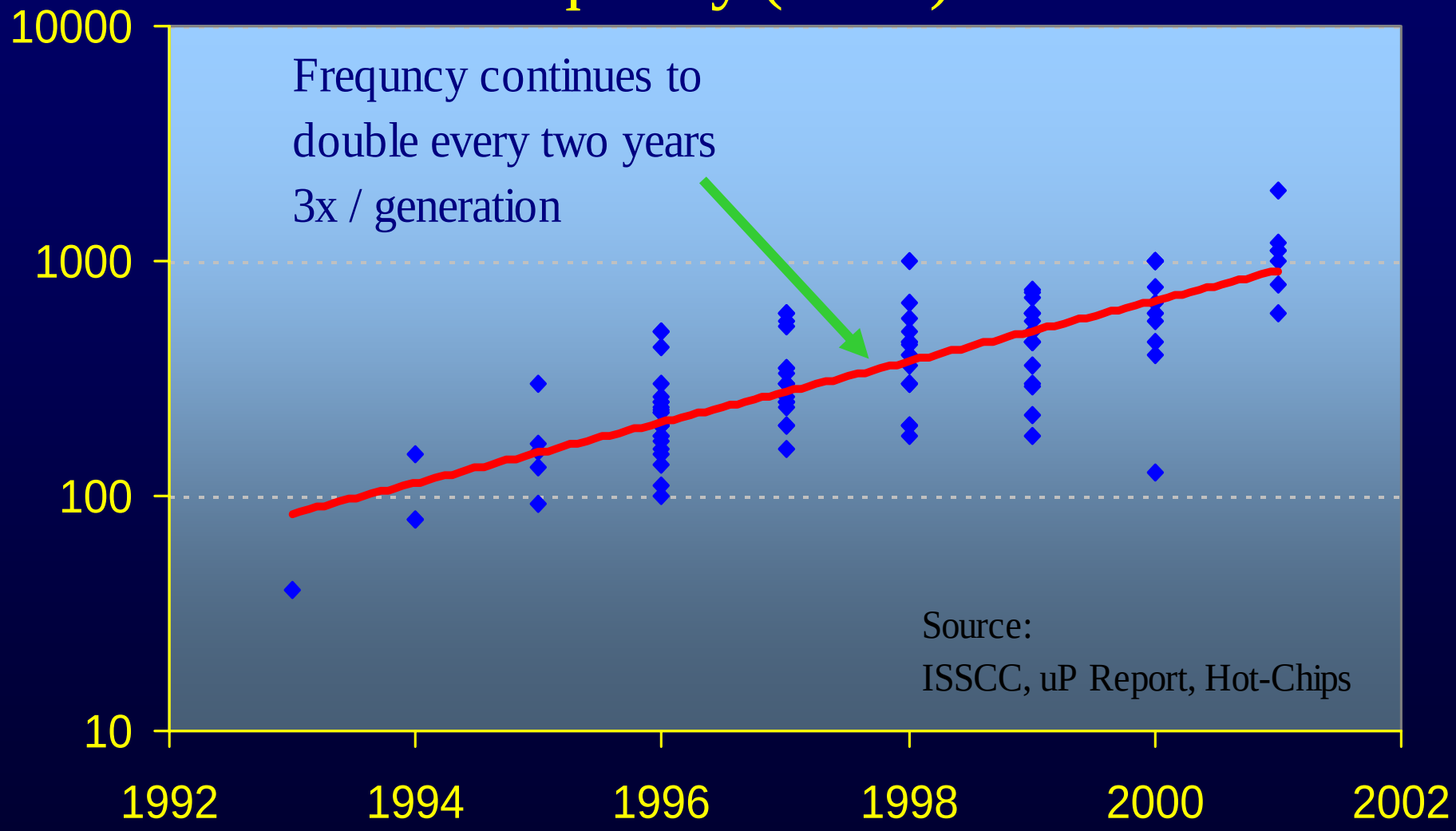
- Transistors per IC doubles every two years
- In less than 30 years
  - 1,000X decrease in size
  - 10,000X increase in performance
  - 10,000,000X reduction in cost
- Heading toward 1 billion transistors before end of this decade

From Robert Yung, Intel Corp., ESSCIRC, Firenze 2002 presentation

Processor Design Challenges

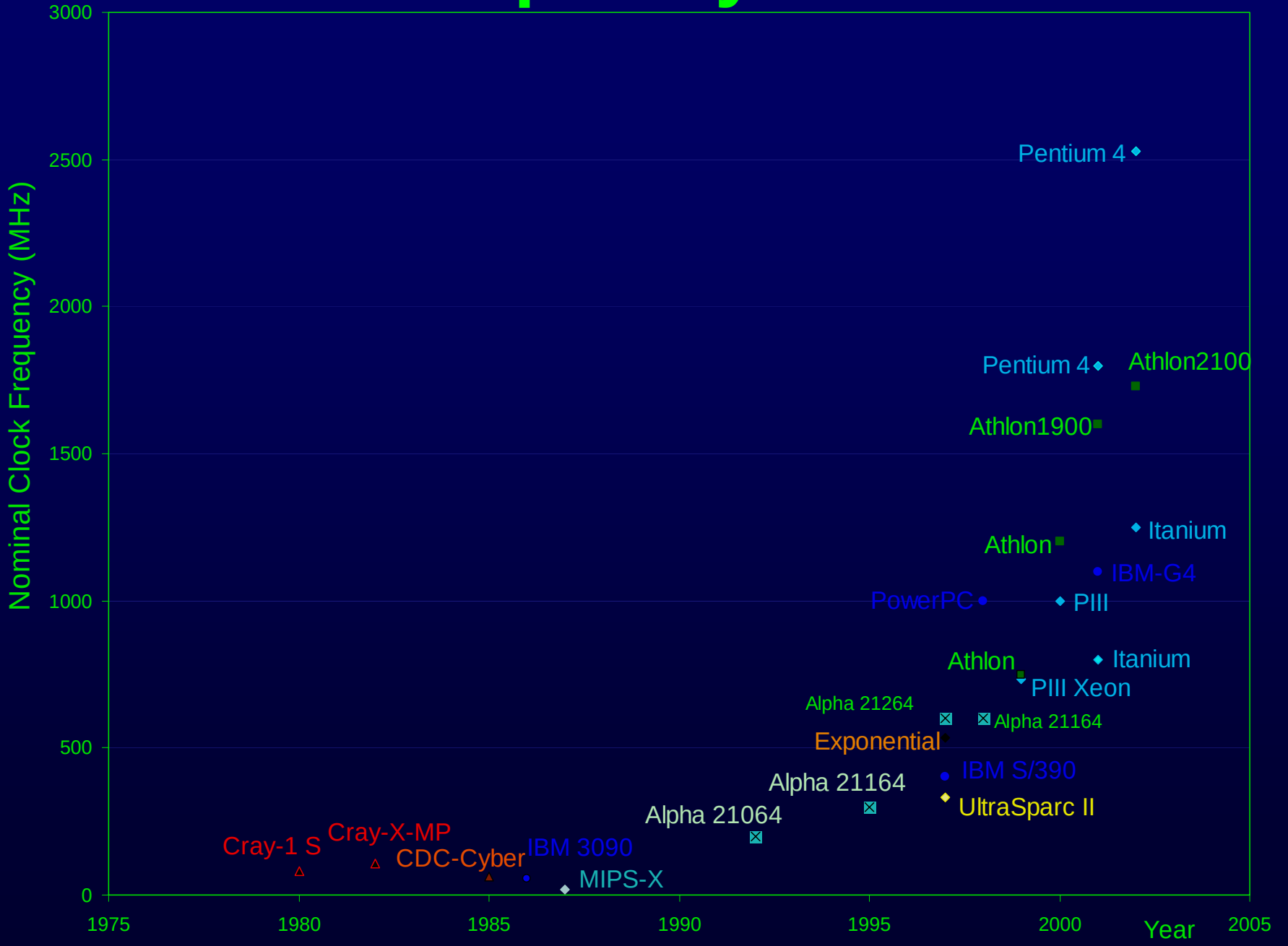
- *Will technology be able to keep up ?*
- *Will the bandwidth keep up ?*
- *Will the power be manageable ?*
- *Can we deliver the power ?*
- *What will we do with all those transistors ?*

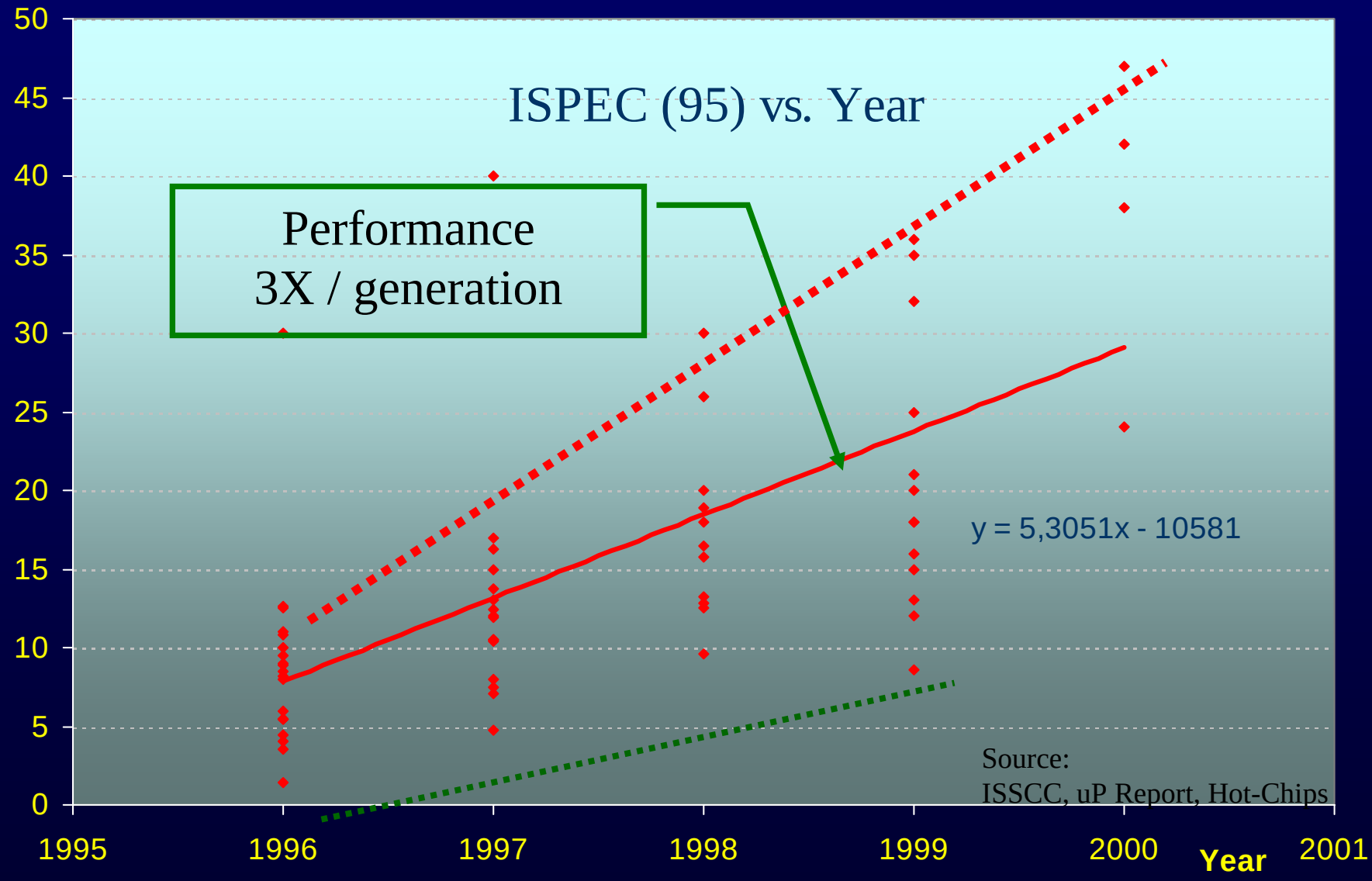
Clock Frequency (MHz) vs. Year

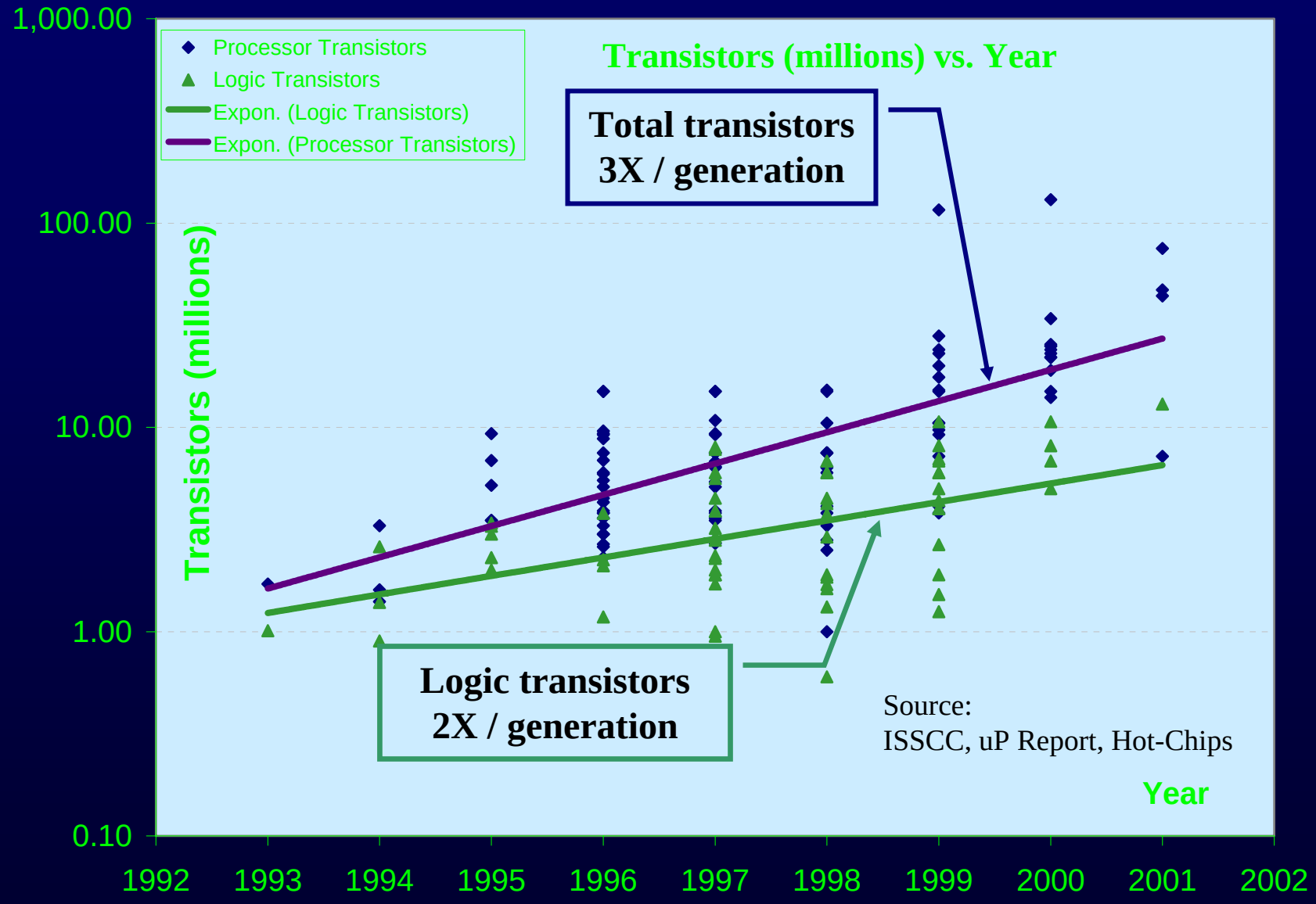


Clock frequency trends

ISSCC-2002



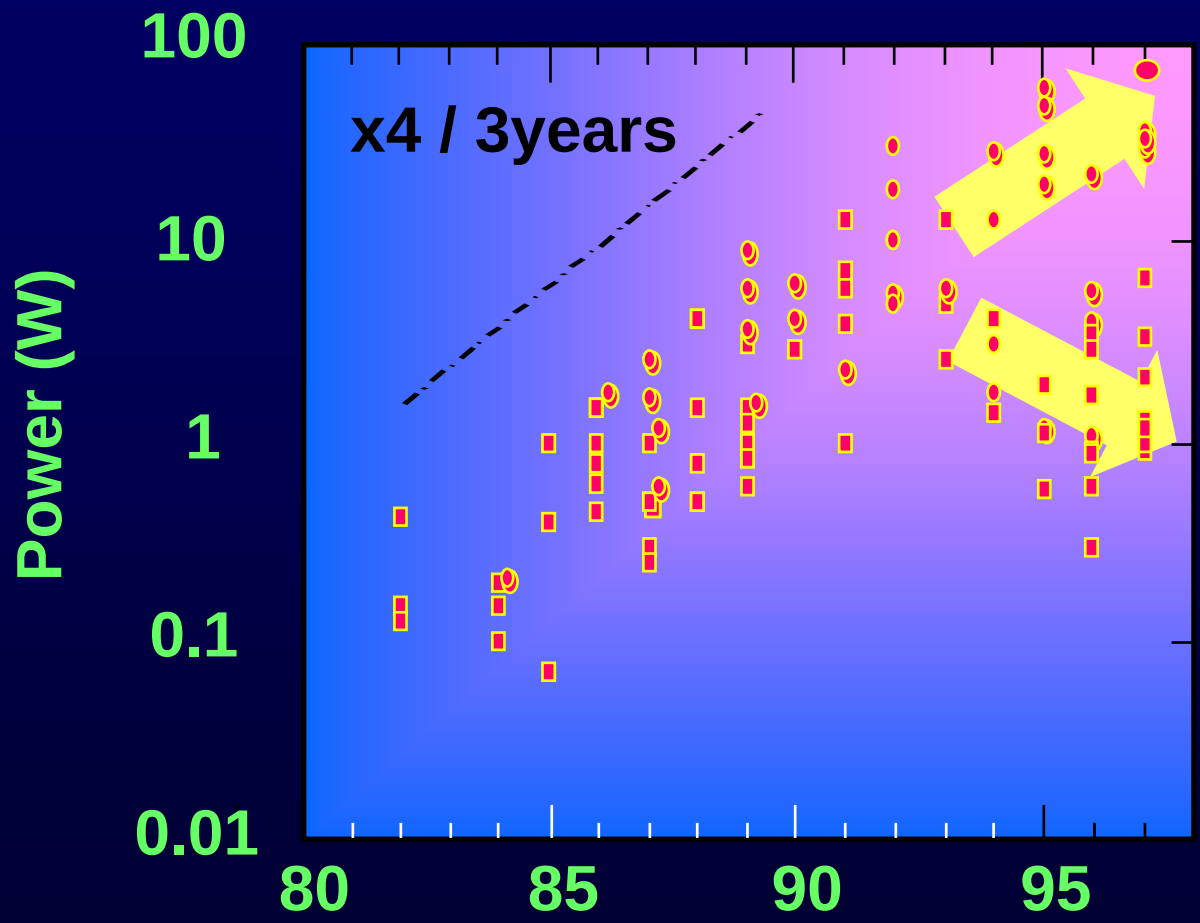

ISPEC (95) vs. Year
Performance
3X / generation
y = 5,3051x - 10581
Source:
ISSCC, uP Report, Hot-Chips
50
45
40
35
30
25
20
15
10
5
0
1995
1996
1997
1998
1999
2000
2001
Year


Transistors (millions) vs. Year
Processor Transistors
Logic Transistors
Expon. (Logic Transistors)
Expon. (Processor Transistors)
Total transistors
3X / generation
Logic transistors
2X / generation
Source:
ISSCC, uP Report, Hot-Chips
Transistors (millions)
Year
1,000.00
100.00
10.00
1.00
0.10
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002

Processor Design Challenges

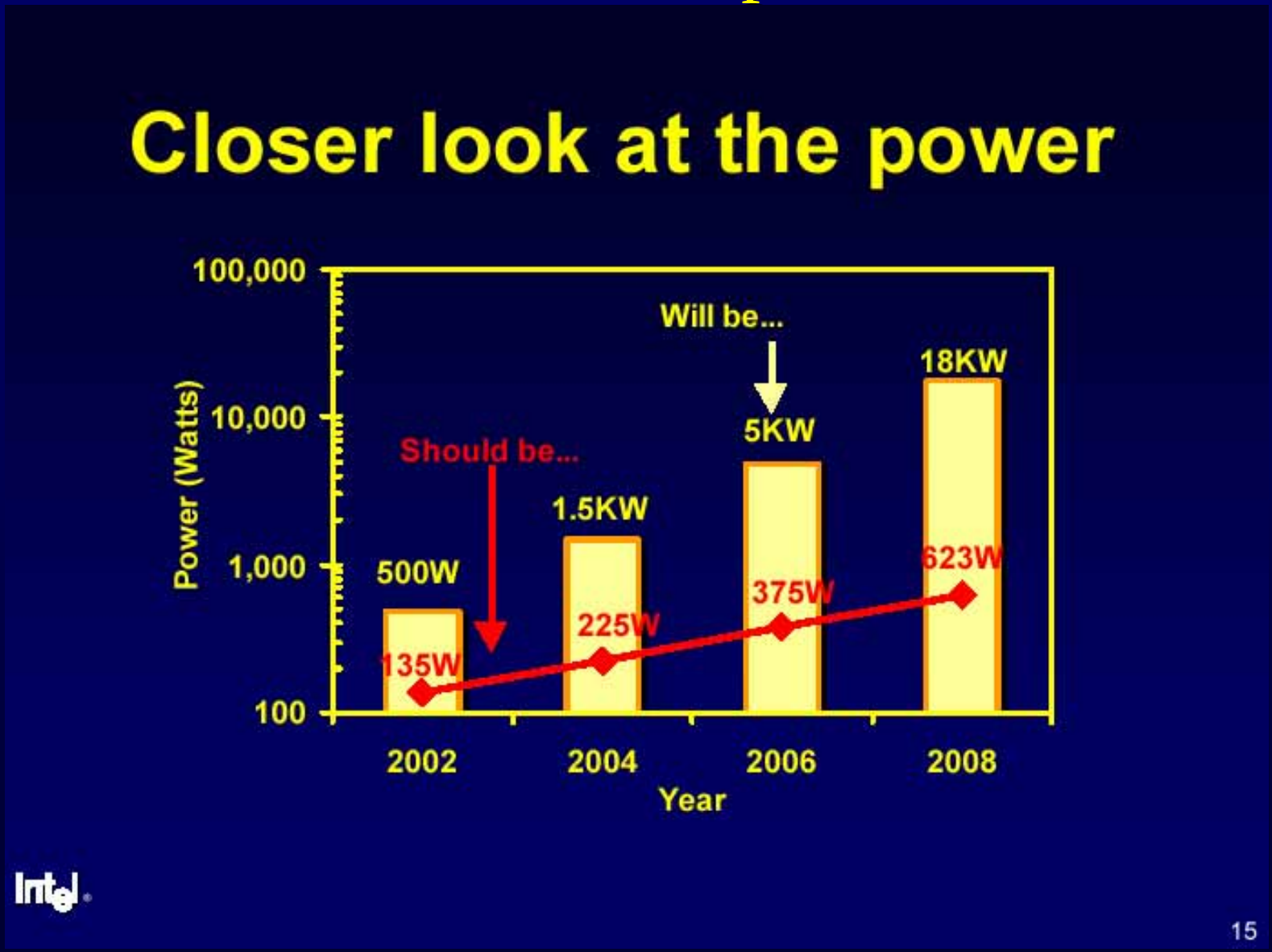
- Performance seems to be tracking frequency increase
- Where are the transistors being used ?
- 3X per generation growth in transistors seems to be uncompensated as far as performance is concerned

Well, it will make up in power …



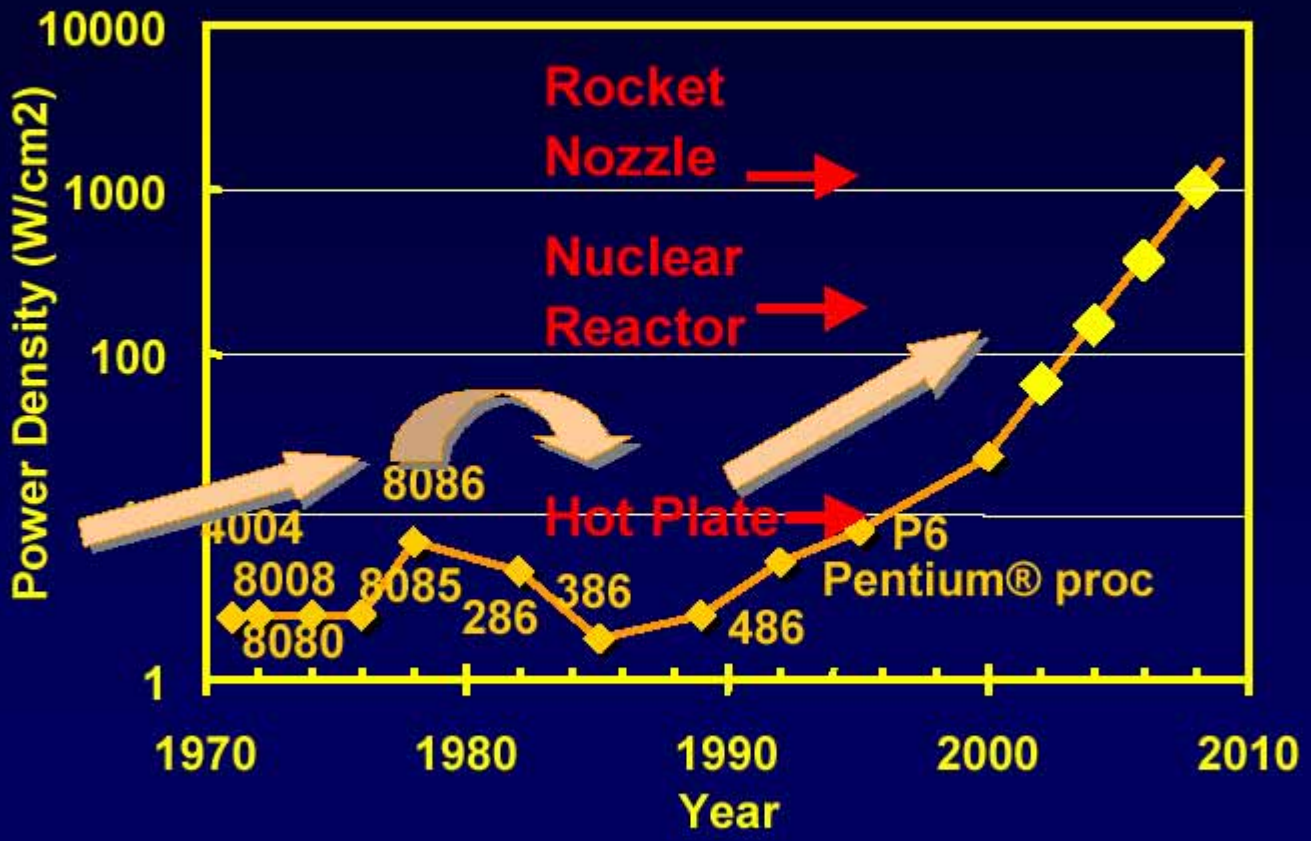
Courtesy of Sakurai Sensei

Gloom and Doom predictions



Source: Shekhar Borkar, Intel

Power density will increase



Power density too high to keep junctions at low temp

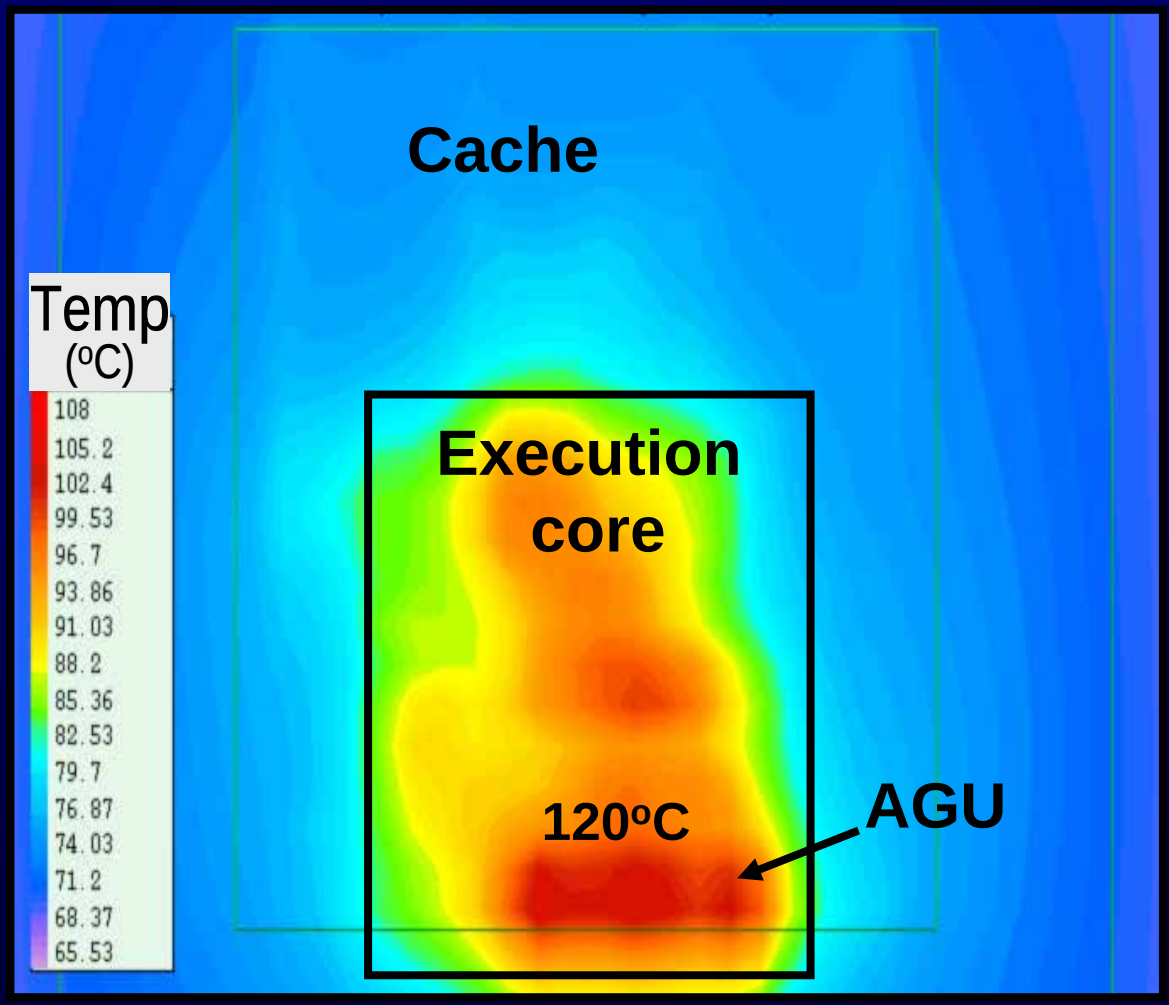
intel

Source: Intel

Power Density

*courtesy of Intel Corp.

Processor thermal map

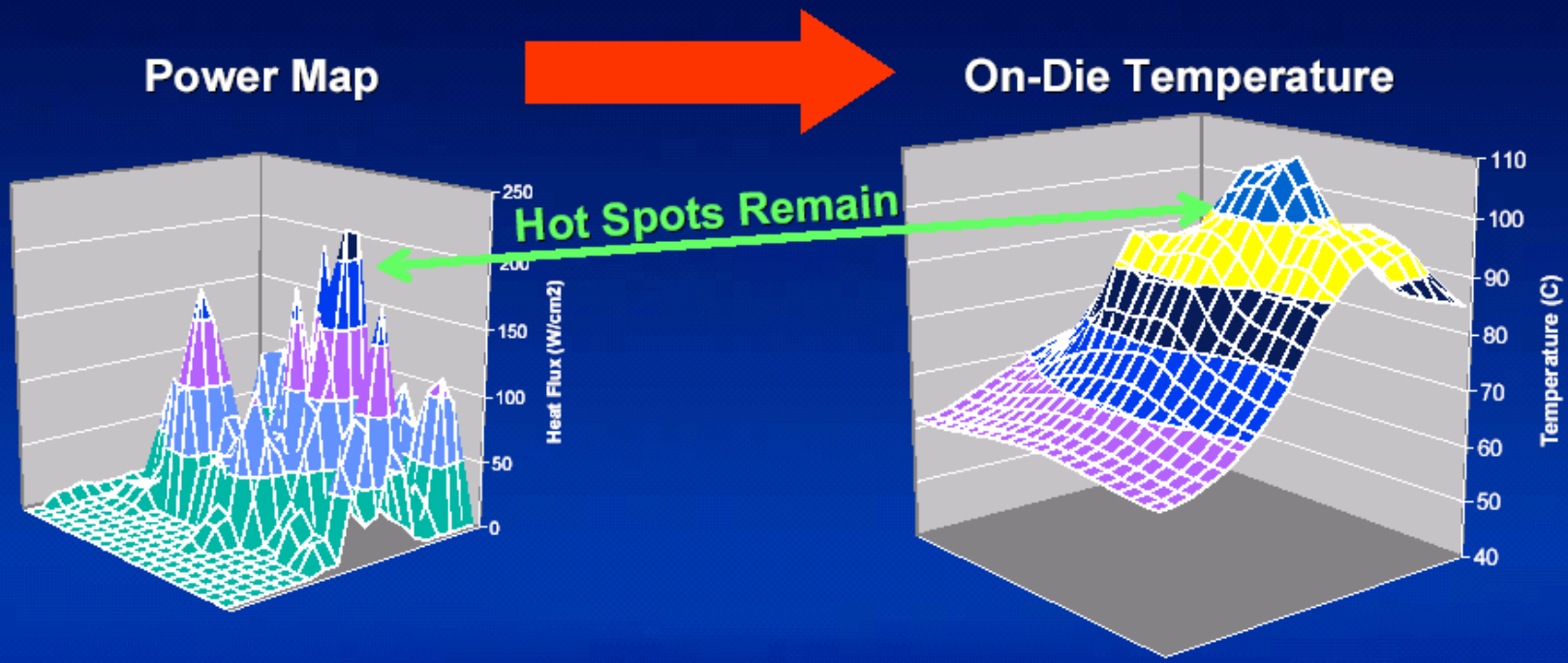


AGUs: performance and peak-current limiters

High activity ⇒ thermal hotspot

Goal: high-performance energy-efficient design

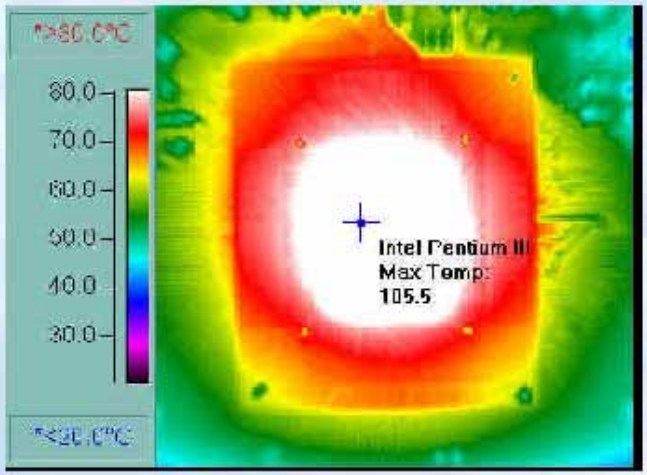
Power Density: The Future



- With high power density, cannot assume uniformity
  - As die temperature increases, CMOS logic slows down
  - At high die temp., long-term reliability can be compromised

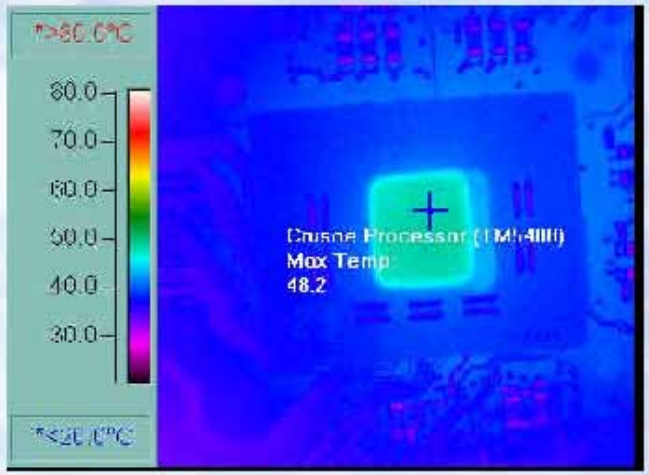
TransMeta Example

Processor Thermal Comparison



Pentium III Playing DVD

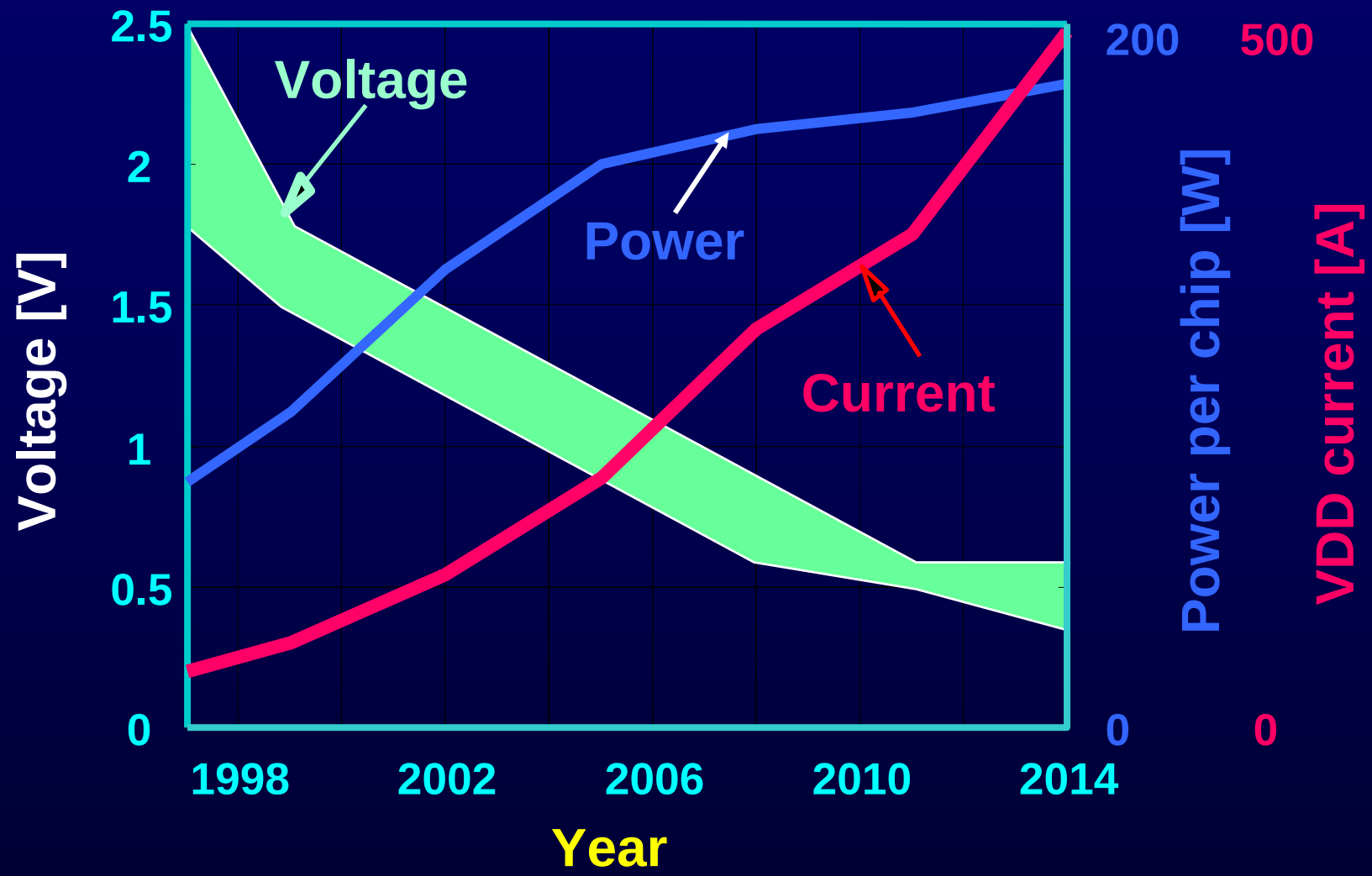
105.5° C
221.9° F



Crusoe Processor Playing DVD

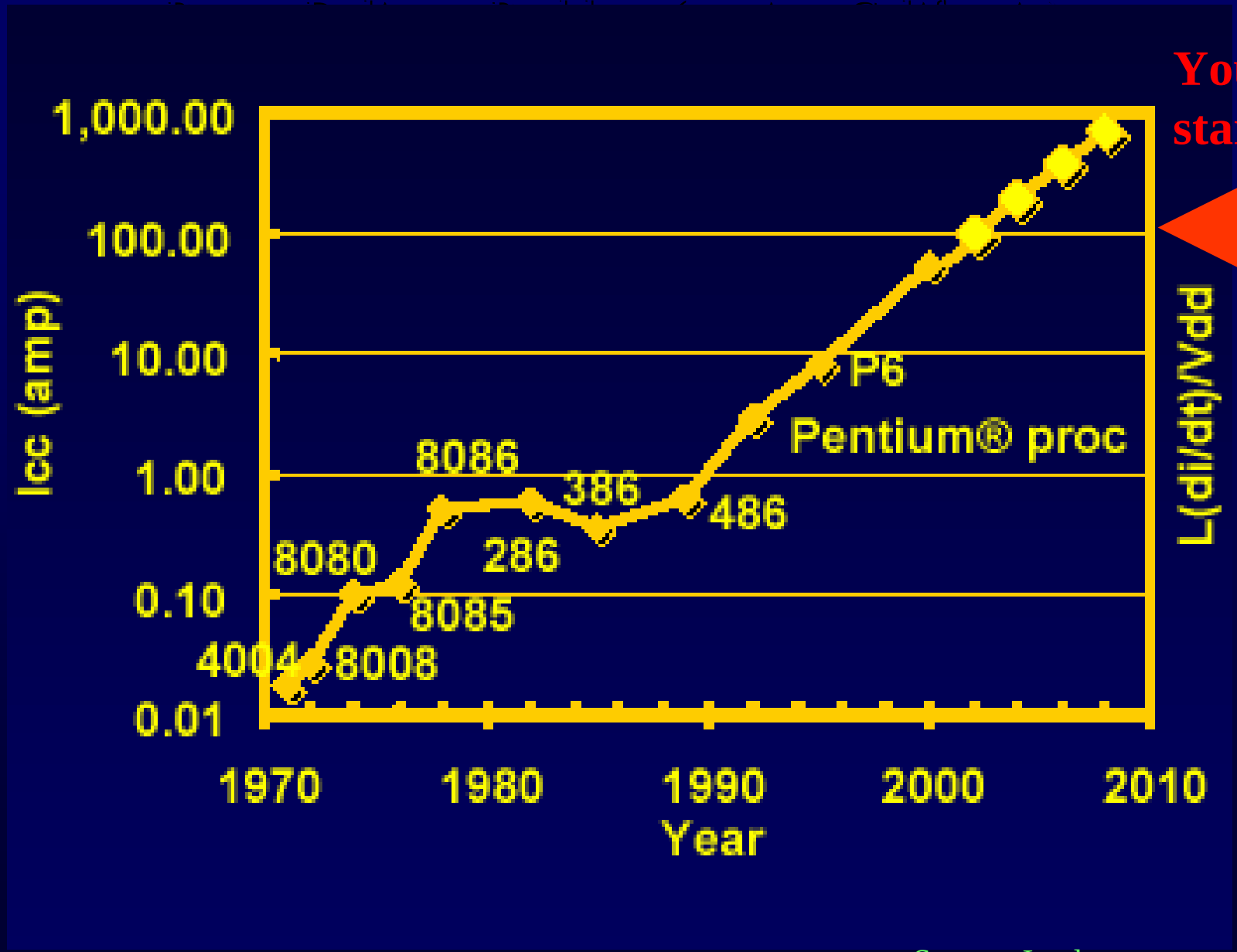
48.2° C
118.8° F

VDD, Power and Current Trend

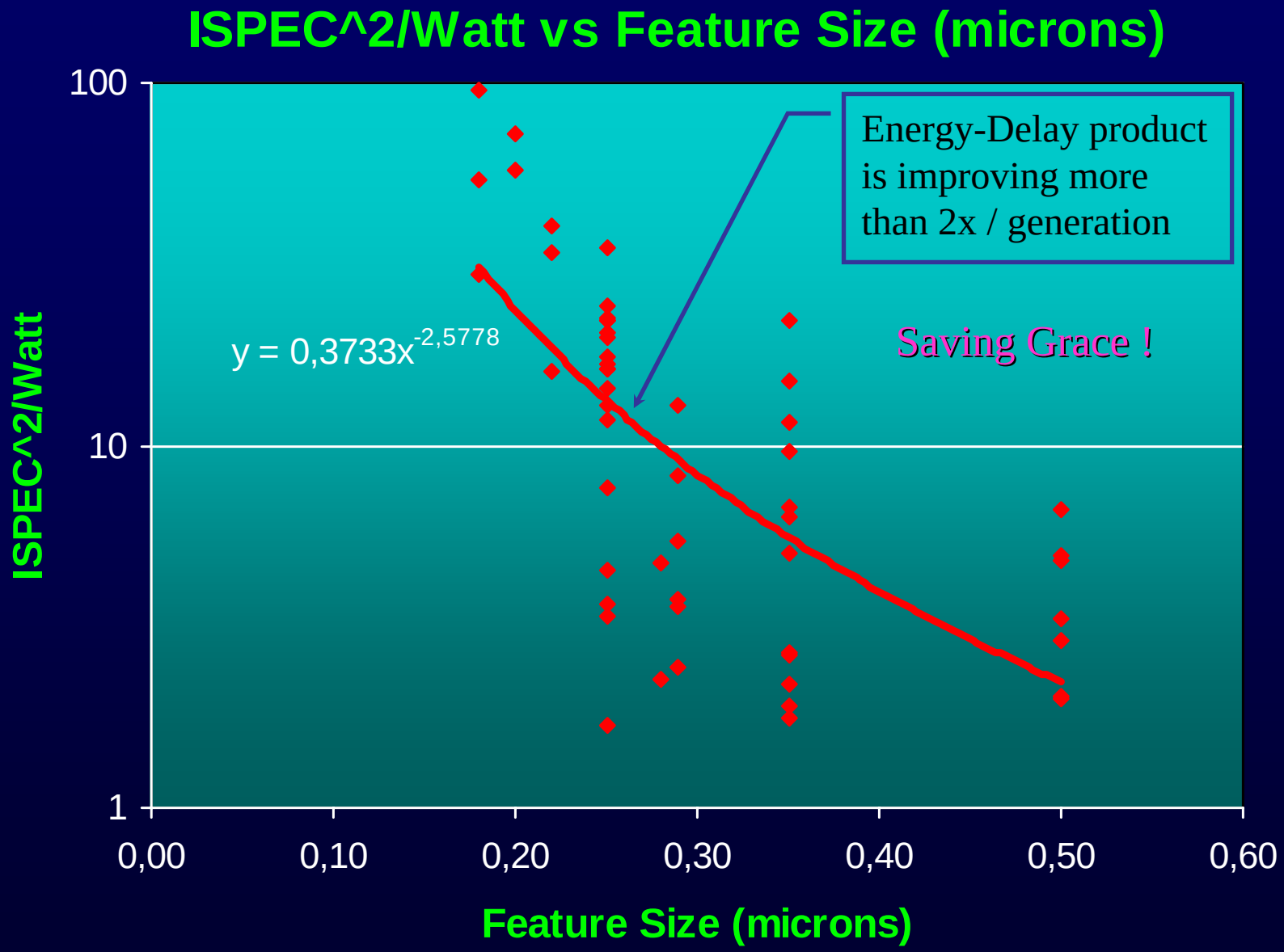


International Technology Roadmap for Semiconductors 1999 update sponsored by the Semiconductor Industry Association in cooperation with European Electronic Component Association (EECA) , Electronic Industries Association of Japan (EIAJ), Korea Semiconductor Industry Association (KSIA), and Taiwan Semiconductor Industry Association (TSIA)

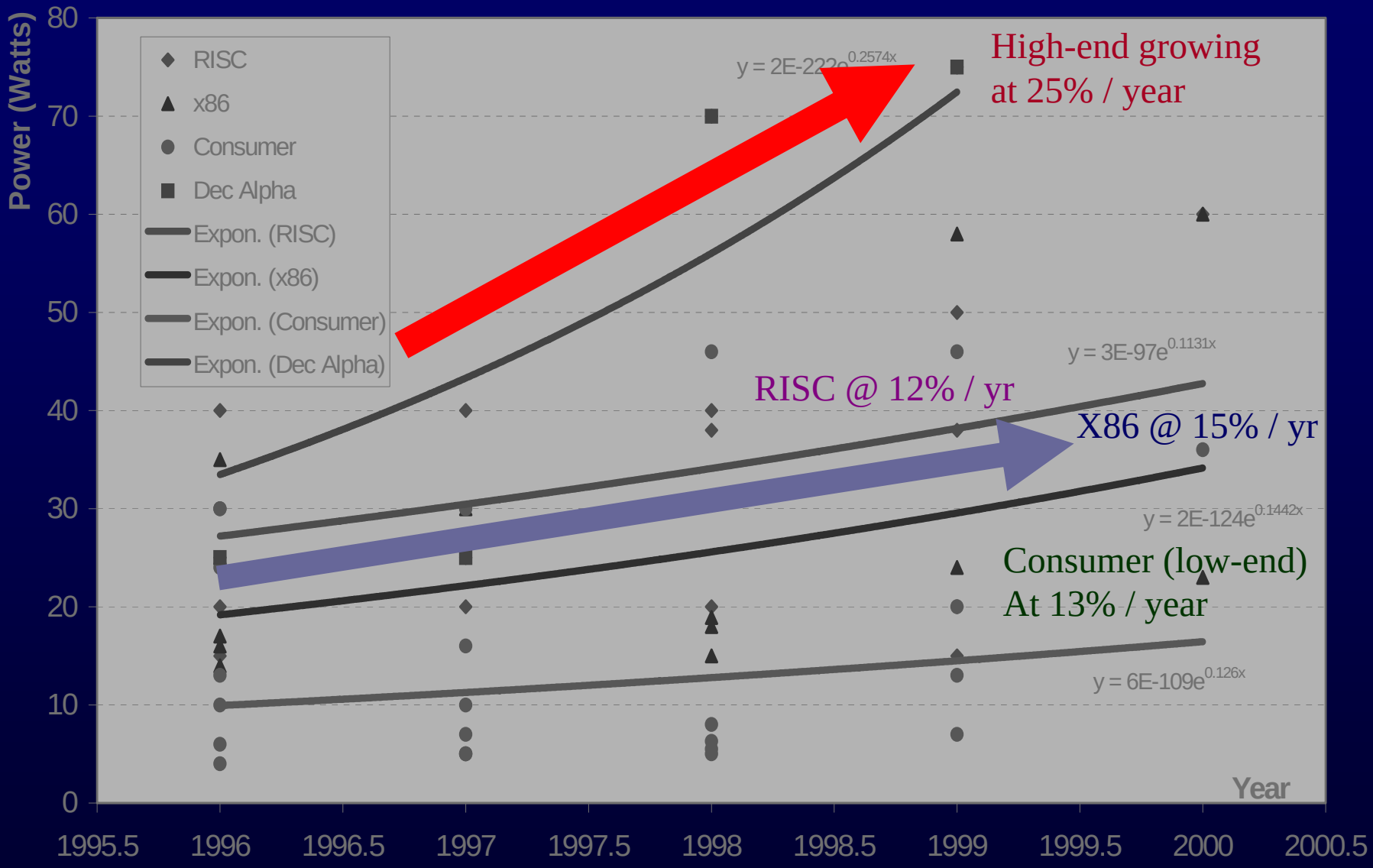
(* Taken from Sakurai's ISSCC 2001 presentation)

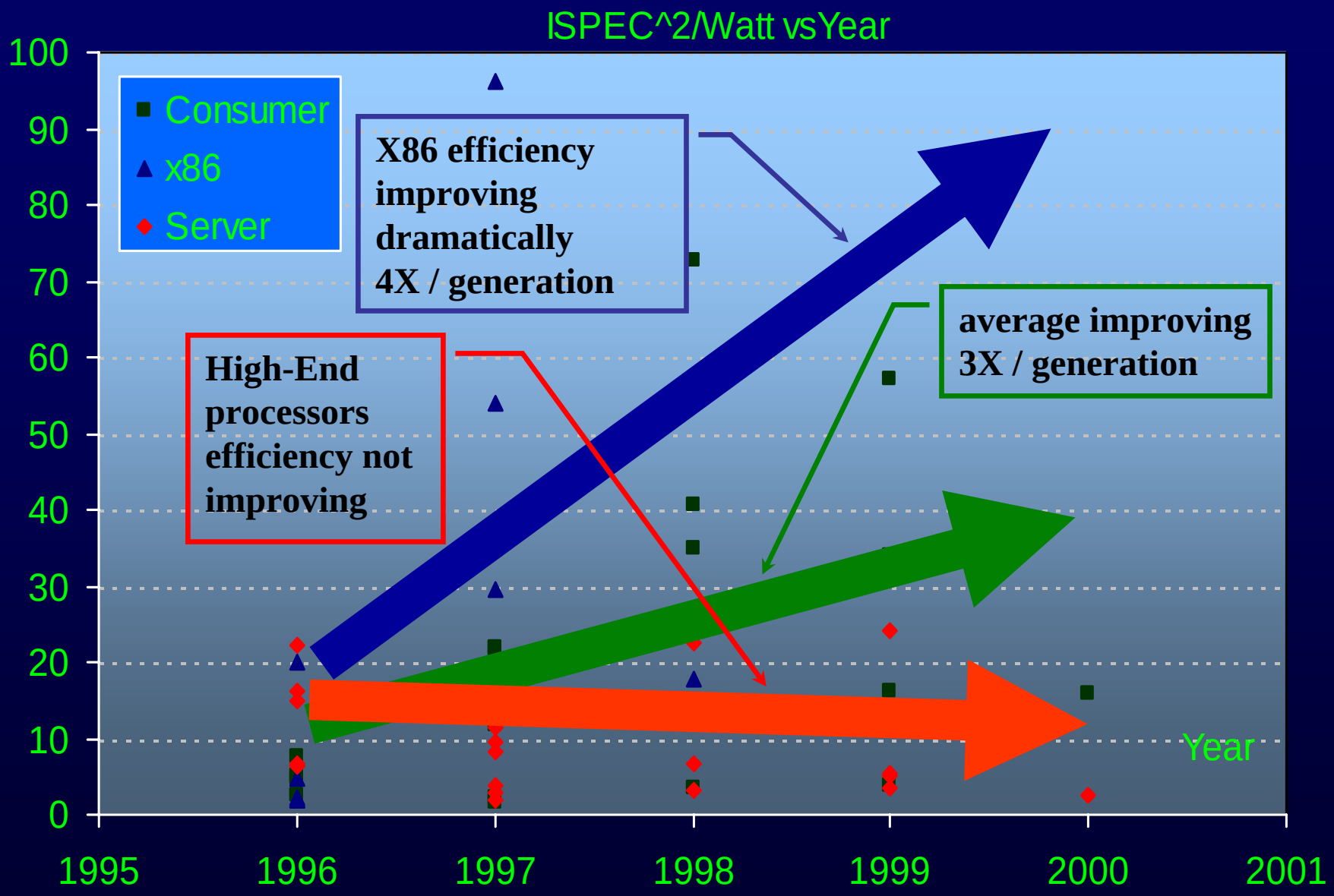


Source: Intel


ISPEC^2/Watt vs Feature Size (microns)
100
10
1
ISPEC^2/Watt
y = 0,3733x-2,5778
Energy-Delay product is improving more than 2x / generation
Saving Grace !
0,00
0,10
0,20
0,30
0,40
0,50
0,60
Feature Size (microns)

Power versus Year




ISPEC^2/Watt vsYear
100
90
80
70
60
50
40
30
20
10
0
1995
1996
1997
1998
1999
2000
2001
Year
Consumer
x86
Server
X86 efficiency improving dramatically 4X / generation
High-End processors efficiency not improving
average improving 3X / generation

Trend in L di/dt:

- di/dt is roughly proportional to

  *I * f*, where *I* is the chip's current and *f* is the clock frequency

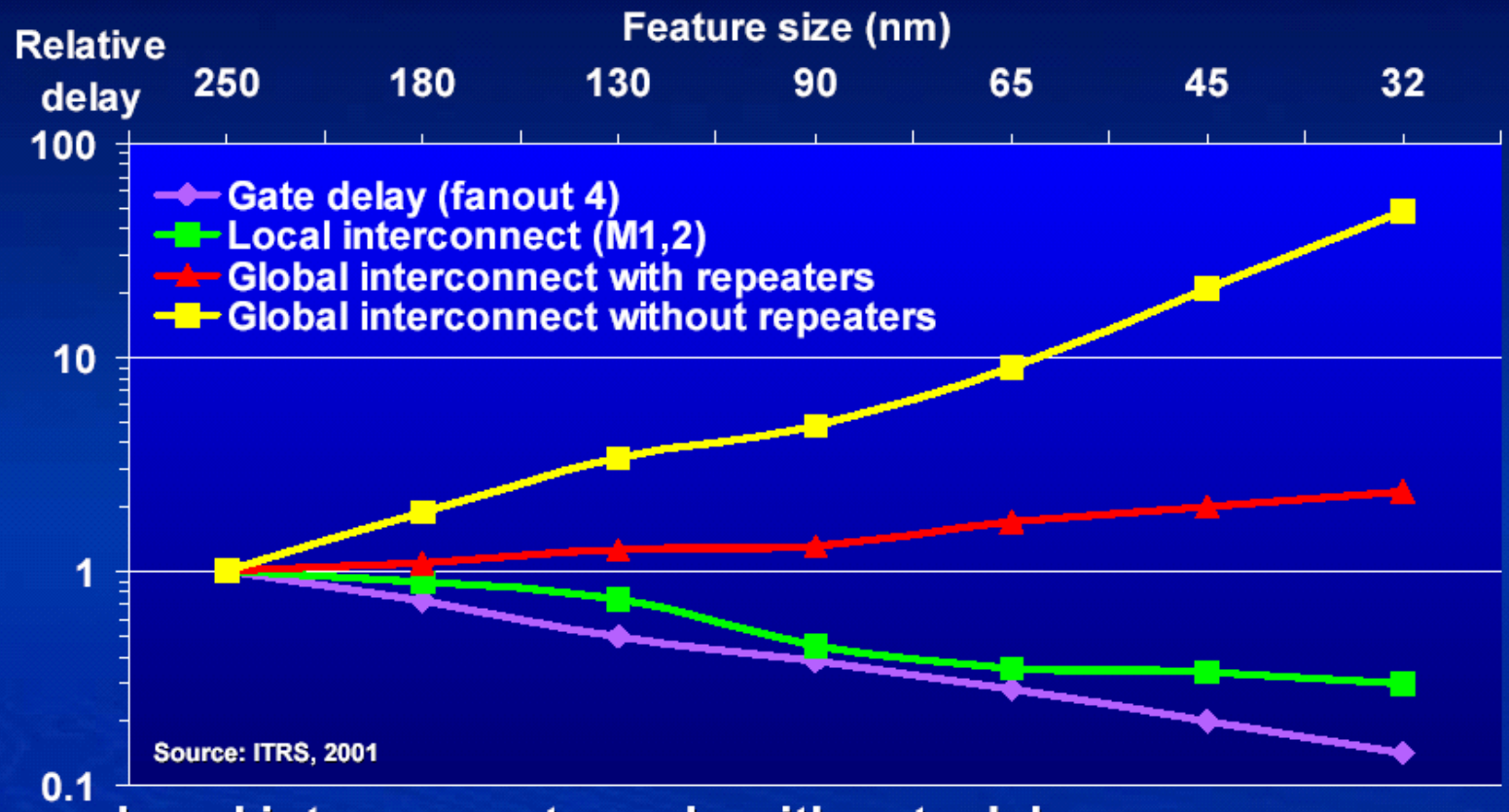
  or *I * Vdd * f / Vdd = P * f / Vdd,* where *P* is the chip's power.
- The trend is:

  P ↑ *f* ↑↑ *Vdd* ↓

  on-chip L ↑ *package L* slightly decreases
- Therefore, L di/dt fluctuation increases significantly.

Source: Shen Lin, Hewlett Packard Labs

On-chip Interconnect Trend



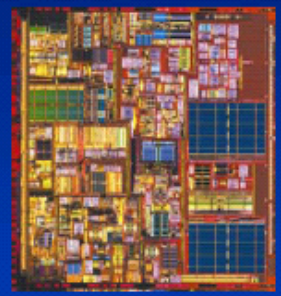
- Local interconnects scale with gate delay
- Intermediate interconnects benefit from low k material
- Global interconnects do not scale because of RC!

More metal layers may not help

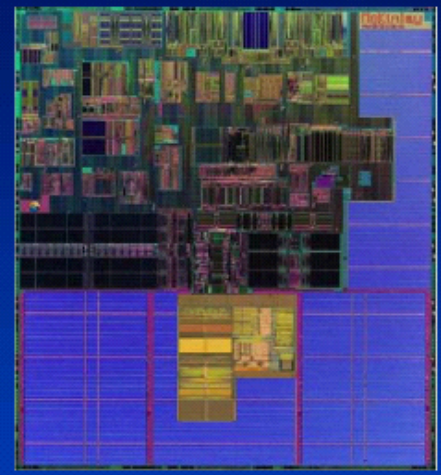

Metal 4
Metal 3
Metal 2

Microprocessor Evolution

- 4004
  - 1971
  - 2300 transistors
  - 10um process
  - 2”, 50mm wafer
  - 12mm^2
  - 108 kHz



- Pentium® 4 processor
  - 2002 (31 yrs)
  - 55M (24K X)
  - 0.13um (1/77 X)
  - 12”, 300mm (6X)
  - 142mm^2 (12 X)
  - 2.8 GHz (26K X)



- Itanium® 2 processor
  - 2002 (31 yrs)
  - 220M (96K X)
  - 0.18um (1/55 X)
  - 12”, 300mm (6X)
  - 421mm^2 (35 X)
  - 1 GHz (9K X)

What to do with all those transistors ?

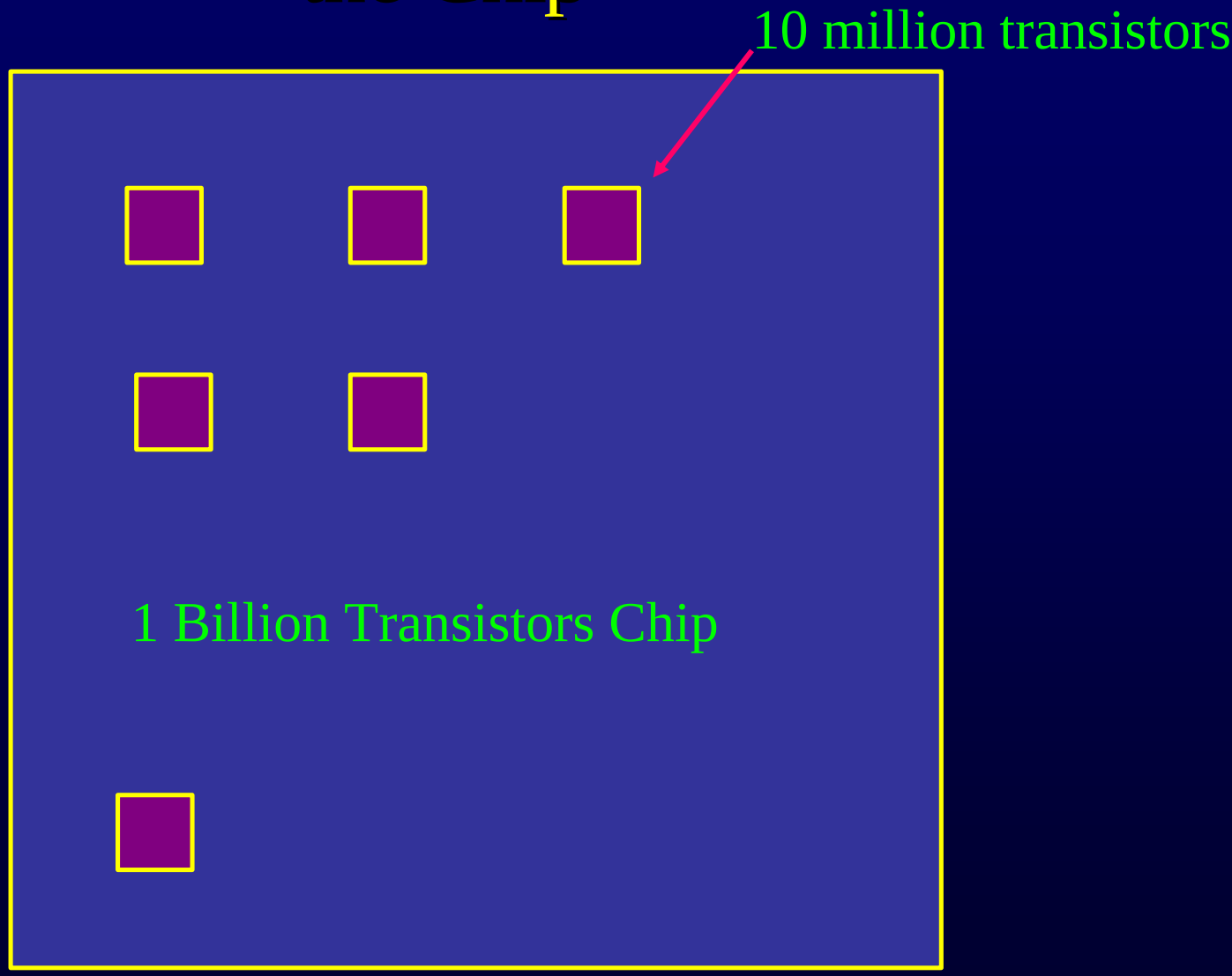
- We have reached 220 Million
- We will reach 1 Billion in the next 5 years !
- Memory transistors will save us from power crisis
- What should the architecture look like ?

Synchronous / Asynchronous Design on the Chip

- 1 Billion transistors on the chip by 2005-6
- 64-b, 4-way issue logic core requires ~2 Million

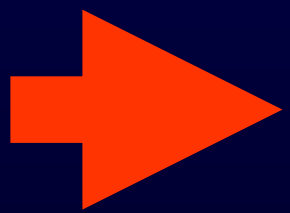
Feature	Digital 21164	MIPS 10000	PowerPC 620	HP 8000	Sun UltraSpar
Frequency	500 MHz	200 MHz	200 MHz	180 MHz	250 MHz
Pipeline Stages	7	5-7	5	7-9	6-9
Issue Rate	4	4	4	4	4
Out-of-Order Exec.	6 loads	32	16	56	none
Register Renam. (int/FP)	none/8	32/32	8/8	56	none
Transistors/ Logic transistors	9.3M/ 1.8M	5.9M/ 2.3M	6.9M/ 2.2M	3.9M*/ 3.9M	3.8M/ 2.0M
SPEC95 (Intg/FlPt)	12.6/18.3	8.9/17.2	9/9	10.8/18.3	8.5/15
Power	25W	30W	30W	40W	20W
SpecInt/Watt	0.5	0.3	0.3	0.27	0.43
*1/Energy*Delay*	6.4	2.6	2.7	2.9	3.6

Synchronous / Asynchronous Design on the Chip



What Drives the Architecture ?

- **Processor to memory speed gap continues to widen**
- **Transistor densities continue to increase**
- **Application fine-grain parallelism is limited**
- **Time and resources required for more complex designs is increasing**
- Time-to-market is as critical as ever

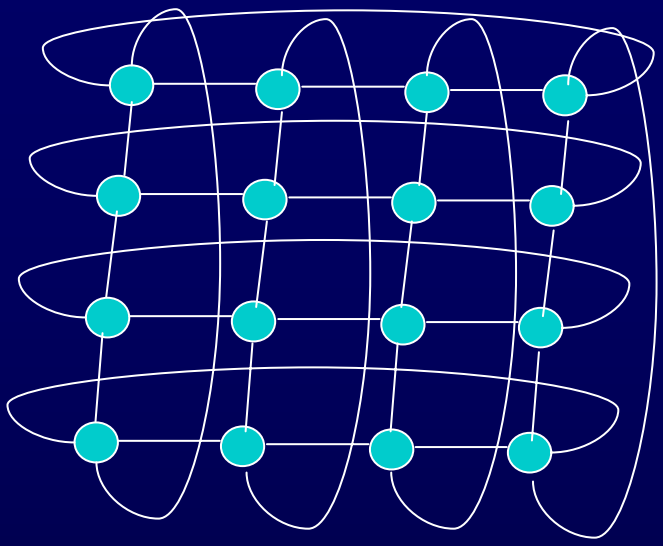


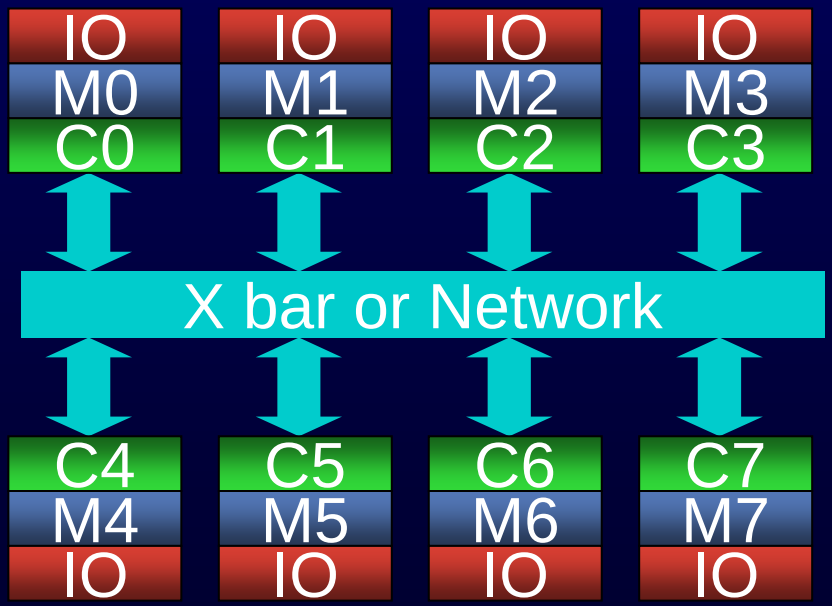
Multiprocessing on the Chip ?

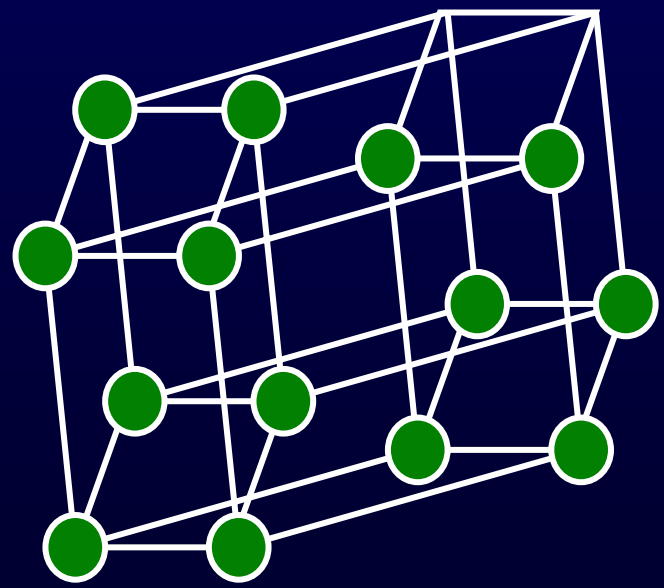
ccNUMA Design

Source: Pete Bannon, DEC

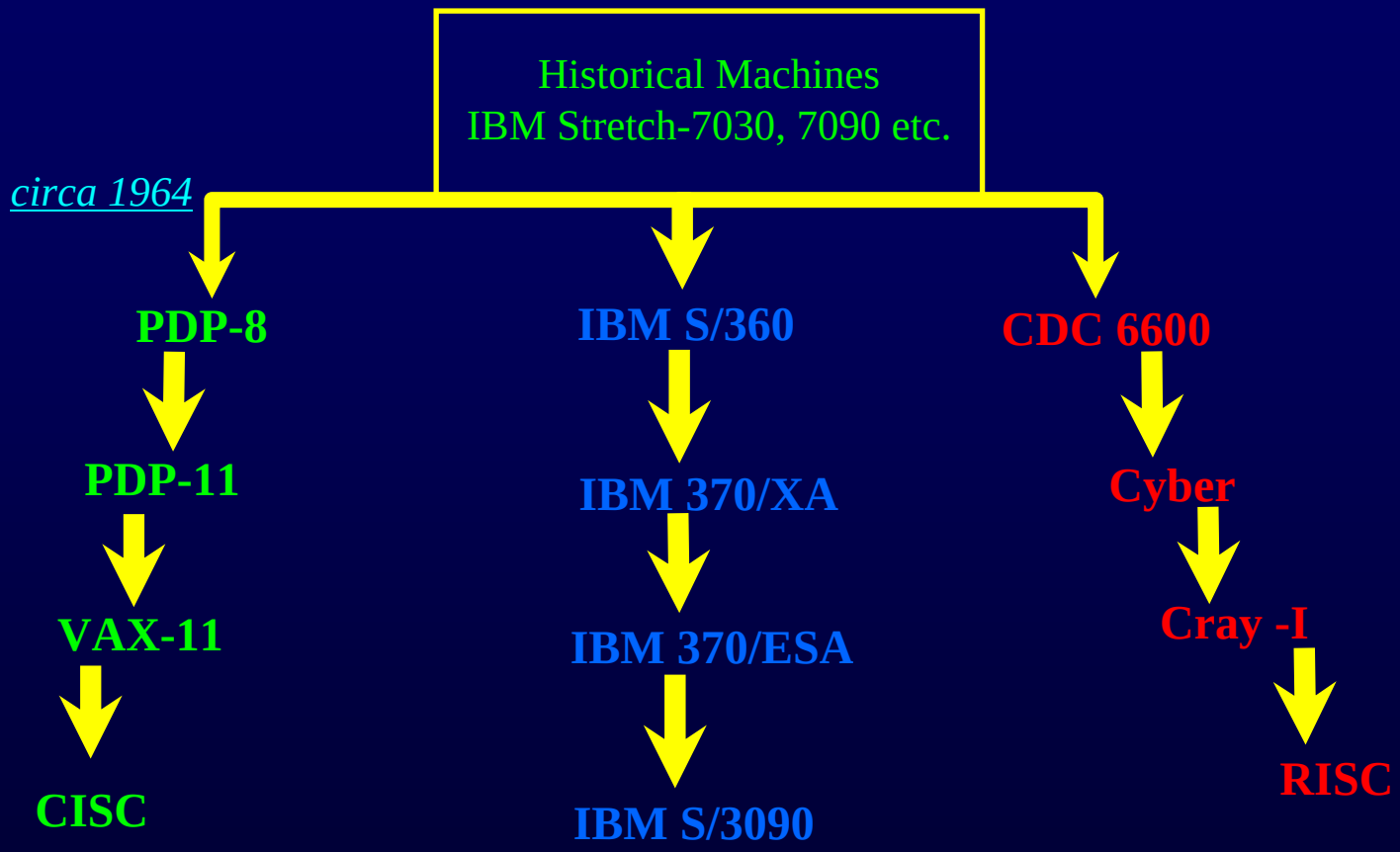
- Metrics
- Topologies
- Cache Coherence







A bit of history



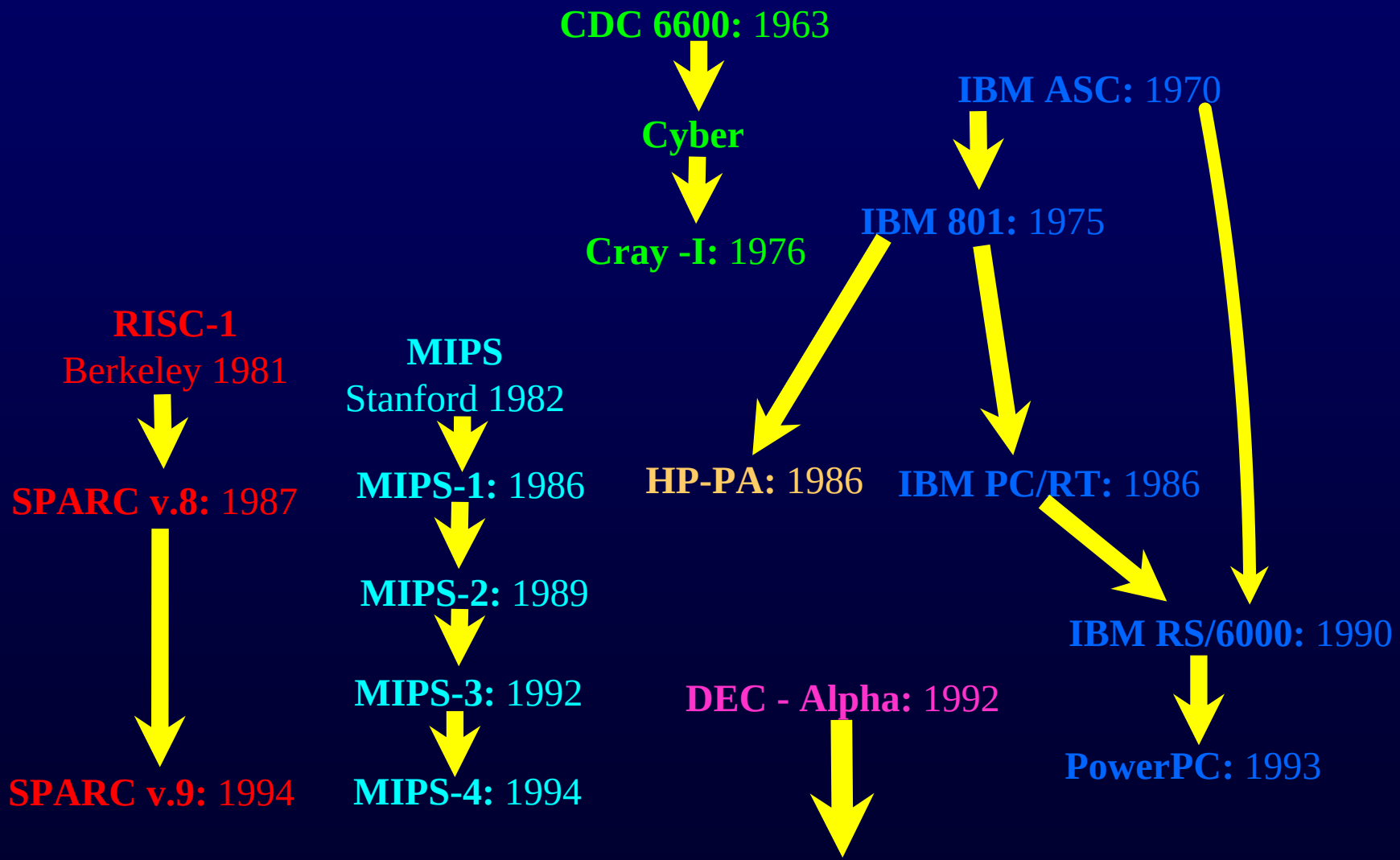
Important Features Introduced

- Separate Fixed and Floating point registers (IBM S/360)
- Separate registers for address calculation (CDC 6600)
- Load / Store architecture (Cray-I)
- Branch and Execute (IBM 801)

Consequences:

- Hardware resolution of data dependencies (Scoreboarding CDC 6600, Tomasulo's Algorithm IBM 360/91)
- Multiple functional units (CDC 6600, IBM 360/91)
- Multiple operation within the unit (IBM 360/91)

RISC: History



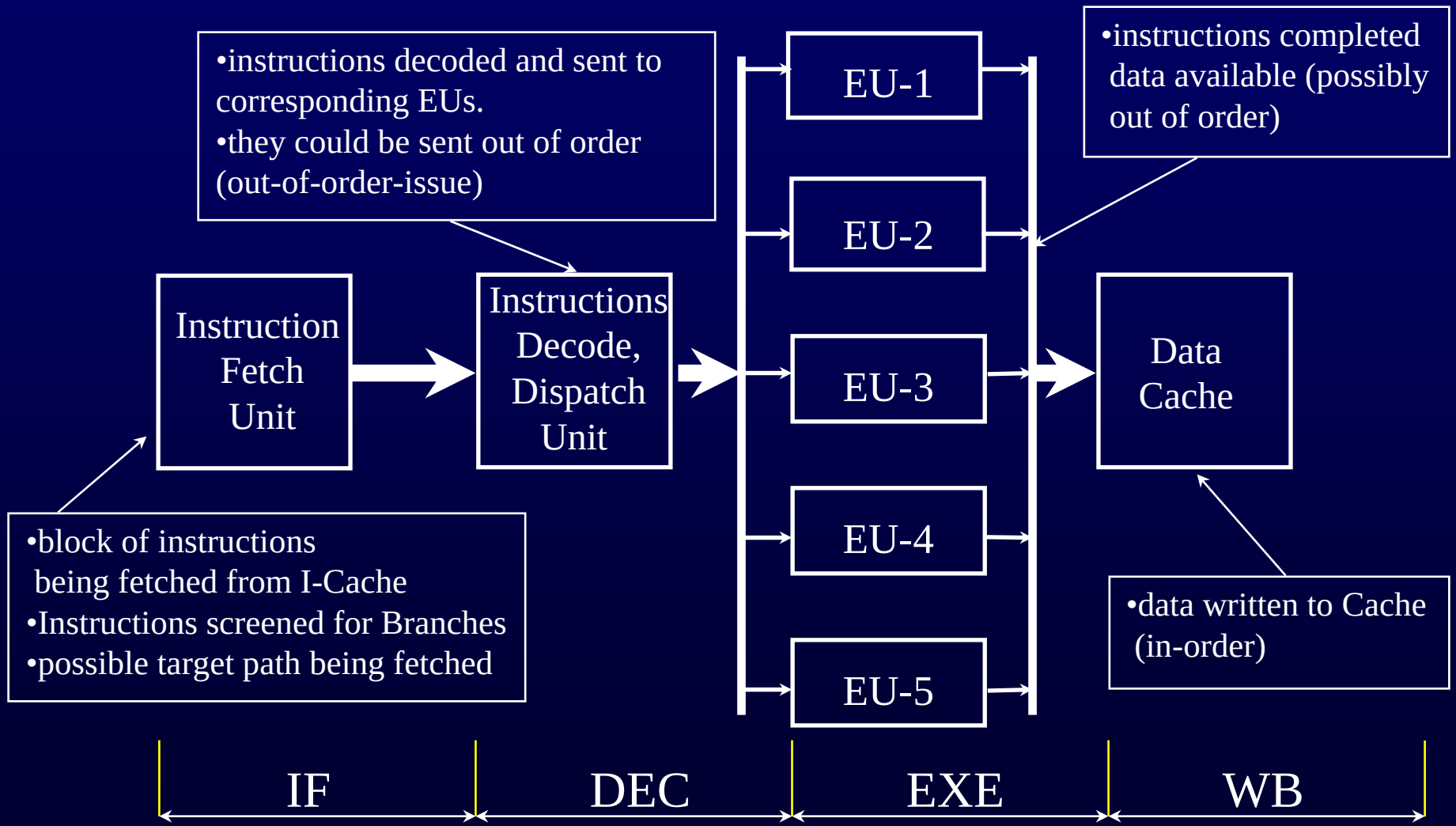
Reaching beyond the CPI of one: The next challenge

- With the perfect caches and no lost cycles in the pipeline the CPI ➔1.00
- The next step is to break the 1.0 CPI barrier and go beyond
- How to efficiently achieve more than one instruction per cycle ?

Again the key is exploitation of parallelism:

- on the level of independent functional units
- on the pipeline level

How does super-scalar pipeline look like ?



Super-scalar Pipeline

- One pipeline stage in super-scalar implementation may require more than one clock. Some operations may take several clock cycles.
- Super-Scalar Pipeline is much more complex - therefore it will generally run at lower frequency than single-issue machine.
- The trade-off is between the ability to execute several instructions in a single cycle and a lower clock frequency (as compared to scalar machine).

- *"Everything you always wanted to know about computer architecture can be found in IBM 360/91"*

Greg Grohosky, Chief Architect of IBM RS/6000

Techniques to Alleviate Branch Problem:
How can the Architecture help ?

- Conditional or Predicated Instructions

  Useful to eliminate BR from the code. If condition is *true* the instruction is executed normally if *false* the instruction is treated as NOP:

```
                    if (A=0) (S=T)                ; R1=A, R2=S, R3=T
        BNEZ R1, L
        MOV  R2, R3        replaced with:        CMOVZ R2,R3, R1
L:      ……..
```

- Loop Closing instructions: BCT (Branch and Count, IBM RS/6000)

  The loop-count register is held in the Branch Execution Unit - therefore it is always known in advance if BCT will be taken or not (loop-count register becomes a part of the machine status)

Super-scalar Issues: *Contention for Data*

Data Dependencies:

- Read-After-Write (RAW)
  - also known as: *Data Dependency* or *True Data Dependency*
- Write-After-Read (WAR)
  - knows as: *Anti Dependency*
- Write-After-Write (WAW)
  - known as: *Output Dependency*

WAR and WAW also known as: *Name Dependencies*

Super-scalar Issues: *Contention for Data*

True Data Dependencies: Read-After-Write (RAW)

An instruction *j* is data dependent on instruction *i* if:

- Instruction *i* produces a result that is used by *j*, or
- Instruction *j* is data dependent on instruction *k*, which is data dependent on instruction *I*

Examples:*

```
SUBI R1, R1, 8   ;decrement pointer
BNEZ R1, Loop  ; branch if R1 != zero
```

```
LD F0, 0(R1)          ;F0=array element
ADDD F4, F0, F2       ;add scalar in F2
SD 0(R1), F4          ; store result F4
```

*[Patterson-Hennessy]

Super-scalar Issues: *Contention for Data*

True Data Dependencies:

Data Dependencies are property of the program. The presence of dependence indicates the potential for hazard, which is a property of the pipeline (including the length of the stall)

A Dependence:

- indicates the possibility of a hazard
- determines the order in which results must be calculated
- sets the upper bound on how much parallelism can possibly be exploited.

i.e. we can not do much about True Data Dependencies in hardware. We have to live with them.

Super-scalar Issues: *Contention for Data*

Name Dependencies are:

- Anti-Dependencies (Write-After-Read, WAR)

  Occurs when instruction *j* writes to a location that instruction *i* reads, and *i* occurs first.

- Output Dependencies (Write-After-Write, WAW)

  Occurs when instruction *i* and instruction *j* write into the same location. The ordering of the instructions (write) must be preserved. (*j* writes last)

In this case there is no value that must be passed between the instructions. If the name of the register (memory) used in the instructions is changed, the instructions can execute simultaneously or be reordered.

The hardware CAN do something about *Name Dependencies* !

Super-scalar Issues: *Contention for Data*

Name Dependencies:

- Anti-Dependencies (Write-After-Read, WAR)

```
ADDD F4, F0, F2      ; F0 used by ADDD
LD F0, 0(R1)         ; F0 not to be changed before read by ADDD
```

- Output Dependencies (Write-After-Write, WAW)

```
LD F0, 0(R1)         ;LD writes into F0
ADDD F0, F4, F2      ; Add should be the last to write into F0
```

This case does not make much sense since F0 will be overwritten, however this combination is possible.

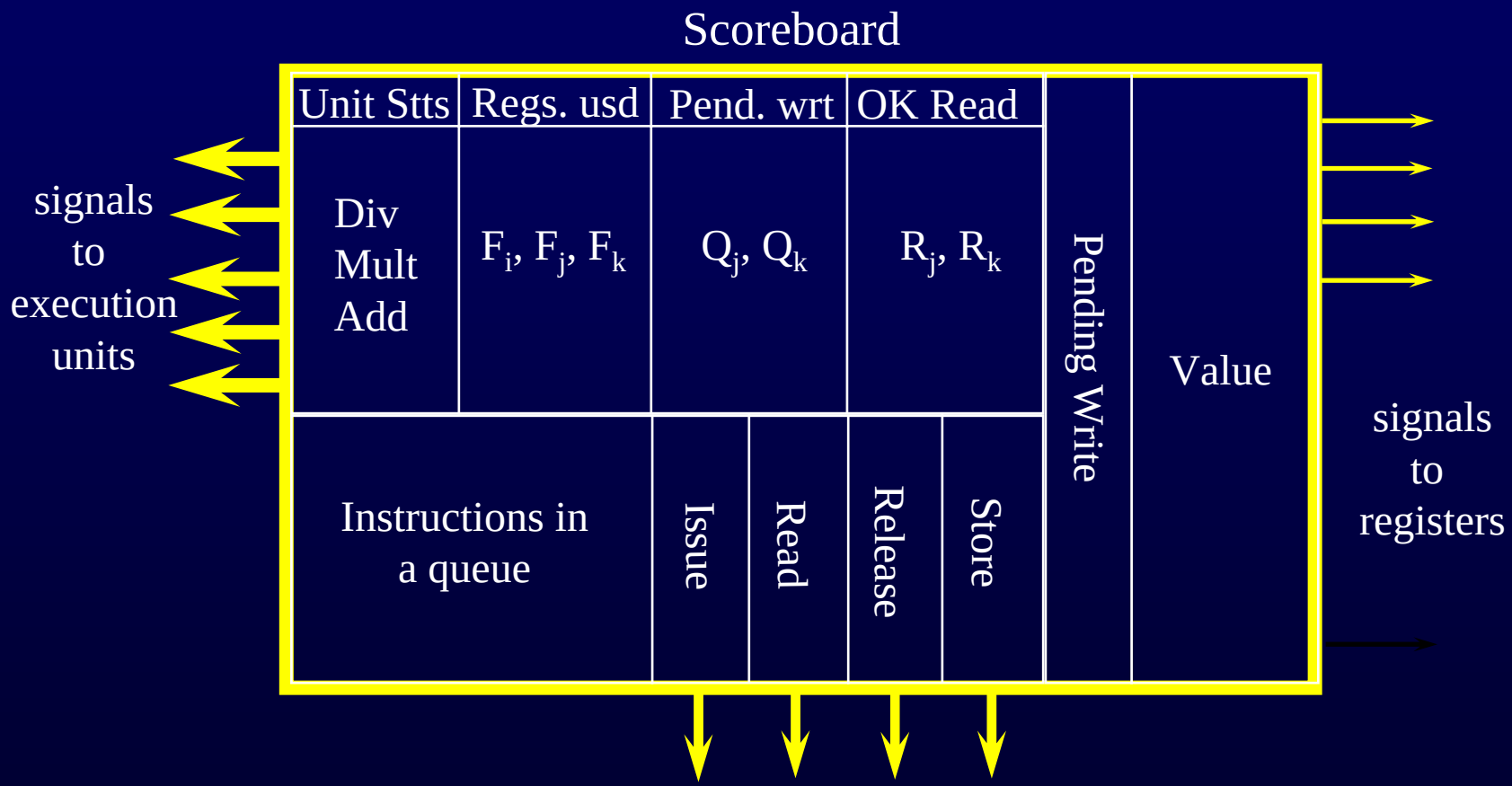
Instructions with name dependencies can execute simultaneously if reordered, or if the name is changed. This can be done: *statically* (by compiler) or *dynamically* by the hardware

Super-scalar Issues: *Dynamic Scheduling*

- Thornton Algorithm (Scoreboarding): CDC 6600 (1964)
  - *One common unit: Scoreboard which allows instructions to execute out of order, when resources are available and dependencies are resolved.*
- Tomasulo's Algorithm: IBM 360/91 (1967)
  - *Reservation Stations used to buffer the operands of instructions waiting to issue and to store the results waiting for the register. Common Data Buss (CDB) used to distribute the results directly to the functional units.*
- Register-Renaming: IBM RS/6000 (1990)
  - *Implements more physical registers than logical (architect). They are used to hold the data until the instruction commit.*

Super-scalar Issues: *Dynamic Scheduling*

Thornton Algorithm (Scoreboarding): CDC 6600



Super-scalar Issues: *Dynamic Scheduling*

Thornton Algorithm (Scoreboarding): CDC 6600 (1964)

Performance:

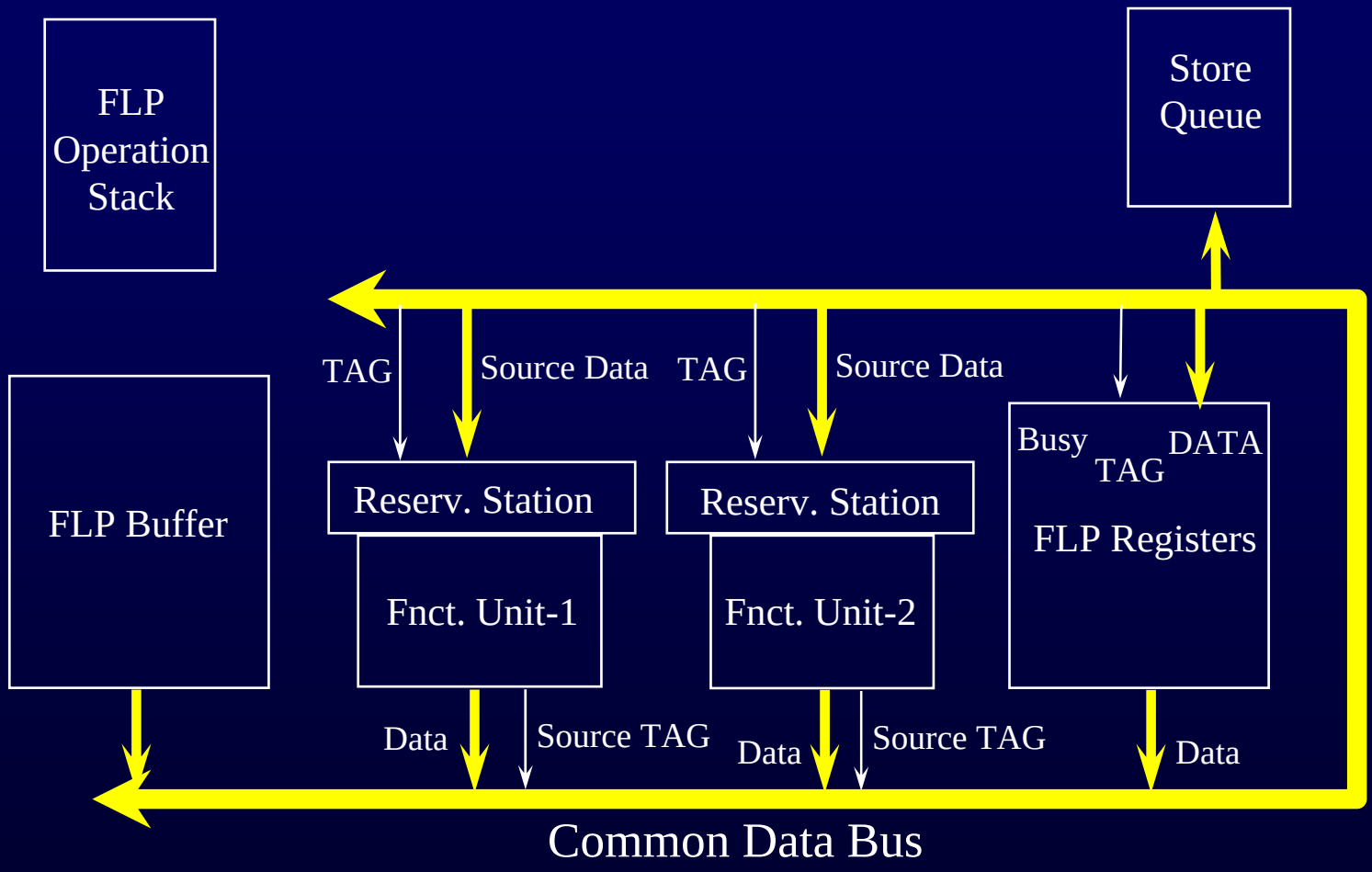
CDC6600 was 1.7 times faster than CDC6400 (no scoreboard, one functional unit) for FORTRAN and 2.5 faster for hand coded assembly

Complexity:

To implement the "scoreboard" as much logic was used as to implement one of the ten functional units.

Super-scalar Issues: *Dynamic Scheduling*

Tomasulo's Algorithm: IBM 360/91 (1967)



Super-scalar Issues: *Dynamic Scheduling*

Tomasulo's Algorithm: IBM 360/91 (1967)

The key to Tomasulo's algorithm are:

- Common Data Bus (CDB)

  CDB carries the data and the TAG identifying the source of the data

- Reservation Station
  - Reservation Station buffers the operation and the data (if available) awaiting the unit to be free to execute. If data is not available it holds the TAG identifying the unit which is to produce the data. The moment this TAG is matched with the one on the CDB the data is taken and the execution will commence.
  - Replacing register names with TAGs "name dependencies" are resolved. (sort of "register-renaming")

Super-scalar Issues: *Dynamic Scheduling*

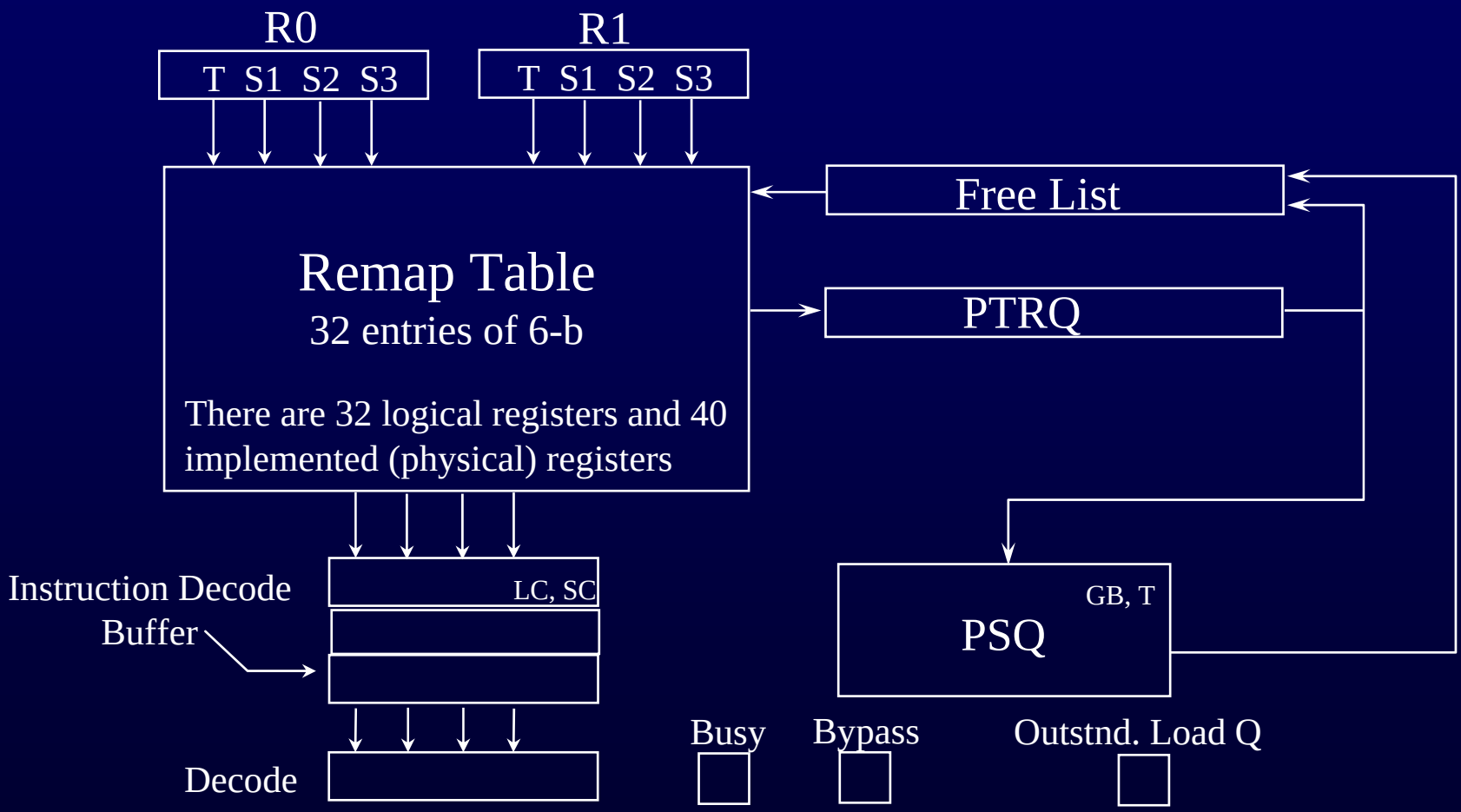
Register-Renaming: IBM RS/6000 (1990)

Consist of:

- Remap Table (RT): providing mapping form logical to physical register
- Free List (FL): providing names of the registers that are unassigned - so they can go back to the RT
- Pending Target Return Queue (PTRQ): containing physical registers that are used and will be placed on the FL as soon as the instruction using them pass decode
- Outstanding Load Queue (OLQ): containing registers of the next FLP load whose data will return from the cache. It stops instruction from decoding if data has not returned

Super-scalar Issues: *Dynamic Scheduling*

Register-Renaming Structure: IBM RS/6000 (1990)



Power of Super-scalar Implementation

Coordinate Rotation: IBM RS/6000 (1990)

```
        FL FR0, sin theta      ;laod rotation matrix
        FL FR1, -sin theta     ;constants
        FL FR2, cos theta      ;
        FL FR3, xdis           ;load x and y
        FL FR4, ydis           ;displacements
        MTCTR I                ;load Count register with loop count

LOOP:   UFL FR8, x(i)                          ;laod x(i)
        FMA FR10, FR8, FR2,  FR3               ;form x(i)cos + xdis
        UFL FR9, y(i)                          ;laod y(i)
        FMA FR11, FR9, FR2,  FR4               ;form y(i)cos + ydis
        FMA FR12, FR9, FR1,  FR10              ;form -y(i)sin + FR10
        FST FR12, x1(i)                        ;store x1(i)
        FMA FR13, FR8, FR0,  FR11              ;form x(i)sin + FR11
        FST FR13, y1(i)                        ;store y1(i)
        BC LOOP                                ;continue for all points
```

$$x1 = x\cos\theta - y\sin\theta$$
$$y1 = y\cos\theta + x\sin\theta$$

This code, 18 instructions worth, executes in 4 cycles in a loop

Super-scalar Issues:
Instruction Issue and Machine Parallelism

- In-Order Issue with In-Order Completion:
  - The simplest instruction-issue policy. Instructions are issued in exact program order. Not efficient use of super-scalar resources. Even in scalar processors in-order completion is not used.
- In-Order Issue with Out-of-Order Completion:
  - Used in scalar RISC processors (Load, Floating Point).
  - It improves the performance of super-scalar processors.
  - Stalled when there is a conflict for resources, or true dependency.
- Out-of-Order Issue with I Out-of-Order Completion:
  - The decoder stage is isolated from the execute stage by the “instruction window” (additional pipeline stage).

Super-scalar Examples:
Instruction Issue and Machine Parallelism

DEC Alpha 21264:

- Four-Way (Six Instructions peak), Out-of-Order Execution

MIPS R10000:

- Four Instructions, Out-of-Order Execution

HP 8000:

- Four-Way, Agressive Out-of-Order execution, large Reorder Window
- Issue: In-Order, Execute: Out-of-Order, Instruction Retire: In-Order

Intel P6:

- Three Instructions, Out-of-Order Execution

Exponential:

- Three Instructions, In-Order Execution

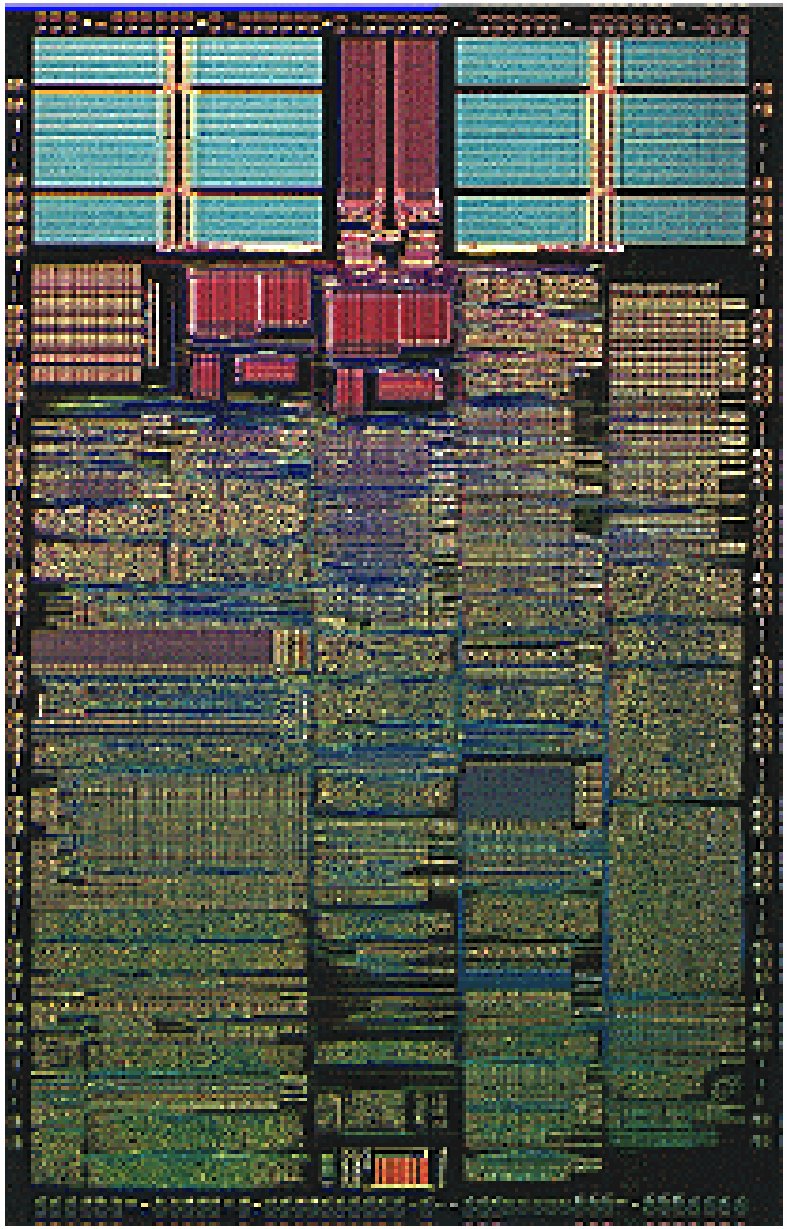
Super-scalar Issues:

The Cost vs. Gain of Multiple Instruction Execution

PowerPC Example:

Feature	601+	604	Difference
Frequency	100MHz	100MHz	same
CMOS Process	.5u 5-metal	.5u 4-metal	~same
Cache Total	32KB Cache	16K+16K Cache	~same
Load/Store Unit	No	Yes	
Dual Integer Unit	No	Yes	
Register Renaming	No	Yes	
Peak Issue	2 + Branch	4 Instructions	~double
Transistors	2.8 Million	3.6 Million	+30%
SPECint92	105	160	+50%
SPECfp02	125	165	+30%

Motorola's PowerPC™ 603 RISC Microprocessor



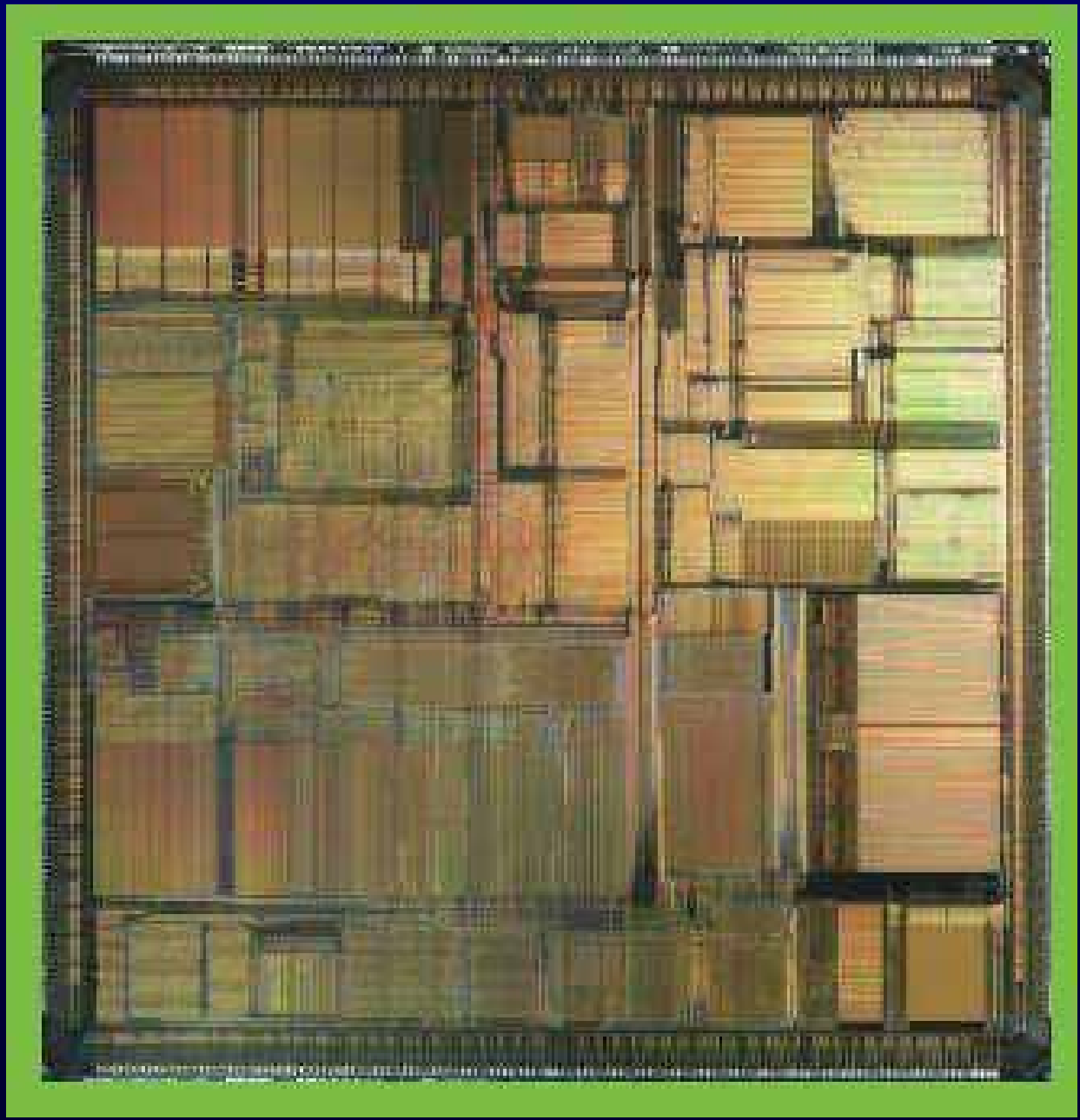
Super-scalar Issues:

Comparisson of leading RISC microrpocessors

Feature	Digital 21164	MIPS 10000	PowerPC 620	HP 8000	Sun UltraSparc
Frequency	500 MHz	200 MHz	200 MHz	180 MHz	250 MHz
Pipeline Stages	7	5-7	5	7-9	6-9
Issue Rate	4	4	4	4	4
Out-of-Order Exec.	6 loads	32	16	56	none
Register Renam. (int/FP)	none/8	32/32	8/8	56	none
Transistors/ Logic transistors	9.3M/ 1.8M	5.9M/ 2.3M	6.9M/ 2.2M	3.9M*/ 3.9M	3.8M/ 2.0M
SPEC95 (Intg/FlPt)	12.6/18.3	8.9/17.2	9/9	10.8/18.3	8.5/15
Perform./ Log-trn (Intg/FP)	7.0/10.2	3.9/7.5	4.1/4.1	2.77*/4.69	4.25/7.5

* cache

Sun Micro. Ultra-SPARC



Super-scalar Issues:
Value of Out-of-Order Execution

Feature	MIPS 5000	MIPS 10000	HP-PA 7300LC	HP 8000	Digital 21164	Digital 21264
Frequency	180 MHz	200 MHz	160 MHz	180 MHz	500 MHz	600 MHz
Pipeline Stages	5	5-7	5	7-9	7	7/9
Issue Rate	2	4	2	4	4	4+2
Out-of-Order Exec.	none	32	none	56	6 loads	20i+15fp
Register-Renam. (int/FP)	none	32/32	none	56	none/8	80/72
Transistors/ Logic transistors	3.6M/ 1.1	5.9M/ 2.3M	9.2M/ 1.7M	3.9M*/ 3.9M	9.3M/ 1.8M	15.2M/ 6M
Cache	32/32K	32/32K	64/64K	none	8/8/96	64/64K
SPEC95 (Intg/FlPt)	4.0/3.7	8.9/17.2	5.5/7.3	10.8/18.3	12.6/18.3	~36/~60
Perform./ Log-Tr (Intg/FP)	3.6/3.4	3.9/7.5	3.2/4.3	2.77*/4.69	7.0/10.2	6.0/10.0

*cache

The ways to exploit instruction parallelism

- Super-scalar:

takes advantage of instruction parallelism to reduce the average number of cycles per instruction.

- Super-pipelined:

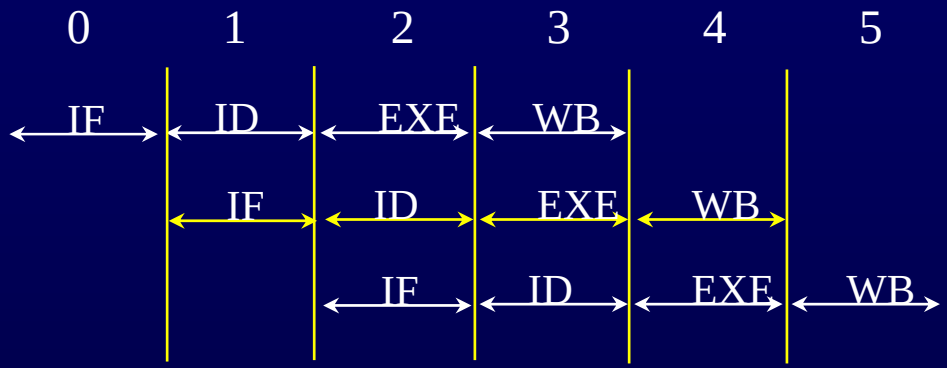
takes advantage of instruction parallelism to reduce the cycle time.

- VLIW:

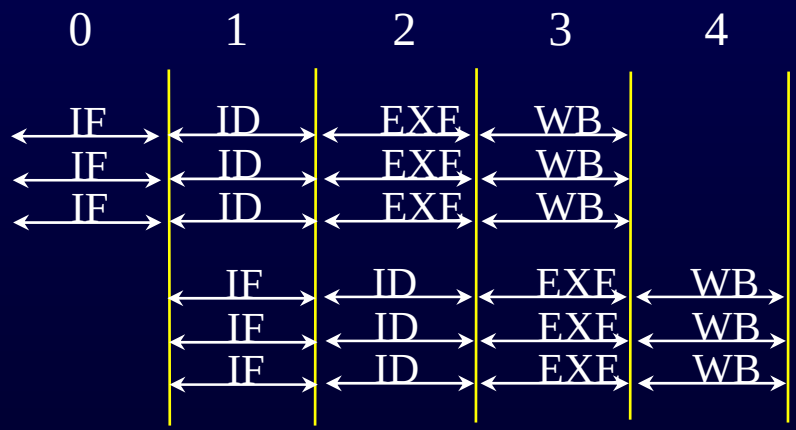
takes advantage of instruction parallelism to reduce the number of instructions.

The ways to exploit instruction parallelism: Pipeline

Scalar:

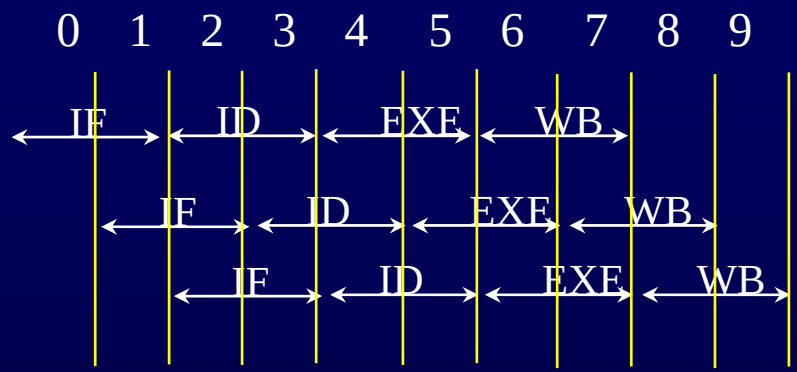


Super-scalar:

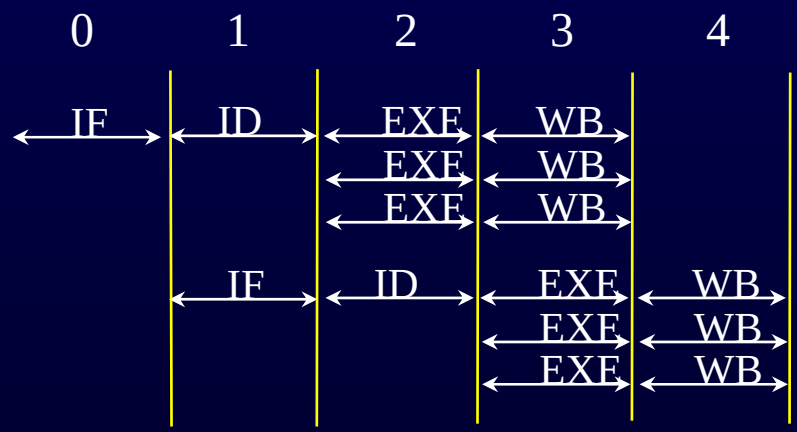


The ways to exploit instruction parallelism: Pipeline

Super-pipelined:



VLIW:



Very-Long-Instruction-Word Processors

- A single instruction specifies more than one concurrent operation:
  - This reduces the number of instructions in comparison to scalar.
  - The operations specified by the VLIW instruction must be independent of one another.
- The instruction is quite large:
  - Takes many bits to encode multiple operations.
  - VLIW processor relies on software to pack the operations into an instruction.
  - Software uses technique called “compaction”. It uses no-ops for instruction operations that cannot be used.

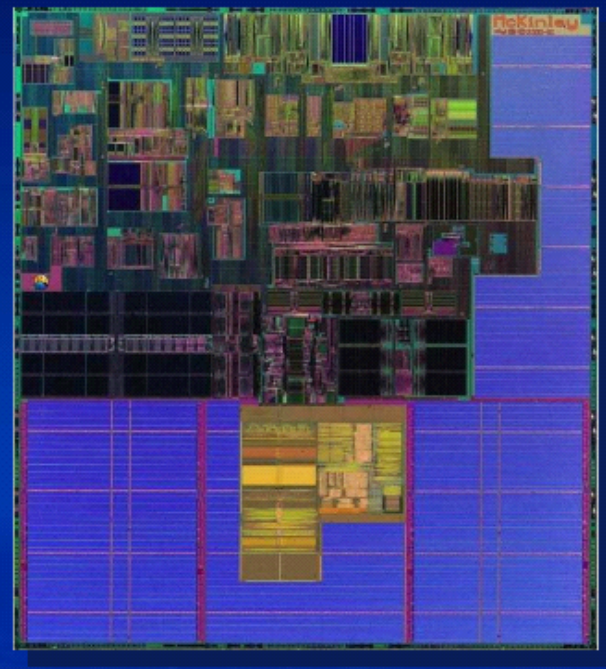
VLIW processor is not software compatible with any general-purpose processor !

Very-Long-Instruction-Word Processors

- It is difficult to make different implementations of the same VLIW architecture binary-code compatible with one another.
  - because instruction parallelism, compaction and the code depend on the processor's operation latencies
- Compaction depends on the instruction parallelism:
  - In sections of code having limited instruction parallelism most of the instruction is wasted
- VLIW lead to simple hardware implementation

Itanium® 2 Processor

- Transistors: 221M
  - Caches, I/O: 3.3MB or ~170M (75%)
  - Core: ~51M (25%)
- Die size: 19.5 x 21.6mm = 421 mm^2
  - Caches, I/O: L3C ~50%; others ~16%
  - Core: 142mm^2 (34%)



Caches becoming an increasing portion of the die because of its performance impact and low power density

Super-pipelined Processors

- In Super-pipelined processor the major stages are divided into sub-stages.
  - The degree of super-pipelining is a measure of the number of sub-stages in a major pipeline stage.
  - It is clocked at a higher frequency as compared to the pipelined processor (the frequency is a multiple of the degree of super-pipelining).
  - This adds latches and overhead (due to clock skews) to the overall cycle time.
  - Super-pipelined processor relies on instruction parallelism and true dependencies can degrade its performance.

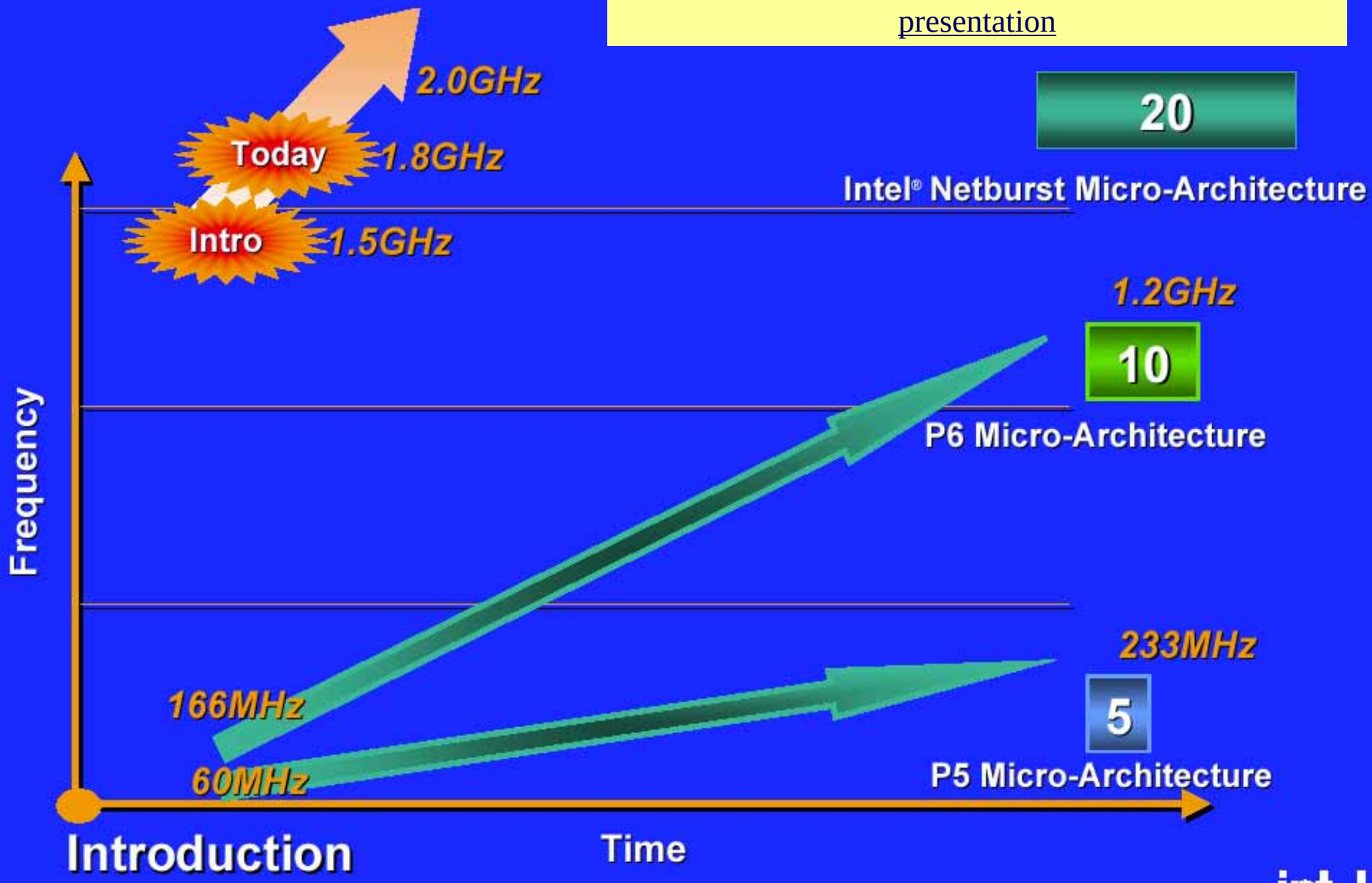
Super-pipelined Processors

- As compared to Super-scalar processors:
  - Super-pipelined processor takes longer to generate the result.
  - Some simple operation in the super-scalar processor take a full cycle while super-pipelined processor can complete them sooner.
  - At a constant hardware cost, super-scalar processor is more susceptible to the resource conflicts than the super-pipelined one. A resource must be duplicated in the super-scalar processor, while super-pipelined avoids them through pipelining.
- Super-pipelining is appropriate when:
  - The cost of duplicating resources is prohibitive.
  - The ability to control “clock skew” is good

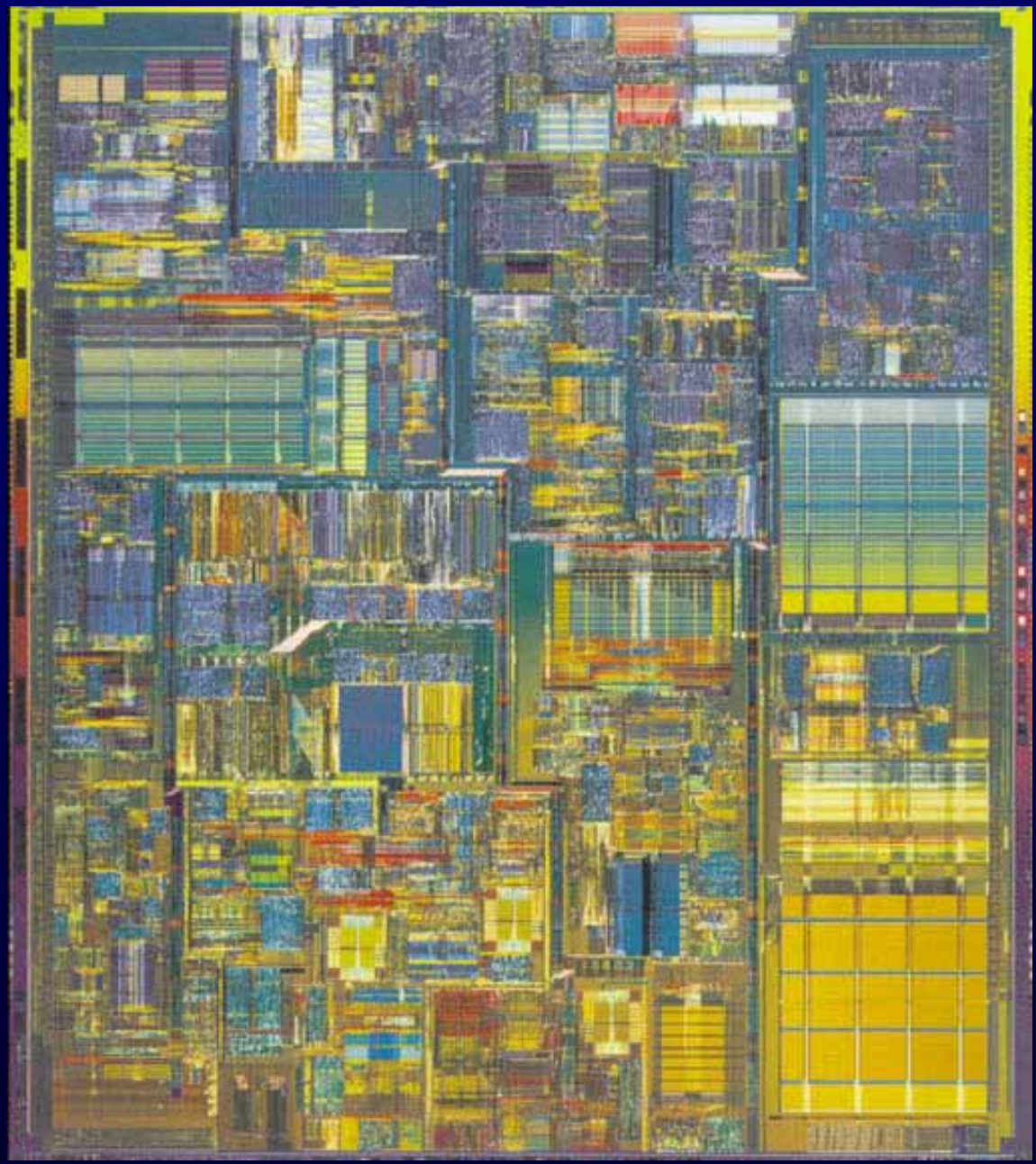
This is appropriate for very high speed technologies: GaAs, BiCMOS, ECL (low logic density and low gate delays).

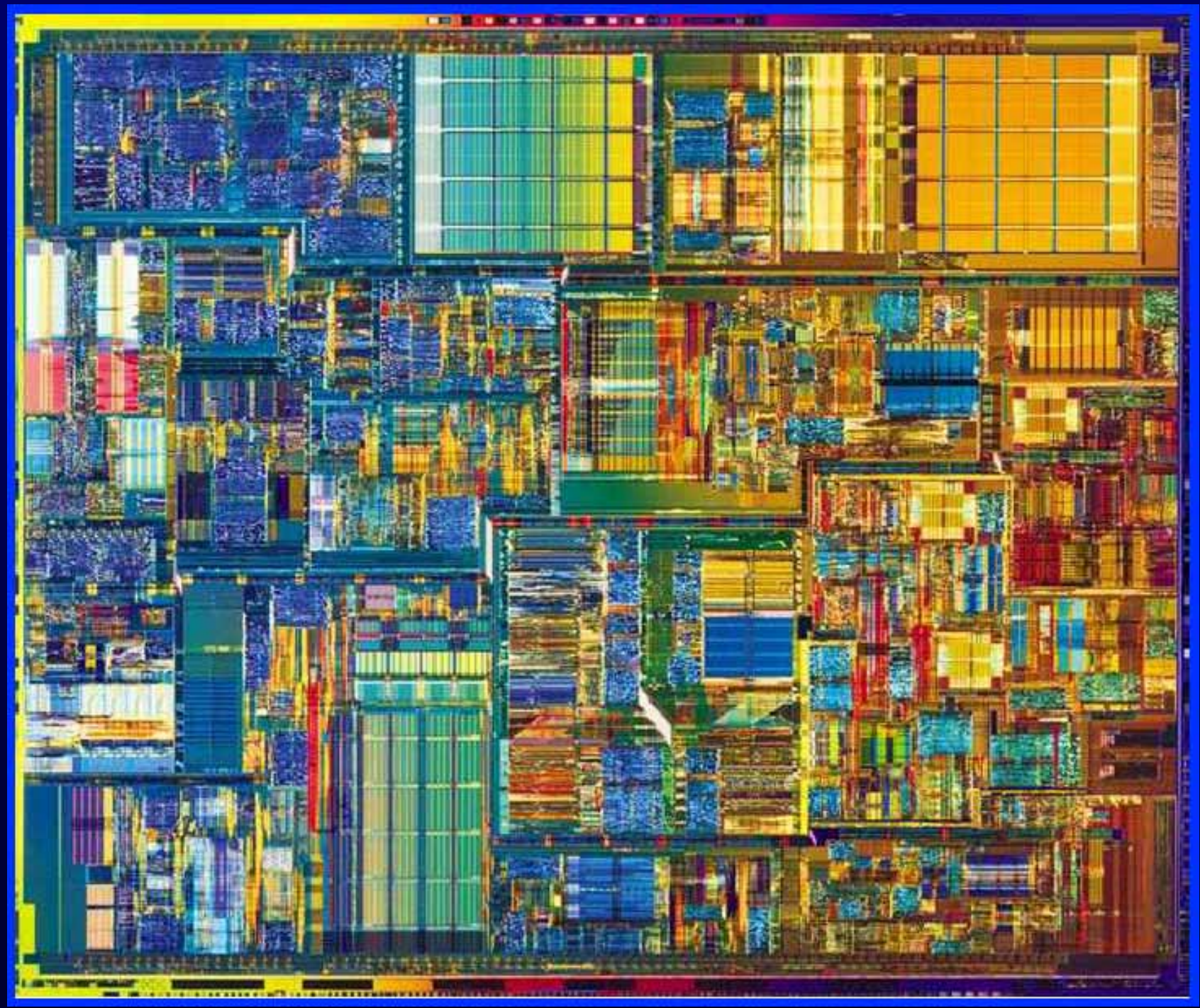
Hyper Pipelined Technology

Courtesy: Doug Carmean, Intel Corp, Hot-Chips-13 presentation

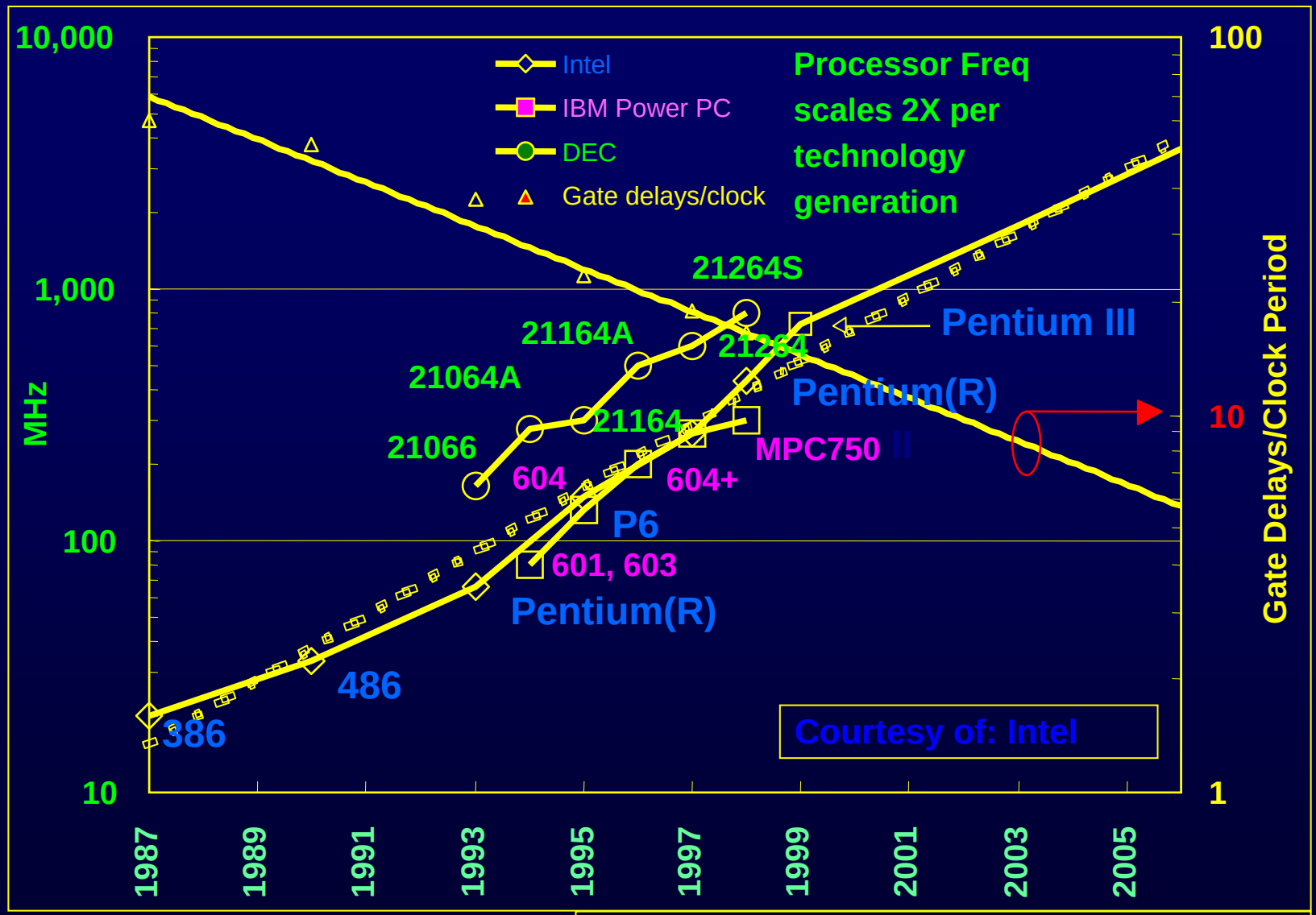


Intel Pentium 4





Pipeline Depth



- Frequency doubles each generation
- Number of gates/clock reduce by 25%

Multi-GHz Clocking Problems

- Fewer logic in-between pipeline stages:
  - Out of 7-10 FO4 allocated delays, FF can take 2-4 FO4
- Clock uncertainty can take another FO4
- The total could be ½ of the time allowed for computation

Consequences of multi-GHz Clocks

- Pipeline boundaries start to blur
- Clocked Storage Elements must include logic
- Wave pipelining, domino style, signals used to clock …..
- Synchronous design only in a limited domain
- Asynchronous communication between synchronous domains

Future Perspective

INTERNET ERA: DSP PLUS ANALOG

2G Cellular Phones

PDAs

3G Cellular Phones

3G Basestations

Digital Hearing

IP Phone

Bluetooth-Enabled Products

VOP Gateway

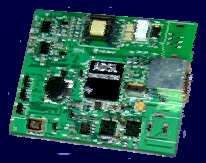
Digital Speakers

Digital Still Camera

Internet Audio

Digital Motor Control

DAB Digital Radio



DSL Modem

Cable Modem

Home Networking

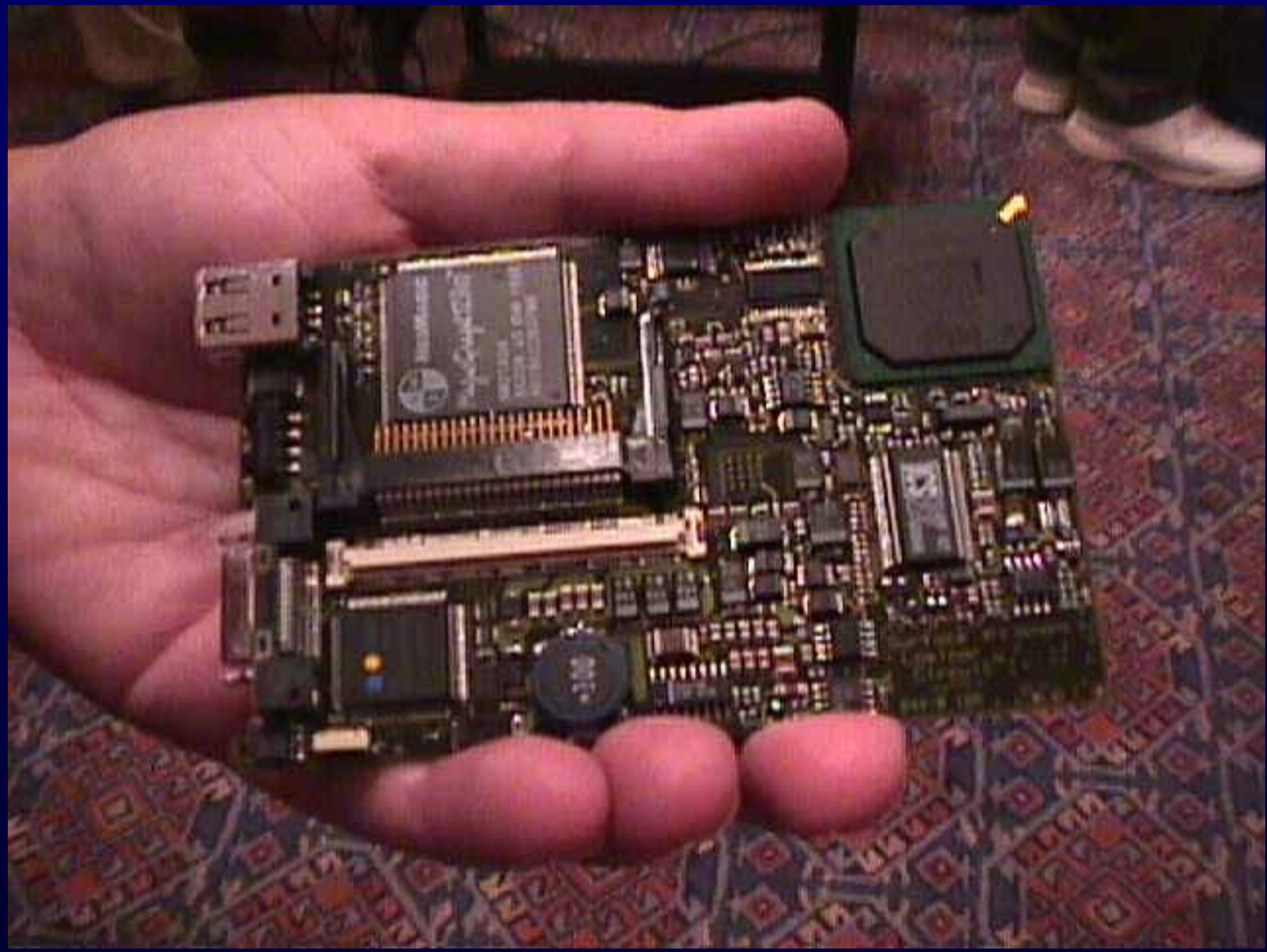
Central Office

Video Server

Pro-Audio

From Dennis Buss, Texas Instruments, ICECS, Malta 2001 presentation

Wearable Computer



Wearable Computer

Wearable Computer

http://www.cs.cmu.edu/~wearable/
http://lcs.www.media.mit.edu/projects/wearables/
http://www.microopticalcorp.com/

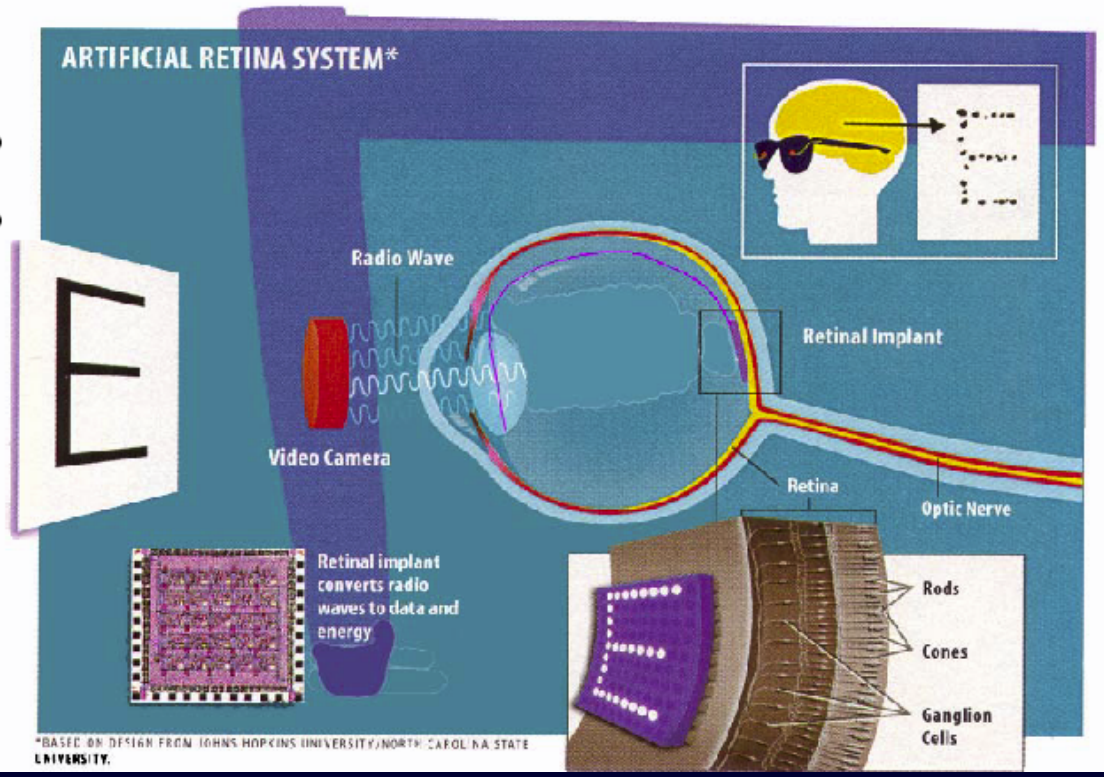
T. Kuroda (21/39)

Digital Ink

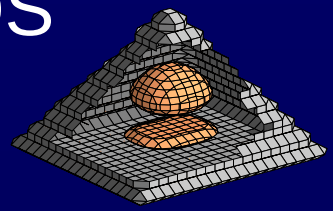
Digital Ink is a sophisticated pen that recognizes and stores the handwriting and drawing of it's user. After writing, the user simply jots the word "send" or "e-mail" followed by a fax number or e-mail address. The documents are wirelessly sent via cellular network to fax machines, desktop computers or even other digital pens. A small digital "ink well" connected to the user's desktop computer serves as home to Digital Ink, and allows the pen's information to be downloaded for future use. Digital Ink reinvents the computer desktop by turning any writing surface - from napkins to paper - into low-tech and socially comfortable computer interfaces. … CMU

T. Kuroda (22/39)

Implantable Computer

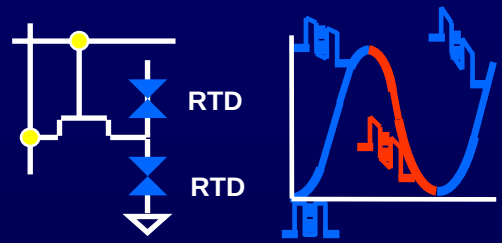


TECHNOLOGY IN THE INTERNET ERA
Future Scaling Beyond Bulk CMOS

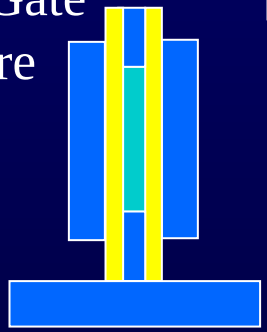


Single Electronics

RTD



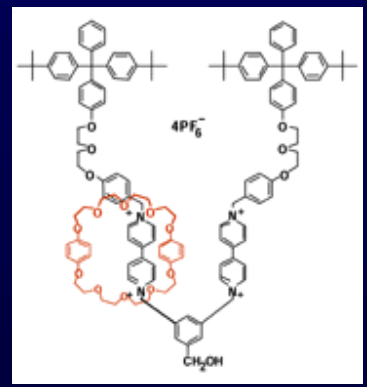
Vertical Gate Structure

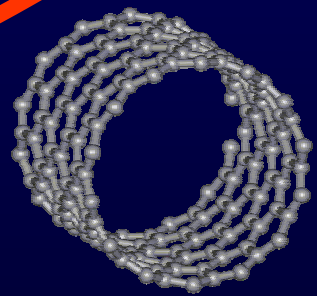


Bulk CMOS

SOI

Molecular Switch





Nanotubes

Today 2020 2040

From Dennis Buss, Texas Instruments, ICECS, Malta 2001 presentation

Year 2010

Extrapolation of the trend with some saturation

Many important interesting application

Home, Entertainment, Office, Translation , Health care

Year 2020???

More assembly technique: 3D

Year 2100

Combination of bio and semiconductor

Ultra small volume
Small number of neuron cells
Extremely low power

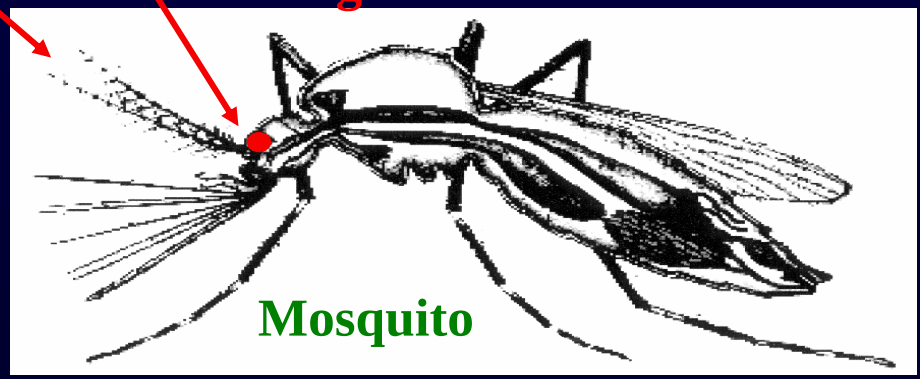
Brain

Sensor

Infrared
Humidity
CO_2

Real time image processing
(Artificial) Intelligence
3D flight control

Long lifetime
by DNA manipulation
Bio-computer



Mosquito

Galaxy

More than 100 billion stars are involved

From Hiroshi Iwai, Toshiba, ISSCC 2000 presentation