Is Computing at a Tipping Point?

A Personal Perspective

FRAN ALLEN
allen@watson.ibm.com

CCSCNE
April 16, 2010

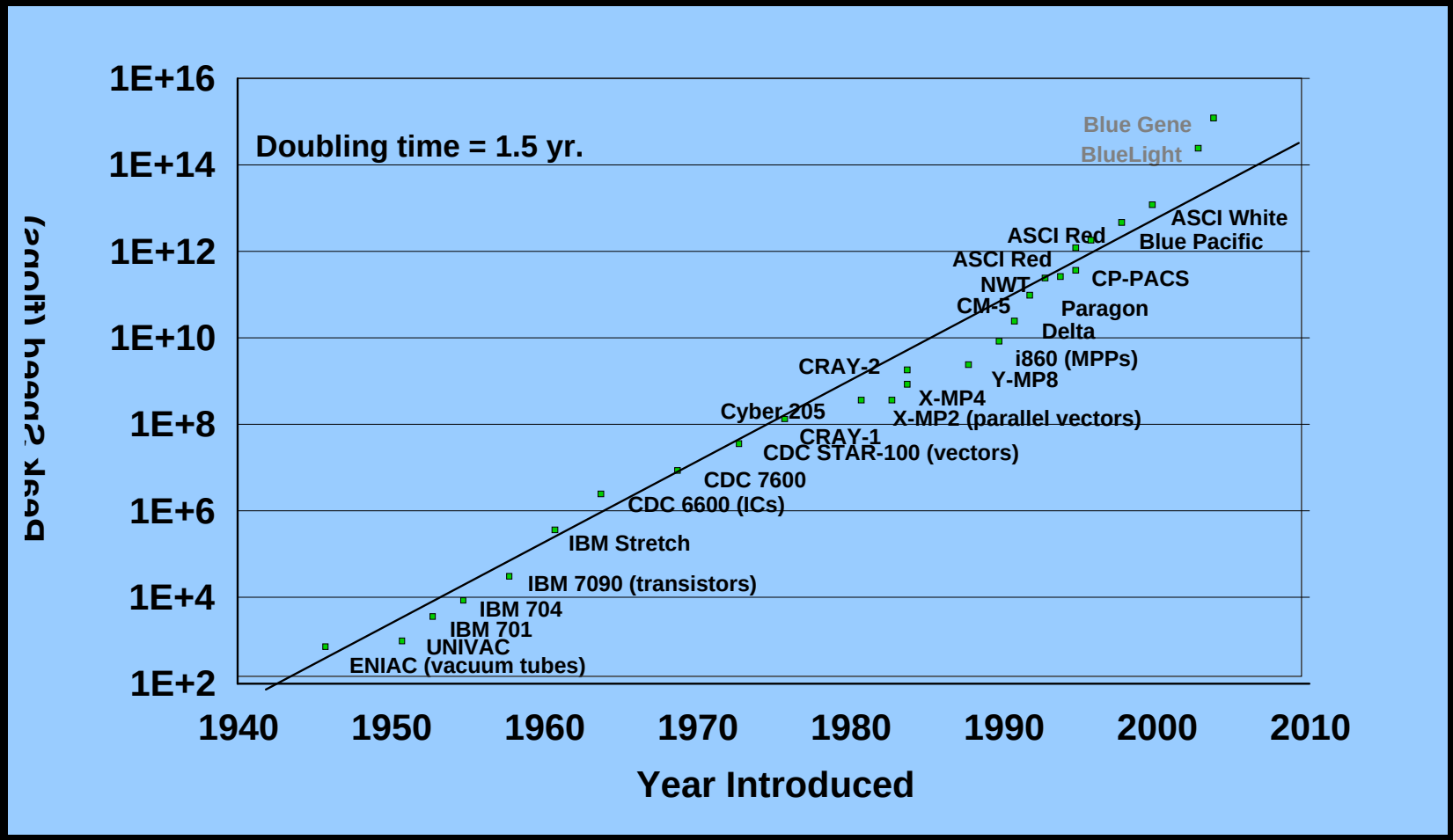
Computing is changing our world

- Computation, communication and information are integrated
- The global digital space makes time and place irrelevant
- Access to information is instantaneous, knowledge will be
- Computing is evolving into a utility

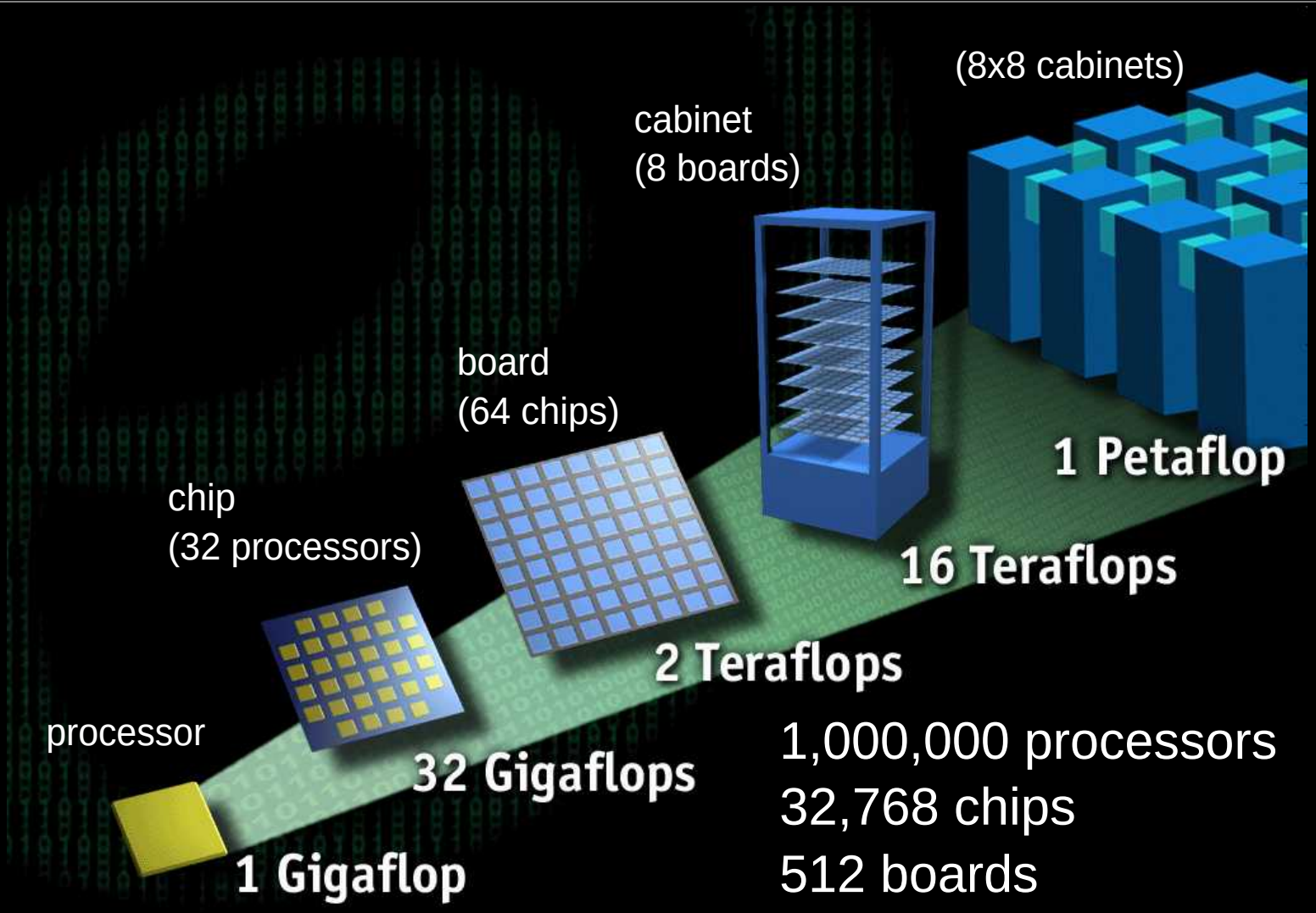
But There are Challenges and Tipping Points

- Moore's law seems to be reaching a limit
  - An Everest of Data
  - Cyber security, Information integrity, ….
  - Universal Access and Ease of use
  - Where have all the women gone??
- Computer Science

Peak Performance Computers by Year



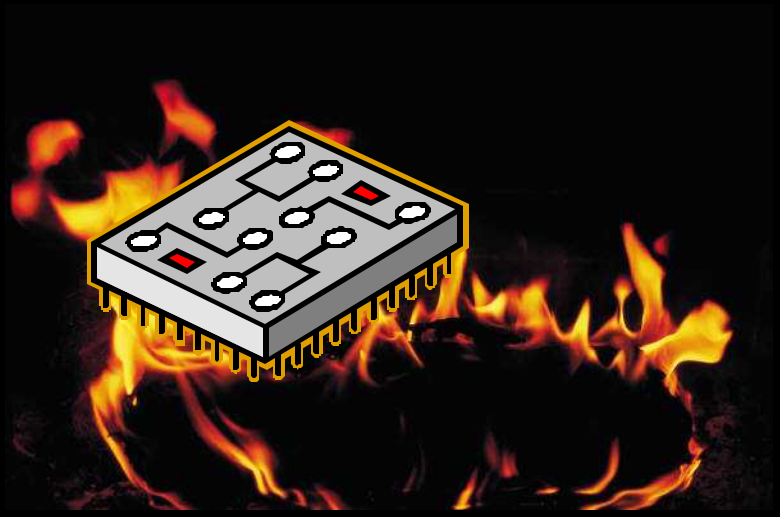
An Idealized Supercomputer



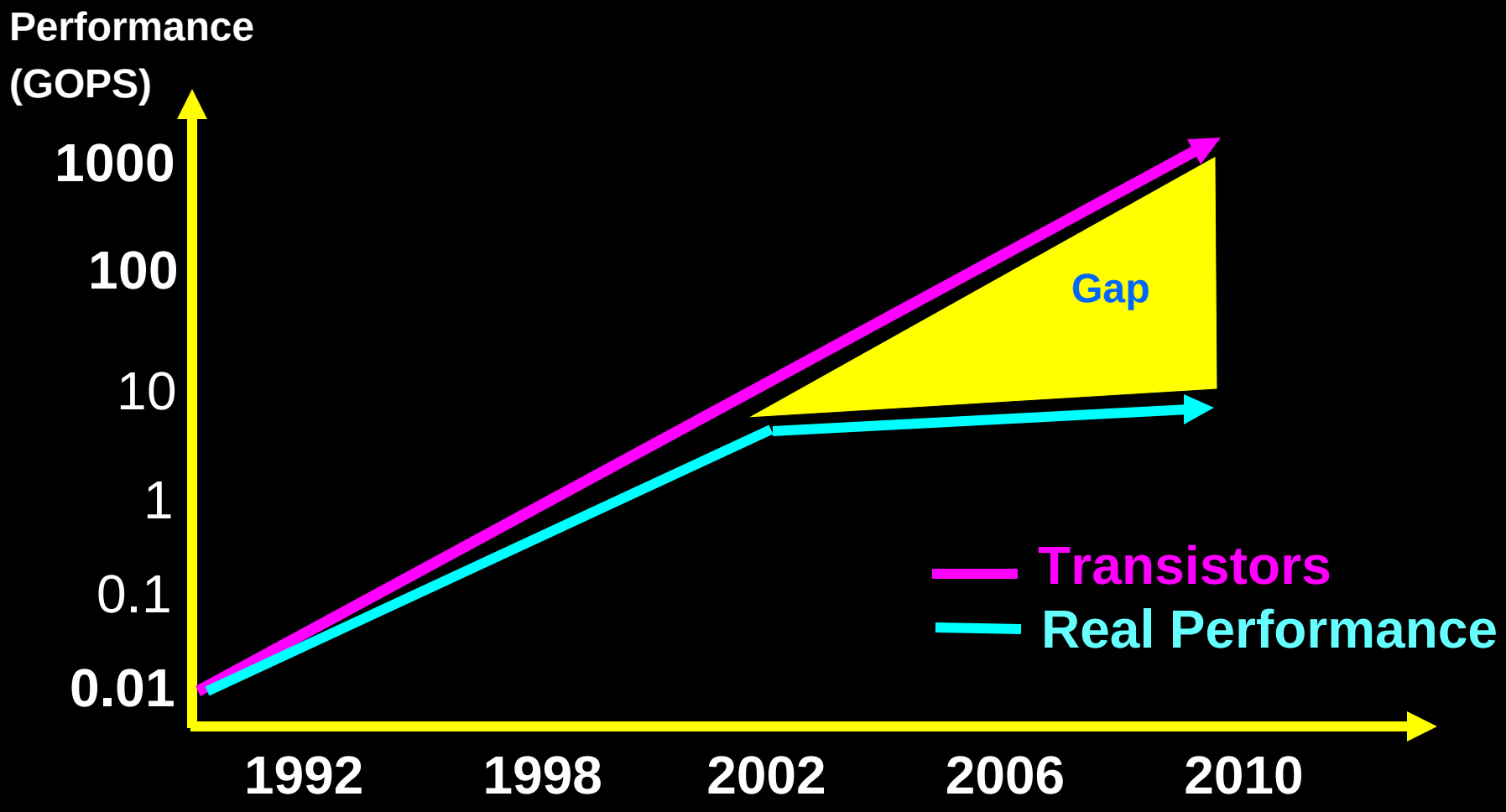
1,000,000 processors
32,768 chips
512 boards

Technology is Hitting a Performance Limit

- Transistors continue to shrink
- More and more transistors fit on a chip
- The chips are faster and faster
- Result: HOT CHIPS!



Real Performance Stops Growing as Fast



Solution: Multicores??

- Simpler, slower, cooler processors (multicores)
- More processors on a chip
- Software (and users) organize tasks to execute in parallel on the processors
- Parallelism will provide the performance!

Parallelism is the new challenge

- High performance computing applications and computers have long used parallelism for performance.

  - Current software cannot provide the parallelism needed

  - Users can't either

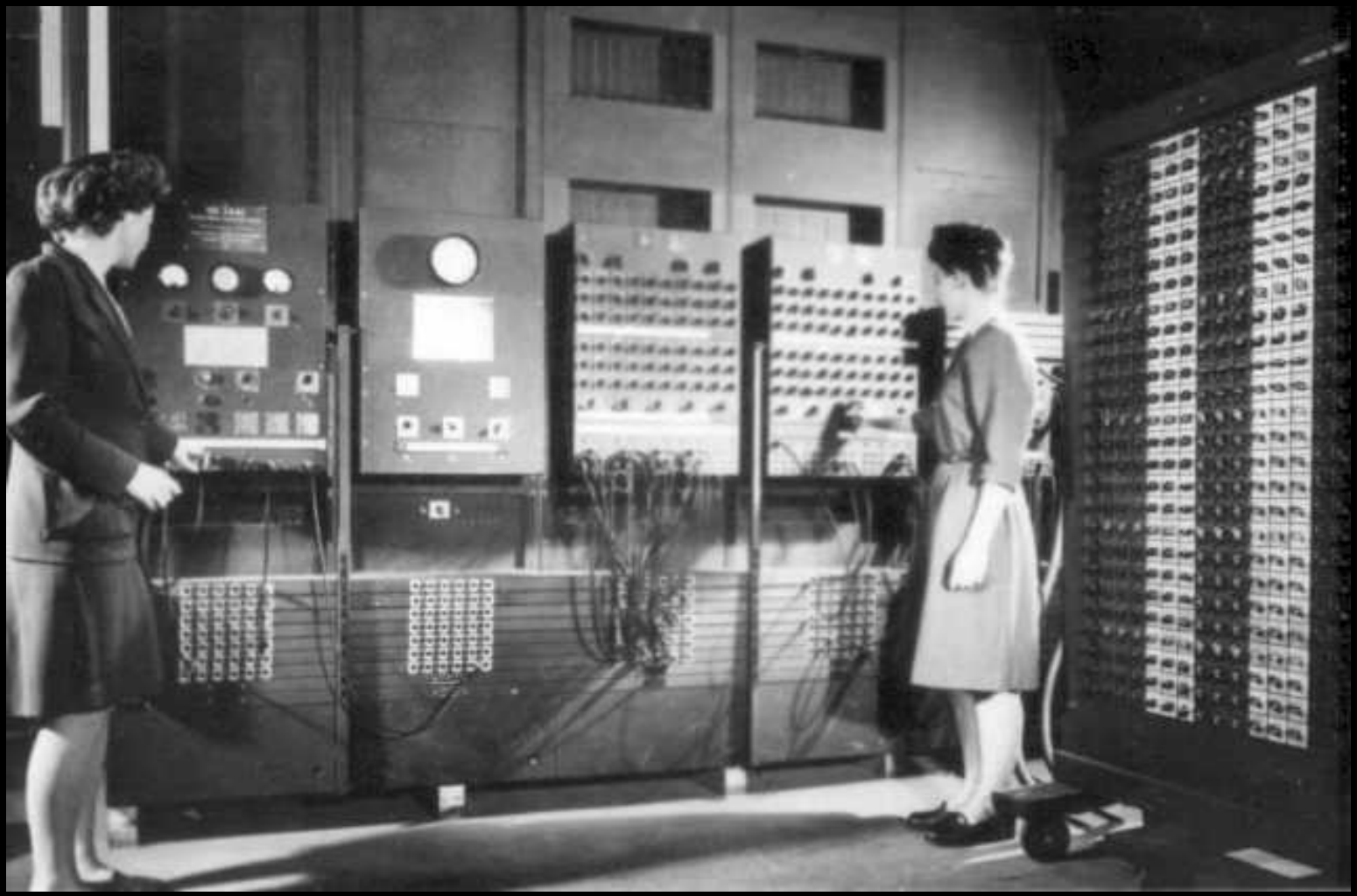
Why Does It Matter?

- "The biggest problem Computer Science has ever faced." John Hennessy
- "The best opportunity Computer Science has to improve user productivity, application performance, and system integrity." Fran Allen

Topics

- The Performance Challenge
- A brief, personal tour of some early languages, compilers, and computers for high performance systems
- Tipping Points

“Computers” using Eniac ~ 1945



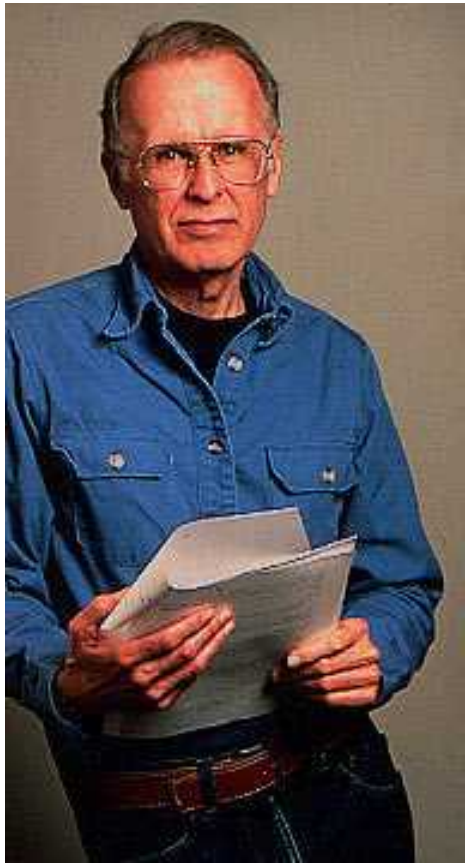
In 1957 I joined IBM Research as a Programmer

1957 IBM Recruiting Brochure

Fortran Project (1954-1957) Goals

- User Productivity
- Program Performance

John Backus



THE FORTRAN GOALS BECAME MY GOALS

The Fortran Language and Compiler

- Released April 15, 1957
- Some features:
  - Beginnings of formal parsing techniques
  - Intermediate language form for optimization
  - Control flow graphs
  - Common sub-expression elimination
  - Generalized register allocation - for only 3 registers!
- Spectacular object code!!

Stretch (1956-1961)

- Goal: 100 times faster than any existing machine
- Main Performance Limitation: Memory Access Time
- Extraordinarily ambitious hardware
- Equally ambitious compiler

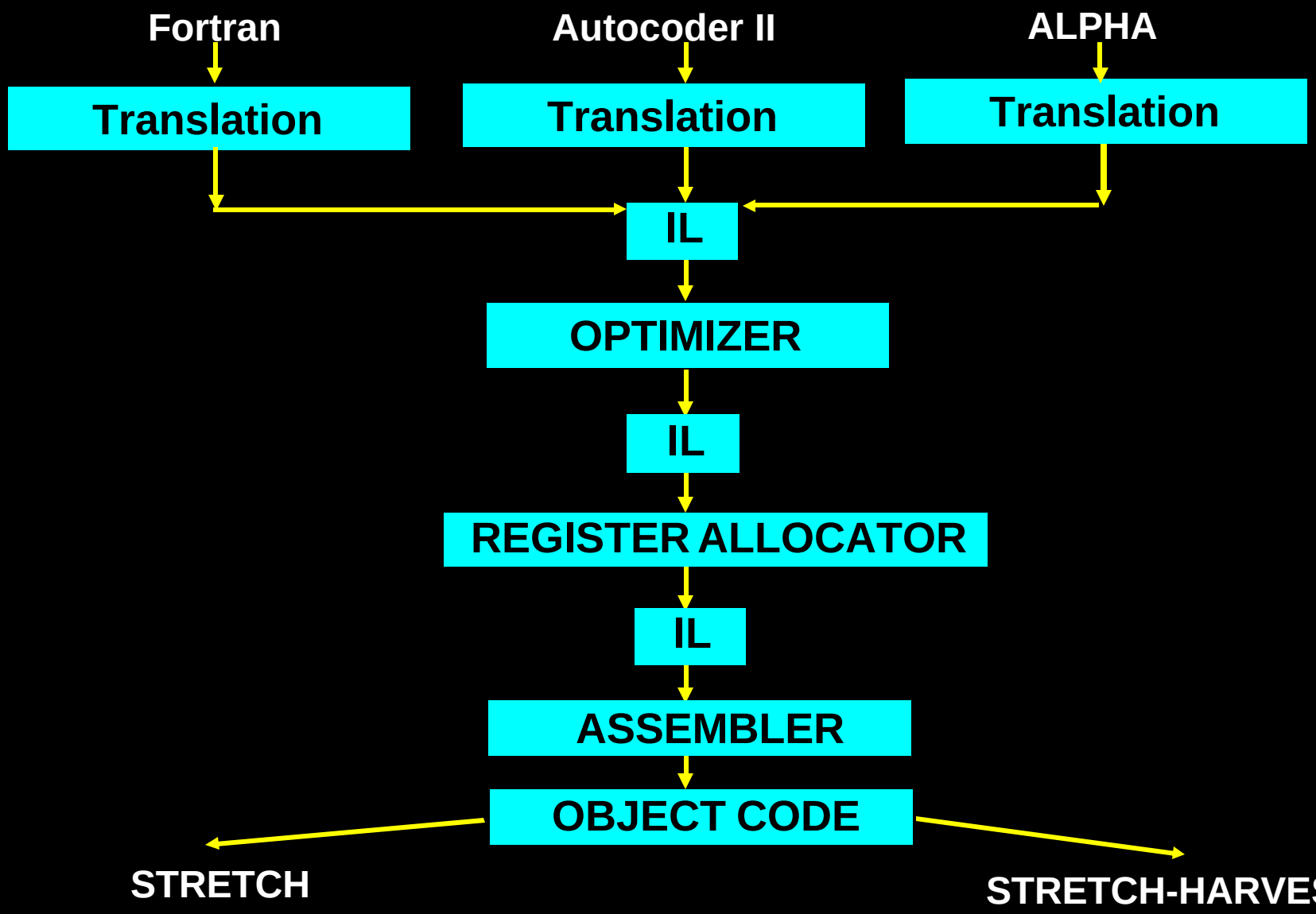
Stretch Concurrency

- Overlapped storage references: up to 6 at a time
- Instruction lookahead unit
  - Up to 11 instructions executing in cpu at the same time
  - Hardware gave the appearance of a sequential machine
- Superscalar??
- Multiprogramming
- I/O and disk operations asynchronous with computation

HARVEST (1958 - 1962)

- Built for NSA for code breaking
- Hosted by Stretch
- Streaming data computation model
- Eight instructions and unbounded execution times
- Only system with balanced I/O, memory and computational speeds (per conversation with Jim Pomerene 11/2000)
- ALPHA: a language designed to fit the problem and the machine

Stretch – Harvest Compiler Organization



Stretch - Harvest Outcomes

- April 1961: Stretch delivered to Los Alamos but
  - Stretch performance off by 50%
  - Considered a failure by IBM
- Feb 1962: Harvest accepted by National Security Agency and used for 14 years
- **Stretch had a huge influence on future systems!**

The State of the Computing in 1965

- Computer Science emerging as a discipline
- Multiple disciplines begin to merge:
  - Information sciences
  - Communications
  - Computing
- Many Programming Languages
- Compilers, >>>>

Topics

- The Multicore Challenge
- A brief, personal tour of some early languages, compilers, and computers for high performance systems
- Hopes and Opportunities

Software and CS Opportunities

- Automatic parallelism and optimization
- Very high level, domain specific languages
- Eliminate C, JAVA…. as general purpose languages
- Automatic optimization of data: data movement, locality, ownership, ….
- New, formal models of parallelism
- Recast compilers

The Importance of Computer Science

- In a talk on “Understanding Science through the Lens of Computation”, Dick Karp said:
  - The algorithmic world view is changing the mathematical, natural, social, and life sciences.
  - Computer Science is placing itself at the center of scientific discourse and exchange of ideas. And this is only the beginning.

My Hopes for the Future

- I hope to see a new generation experiencing the excitement I feel for our field.
- I hope to see universal access and participation.
- I hope Computer Science will become the Queen of the Sciences!!!

A Poem by Yeh Chien-Ying, Minister of Defense, Peoples Republic of China

STORM THE PASS

A city's besiegers

Fear not its strength.

And those attacking a subject

Must not fear its difficulties.

Determined battle will take us

Through obstacles, through the pass,

to the mountain tops of science.

(Published in Ren Min Ri Bao, September 21, 1977)

END OF TALK

BEGINNING OF A NEW ERA IN COMPUTING and COMPUTER SCIENCE