

# INICIOS Y EVOLUCIÓN DE LOS PROCESADORES GRÁFICOS (GPUs)

JGPU 2020

Albert García-García < [agarcia@dtic.ua.es](mailto:agarcia@dtic.ua.es) >

# CONTENIDO

**La Ley de Moore**

**El Pipeline Gráfico**

**La Unidad de Procesamiento Gráfico (GPU)**

**Primeros Pasos en Computación sobre GPUs**

**La Arquitectura CUDA**

# LA LEY DE MOORE

GORDON MOORE



# LA LEY DE MOORE

GORDON MOORE

**Cramming More Components onto Integrated Circuits.** Gordon E. Moore, 1965

**“ THE NUMBER OF TRANSISTORS ON A CHIP DOUBLES EVERY 12 MONTHS”**

– GORDON MOORE, COFUNDADOR DE INTEL, 1965

**“ THE NUMBER OF TRANSISTORS ON A CHIP DOUBLES EVERY 24 MONTHS”**

– GORDON MOORE, COFUNDADOR DE INTEL, 1975

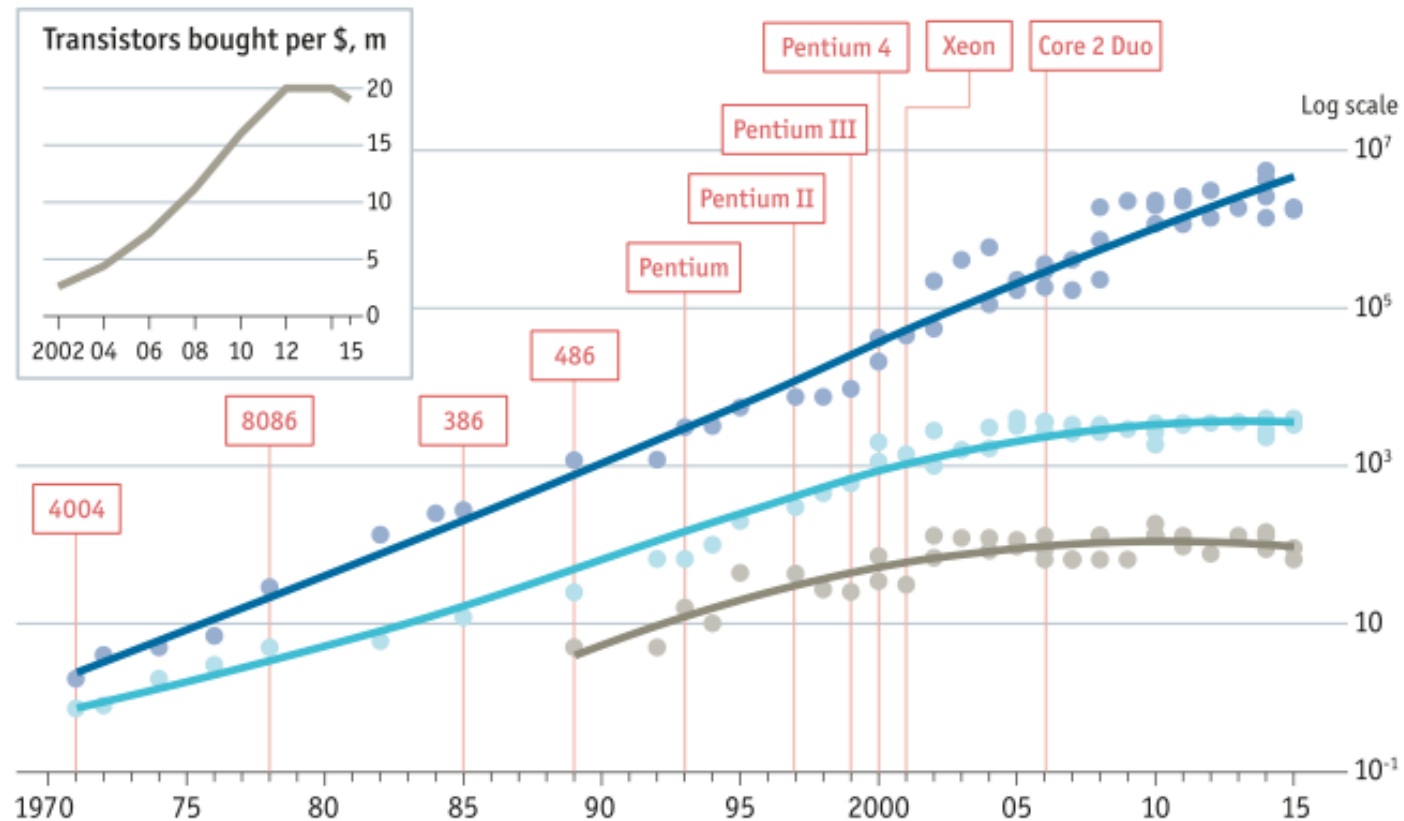


# LA LEY DE MOORE

CADA DOS AÑOS, APROXIMADAMENTE, SE DUPLICA EL NÚMERO DE TRANSISTORES

## Stuttering

● Transistors per chip, '000 ● Clock speed (max), MHz ● Thermal design power\*, w □ Chip introduction dates, selected

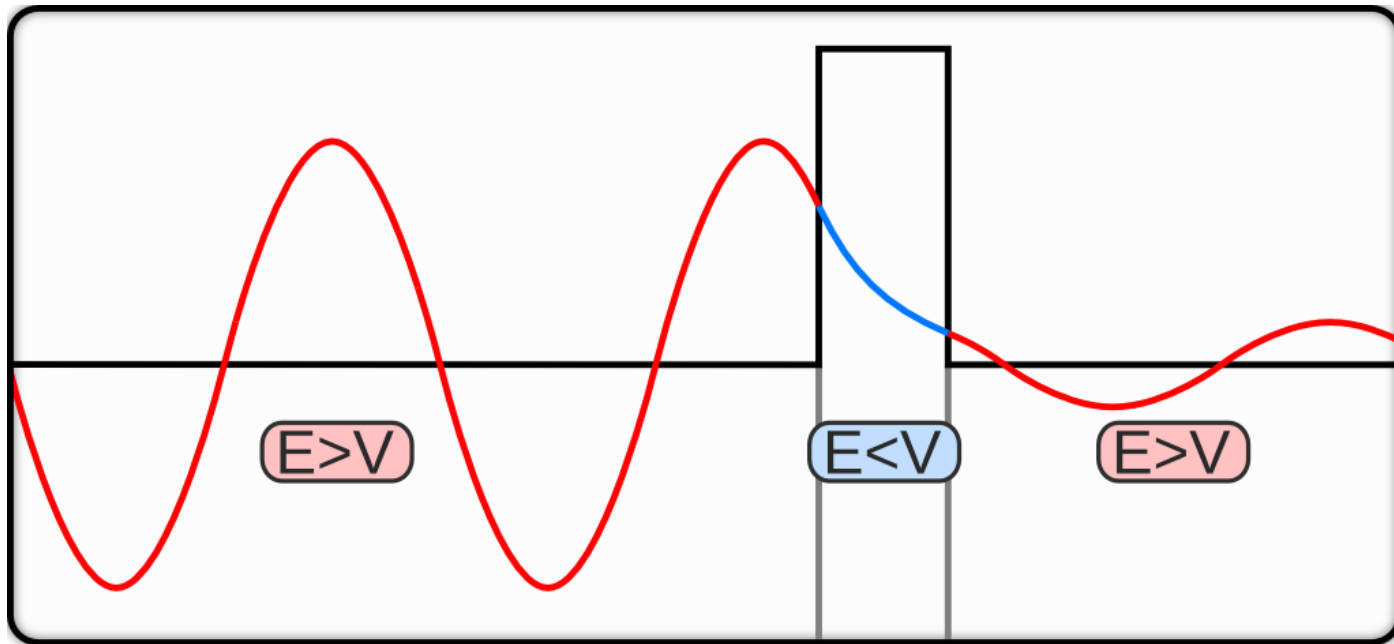


Sources: Intel; press reports; Bob Colwell; Linley Group; IB Consulting; *The Economist*

\*Maximum safe power consumption

# LA LEY DE MOORE

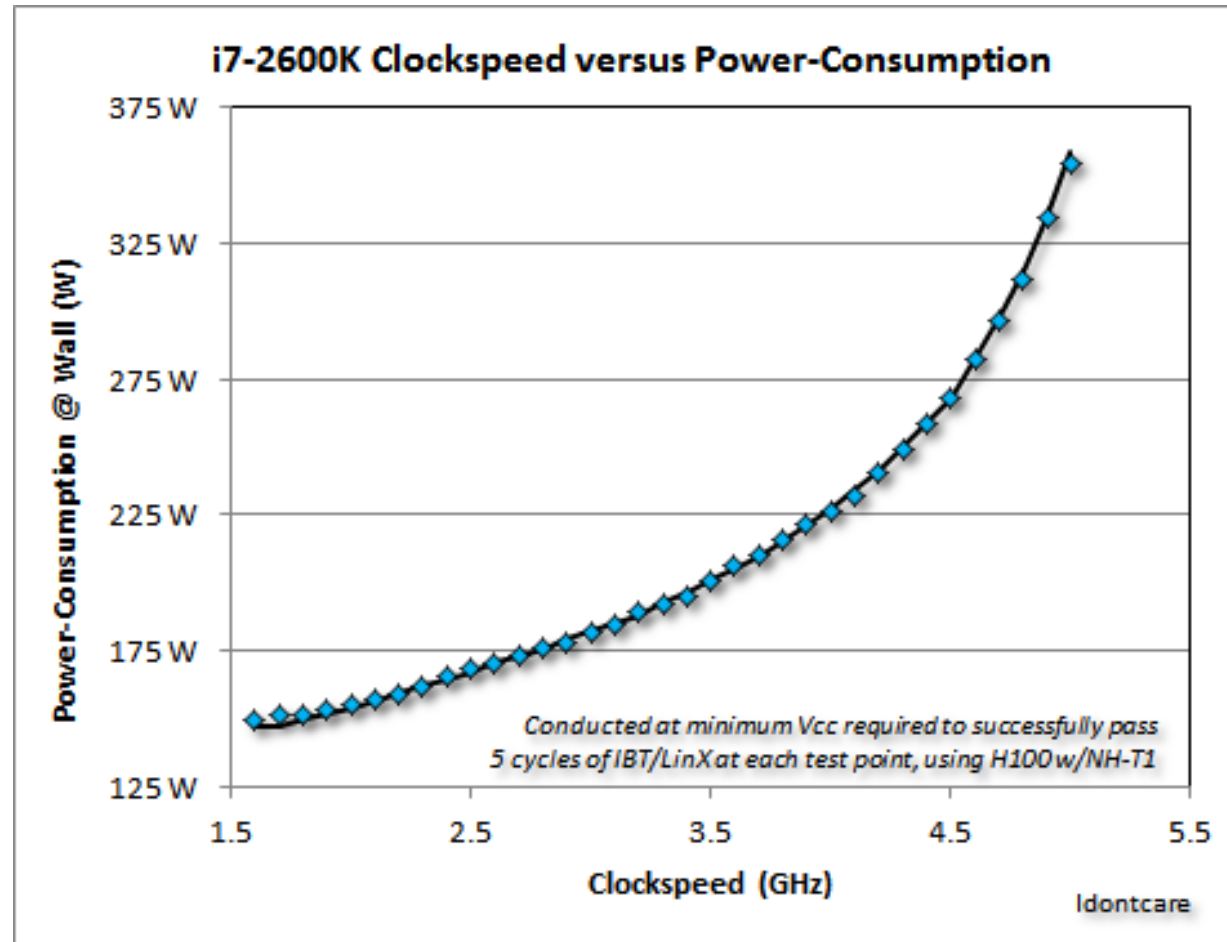
## PROBLEMAS: TAMAÑO DEL TRANSISTOR Y EFECTO TÚNEL





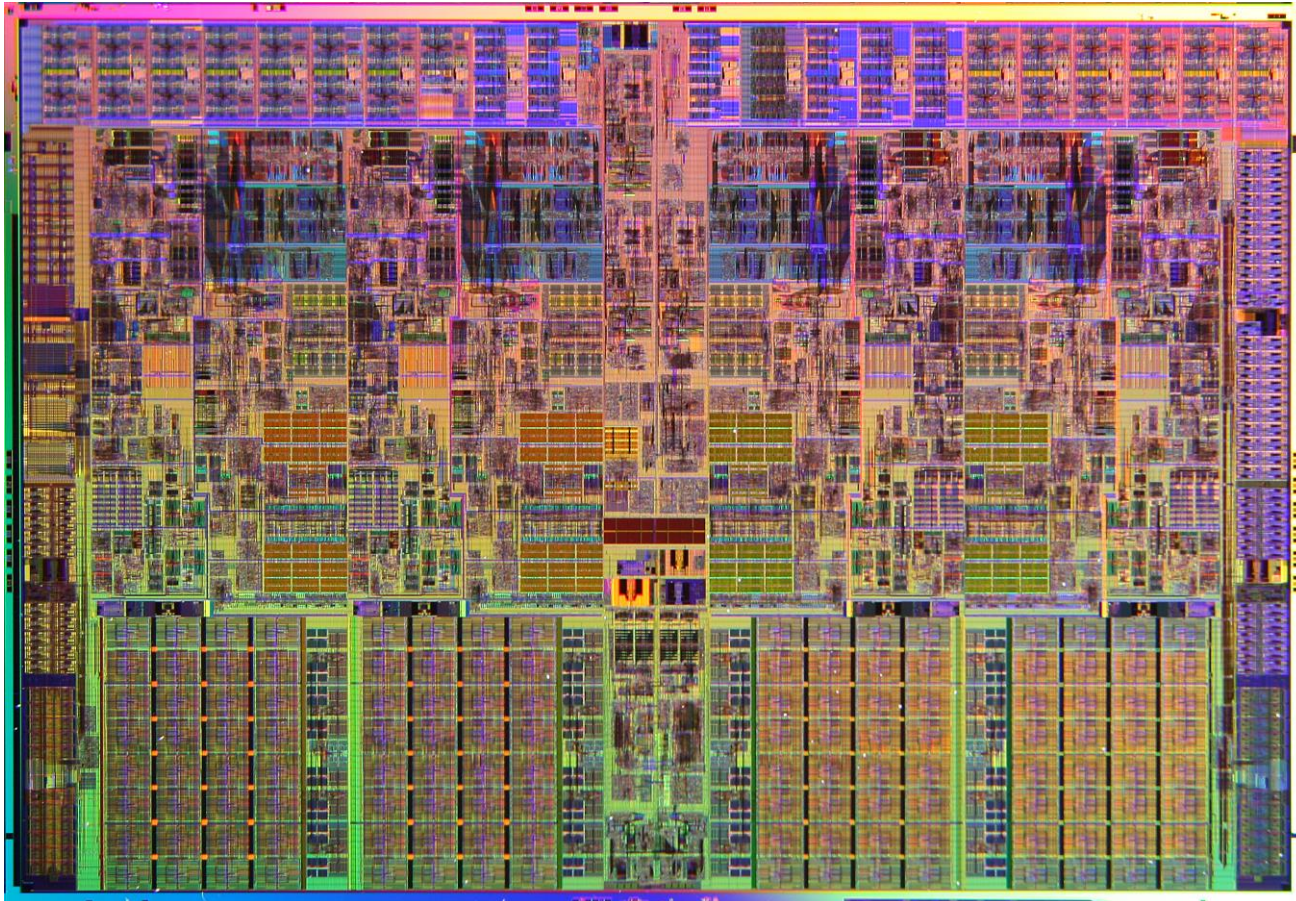
# LA LEY DE MOORE

## PROBLEMAS: CONSUMO ENERGÉTICO Y DISIPACIÓN DE CALOR



# LA LEY DE MOORE

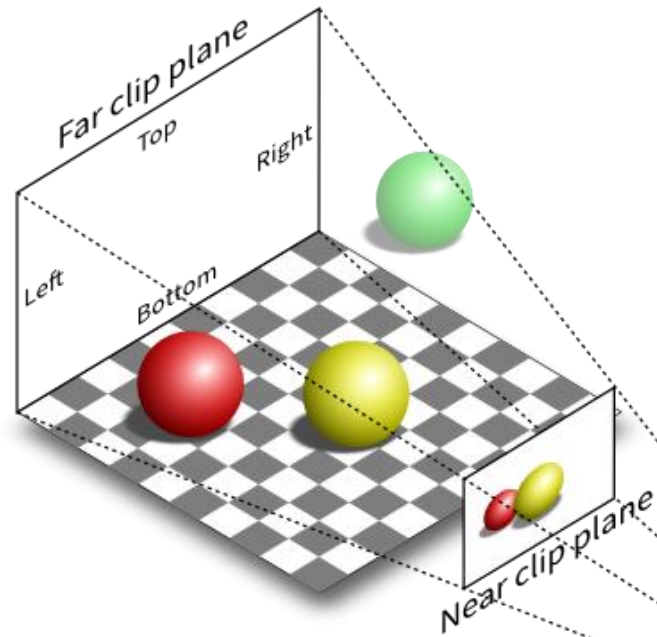
¿CÓMO CONTINUAR ESCALANDO? MULTICORE



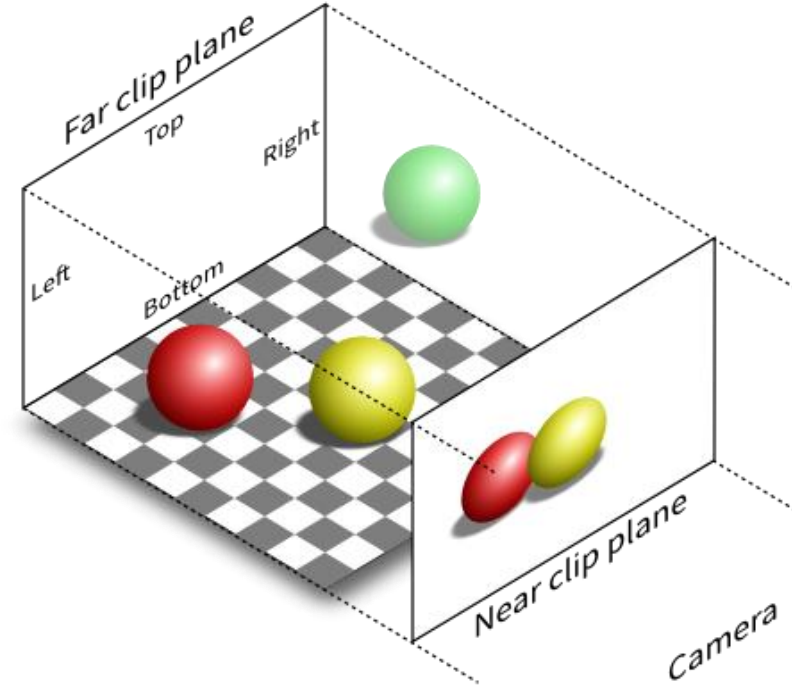


# EL PIPELINE GRÁFICO

## RENDERING



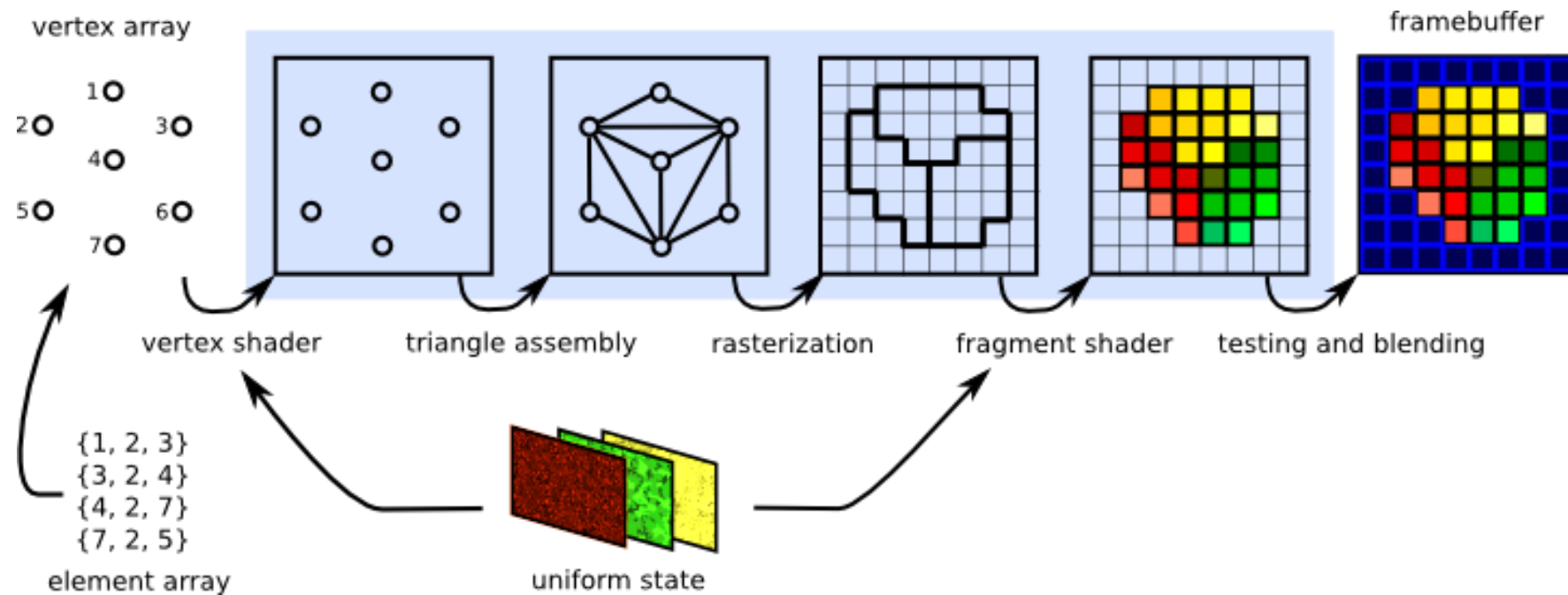
Perspective projection (P)



Orthographic projection (O)

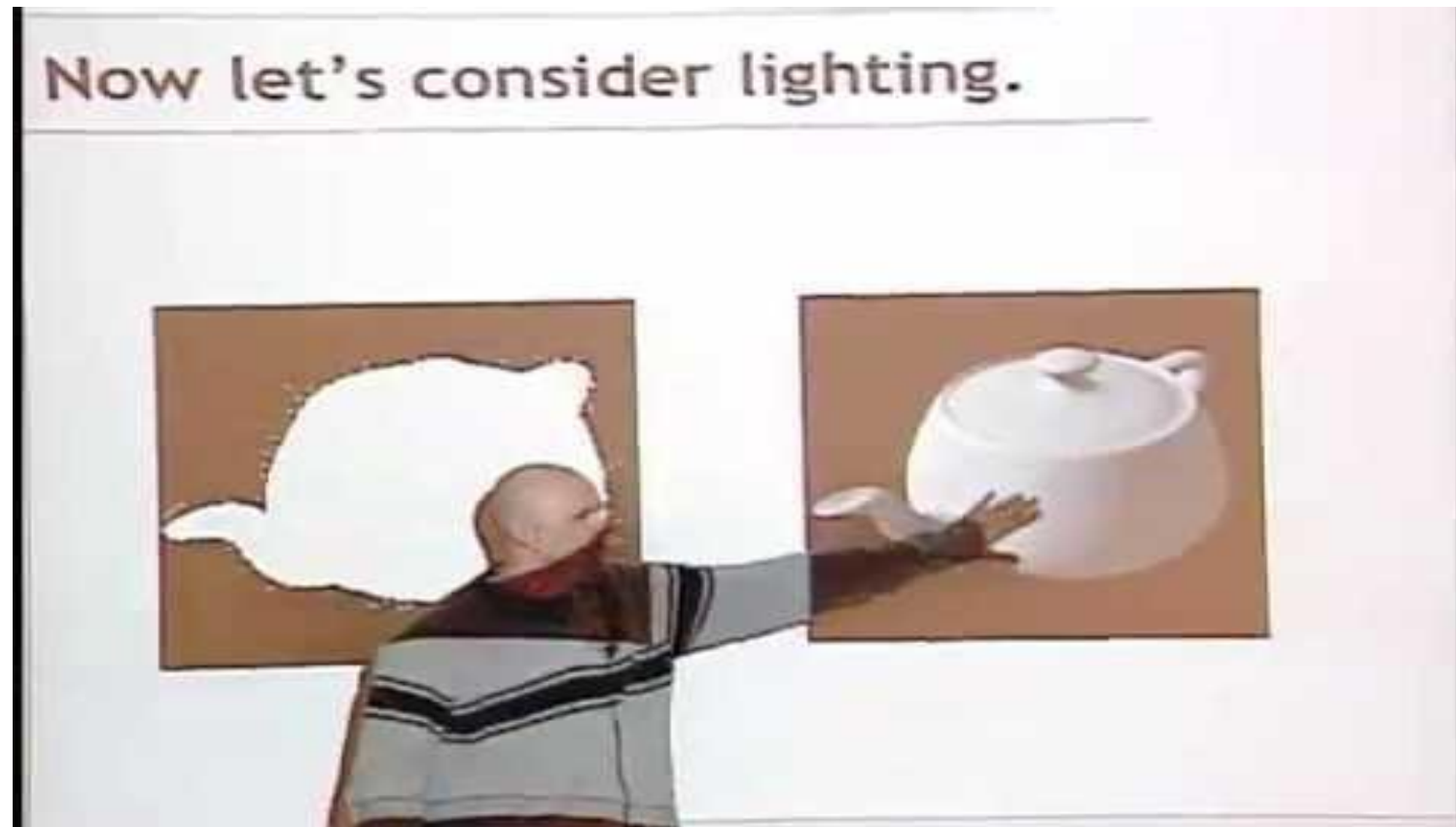
# EL PIPELINE GRÁFICO

## RENDERING



# EL PIPELINE GRÁFICO

OpenGL GRAPHICS PIPELINE OVERVIEW por Owens



# EL PIPELINE GRÁFICO

## RAY TRACING





# LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

GEFORCE 256 (1999)





# LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

## GEFORCE 256 (1999)

"A SINGLE-CHIP PROCESSOR WITH INTEGRATED TRANSFORM, LIGHTING, TRIANGLE SETUP/CLIPPING, AND RENDERING ENGINES THAT IS CAPABLE OF PROCESSING A MINIMUM OF 10 MILLION POLYGONS PER SECOND."

# LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

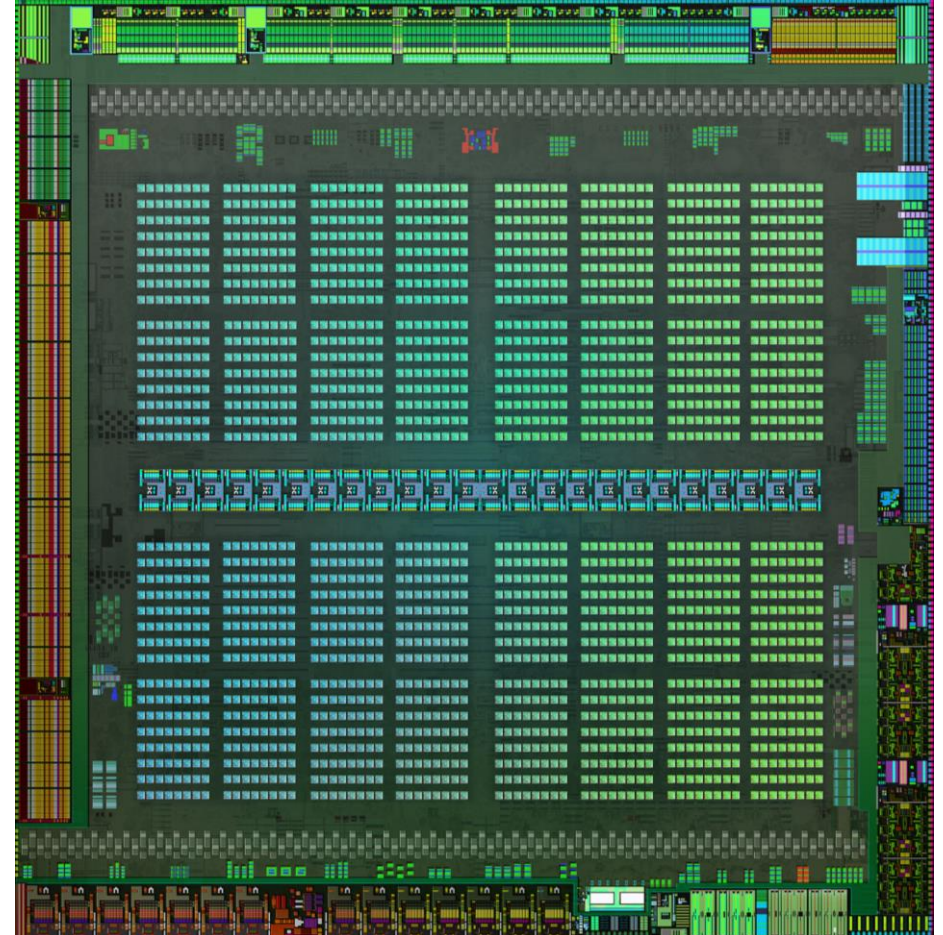
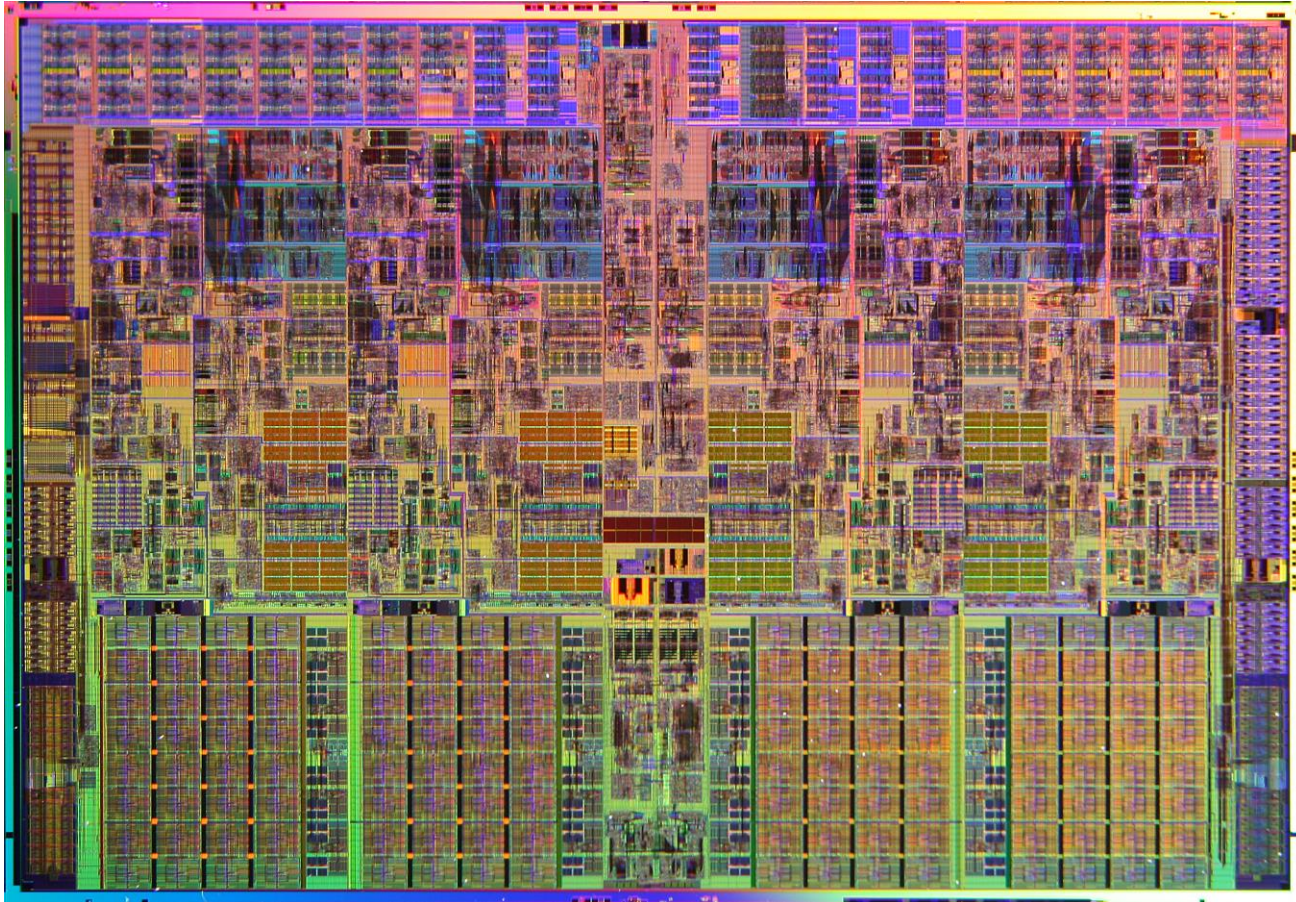
## GEFORCE 256 TECH DEMO





# LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

## LA ESENCIA DE LA GPU





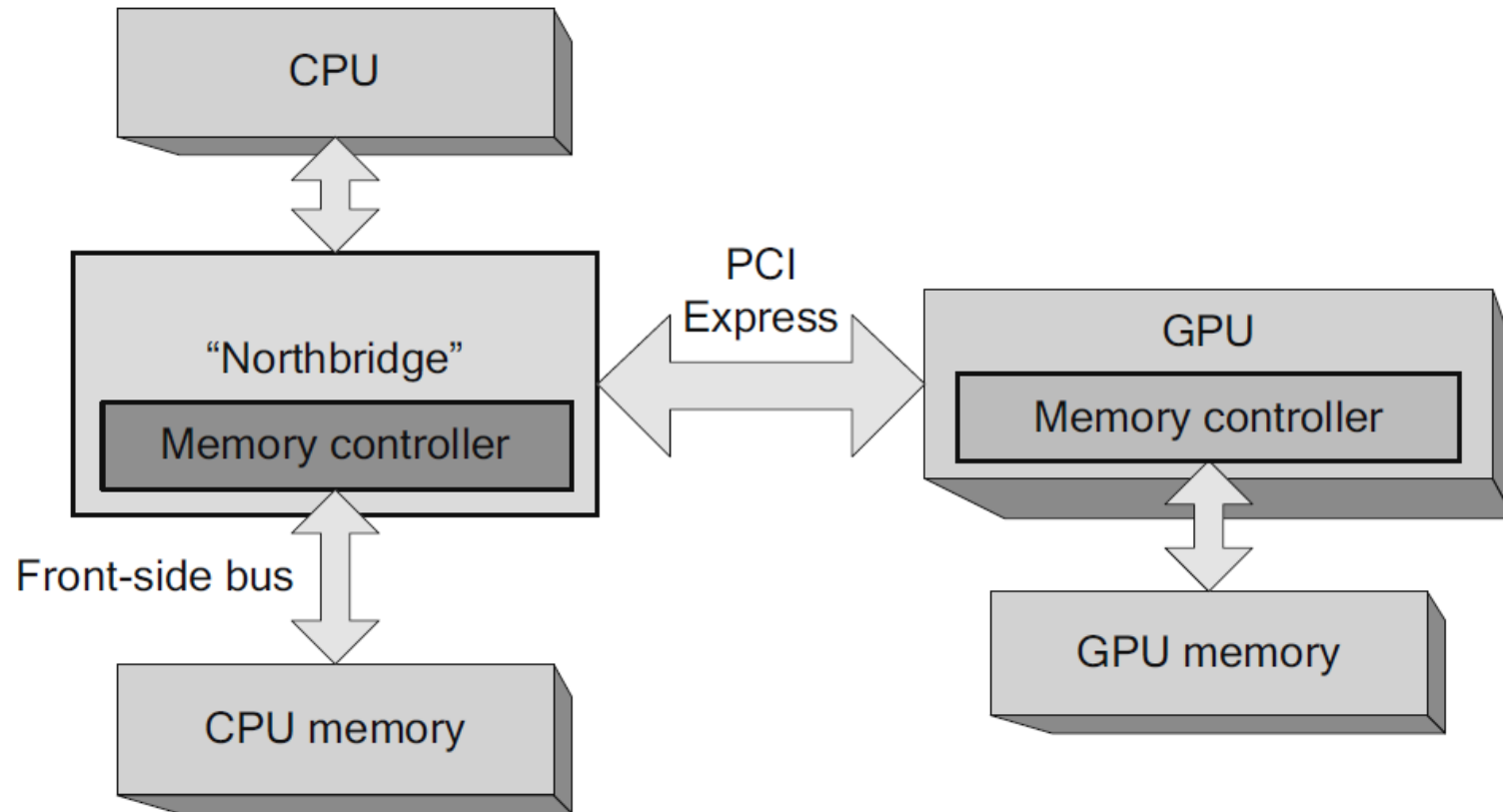
# LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

DEJEMOS QUE LOS CAZADORES DE MITOS LO EXPLIQUEN



# LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

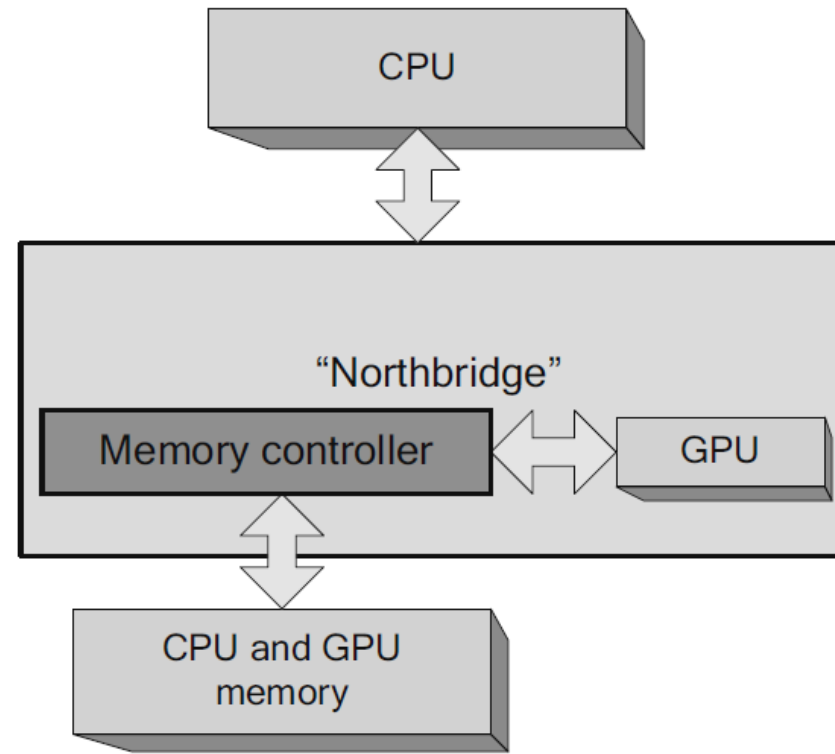
## ARQUITECTURA CPU/GPU TÍPICA





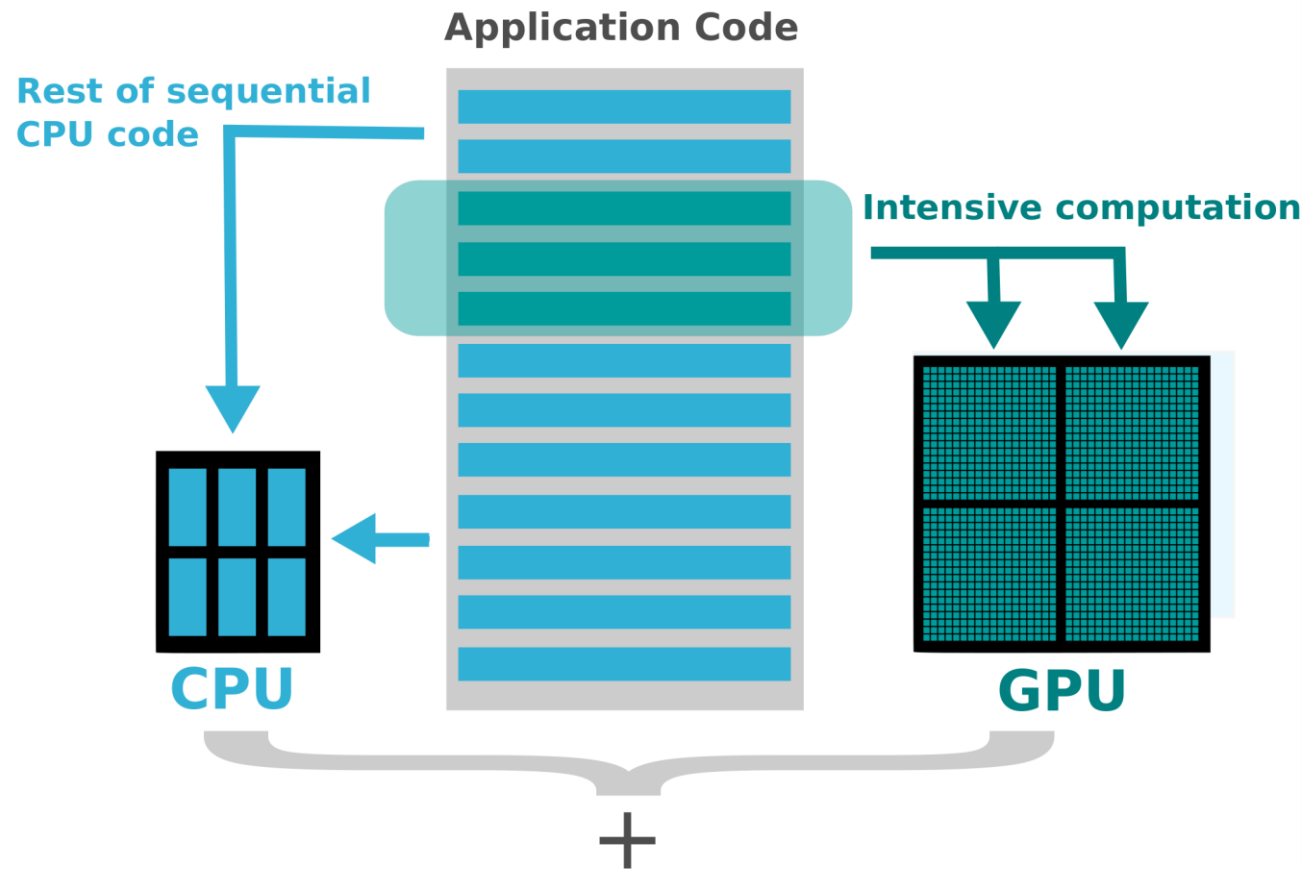
# LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

## ARQUITECTURA CPU/GPU INTEGRADA



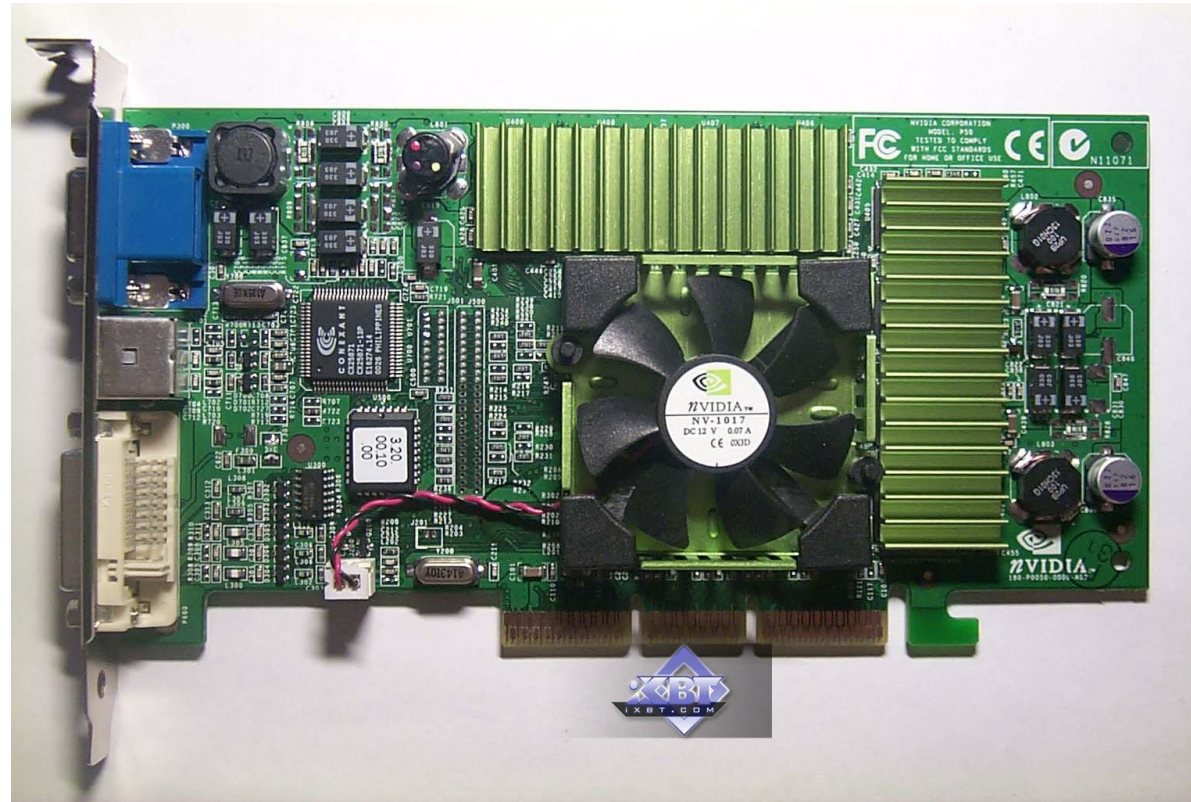
# LA UNIDAD DE PROCESAMIENTO GRÁFICO (GPU)

## COMPUTACIÓN HETEROGÉNEA



# PRIMEROS PASOS EN COMPUTACIÓN SOBRE GPU<sub>s</sub>

## GEFORCE 3 CON VERTEX Y PIXEL SHADERS PROGRAMABLES (2001)



# PRIMEROS PASOS EN COMPUTACIÓN SOBRE GPUS

## GEFORCE 3 TECH DEMO



# PRIMEROS PASOS EN COMPUTACIÓN SOBRE GPUS

## ENGAÑAR A LA GPU EMPLEANDO APIs GRÁFICAS (OPENGL O DIRECTX)

```
float saxpy (
    float2 coords : TEXCOORD0,
    uniform sampler2D textureY,
    uniform sampler2D textureX,
    uniform float alpha ) : COLOR
{
    float result;
    float yval=y_old[i];
    float y = tex2D(textureY,coords);
    float xval=x[i];
    float x = tex2D(textureX,coords);
    y_new[i]=yval+alpha*xval;
    result = y + alpha * x;
    return result;
}
```

$$Y = Y + \text{alpha} * X$$



# PRIMEROS PASOS EN COMPUTACIÓN SOBRE GPUS

## LIMITACIONES QUE IMPIDIERON EL PROGRESO

CURVA DE APRENDIZAJE DE OPENGL/DIRECTX Y ESFUERZO DE TRADUCCIÓN

NECESIDAD DE APRENDER LENGUAJES DE SHADING (CG, GLSL)

SOPORTE DE FLOAT O DOUBLE NO GARANTIZADO

LIMITACIONES EN LOS PATRONES DE ESCRITURA Y LECTURA DE MEMORIA

CARENCIA DE HERRAMIENTAS DE DEPURACIÓN Y CONTROL DE ERRORES

RECURSOS LIMITADOS: MEMORIA, VELOCIDAD, FLEXIBILIDAD...

# ARQUITECTURA CUDA

GEFORCE 8800 GTX (2007)



# ARQUITECTURA CUDA

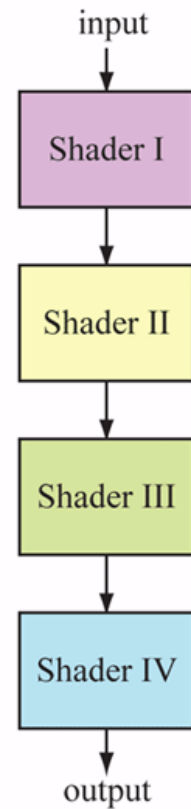
## GEFORCE 8800 GTX TECH DEMO



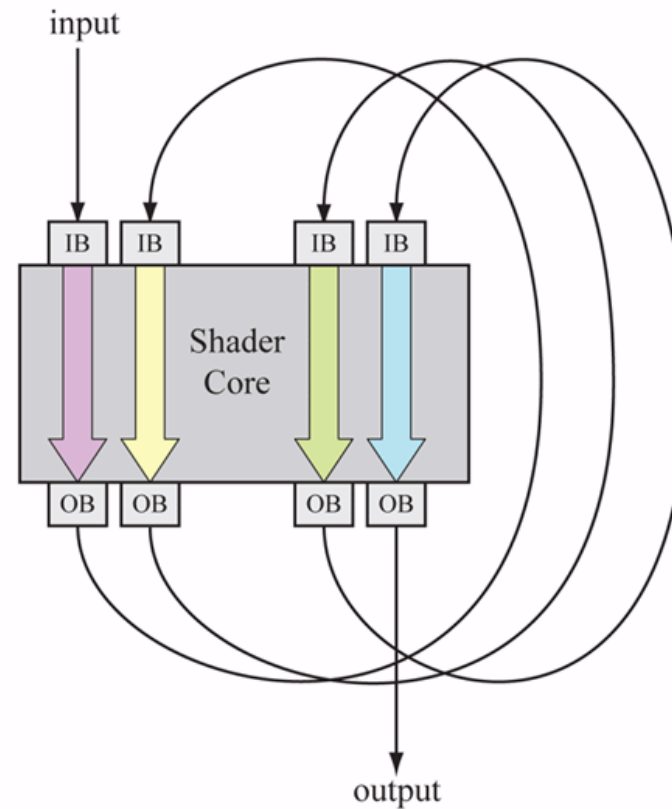
# ARQUITECTURA CUDA

## SHADERS UNIFICADOS

**Non-Unified Architecture**

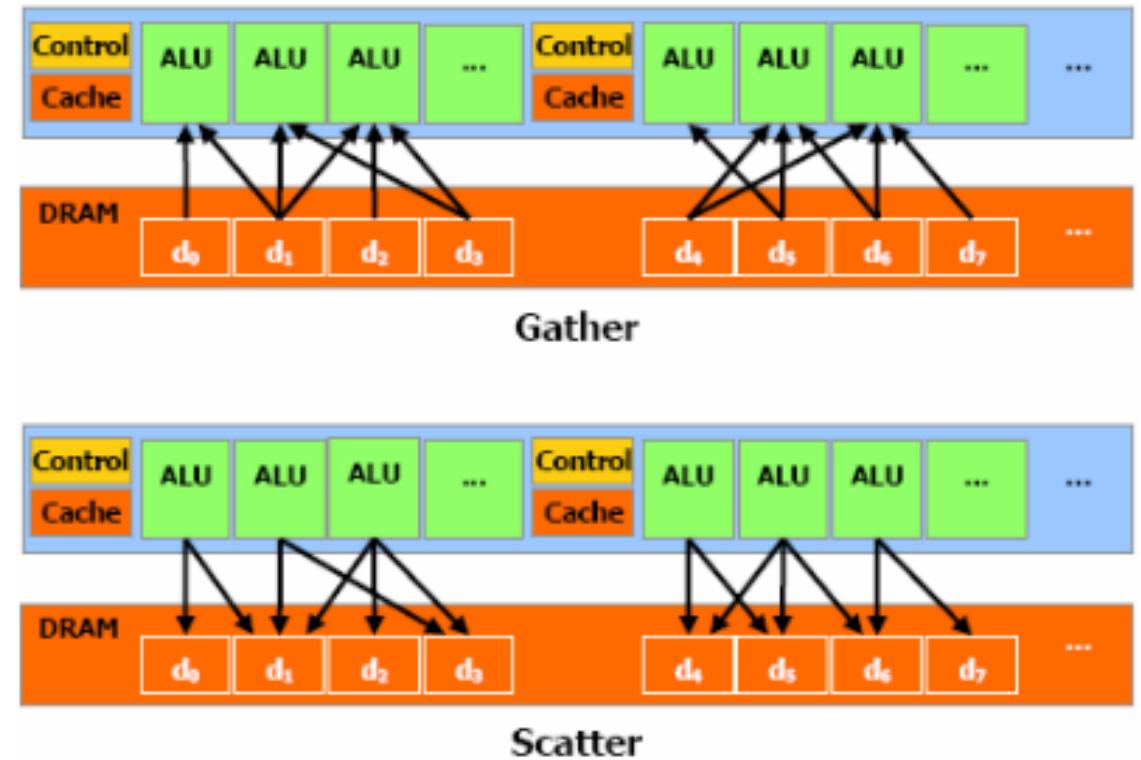
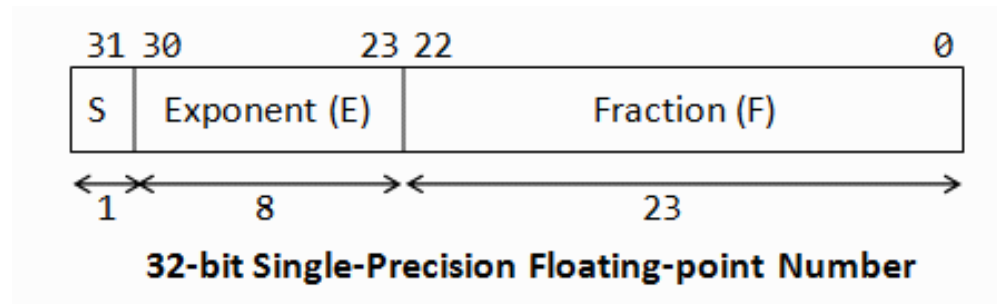


**Unified Shader Architecture**



# ARQUITECTURA CUDA

## PRECISIÓN FLOAT Y PATRONES DE ACCESO





# ARQUITECTURA CUDA

## ECOSISTEMA

ARQUITECTURA HARDWARE PROPIA

DRIVER ESPECIALIZADO PARA LA GPU

LENGUAJE DE PROGRAMACIÓN FLEXIBLE (BASADO EN C++ INICIALMENTE)

COMPILADOR Y ENTORNO DE DESARROLLO Y DEPURACIÓN

DOCUMENTACIÓN, TUTORIALES, DONACIONES

# GEFORCE® RTX

## GRAPHICS REINVENTED



# TURING

## REAL-TIME RAYTRACING



# TURING

## REAL-TIME SIMULATIONS





# APLICACIONES

¿QUÉ NECESIDAD HAY DE UN PROCESADOR GRÁFICO PARA CÓMPUTO GENERAL?

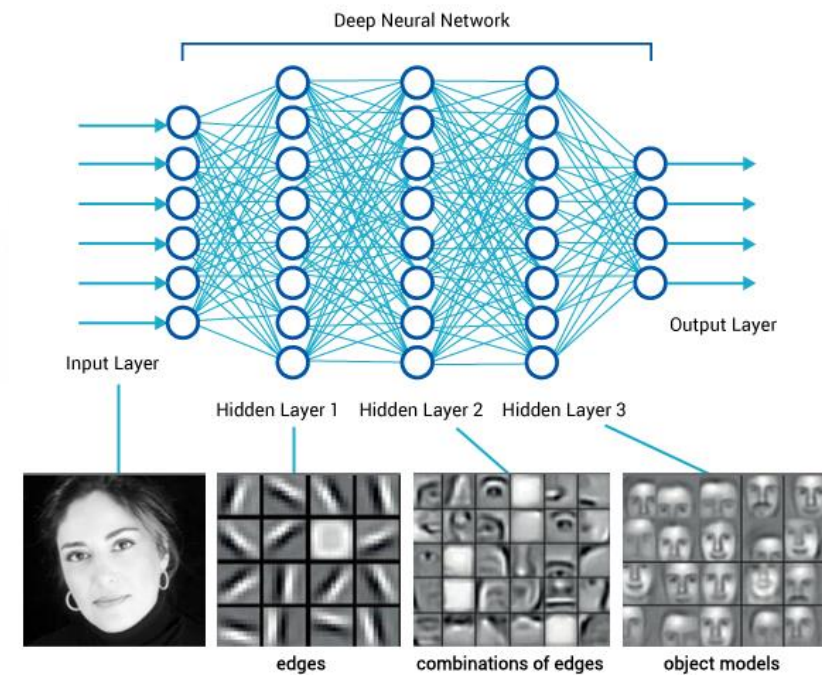
AR/VR



Coches  
Autónomos



Deep Learning



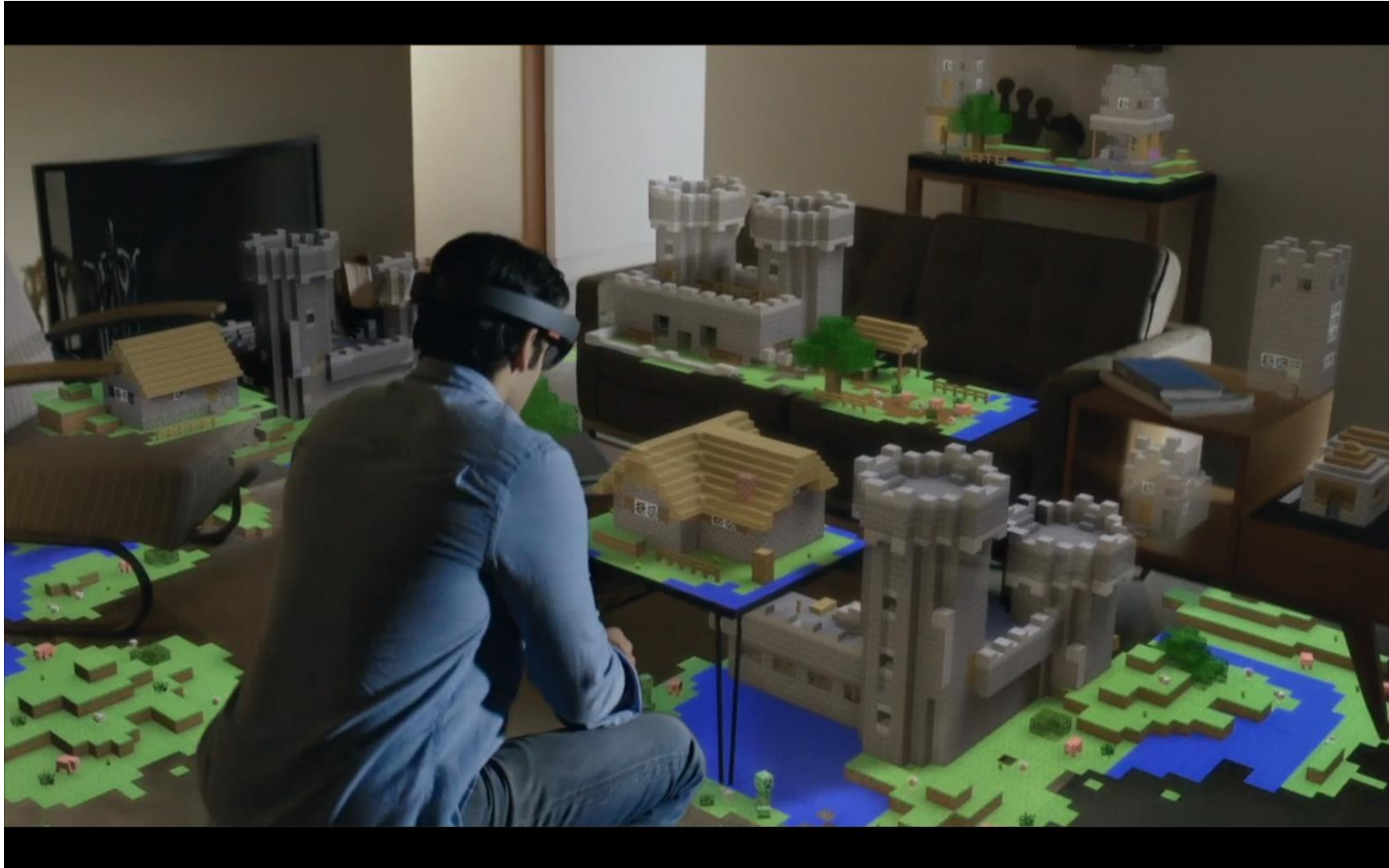
# APLICACIONES

## AR/VR



# APLICACIONES

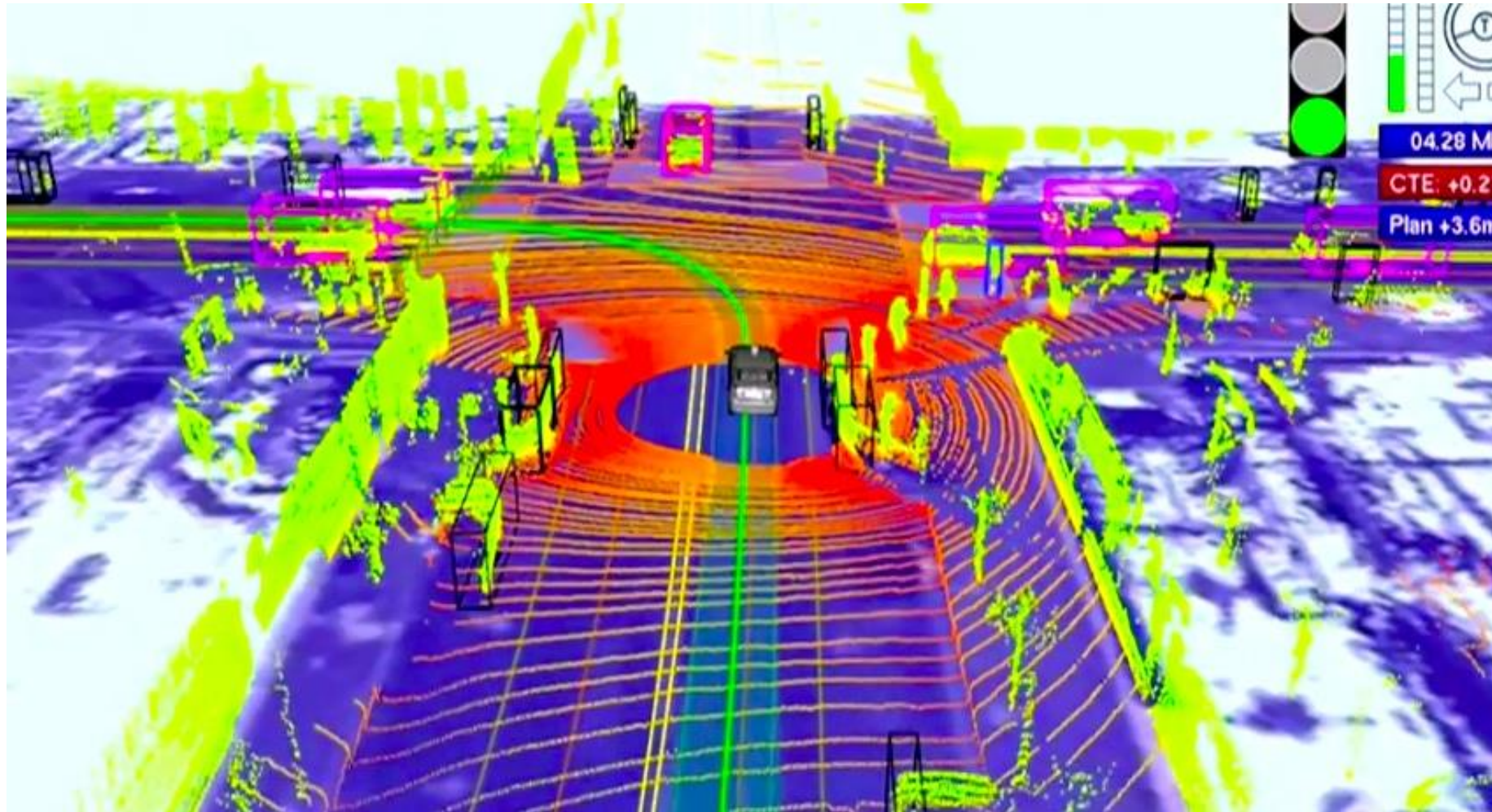
## AR/VR





# APLICACIONES

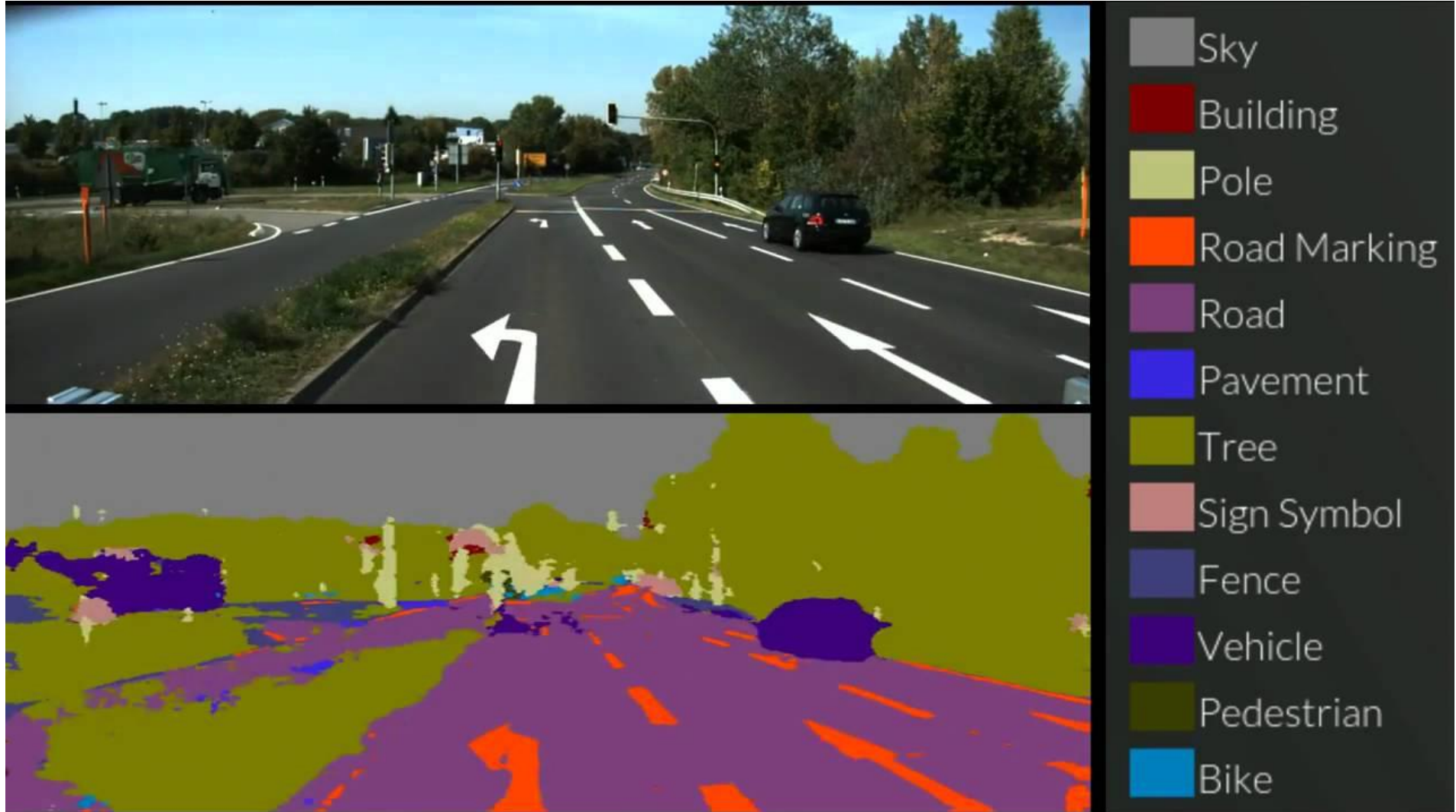
## COCHES AUTÓNOMOS





# APLICACIONES

## COCHES AUTÓNOMOS



# APLICACIONES

## DEEP LEARNING



# APLICACIONES

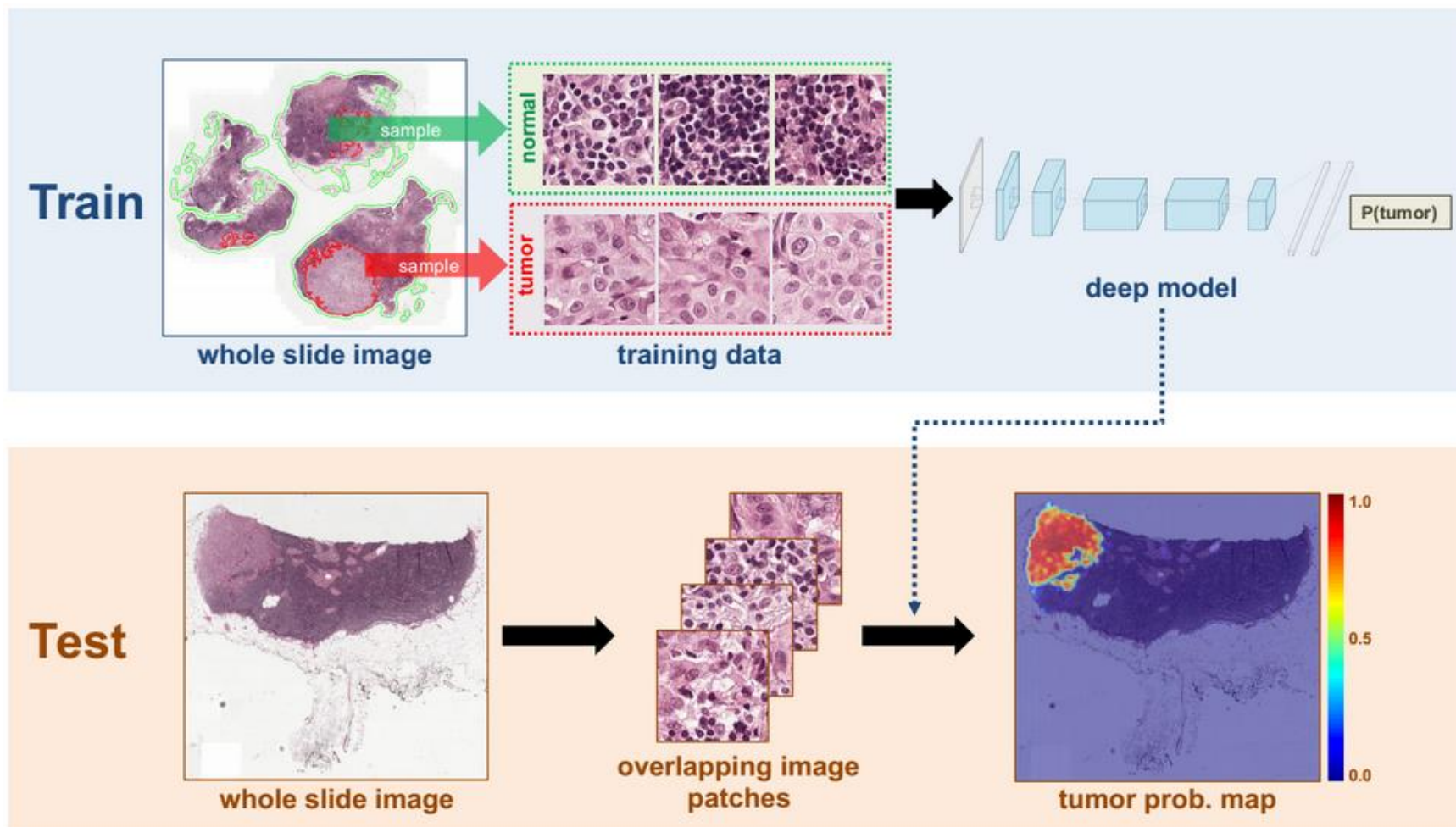
## DEEP LEARNING





# APLICACIONES

## DEEP LEARNING





# 10X GROWTH IN GPU COMPUTING

2008

150,000  
CUDA Downloads

27  
CUDA Apps

60  
Universities Teaching

4,000  
Academic Papers

6,000  
Tesla GPUs

77  
Supercomputing Teraflops



2015



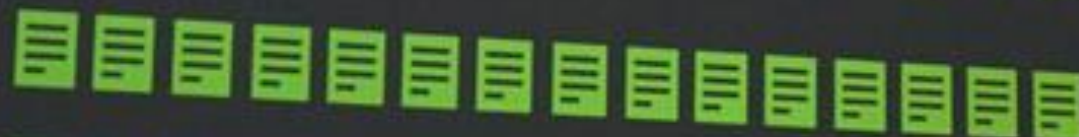
3 Million  
CUDA Downloads



319  
CUDA Apps



800  
Universities Teaching



60,000  
Academic Papers



# INICIOS Y EVOLUCIÓN DE LOS PROCESADORES GRÁFICOS (GPUs)

JGPU 2020

Albert García-García < [agarcia@dtic.ua.es](mailto:agarcia@dtic.ua.es) >