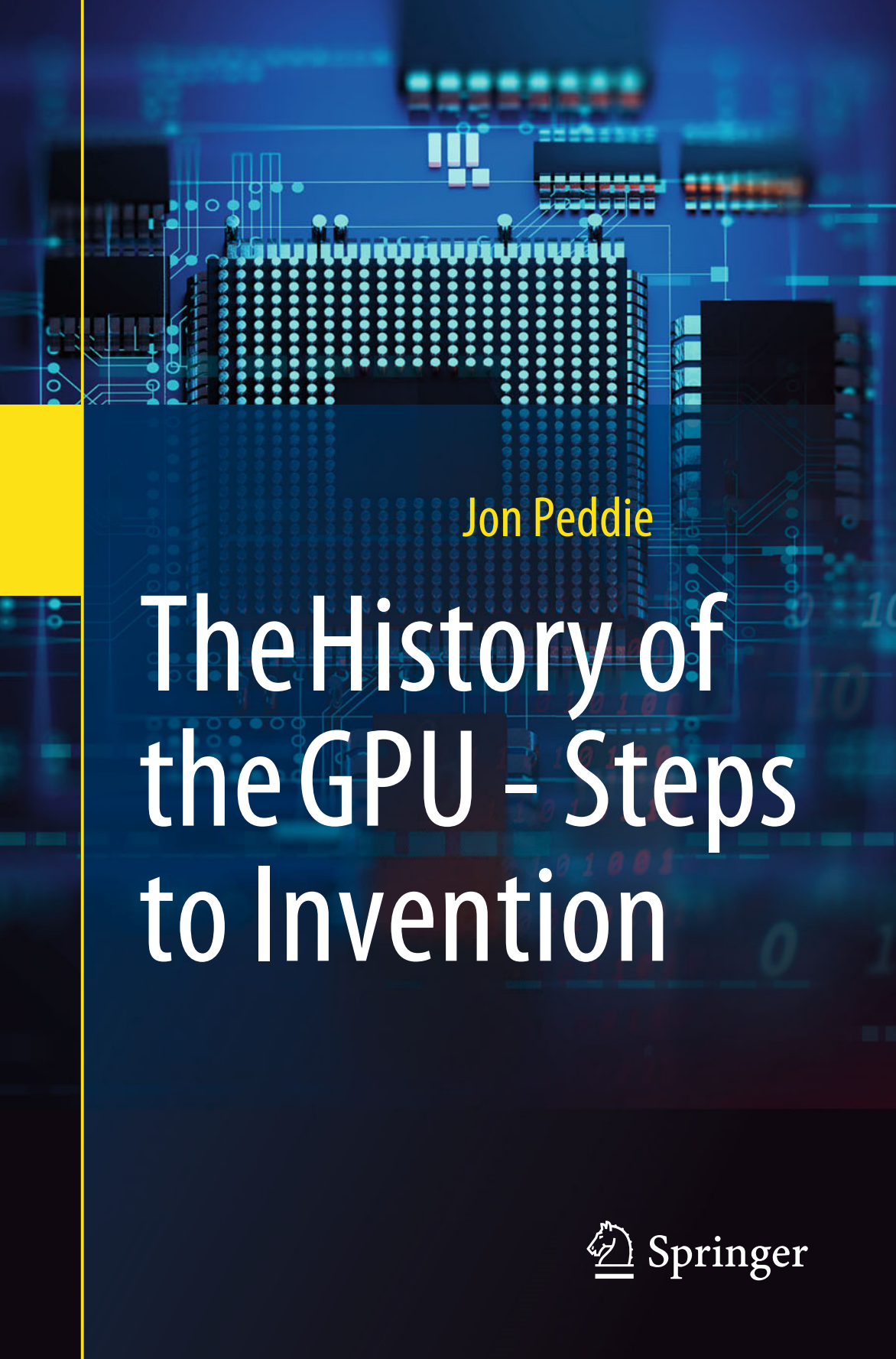
Jon Peddie
The History of the GPU - Steps to Invention
Springer

The History of the GPU - Steps to Invention

Jon Peddie

The History of the GPU - Steps to Invention

Springer

Jon Peddie
Jon Peddie Research
Tiburon, CA, USA

ISBN 978-3-031-10967-6 ISBN 978-3-031-10968-3 (eBook)
https://doi.org/10.1007/978-3-031-10968-3

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Foreword

History often elicits strong responses whether it is studied in school, the subject of documentary films and books, or passed orally from generation to generation. No matter the source, no history can cover every event for any one person. My own memory demonstrates that daily.

I believe that history is an essential subject. Understanding what happened in the past gives insight into what worked and (perhaps more importantly) what didn't work and why. In addition, history provides context for current events. We learn from history in important ways.

Computing itself is a relatively new field. Many science and engineering fields are significantly older and their history has been documented extensively. There are substantive debates about what counts as the first digital computer. Suffice it to say that digital computers are not much more than 100 years old.

Computer graphics is an even newer field. It integrates disparate display technologies, digital and analog computers, and a human's innate capability to see pictures on a flat screen. Verne Hudson from Boeing-Wichita coined the term circa 1960. His collaborator, Bill Fetter, popularized it.

Jon's book complements a spate of recent publications devoted to the history of different aspects of computer graphics. Books by Peddie, Masson, and Carlson describe the field in general. Smith traces the evolution of the pixel. Llach looks at graphics in building and architecture, Weisberg the history of CAD, and Gaboury the influence of the University of Utah. This is just a sampling.

What I find interesting about the authors is that many are intimately involved with the field rather than historians per se. A number of them are pioneers or students of pioneers who have first-hand knowledge of the history they are documenting. These authors write with both authority and immediacy.

This book provides a broad view of the graphics processing unit. Jon has been involved with special purpose graphics processing technology since day one. He does an excellent job documenting processors dedicated to generating better images faster. Like any history, it's not complete. The book does provide a coherent, well-organized view of the evolution of a valuable technology. Jon emphasizes how GPUs evolved from custom processors devoted to picture generation to general-purpose parallel

processors. It provides context that helps the reader better understand how GPUs fit into the computer graphics world.

I was totally unaware of Hudson and Fetter and the existence of computers and computer graphics until the late 1960s. I didn't enter high school until 1962. My curriculum included Latin, Greek, and little science. Therefore, I could barely spell "computer." Ironically, I retired from Boeing as a Senior Technical Fellow in visualization and interactive techniques after a 35-year career.

The computer graphics bug bit me as a Johns Hopkins undergrad in 1969. Bill Huggins, who had spent his sabbatical learning computer animation with Bell Labs pioneers, recruited me to make computer-animated educational films. The process was arduous. It involved punched cards, line printer keyframes, a microfilm recorder (located in Brooklyn NY), an assembly language animation "language," and an IBM 7094 mainframe. There were no interactive devices for animators/programmers, no color output, no shaded images, and no sound. Just white lines on a black background. And I loved it!

My early career let me create more animated films and learn about interactive graphics at Battelle-Columbus Labs. I became aware that a digital computer can display one frame at a time whether the frame is part of a projected film or displayed on a graphics screen. The human visual system does the rest and gives a person the illusion of continuous motion as long as each image is shown quickly enough.

For the film, a projector shows frames fast enough (24–30 Hz) to make the motion seem continuous. Images on interactive device screens must be redrawn at the same rate or faster. Current interactive devices established a redraw rate at 60+ Hz. The requirement to draw new frames interactively ultimately led to the work with GPUs. A film may take compute-centuries to produce enough frames for a full-length animated film. Projectors are responsible for showing the frames fast enough.

GPUs help reduce compute-centuries for a film to something more reasonable by improving overall throughput. Interactivity pushes compute performance even harder. In today's interactive graphics world, GPUs must compute a completely new frame fast enough to create the illusion of continuous motion. Put another way, the image generation compute task, the task GPUs perform, must determine the color of each pixel on each frame fast enough to convince the human visual system that image transformations (either 2D or 3D) are continuous.

My work at Boeing emphasized acceptable interactive performance. I was able to work at a Boeing scale (interactively working with the complete digital design of a commercial airplane like a 787, ~2 billion polygons) on a GPU-equipped PC to make end-users think the task was easy. I often measure success by making the difficulty of complicated behind-the-scenes tasks seem simple when in actual use.

I think Jon's discussion about GPU evolution to become a generalized parallel processor adds real value. It confirms my belief that the most successful and powerful technologies are those that can be generalized and applied to problems the original developers never foresaw. GPUs fit that profile.

Pay careful attention to the lessons learned from GPU evolution and generalization. Those lessons can be applied to the reader's own work. And understand how forthcoming generations of GPUs can be extended to provide even more value in the future.

Sammamish, WA, USA D. J. Kasik
June 2022

Preface

This is the first book in the three-book series on the History of the GPU.

History books are challenging to write. Technical history books are incredibly challenging. Why? Because things don't happen in an orderly sequence. Although one might think that event A leads to event B, often A leads to D, and B leads to C, but C leads to G.

Because the integrated graphics processing unit (GPU) has been employed in so many systems (platforms) and evolved since 1996, how do you tell a 2D story in a linear presentation such as the book?

One possibility is to list everything chronologically. Another approach is to list things by platform. And yet another choice is to list items by company, or by applications.

I have chosen a combination of all three.

This first book in the series covers the developments that lead up to the integrated GPU, from the early 1960s to the late 1990s

The book has two main sections, the PC platform and other platforms. Other platforms include workstations and game machines.

Each chapter is designed to be read independently, hence there may be some redundancy. Hopefully, each one tells an interesting story.

In general, a company is discussed and introduced in the year of its formation. However, a company may be discussed in multiple time periods in multiple chapters depending on how significant their developments were and what impact they had on the industry.

The History of the GPU - Steps to Invention

I mark the GPU's introduction as the first fully integrated single chip with hardware geometry processing capabilities—transform and lighting. Nvidia gets that honor on the PC by introducing their GeForce 256 based on the NV10 chip in October 1999. However, Silicon Graphics Inc. (SGI) introduced an integrated GPU in the Nintendo 64 in 1996, and ArtX developed an integrated GPU for the PC a month after Nvidia. As you will learn, Nvidia did not introduce the concept of a GPU, nor did they

develop the first hardware implementation of transform and lighting. But Nvidia was the first to bring all that together in a mass-produced single-chip device.

The evolution of the GPU did not stop with the inclusion of the transformation and lighting (T&L) engine because the first era of such GPUs had fixed-function T&L processors—that was all they could do and when they were not doing that they sat idle using power. The GPU kept evolving and has gone through six eras of evolution ending up today as a universal computing machine capable of almost anything.

However, to fully appreciate and hopefully understand what wonderful development the GPU has been, it is necessary to know where, why, and how it was developed. To do that I start the story with the early computers from the late 1950s and 1960s.

Now GPUs are ubiquitous.

What Is In and Not In These Books

As a public speaker and former engineer, you can tell from the above diagram; I like block diagrams. I have attempted to illustrate all the innovative GPUs and some of their predecessors with block diagrams. In some cases, I could not find sufficient data to construct a diagram; in some cases, the best I could do was a system-level diagram where the GPU is just a block.

In these books, you won't find any formulas (no math), code examples, operating application examples, user interface illustrations, and hopefully no commercials or propaganda.

Notable quotes and long quotations are presented indented to identify them as important and separate from the text.

At the end is the glossary. Not every term used in the book is in the glossary as many of the explanations are in the body text.

There is also a list of acronyms. The tech industry loves acronyms, and they can save time in communicating; they can also be very confusing. The acronym lists the acronym and a brief description.

Significant Things

One of my goals for these books was to identify those developments that I (and hopefully others) thought were inflection points and disruptive results—things that moved the industry and or changed its direction. I marked those milestones in bold italics.

The introduction of the GPU was just such a thing. It has profoundly and forever changed how computers work and are used.

I hope you find this and the following books interesting and informative. I have personally lived through almost all of it and have known most of the people

mentioned. Many are acquaintances, and many are friends. Many of the people mentioned have generously contributed to this book with fact-checking, storytelling, and encouragement. However, it's necessary to say that any mistakes or inaccuracies are all my own.

The Author

A Lifetime of Chasing Pixels

I have been working in computer graphics since the early 1960s, first as an engineer, then as an entrepreneur (I found four companies and ran three others), ending up in a failed attempt at retiring in 1982 as an industry consultant and advisor. Over the years, I watched, advised, counseled, and reported on developing companies and their technology. I saw the number of companies designing or building graphics controllers swell from a few to over forty-five. In addition, there have been over thirty companies designing or making graphics controllers for mobile devices.

I've written and contributed to several other books on computer graphics (seven under my name and six co-authored). I've lectured at several universities around the world, written uncountable articles, and acquired a few patents, all with a single, passionate thread—computer graphics and the creation of beautiful pictures that tell a story. This book is liberally sprinkled with images—block diagrams of the chips, photos of the chips, the boards they were put on, and the systems they were put in—and pictures of some of the people who invented and created these marvelous devices that impact and enhance our daily lives—many of them I am proud to say are good friends of mine.

I laid out the book in such a way (I hope) that you can open it up to any page and start to get the story. You can read it linearly; if you do, you'll probably find a new information and probably more than you ever wanted to know. My email address is in various parts of this book, and I try to answer every one, hopefully within 48 h. I'd love to hear comments, your stories, and your suggestions.

The following is an alphabetical list of all the people (at least I hope it's all of them) who helped me with this project. A couple of them have passed away, sorry to say. Hopefully, this book will help keep the memory of them and their contributions alive.

Thanks for reading

Jon Peddie—*Chasing pixels, and finding gems*

Acknowledgments and Contributors

The following people helped me with editing, interviews, data, photos, and most of all encouragement. I literally and figuratively could not have done this without them.

Anand Patel—Arm

Andrew Wolfe—S3

Ashraf Eassa—Nvidia

Atif Zafar—Pixilica

Borger Ljosland—Falanx

Brian Kelleher—DEC, and finally Nvidia

Bryan Del Rizzo—3dfx & Nvidia

Carrell Killebrew—TI/ATI/AMD

Chris Malachowsky—Nvidia

Curtis Priem—Nvidia

Dado Banatao—S3

Dan Vivoli—Nvidia

Dan Wood—Matrox, Intel

Daniel Taranovsky—ATI

Dave Erskine—ATI & AMD

Dave Kasik—Boeing

Dave Orton—SGI, ArtX, ATI & AMD

David Harold—Imagination Technologies

Edvaed Sergard—Falanx

Emily Drake—Siggraph

Eric Demers—AMD/Qualcomm

Frank Paniagua—Video Logic

Gary Tarolli—3dfx

George Sidiropoulos—Think Silicon

Gerry Stanley—Real3D

Henry C. Lin—Nvidia

Henry Chow—Yamaha & Giga Pixel

Henry Fuchs—UNC

Henry Quan—ATI

Hossain Yassaie—Imagination Technologies

Iakovos Istamoulis—Think Silicon

Ian Hutchinson—Arm

Jay Eisenlohr—Rendition

Jay Torberg—Microsoft

Jeff Bush—Nyuzi

Jeff Fischer—Weitek & Nvidia

Jem Davis—Arm

Jensen Huang—Nvidia

Jim Pappas—Intel

Joe Curley—Tseng/Intel

John Poulton—UNC & Nvidia

Jonah Alben—Nvidia

Karl Guttag—TI

Karthikeyan (Karu) Sankaralingam—University of Wisconsin-Madison

Kathleen Maher—JPA & JPR

Ken Potashner—S3 & SonicBlue

Kristen Ray—Arm

Lee Hirsch—Nvidia

Luke Kenneth Casson Leighton—Libre-GPU

Mark Kilgard—Nvidia (Iris GL)

Mary Whitton—Iknoas

Megan Zea—PCI SIG

Melissa Scuse—Arm

Mike Diehl—HP

Mike Mantor—AND

Mikko Alho—Siru

Mikko Nurmi—Bitboys

Neal Leavitt—Editing

Neil Trevett—3Dlabs & Khronos

Nick England—Iknoas

Pedro Duarte—Universities of Coimbra

Peter McGuinness—SGS Thompson

Peter L. Segal—AT & T

Petri Norlund—Bitboys

Phil Roges—ATI

Richard Huddy—ATI

Richard Selvaggi—Tseng Labs

Rick Bergman—ATI/AMD

Robert Dow—JPR

Ross Smith—3dfx

Ruchika Saini—editing

Sasa Marinkovic—ATI & AMD

Simon Fenny—Video Logic & Imagination Technologies

Stefan Demetrescu—Stanford

Stephen Morein—Stellar

Steve Brightfield—SiliconArts

Steve Edelson—Edson Labs

Tatsuo Yamamoto—Sega/DMP

Tim Leland—Qualcomm

Timothy Miller—Traversal Technology

Tom Forsyth—3Dlabs

Tony Tamasi—3dfx & Nvidia

Trevor Wing—Video Logic

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Introduction

Over the years, the computer's central processing unit (CPU) eventually incorporated every coprocessor developed to augment and add to its function except for one major processor, the GPU. A GPU is a specialized processor developed initially to accelerate graphics rendering and geometry transformations.

The CPU has even incorporated graphics processing. However, the CPU has not terminated the GPU's stand-alone value as it did with floating-point processors, digital signal processors (DSPs), video Codecs, and other accelerators. The GPU survives as a stand-alone coprocessor because the GPU scales almost infinitely—adding transistors to create thousands of processor cores. The only asymptote a GPU might face is inter-processor communications. Clustering groups of processors (shaders) overcomes that barrier. Coherent caches have also scaled well and address the GPU's inter-processor communications bottleneck. GPUs have been a significant beneficiary of Moore's law [3] (which postulates that the number of transistors on a chip doubles and cuts the price in half approximately every 1.5 to 2 years).

The GPU is a wonderful device and has made tremendous contributions to the computer's capabilities.

GPUs can process data simultaneously, which is known as parallel processing. As a result, GPUs are used in applications beyond gaming and simulation. Applications as far-ranging as artificial intelligence (AI), machine learning (ML), CAD, and compute-intensive tasks use GPUs. GPUs have been used as accelerators for photo and video editing and high-performance computers (HPC) and supercomputers.

GPUs were initially stand-alone discrete hardware units (dGPU). Later, in 2010, they were added to the CPU (iGPU) but still held their place as a stand-alone device. GPUs can contain specialized AI elements and multimedia accelerators for video and audio, and ray tracing accelerators. As the GPU advanced beyond its original role as a graphics processor and found use in pure computing applications not requiring a display, its additional capabilities have been called General-Purpose GPU (GPGPU), or GPU compute (cGPU). It is a false term because there is nothing general-purpose

J. Peddie, *The History of the GPU - Steps to Invention*,
https://doi.org/10.1007/978-3-031-10968-3_1

about a GPU. It cannot run an operating system, manage disk drives and peripherals, or boot up a system, nor is there anything general-purpose about a parallel processor. Those are the jobs of the CPU. GPUs are specialized devices with specific and specialized parallel processing capabilities.

The terms *semiconductor, integrated circuit (IC),* and *chip* will be used interchangeably in this book and should be considered synonyms.

What does a GPU do? It is almost everything needed to calculate in parallel from geometry processing to image processing to AI training and accelerated computing.

Computer graphics is about geometry; as Pixar cofounder Alvy Ray Smith says in his book, *A Biography of the Pixel*, "Computer graphics is geometry in, pixels out [4]." There's much more to the GPU, however. Image processing is pixels in, pixels out, whereas AI training and compute-acceleration is data in, data out—and a GPU does all of that and more.

And David Kasik of Boeing says:

> I consider computer graphics to be about creating, displaying, and modifying visual content to communicate to others. The sources are much broader than geometry: cameras, simulations, brain waves, sounds, etc. All forms of data can have a visual manifestation.

The integration of the rendering engine and the transformation and lighting (T&L) engine into a graphics controller converting it into a GPU was done to solve a geometry problem. Therefore, you will find a lot of discussion about geometry but no math in this book.

Transformation and lighting are critical components of computer graphics and the GPU. Transform means to convert the coordinates of the 3D model to the coordinates of the viewing or display device. The coordinates are described by the vertices of objects in a scene. Lighting refers to the simulation of light in a scene—on the objects in a scene—and the effect of light from one object to another and the scene. It is a complicated process.

As will be explained in later chapters, the GPU will find its way into general compute applications as a parallel processor, totally devoid of any graphics functions.

Graphics processing units were developed for three-dimensional (3D) computer graphics applications, first for CAD and then for games. The historical development of computer graphics gives an overview of the influences that heralded the development and invention of the GPU. The following is an overview of those developments, which will provide a foundation for appreciating the GPU's development and evolution.

Displays and Pixels

In computer graphics and digital imaging, a pixel is the smallest addressable element in a raster image, or the smallest addressable element in a digital display such as an LCD. A pixel is not addressable in geometry. Geometry is at the front end of a processing pipeline and the output is a raster scan, sized to a specific physical display, and measured in pixels.

A GPU's primary function is to drive a display (although GPUs used for compute acceleration do not drive a display). GPUs and graphics controllers before them