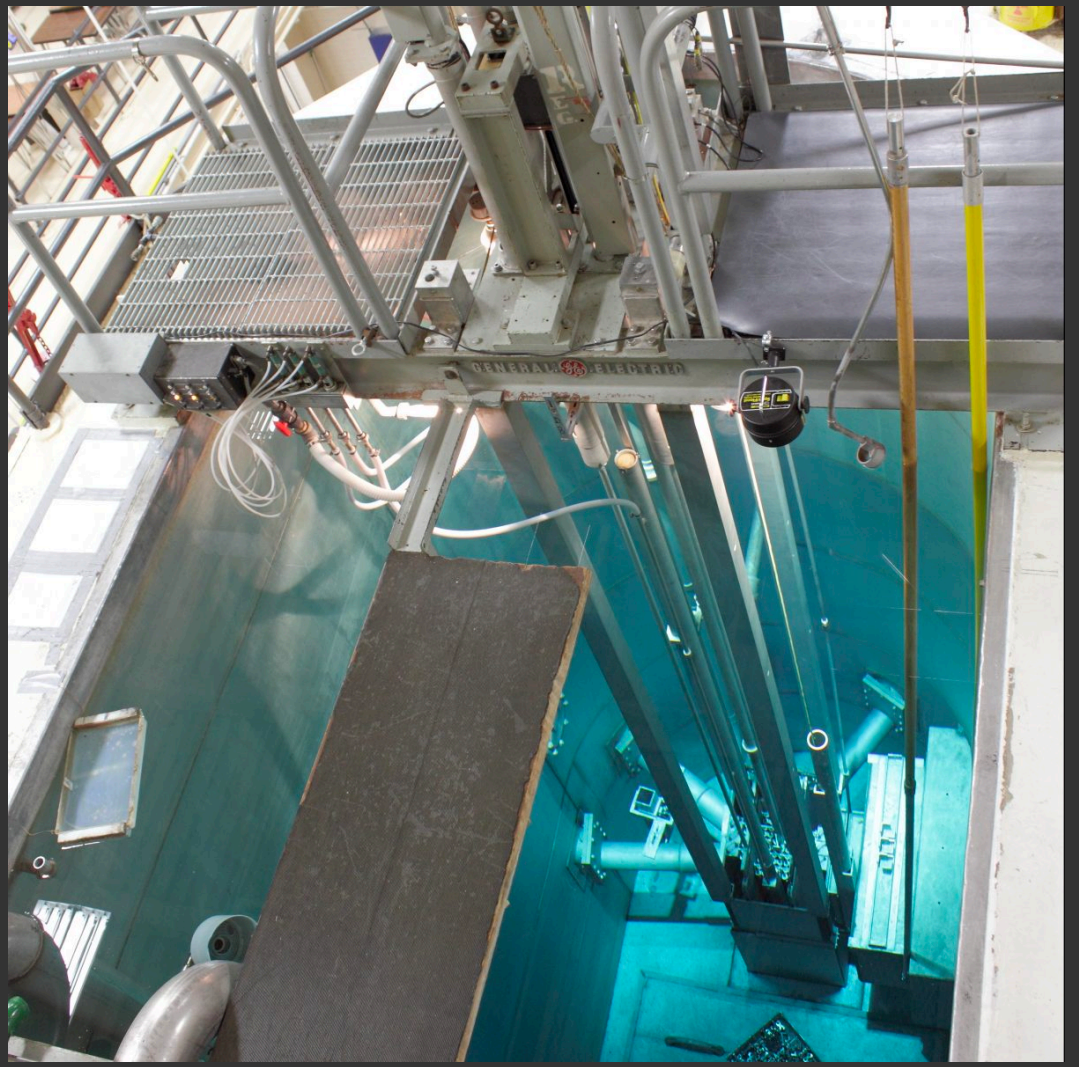
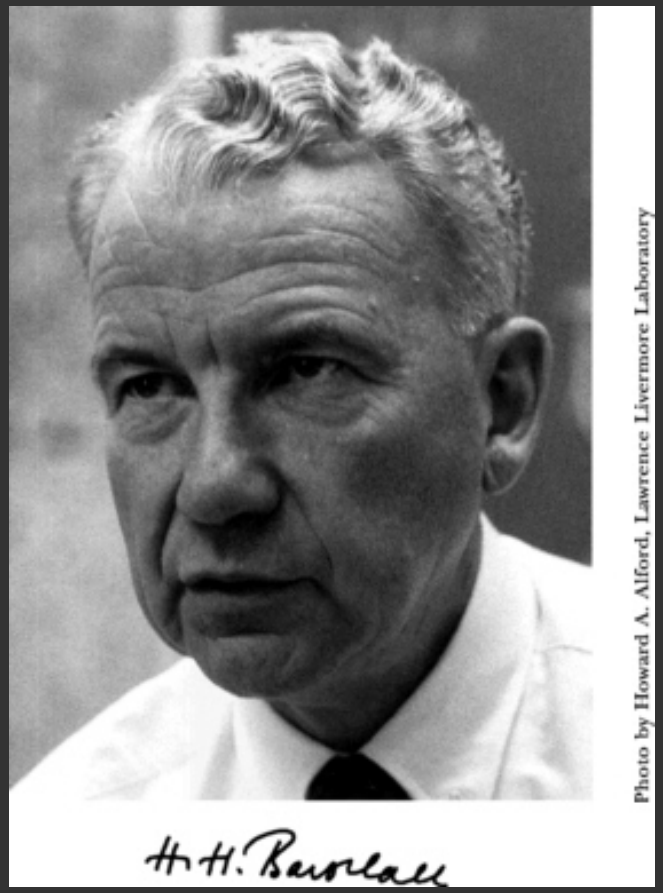
A Menagerie of High-Performance Graphics Systems

Pat Hanrahan
Stanford University

Salishan Conference on High Speed Computing
April 25, 2023

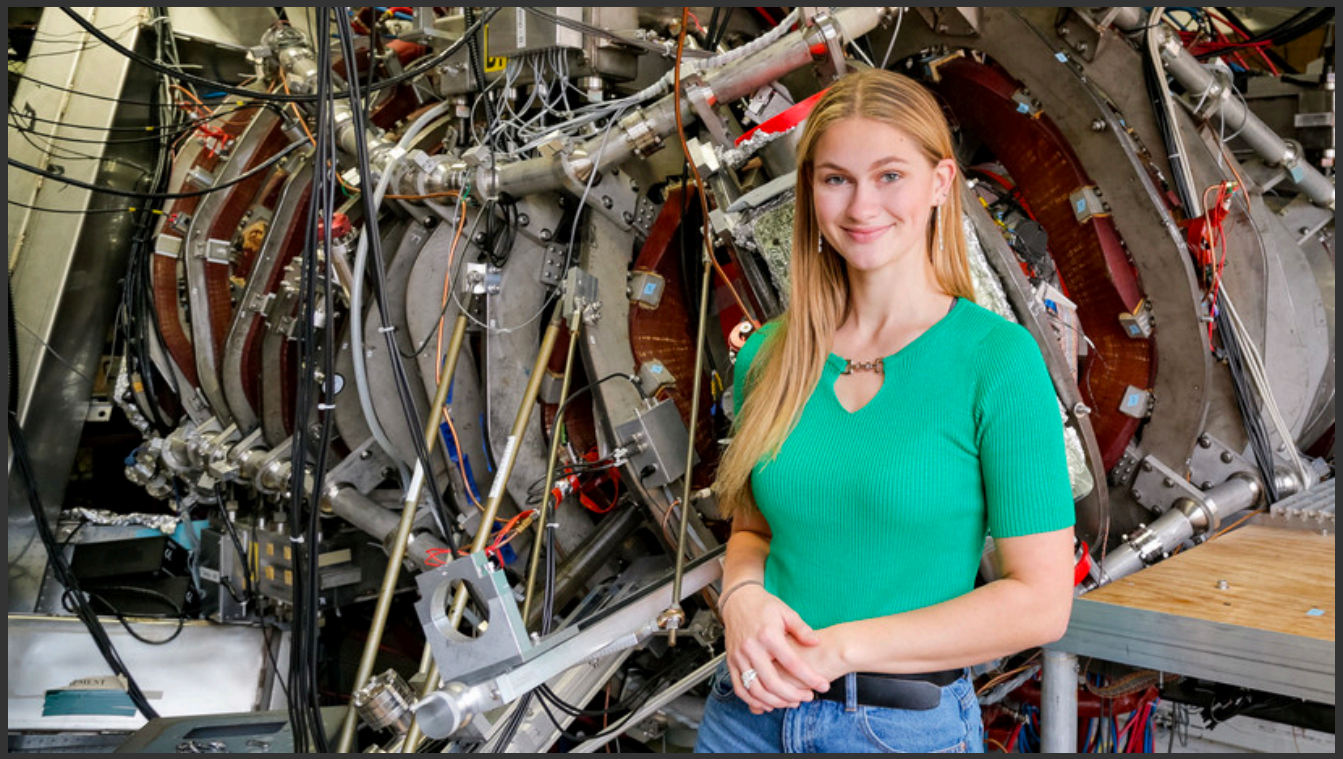
University of Wisconsin
1 MW TRIGA Nuclear Reactor





H. H. Barschall

Max Carbon



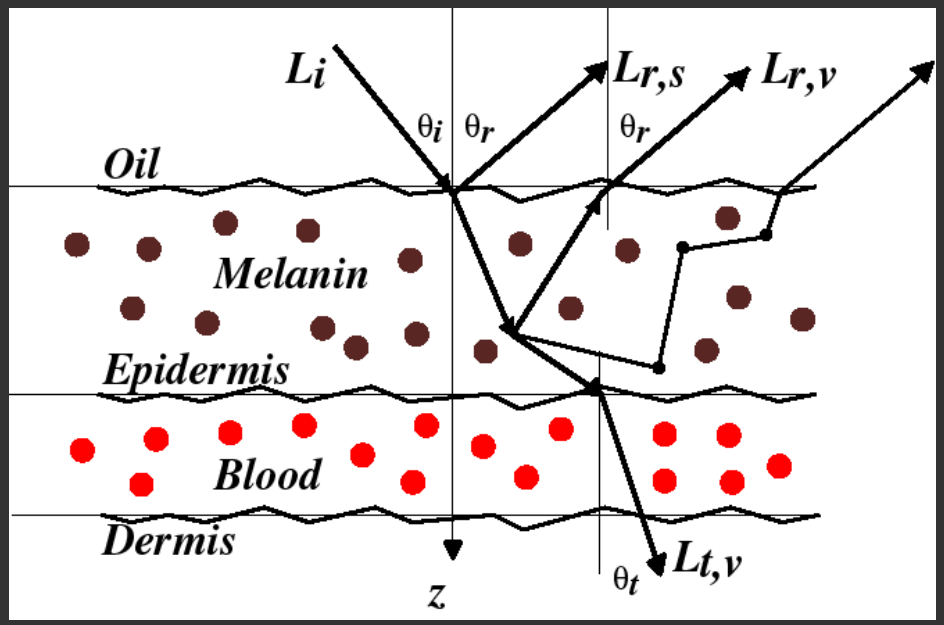
Grace Stanke

BS Engineering Physics / Nuclear Engineering 2023

Miss Wisconsin 2022 and Miss America 2023

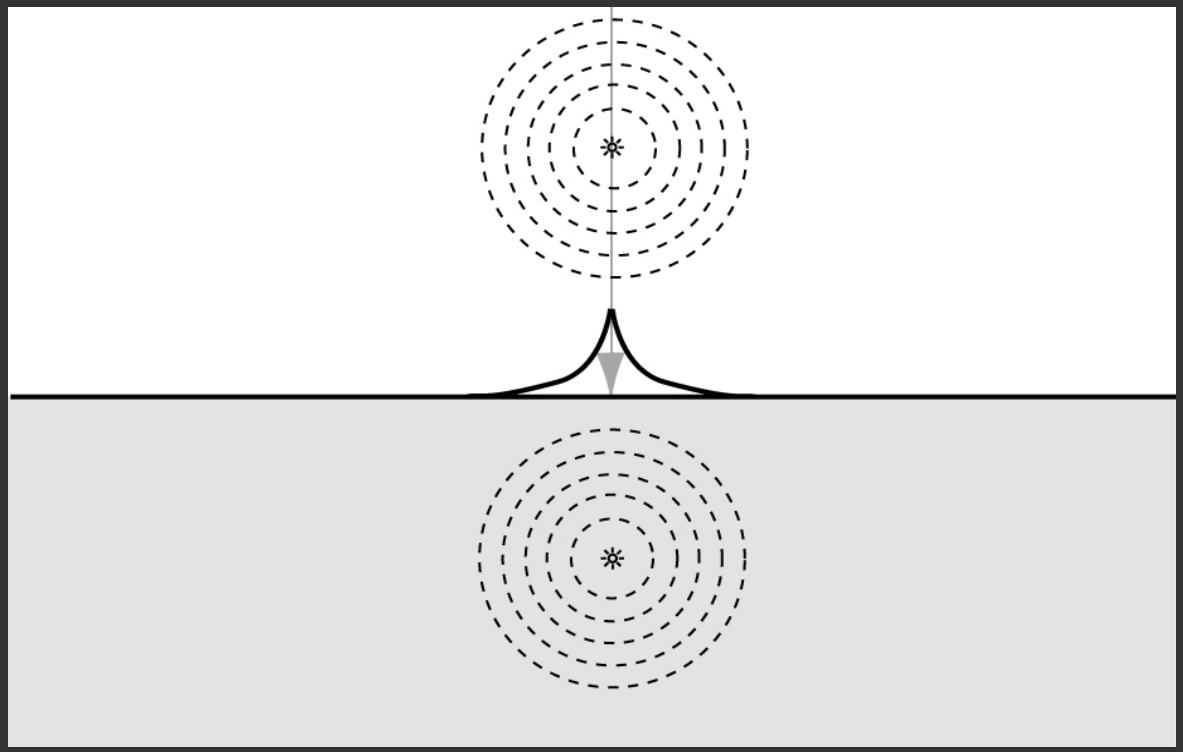
Physical Simulation

Reflection from Layered Surfaces



Hanrahan and Krueger

Diffusion Approximation



Two virtual light sources

Diffuse Milk

Skim Milk

Whole Milk

The Big Idea in Computer Graphics

Ivan Sutherland

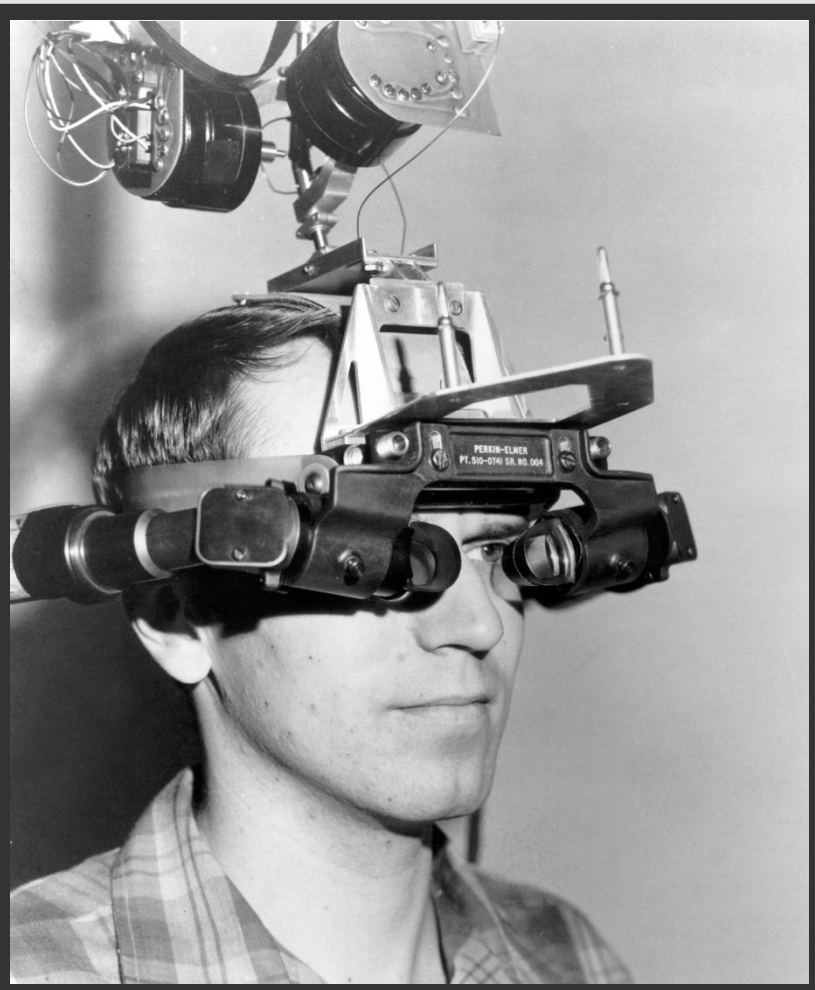
The Ultimate Display

1965

The display is a window

into a virtual world

E&S built flight simulators



29 hours / frame - 100 million hours total of CPU time for the movie

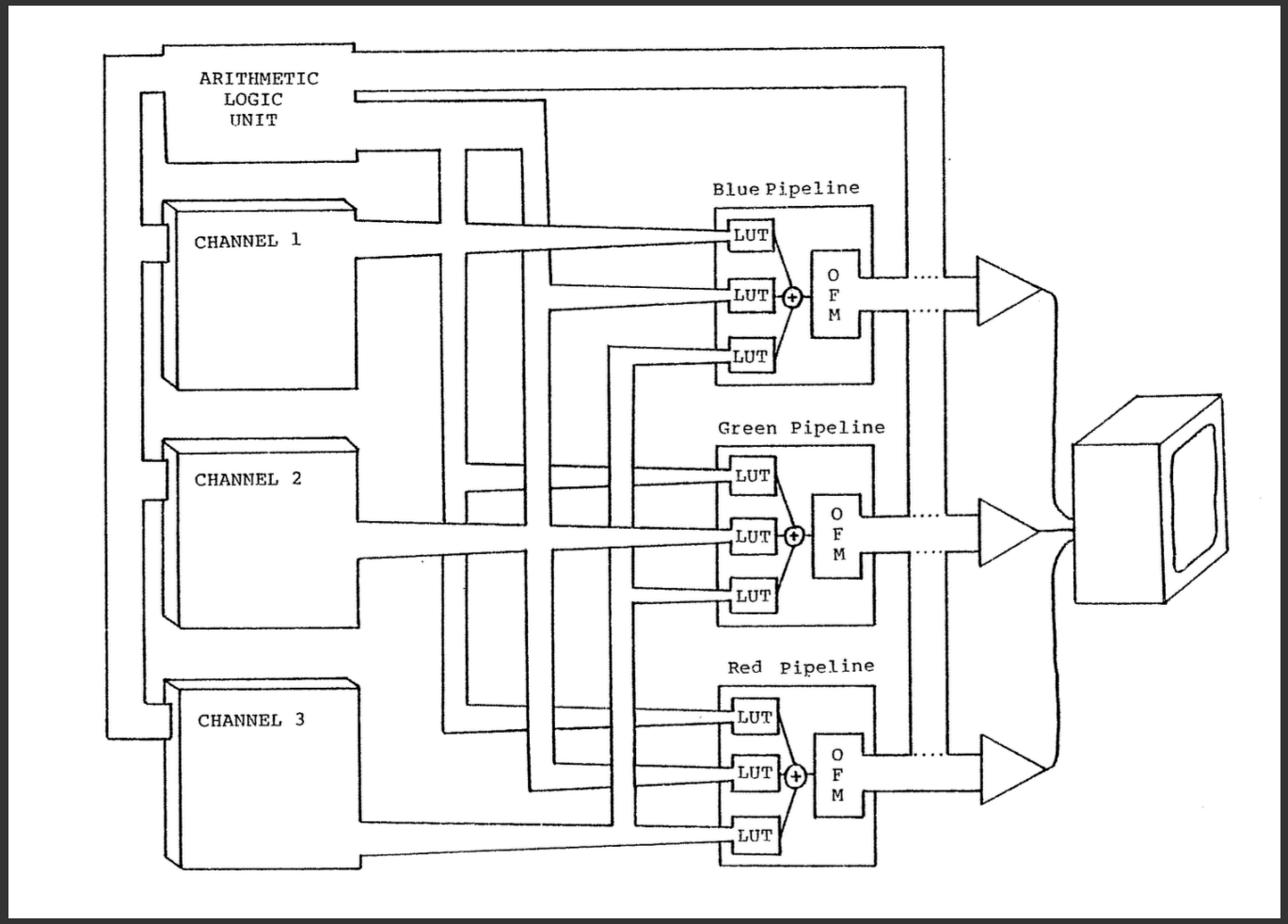
29 hours/frame = 29 x 60 x 60 seconds/frame = 104,400 seconds/frame

100K seconds/frame x 10 GFLOPs = ~1 PFLOP

1 PFLOP / 1 Megapixel = ~1 GFLOP / pixel

My Desperate Quest for Cycles

Photograph by Donna Cox (1983)



Stanford Technology Corporation 70F
circa 1977

PROCEDURES FOR PARALLEL ARRAY PROCESSING ON A PIPELINED DISPLAY TERMINAL

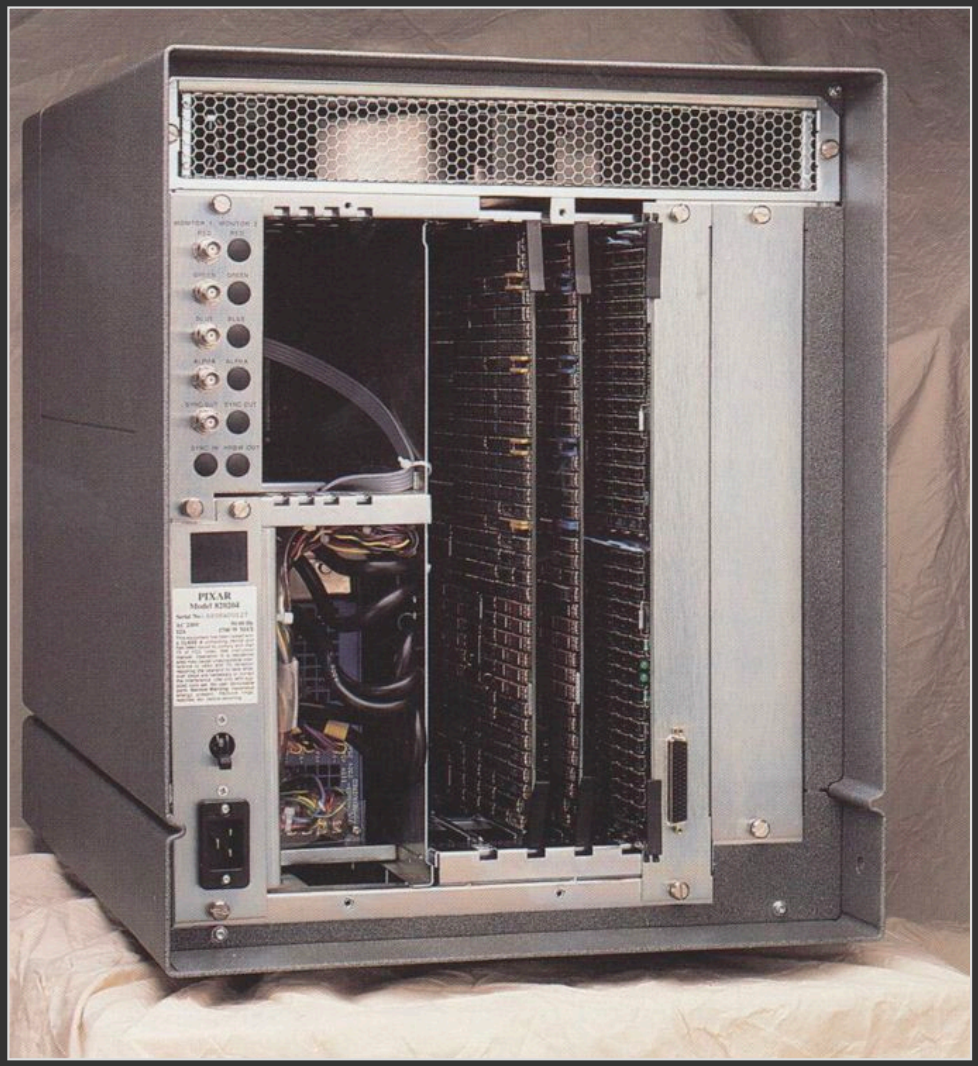
by

Pat Hanrahan

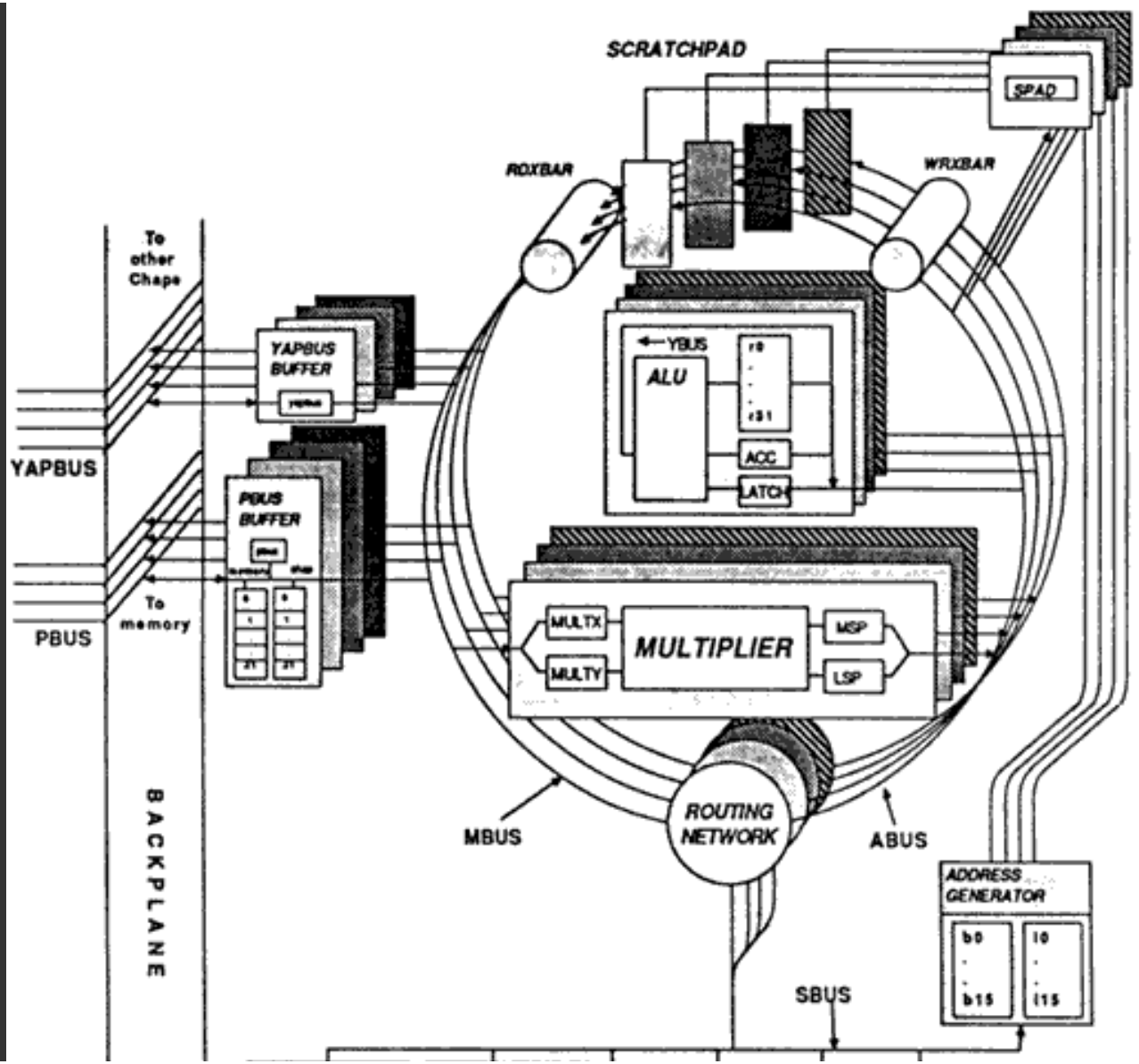
Computer Sciences Technical Report #490

December 1982

```
procedure Life;
 const
  world = 1; eq2 = 2; eq3 = 3;
 function alive : boolean;
  begin
   alive := ||tst
  end;
 procedure generation;
  procedure neighbors;
   begin
        ||clr;
        ||adi(world,1,-1, 1);
        ||adi(world,1, 0, 1);
        ||adi(world,1, 1, 1);
        ||adi(world,1,-1, 0);
        ||adi(world,1, 1, 0);
        ||adi(world,1,-1,-1);
        ||adi(world,1, 0,-1);
        ||adi(world,1, 1,-1);
   end;
  begin
   neighbors;
   ||eq( 2 ); ||mov(status,eq2);
   ||eq( 3 ); ||mov(status,eq3);
   ||lda(world);
   ||and(eq2,0,0);
   ||or (eq3,0,0);
   ||sta(world);
  end;
 begin
  boot;
  ||shw(world);
  while alive do generation;
 end;
```



Pixar Image Computer 1986


SCRATCHPAD
SPAD
ROXBAR
WRXBAR
To other Chips
YAPBUS BUFFER
YAPBUS
PBUS BUFFER
PBUS
To memory
YBUS
ALU
r0
r31
ACC
LATCH
MULTX
MULTY
MULTIPLIER
MSP
LSP
MBUS
ROUTING NETWORK
ABUS
ADDRESS GENERATOR
b0
b15
i0
i15
SBUS
BACKPLANE

Features

4-way SIMD (used in future graphics/media processors)

- RGBA and XYZW

Tesselated XY memory

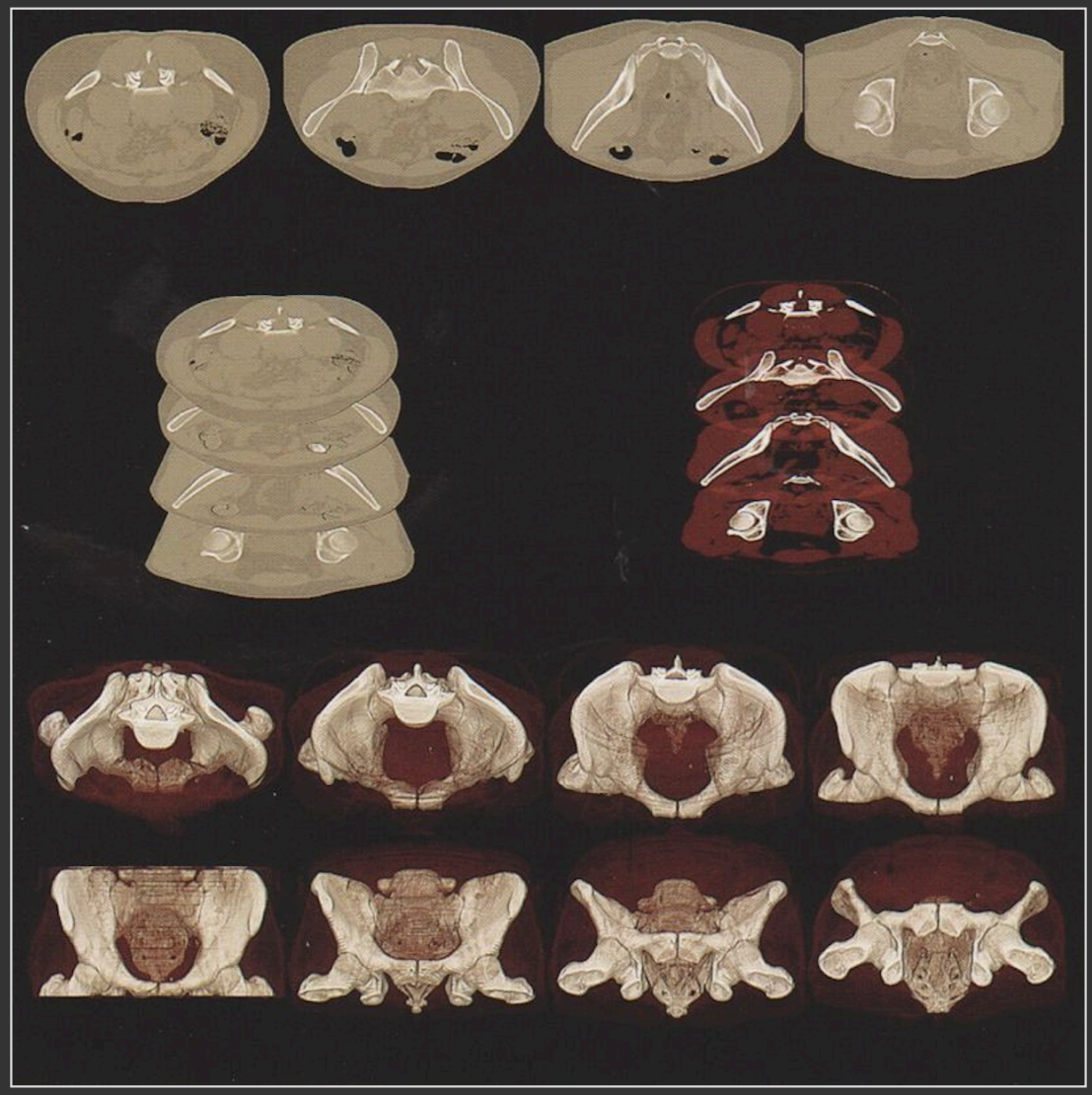
- Fetch row, column, broadcast, indexed

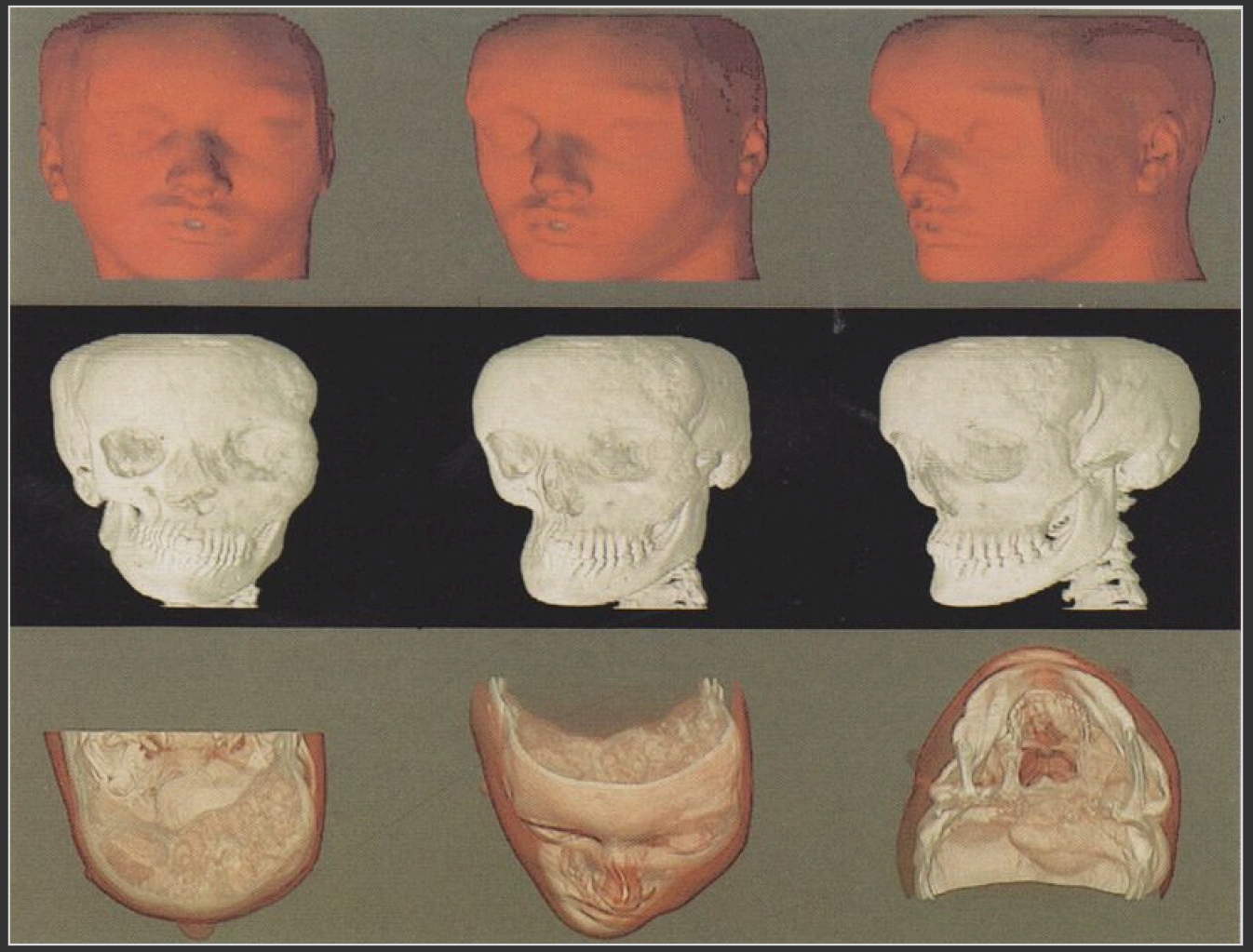
Predication with structured control flow

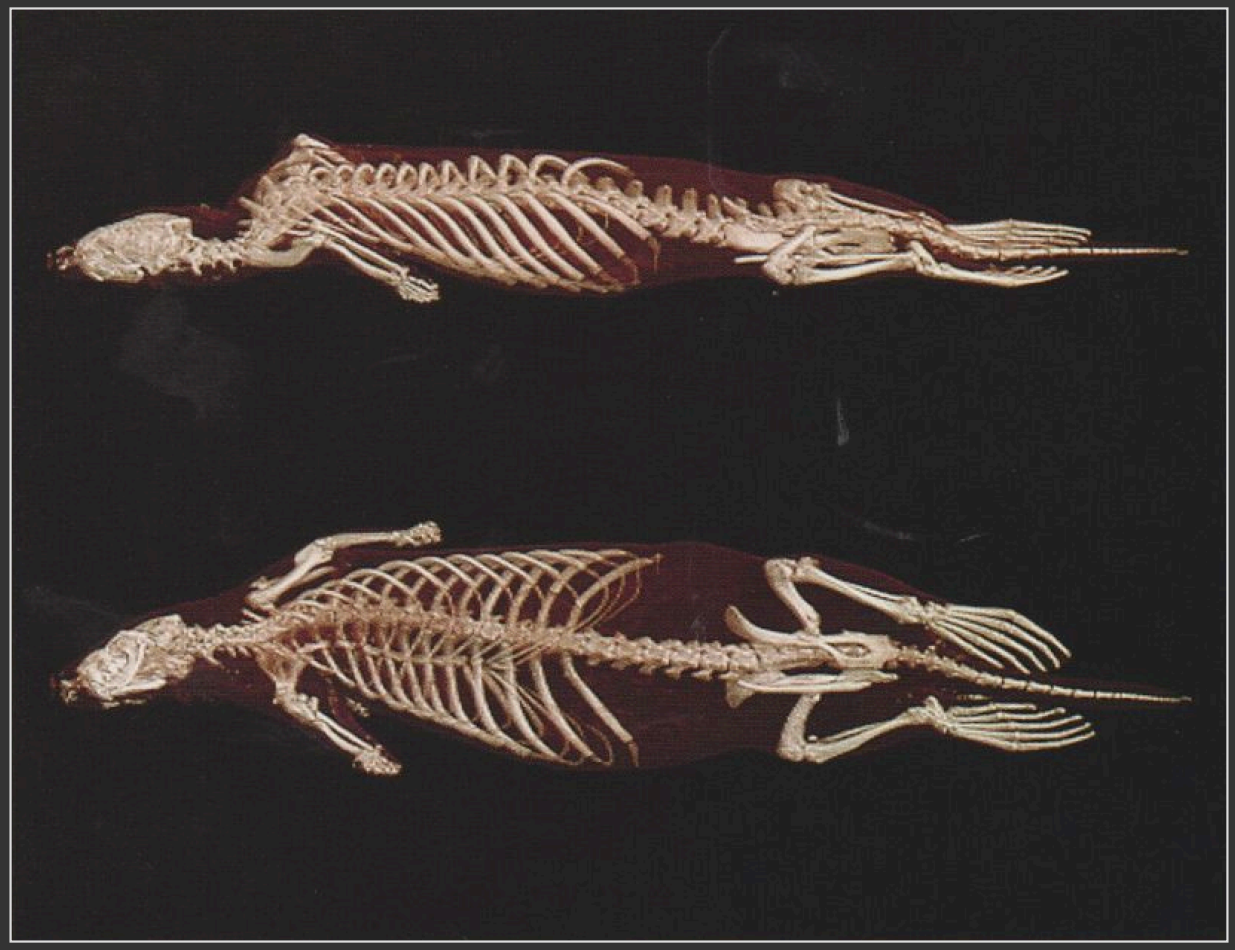
- if-then-else, while, ...
- Efficiently skip instructions if the predicate is all 0's

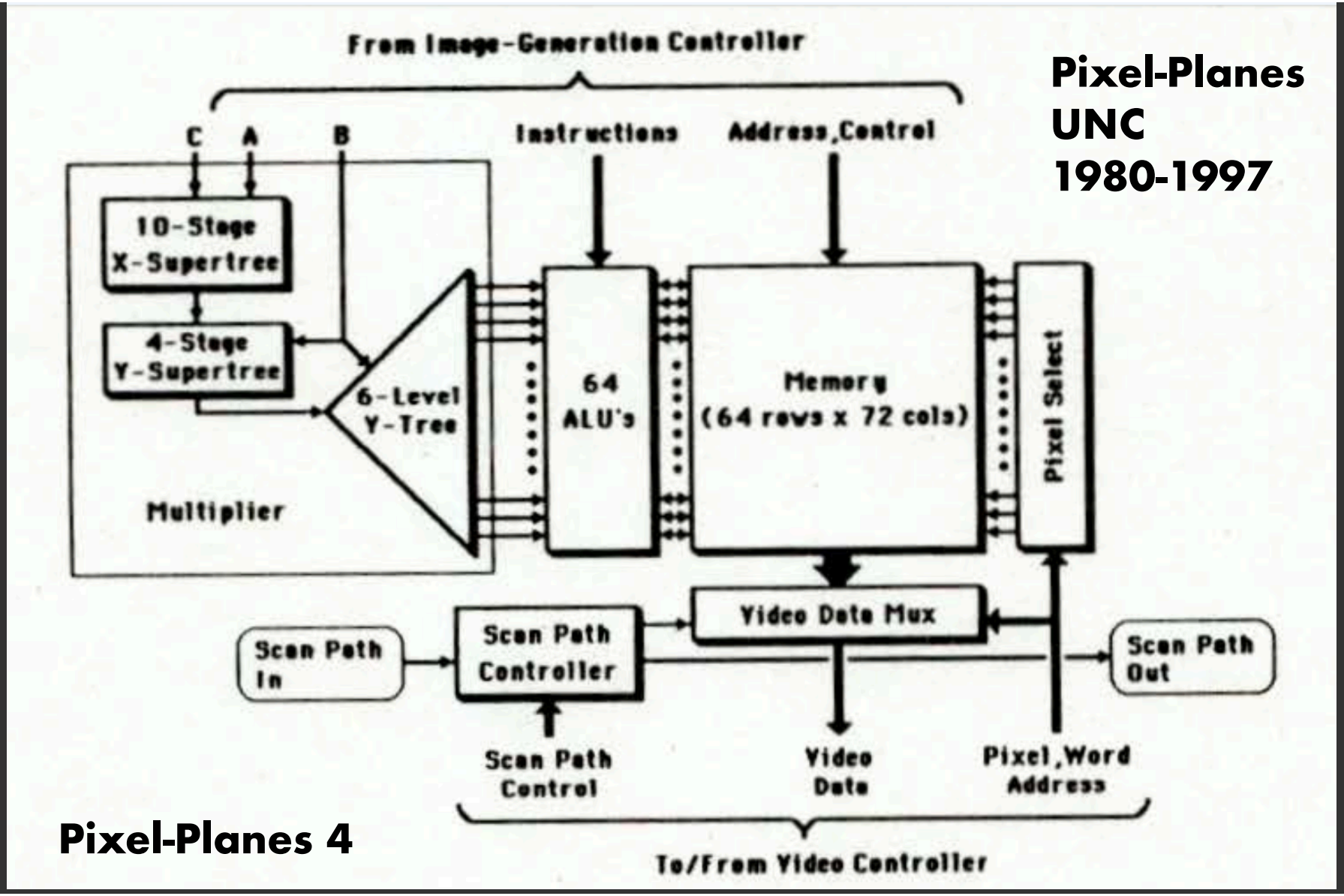
Chap: A SIMD Graphics Processor, A. Levinthal, T. Porter

Parallel Computers for Graphics Applications, A. Levinthal, P. Hanrahan, M. Paquette, J. Lawson





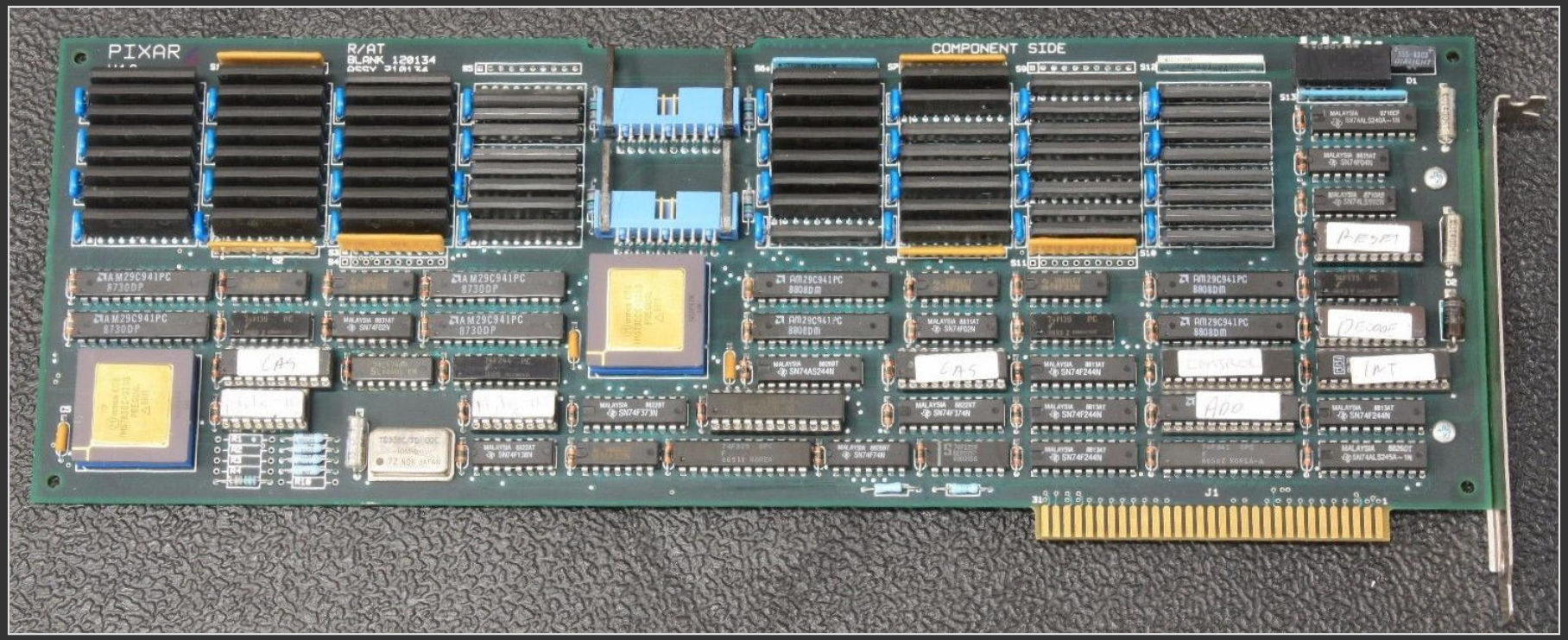




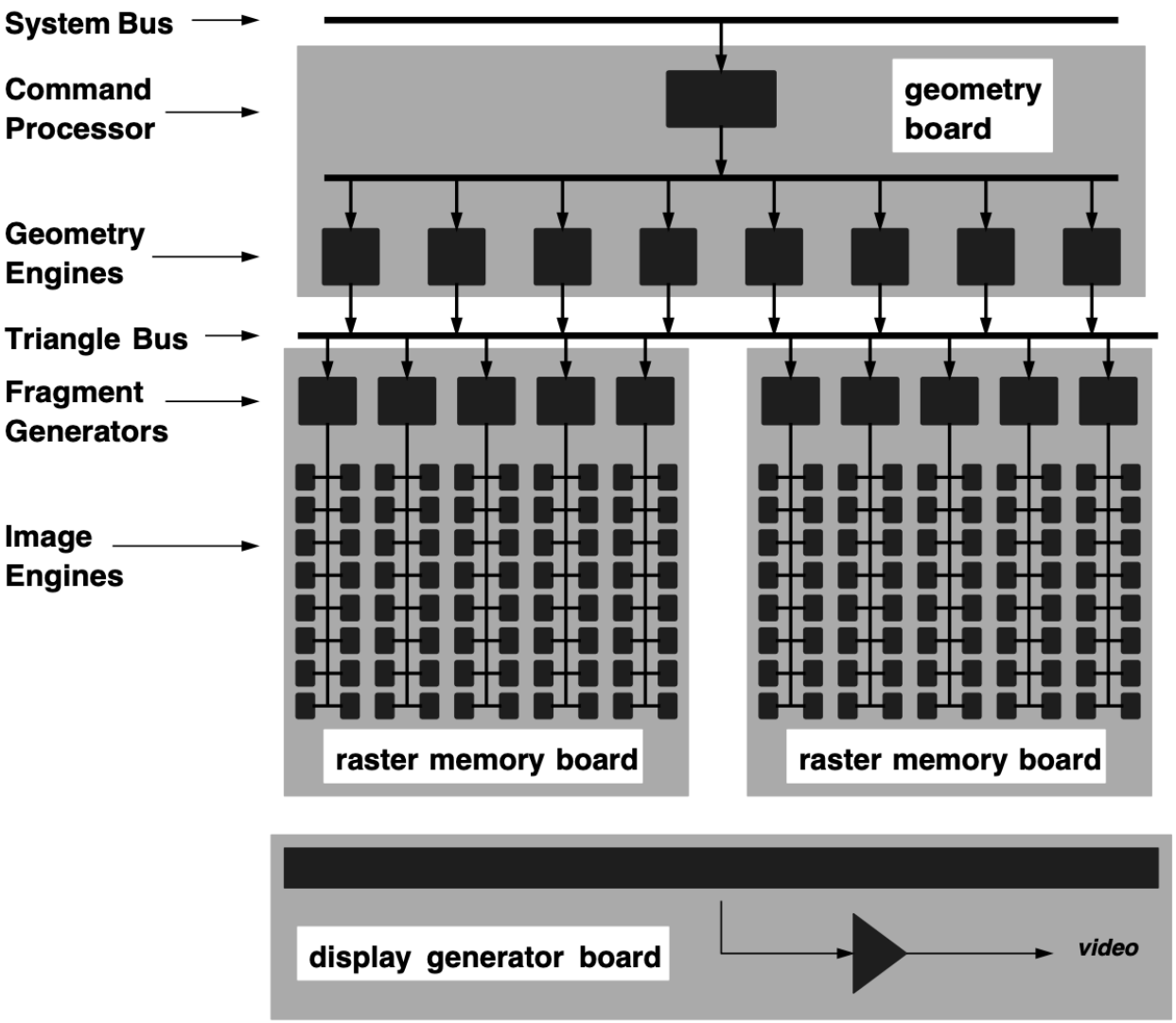
Pixel-Planes
UNC
1980-1997

Pixel-Planes 4

Pixar Dual Inmos T800-2 Transputer Board 1987



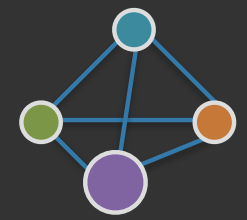
Evolved from the Reyes Machine

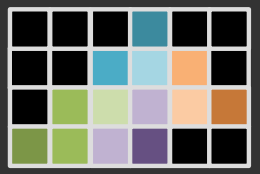


RealityEngine Graphics
K. Akeley
1993

SGI
- GT
- GTX
- VGX
- RE
- IR

The OpenGL Graphics Pipeline





Application

Command

Vertex

Rasterization

Texture

Fragment

Display

OpenGL Specifies an Architecture

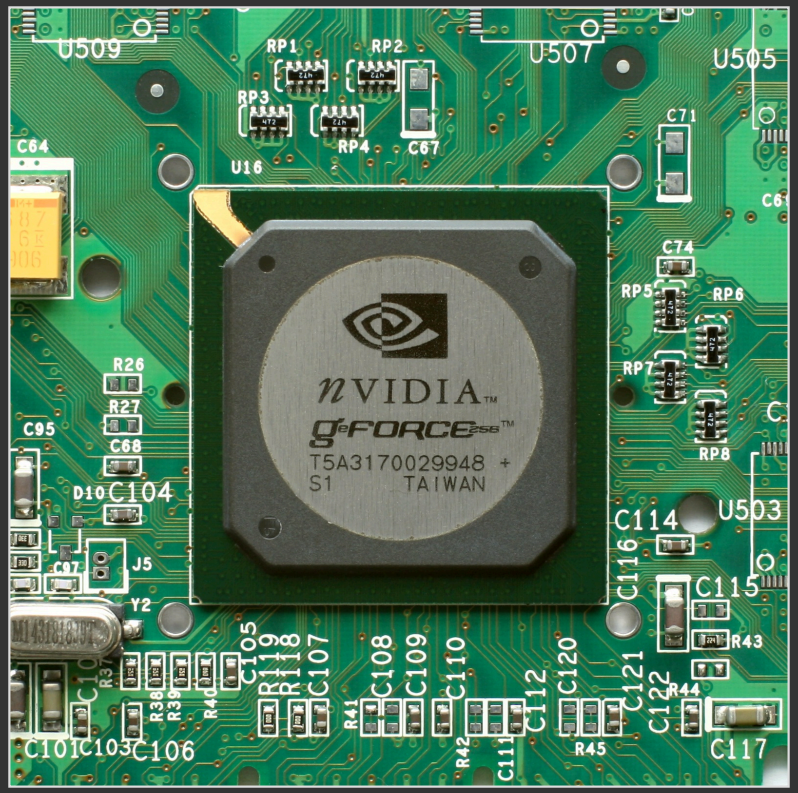
Describes an abstract graphics pipeline

- Specifies graphics state including the frame buffer
- Specifies how OpenGL API calls update the state
- Similar to a CPU instruction set architecture

Architecture specification vs implementation

a la Fred Brooks (1931-2022)

NVIDIA Geforce 256 (1999)
First GPU (complete graphics pipeline on a single chip)



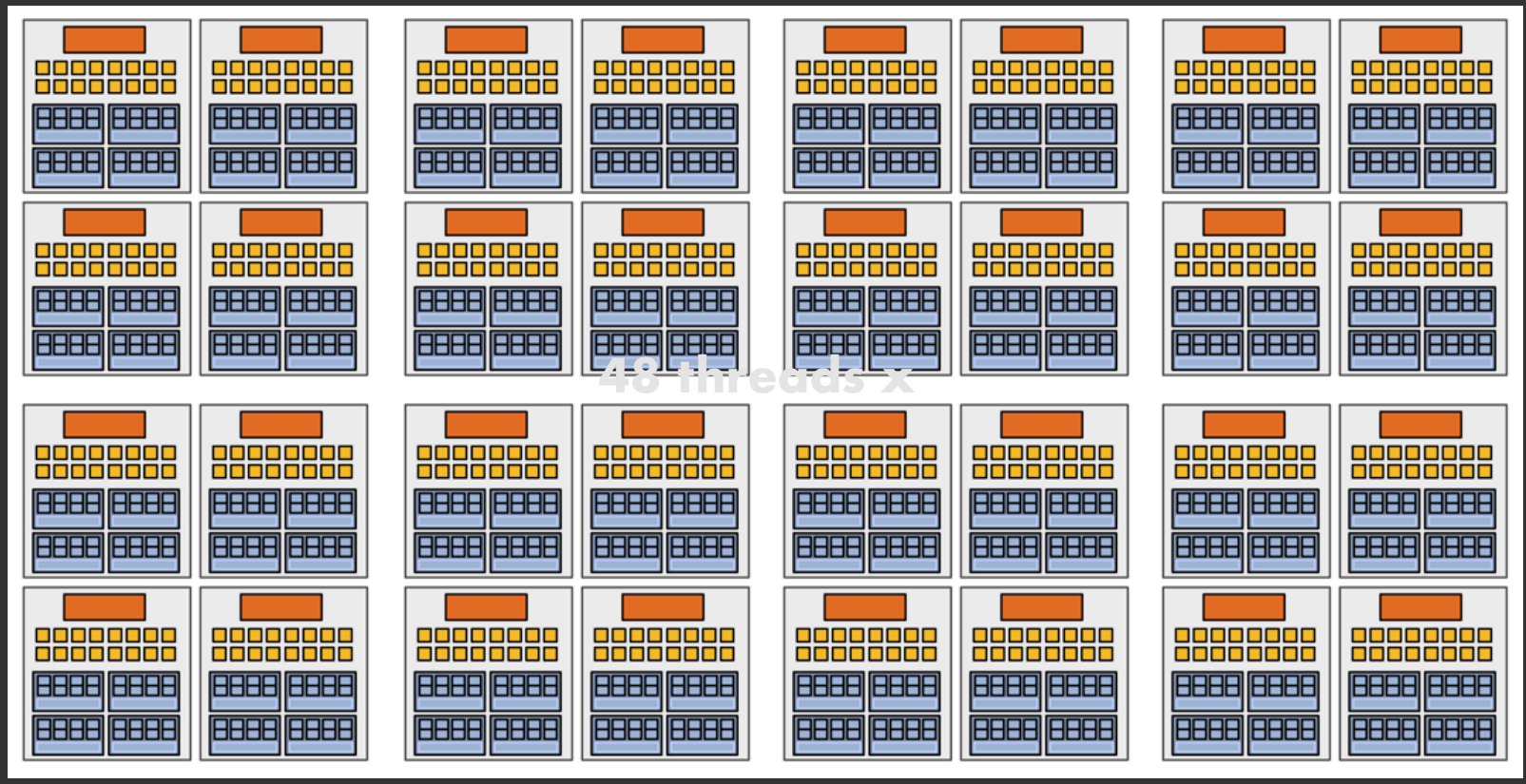
TSMC 220 nm
17 million transistors

Programmable Vertex and Fragment Stages

Fragment shader assembly language program

```
DCL       t0.xy          # Interpolate t0.xy
DCL       v0.xyzw        # Interpolate v0.xyzw
DCL_2D    s0             # Declaration - no code
TEX1D     r0, t0, s0     # TEXTURE LOAD!
MUL       r1, r0, v0     # Multiply
MOV       oC0, r1        # Store to framebuffer
```

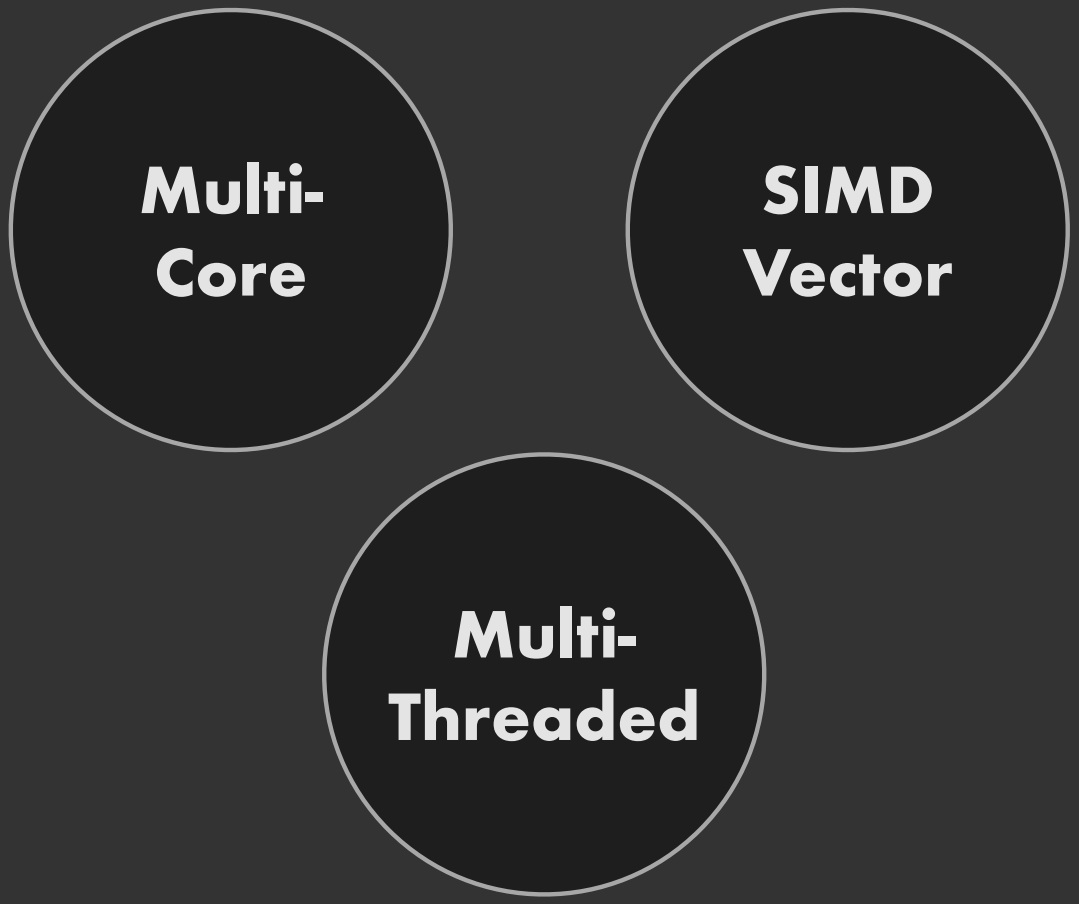
Unified Shaders (NVIDIA GeForce 8, 2007)

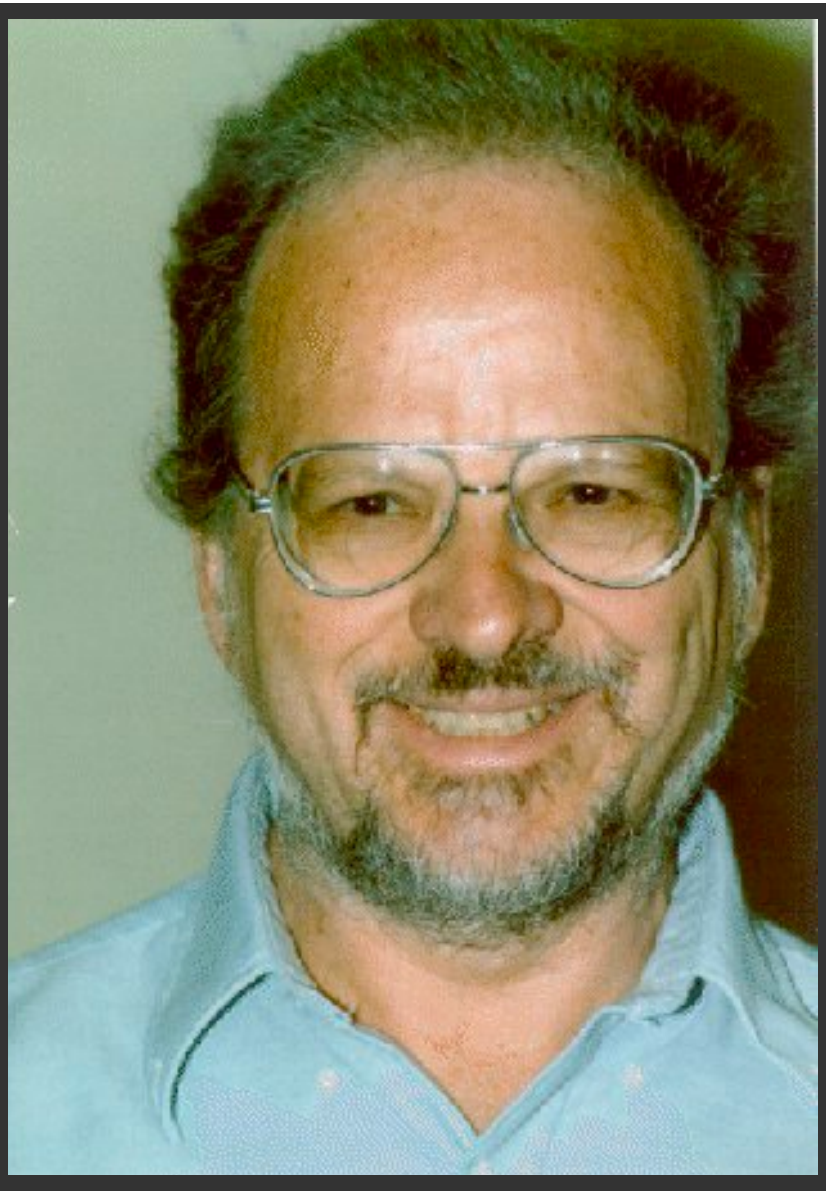


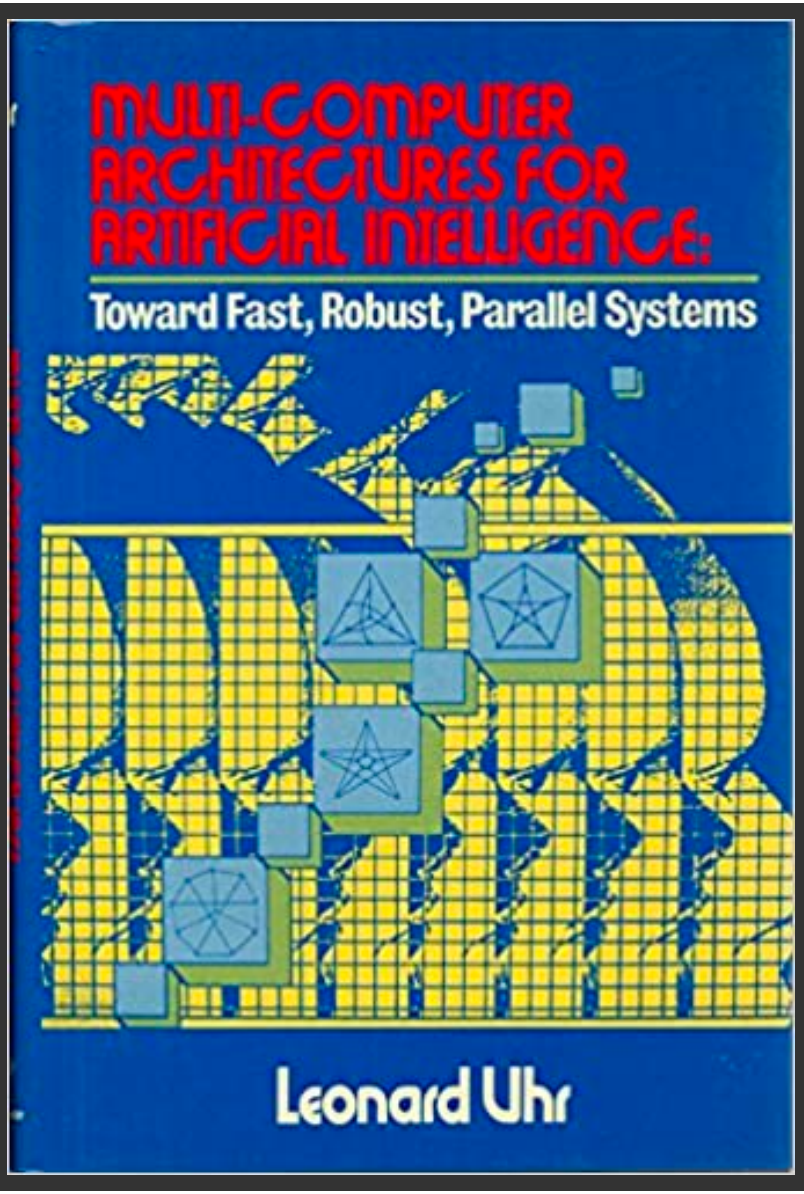
16 cores x 32 SIMD ALUs x 2 flops/cycle x 1 GHz = 1 TFLOP

512 shader cores x 48 threads/core = 24,576 threads

GPUs use Multiple Forms of Parallelism






MULTI-COMPUTER ARCHITECTURES FOR ARTIFICIAL INTELLIGENCE:
Toward Fast, Robust, Parallel Systems
Leonard Uhr

1987

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

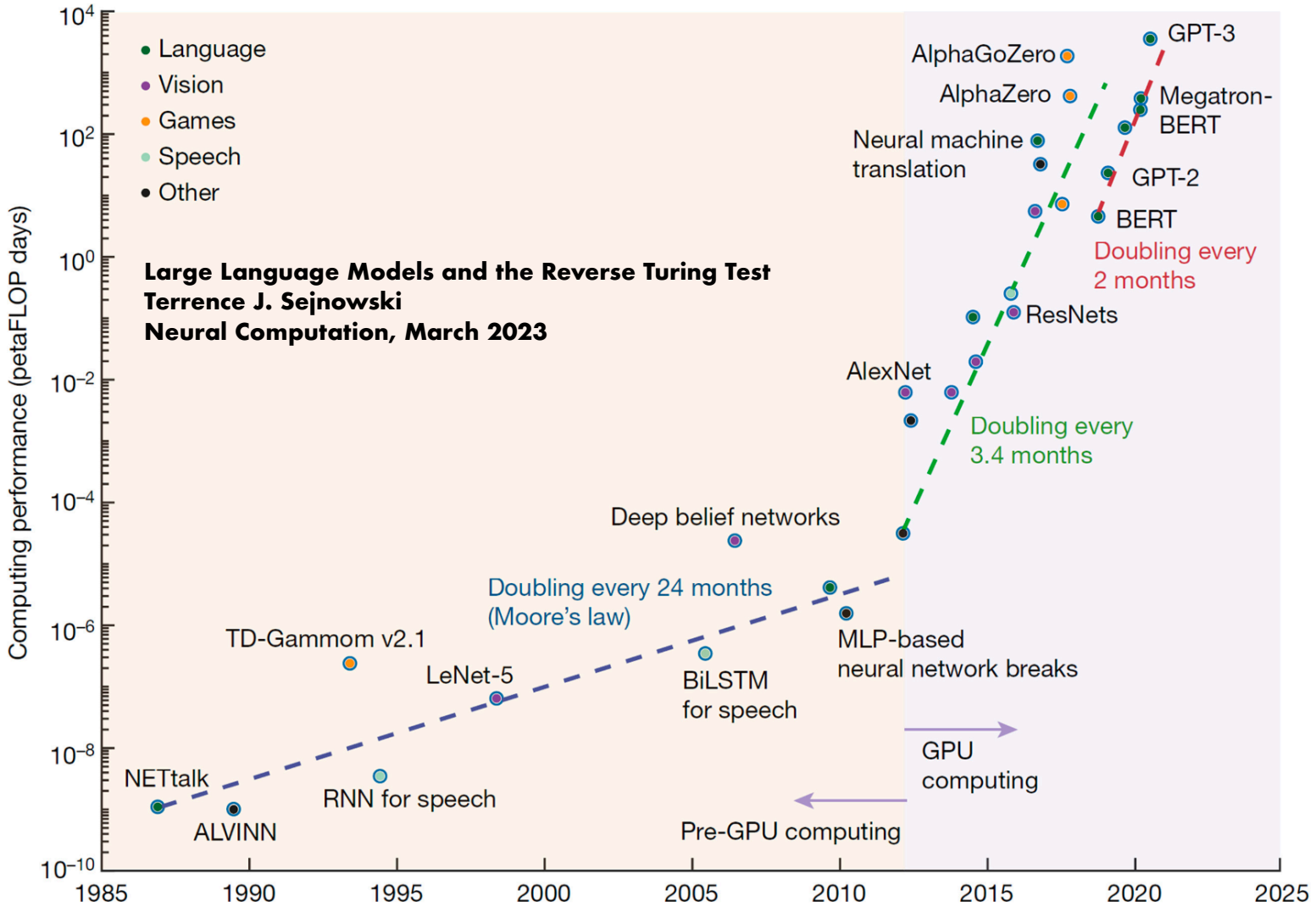
Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

2012


Language
Vision
Games
Speech
Other
Large Language Models and the Reverse Turing Test
Terrence J. Sejnowski
Neural Computation, March 2023
Computing performance (petaFLOP days)
10^4
10^2
10^0
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
1985
1990
1995
2000
2005
2010
2015
2020
2025
GPT-3
AlphaGoZero
AlphaZero
Megatron-BERT
Neural machine translation
GPT-2
BERT
Doubling every 2 months
ResNets
AlexNet
Doubling every 3.4 months
Deep belief networks
Doubling every 24 months (Moore's law)
TD-Gammom v2.1
LeNet-5
BiLSTM for speech
MLP-based neural network breaks
GPU computing
NETtalk
RNN for speech
ALVINN
Pre-GPU computing

GPT and GPUs

AlexNet “takes between five and six days to train on two GTX 580 3GB GPUs”

MetaAI research used 2,048 80gb A100s for 5 months to train the LLaMa suite of language models. This is 7.4m A100 hours

OpenAI uses 285,000 CPU cores and 10,000 A100s (currently upgrading to 30,000) to train GPT-4

According to Microsoft the system would rank #5 in the top 500 list

`https://news.microsoft.com/source/features/ai/openai-azure-supercomputer/`

Parallel Worlds

The metaverse will require scalable multiphysics simulation

- Monte Carlo path tracing for lighting simulation
- Rigid bodies with collisions and friction
- Water, clouds, wind, cloth, muscles, hair, ...

Data analysis and machine learning

- Generative AI: GPT and stable diffusion
- Reinforcement learning (predictive simulation and ML)

The Quest for Cycles Continues!

Thank You

Resources

P. Hanrahan, E Catmull, The Design of RenderMan

P. Hanrahan, Shading Languages and the Emergence of Programmable Graphics Systems