



Jon Peddie

# The History of the GPU – Eras and Environment

 Springer

# The History of the GPU - Eras and Environment

Jon Peddie

# The History of the GPU - Eras and Environment

Jon Peddie  
Jon Peddie Research  
Tiburon, CA, USA

ISBN 978-3-031-13580-4      ISBN 978-3-031-13581-1 (eBook)  
<https://doi.org/10.1007/978-3-031-13581-1>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

# Foreword

Computer graphics has attracted the world's leading computer scientists and the most prominent computer companies. Computer graphics is attractive for its grand computer science challenge, visceral beauty, and broad market impact by sitting at the intersection of computing and art, the simulation of light, physics, and virtual worlds.

The industry was chaotic and fast-changing, with countless companies innovating. In less than three decades, 3D graphics became standard in computers, evolved from a fixed function accelerator into a programmable shading GPU, became a technology juggernaut, and real-time ray tracing became a reality. The GPU became general purpose, and GPU computing was born, democratizing scientific computing, and enabling deep learning and the modern AI era. No one would have believed a chip design that started out running Quake would evolve to become the engine for conversational AI, self-driving cars, climate simulation, and countless applications that impact most modern life.

As the longest running historian of the graphics industry, Jon masterfully lays out the lineage and broad “family tree” of the GPU and the markets and industries it has come to serve. Its adoption, the rapid rate of its improvements, and breadth and depth of its application make it worthy of having its “origins story” explored in the detail and level of completeness for which Jon's books are known.

As you read Jon's book, you will benefit from his insight and ability to “paint the picture” of how we got here, what it means, and where we might go. The reader will see the awe-inspiring amount of innovation, brilliance, determination, guts, and effort from casts of thousands over decades that have made the GPU worthy of such a treatise. Enjoy!!

Nvidia, California, US

Jenson Huang  
Curtis Priem

# Preface

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

The integrated Graphics Processing Unit (GPU) has been employed in many systems (platforms) and evolved since 1996.

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

The book has two main sections, the PC platform, and other platforms. Other platforms include workstations, game machines, and others, which include vehicles—GPUs are used everywhere in almost everything.

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.

History of the GPU		
Steps to Invention Book 1	Erass and Environment Book 2	New Developments Book 3
1. Preface	1. Preface	1. Preface
2. History of the GPU	2. Race to build the first GPU	2. Second Era of GPUs (2001-2006)
3. 1980-1990 Graphics Controllers on Other Platforms	3. GPU Functions	3. Third to Fifth Era of GPUs
4. 1980-1989 Graphics Controllers on PCs	4. Major Era of GPUs	4. Mobile GPUs
5. 1990-1995 Graphics Controllers on PCs	5. First Era of GPUs	5. Game Console GPUs
6. 1990-1999 Graphics Controllers on Other Platforms	6. GPU Environment-Hardware	6. Compute GPUs
7. 1996-1999 Graphics Controller on PCs	7. Application Program Interface (API)	7. Open GPUs
8. What is a GPU	8. GPU Environment-Software Extensions	8. Sixth Era of GPUs

The History of the GPU - Eras and Environment

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, however, 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.

## 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 founded 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 45. In addition, there have been over 30 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 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 everyone, 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.

Ashraf Eassa—Nvidia  
Andrew Wolfe—S3  
Anand Patel—Arm  
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 Orton—SGI, ArtX, ATI & AMD  
David Harold—Imagination Technologies  
Dave Kasik—Boeing  
Emily Drake—Siggraph  
Edvaed Sergard—Falanx  
Eric Demers—AMD/Qualcomm  
Frank Paniagua—Video Logic  
Gary Tarolli—3dfx  
Gerry Stanley—Real3D  
George Sidiropoulos—Think Silicon  
Henry Chow—Yamaha & Giga Pixel  
Henry Fuchs—UNC  
Henry C. Lin—Nvidia  
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

Jonah Alben—Nvidia  
John Poulton—UNC & Nvidia  
Karl Gutttag—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 Nurmi—Bitboys  
Mikko Alho—Siru  
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  
Steve Brightfield—SiliconArts  
Steve Edelson—Edson Labs  
Stefan Demetrescu—Stanford  
Stephen Morein—Stellar  
Tatsuo Yamamoto—Sega/DMP  
Tim Leland—Qualcomm  
Timothy Miller—Traversal Technology  
Tom Forsyth—3Dlabs  
Tony Tamasi—3dfx & Nvidia  
Trevor Wing—Video Logic

# Contents

<b>1</b>	<b>Introduction</b>	1
1.1	Nvidia's NV10—First Integrated PC GPU (September 1999)	2
1.1.1	Nvidia's Non GPU GPU	9
1.1.2	The S3 Nvidia Patent Suit and Silicon Valley Gossip	10
1.1.2.1	Meanwhile, Back in Reality?	11
1.1.2.2	Nvidia and S3 Enter into a Cross-Licensing Agreement	12
1.2	Summary and Conclusion	14
1.3	Epilog—All the Others	15
	References	16
<b>2</b>	<b>The GPUs' Functions</b>	19
2.1	The Pipeline	20
2.1.1	The Meaning of Real Time	21
2.1.2	Binning and Branding	22
2.1.3	The Frame Buffer	23
2.1.3.1	GPUs and Memory	25
2.1.3.2	Resizable Base Address Register (2010)	26
2.1.4	Object-Level Clipping	27
2.1.4.1	Looking Back at the Geometry Processor's Origins	27
2.1.4.2	Digital Signal Processors Used for Geometry	30
2.2	The Rendering Equation	31
2.3	The Geometry Creation	32
2.4	The Software Story: The All-Important APIs	36
2.4.1	The Triangle Setup	37
2.4.2	Drawing and Shading	37
2.4.3	Triangle Sorting	37
2.4.4	Texture Mapping	38
2.4.4.1	3D Texture Filtering	41

2.5	Fill Rate, Rendering Pipelines, and Triangle Size .....	41
2.5.1	Rendering Techniques .....	43
2.5.1.1	Polygon Rendering .....	45
2.5.1.2	Scan-Line Rendering .....	45
2.5.1.3	Immediate Mode Rendering .....	45
2.5.1.4	Tile Rendering .....	46
2.5.1.4.1	Tile-Based Deferred Rendering .....	46
2.5.1.4.2	Immediate Mode Tile Rendering .....	48
2.5.1.5	Ray-Traced Rendering .....	49
2.5.2	Instancing .....	50
2.5.3	Aliasing .....	50
2.5.4	Scaling .....	53
2.5.5	Environment Mapping .....	55
2.6	Generating the Image: Hardware Issues .....	56
2.6.1	VPU—Visual Processing Unit .....	57
2.6.2	Multi-display .....	57
2.7	Crypto GPU .....	58
2.8	Audio, In, Out, In Again .....	58
2.9	Conclusions .....	59
	References .....	60
<b>3</b>	<b>The Major GPU Eras .....</b>	<b>63</b>
3.1	The First Era—Transform and Lighting—DirectX 7 (1999) .....	66
3.1.1	Shading and Shaders .....	66
3.1.1.1	Shading .....	68
3.1.2	Geometry Processing .....	70
3.2	The Second Era—Programmable Shaders—DirectX 8 and 9 (2001–2006) .....	72
3.2.1	Pixel (Fragment) Shader Stage .....	73
3.2.2	How Many Shaders? Is There a Limit? .....	74
3.3	The Third Era—The Unified Shader—DirectX 10 and 11 (2006–2009) .....	75
3.3.1	Geometry Shader (2006) .....	78
3.4	The Fourth Era—Compute Shaders—DirectX 11 (2009–2015) .....	79
3.4.1	Tessellation Shader (October 2009) .....	80
3.4.2	Summary .....	83
3.5	The Fifth Era—Ray Tracing and AI—DirectX 12 (2015–2020) ...	83
3.5.1	Ray Tracing Shaders .....	84
3.5.2	Real-Time Ray Tracing with AI .....	87
3.5.3	Variable Rate Shading—2019 .....	89
3.6	The Sixth Era—Mesh Shaders—DirectX 12 Ultimate (2020) .....	91
3.6.1	Primitive and Mesh Shaders—2017–2020 .....	92

3.6.2	Sampler Feedback .....	93
3.7	Summary on Shading .....	93
3.7.1	Mobile .....	93
3.7.1.1	GPU Sources for Mobile Devices .....	94
3.7.2	GPU-Compute .....	96
3.8	FLOPS Versus Fraps: Cars and GPUs .....	96
3.8.1	Why Good Enough is Not .....	98
3.9	Conclusion .....	100
	References .....	101
<b>4</b>	<b>The First Era of GPUs .....</b>	<b>105</b>
4.1	The Golden Age—Transform, and Lighting Changes the Industry (1999–2001) .....	107
4.1.1	On Being First .....	108
4.2	First Era Discrete PC-Based GPUs .....	108
4.2.1	Glaze3D Bitboys 2.0 (1999–2001) .....	109
4.2.1.1	Infineon .....	112
4.2.1.2	Bitboys Gets the Axe .....	114
4.2.1.3	Bitboys Gets Hammered .....	115
4.2.1.4	The Final Blow .....	116
4.2.2	S3 Savage 2000 (November 1999) .....	116
4.2.2.1	S3 Chrome .....	119
4.2.2.2	Epilogue: The Curious Trail to Zhaoxin .....	120
4.2.3	ATI and Nvidia: First Era GPUs (1999–2002) .....	121
4.2.4	ATI Radeon R100—256 (April 2000) .....	121
4.2.4.1	Pixel Tapestry Architecture .....	125
4.2.5	Nvidia’s NV15—GeForce 2 GTS (April 2000) .....	128
4.2.6	STMicroelectronics—Imagination Technologies Kyro II (2001–2002) .....	128
4.2.6.1	PowerVR3 STG4000 Kyro—2001 .....	132
4.2.6.2	PowerVR3 STG4500 Kyro II—2001 .....	132
4.2.6.3	The End .....	134
4.2.6.4	Summary .....	134
4.3	The Development and History of the Integrated GPU (1999–) .....	135
4.3.1	ArtX .....	136
4.3.1.1	ArtX: First Company to Announce a PC-Based iGPU .....	137
4.3.1.2	ATI Acquires ArtX (February 2000) .....	140
4.3.2	ATI’s First IGP (March 2000) .....	141
4.3.3	SiS’ First PC-Based IGP (December 2000) .....	142
4.3.4	Nvidia’s nForce 220 IGP (June 2001) .....	145
4.3.5	ATI’s IGP 320 (2002) .....	146
4.4	IGP Conclusion .....	146
4.5	The Expansion Years (2001–2016) .....	147

4.5.1	The Collapse and the Rise of Graphics Chip Companies .....	148
4.6	Conclusion .....	149
	References .....	149
<b>5</b>	<b>The GPU Environment—Hardware .....</b>	<b>151</b>
5.1	It Takes a Village to Build a GPU .....	151
5.2	Semiconductor Technology .....	152
5.2.1	Intel Introduces Angstroms .....	153
5.2.1.1	Fins to Sheets .....	155
5.2.1.2	GPU Memory .....	156
5.2.1.2.1	Shared Versus Private Memory .....	158
5.2.1.3	Memory Type and GPU .....	159
5.2.2	Chiplets .....	160
5.3	PC Bus Architectures .....	165
5.3.1	Industry Standard Architecture: 1981 .....	167
5.3.2	Micro Channel Architecture: 1987 .....	168
5.3.3	Extended ISA: 1988 .....	168
5.3.4	VESA Local Bus: 1992 .....	169
5.3.5	Peripheral Component Interconnect: 1992 .....	169
5.3.6	Accelerated Graphics Port: 1997 .....	172
5.3.7	Peripheral Component Interconnect Express: 2003 .....	172
5.3.8	Other I/O .....	176
5.4	GPU Video Outputs .....	177
5.4.1	VGA: 1987 .....	179
5.4.2	DVI (1999–) .....	179
5.4.3	HDMI (2002–) .....	179
5.4.4	High Dynamic Range (2015) .....	181
5.4.5	DisplayPort .....	182
5.4.6	Seeing More .....	183
5.4.7	Virtual Reality Headsets .....	183
5.4.8	Augmented Reality Glasses .....	184
5.4.9	Mixed Reality Headsets .....	185
5.4.10	Monitor Synchronization: 2013–2015 .....	186
5.4.10.1	Those Damn Scalers .....	192
5.4.10.2	Flickering .....	193
5.4.10.3	Adaptive Sync, FreeSync, and G-Sync .....	194
5.5	Multiple AIBs in a System .....	194
5.5.1	Multi-GPIs (1996) .....	196
5.6	Conclusion .....	199
	References .....	199

<b>6</b>	<b>Application Program Interface (API)</b>	<b>201</b>
6.1	Application Program Interface	202
6.1.1	APIs and OSs	203
6.1.1.1	Chaos in the Mobile Market	207
6.1.1.2	DirectX Shaders	207
6.1.1.3	Comparison of Vertex Shaders	209
6.1.2	History of DirectX	209
6.1.2.1	Hardware Feature Levels	214
6.1.2.2	Microsoft's Japanese Bigotry	214
6.1.3	The History of OpenGL	215
6.1.4	The Fahrenheit Project	215
6.1.5	Low-Level APIs	217
6.1.5.1	Mantle	218
6.1.5.2	Metal	219
6.1.5.3	Vulkan	220
6.1.5.3.1	Cross-Platform Support	220
6.1.6	WebGPU	222
6.1.7	DirectX 12	223
6.1.7.1	Ultimate	223
6.1.7.1.1	Ray Tracing	224
6.1.7.1.2	Variable Rate Shading	225
6.1.7.1.3	Getting to Mesh Shaders	226
6.1.7.1.4	HW T&L Engines, 1981 to 1996: IRIS GL to Direct X 7.0	226
6.1.7.1.5	Vertex and Pixel Shaders, 1997 to 2008: Direct3D 10	227
6.1.7.1.6	Tessellation	229
6.1.7.1.7	The Pipeline Expands	230
6.1.7.1.8	Unified Shader, 2006–2010: DirectX 9.0 and OpenGL 3.3	232
6.1.7.1.9	Getting to Compute: Task Shaders 2016–2017 (DirectX 12, Vulkan 2)	233
6.1.8	Microsoft DirectX Shader Model 4.0: Enhancements	234
6.1.8.1	Geometry Shaders	234
6.1.8.1.1	Mesh Shaders, 2018–2020: DirectX 12 Ultimate, Vulkan Extension	235
6.1.8.1.2	Meshlets	237
6.1.8.1.3	Benchmarking	242
6.1.8.1.4	Interactive Mode	242
6.1.8.1.5	Mesh Demo	243

6.2	Conclusion	247
	References	248
<b>7</b>	<b>The GPU Environment—Software Extensions and Custom Features</b>	<b>251</b>
7.1	Software Libraries and Tools	251
7.1.1	Ambient Occlusion	252
7.1.2	Nvidia’s DLSS (February 2019)	253
7.1.3	AMD’s Fidelity FX Super Resolution (May 2021)	257
7.1.4	Intel’s XeSS (March 2022)	259
7.2	More Than a Driver	261
7.2.1	SYCL	261
7.2.2	GLSL	262
7.2.3	HLSL	262
7.2.4	SPIR-V	263
7.2.5	Textures	263
7.3	Summary	266
7.4	Software Development Kits for Developers	266
7.4.1	Nvidia’s GameWorks (2014–)	267
7.4.2	AMD’s FX Library (2014)	270
7.4.3	AMD’s GPUOpen (2015–)	271
7.4.4	Application Enhancement Software	273
7.4.5	AMD’s Gaming Evolved	274
7.4.6	Nvidia’s GeForce Experience	276
7.5	Conclusion	279
7.6	Summary	279
	References	280
	<b>Appendix A: Acronyms</b>	<b>283</b>
	<b>Appendix B: Definitions</b>	<b>287</b>
	<b>Index</b>	<b>327</b>



# List of Figures

Fig. 1.1	Die shot of Nvidia's first GPU, the NV10. Reproduced with permission from Curtis Priem	3
Fig. 1.2	Nvidia's GeForce 256 (NV10) block diagram with OpenGL	3
Fig. 1.3	VisionTek GeForce 256 DDR. <i>Source</i> Hyins for Wikipedia	4
Fig. 1.4	Dan Vivoli named the GPU	5
Fig. 1.5	Pat Gelsinger launched Intel's IDF in 1999. <i>Source</i> Intel	6
Fig. 1.6	The Utah teapot, a model by Martin Newell (1975) and used ever since. Reproduced with permission from Dhatfield for Wikipedia	7
Fig. 1.7	An example of a texture mapped to the faces of a cube box, also called a <i>skybox</i> . Reproduced with permission from Ariegee for Wikipedia	8
Fig. 1.8	Jen-Hsun (Jensen) Huang introduced the GPU in 1999. <i>Source</i> Nvidia	9
Fig. 1.9	The founders of Nvidia: Curtis Prien, Jensen Huang, and Chris Malachowsky. Reproduced with permission from Curtis Prien	14
Fig. 1.10	The rise and fall of graphics chip suppliers	15
Fig. 2.1	The graphics pipeline has been functionally the same since the 1970s	20
Fig. 2.2	A digital image of Richard Shoup, rendered on the Super Paint system, April 1973. <i>Source</i> The Richard G. Shoup Estate	24
Fig. 2.3	A viewing frustum. <i>Source</i> Computer desktop encyclopedia 1988 intergraph computer	28
Fig. 2.4	The silicon graphics inc. geometry engine-circa 1980s. <i>Source</i> Wikipedia: <a href="http://www.Shieldforyoureyes">http://www.Shieldforyoureyes</a>	28
Fig. 2.5	Rendering equation light characteristics	31
Fig. 2.6	Wire-frame models of a cube, icosahedron, and approximate sphere. <i>Source</i> Wikipedia	33
Fig. 2.7	Three, 2D views and a perspective view. <i>Source</i> Wikipedia	33

Fig. 2.8	An eight-sided faceted circle approximation (left) and a perfect circle (right) . . . . .	34
Fig. 2.9	Compare Lora Croft from 1996 to 2014—a measure of how CG has improved due to hardware that enabled more powerful and advanced software. <i>Source</i> Wikipedia . . . . .	35
Fig. 2.10	Hitman codename 47 from the 2016 version of the game (left) compared to Hitman 3 2021. <i>Source</i> Wikipedia . . . . .	35
Fig. 2.11	Comparison of the graphics quality of Call of Duty 2003–2021. Reproduced with permission from activision . . . . .	36
Fig. 2.12	Mipmaps . . . . .	39
Fig. 2.13	Example of texture mapping. Taking an image (left) overlaying on a 3D model (center) and creating a realistic image (left) . . . . .	40
Fig. 2.14	Comparison of Trilinear versus Anisotropic filtering—notice the street tiles are in focus to the vanishing point in the right image. <i>Source</i> Cobblestones. JPG Wikipedia Thomas . . . . .	41
Fig. 2.15	Fully rendered image of a 2021 Audi E-Tron GT. Reproduced with permission from TurboSquid . . . . .	42
Fig. 2.16	Wire-frame model of an Audi E-Tron GT. Reproduced with permission from TurboSquid . . . . .	42
Fig. 2.17	A Red, Blue, Green (RGB)-shaded triangle. <i>Source</i> Tilmann R, Public domain, via Wikimedia commons . . . . .	43
Fig. 2.18	Texture mapping is a way to add realism to 3D models . . . . .	43
Fig. 2.19	A sphere without bump mapping (left). A bump map (middle) is applied to the sphere on the left. The sphere with the bump map is shown on the right. It appears to have a mottled surface resembling an orange and a dent to represent where the stem was attached. <i>Source</i> Brion VIBBER, McLoaf, Vierge Marie, CC BY-SA 3.0, via Wikimedia Commons . . . . .	44
Fig. 2.20	A tile-based rendering (TBR) graphics pipeline . . . . .	47
Fig. 2.21	The rays in ray tracing . . . . .	49
Fig. 2.22	An example of instancing to create a field of sunflowers. Reproduced with permission from <a href="http://www.Blender.org">http://www.Blender.org</a> . . . . .	50
Fig. 2.23	Changing the intensity or color of the pixels between the line and the background gives the effect of an even line; the eye (brain) is tricked into seeing a smooth line. <i>Source</i> Blender . . . . .	51
Fig. 2.24	Bilinear upscaling filters out most of the raster Jaggies. Reproduced with permission from AMD . . . . .	53
Fig. 2.25	AMD’s FidelityFX super-resolution sharpened and filtered the image. Reproduced with permission from AMD . . . . .	54
Fig. 2.26	An unfiltered or sharpened image at the monitor’s native resolution. Reproduced with permission from AMD . . . . .	54

Fig. 2.27	A scene of a park. <i>Source</i> Springer .....	55
Fig. 2.28	Reflection mapping is used to place objects into scenes. <i>Source</i> Springer [50] .....	55
Fig. 2.29	Images of GPU dies with multi-thousand shader processors. Reproduced with permission from AMD (left) and Nvidia .....	56
Fig. 2.30	The many names of a GPU .....	59
Fig. 3.1	Performance of graphics leading up to GPUs. Graphics hardware had been progressing at a rate faster than Moore's law. Data Courtesy of John Poulton, UNC Chapel Hill .....	64
Fig. 3.2	The six eras of GPU evolution and development .....	64
Fig. 3.3	DirectX basic pipeline .....	64
Fig. 3.4	A shaded triangle. <i>Source</i> TilmannR, Public domain, via Wikimedia Commons .....	68
Fig. 3.5	Shader stages relative to memory (dark blue boxes), the white boxes are not shaders .....	68
Fig. 3.6	The first single chip GPU with an integrated T&L engine, Nvidia's NV10 used in the GeForce 256, circ 1999. Reproduced with permission from Konstantin Lanzet, Wikipedia .....	71
Fig. 3.7	Raja Koduri. <i>Source</i> Intel .....	71
Fig. 3.8	Vertex shading in the DirectX 8 pipeline .....	73
Fig. 3.9	A scan-converted triangle quantized by pixel size .....	73
Fig. 3.10	Mike Mantor, Corporate Fellow and Chief GPU Architect at AMD. Reproduced with permission from Expreview.com ...	75
Fig. 3.11	ATI/AMD Xenos block diagram .....	76
Fig. 3.12	A typical unified architecture GPU, with 112 streaming processor cores, organized as 14 multithreaded streaming multiprocessors. Nvidia's Tesla GeForce 8800 [15] .....	77
Fig. 3.13	DirectX 11 rendering pipeline featuring tessellation (gray box) .....	78
Fig. 3.14	Tessellation in the chamber for the circumcision of the princes in the Imperial Topkapı Sarayı, Istanbul, fifteenth century .....	81
Fig. 3.15	Catmull-Clark subdivision of a cube. Reproduced with permission from Romainbehar, Wikipedia .....	81
Fig. 3.16	The first GPU with integrated hardware tessellation was the ATI Radeon 8500. Reproduced with permission from Palit, ixbt.com .....	82
Fig. 3.17	ATI's tessellation pipeline .....	82
Fig. 3.18	Appel projected light at a 3D computer model and displayed the results on a plotter using a form of tone-mapping to create light and dark areas. <i>Source</i> Arthur Appel .....	84
Fig. 3.19	The ray tracing paradigm .....	85

Fig. 3.20	Bounding boxes enclosing a patch from the teapot model are tested for an intersection with a ray	86
Fig. 3.21	Shader evaluation pipeline	86
Fig. 3.22	Ray-traced image of a Mercedes-Benz SS Roadster. Reproduced with permission from Volkan Kaçar	87
Fig. 3.23	DLSS off and on.n <i>Source</i> Nvidia	88
Fig. 3.24	Reduced resolution rendering followed by upscaling gives an enhanced image at high frame rates. Reproduced with permission from Nvidia	90
Fig. 3.25	Variable rate shading architecture. Reproduced with permission from Nvidia	91
Fig. 3.26	GPU taxonomy.n Reproduced with permission from JPR	94
Fig. 3.27	Mobile gaming device	95
Fig. 3.28	Mobile GPU design sources	95
Fig. 3.29	Comparison of GFLOPS of GPUs over time	97
Fig. 3.30	Simplified AIB block diagram	98
Fig. 3.31	Simplified integrated GPU block diagram	99
Fig. 3.32	AMD's view of the eras of the GPU. Reproduced with permission by AMD	100
Fig. 3.33	GPU architectural introductions through the eras	101
Fig. 4.1	The rise and fall of PC graphics chip suppliers versus market growth	106
Fig. 4.2	Block diagram of a PC	106
Fig. 4.3	The Bitboys: Petri Nordland, Mika Tuomi, and Kai Tuomi (1999)	112
Fig. 4.4	Bitboys' Glaze3D block diagram with triangle setup engine	113
Fig. 4.5	Bitboys' Glaze3D pixel pipeline	114
Fig. 4.6	Bitboys' Glaze3D prototype circa 1999. <i>Source</i> Petri Nordlund	115
Fig. 4.7	S3 Savage 2000 AIB with T&L. <i>Source</i> Swaaye-Wikipedia	117
Fig. 4.8	Die photo of ATI's R100 chip. <i>Source</i> Fritzchens Fritz Wikipedia	123
Fig. 4.9	Example of vertex skinning. <i>Source</i> ATI	123
Fig. 4.10	ATI R100 Radeon 7000 AIB. The R100 chip is under the fan. <i>Source</i> ATI	124
Fig. 4.11	Block diagram of the ATI R100 GPU	127
Fig. 4.12	ATI's multi-texturing capabilities let developers add more special effects without impacting performance. <i>Source</i> ATI	128
Fig. 4.13	Block diagram of the ATI R100 Charisma engine	129
Fig. 4.14	Tim Chambers (courtesy of Bill DeBerry Funeral Directors—Denton TX)	130
Fig. 4.15	Changing of partners	131
Fig. 4.16	Imagination Technologies Kryo-based AIB. <i>Source</i> Trio3D Wikipedia	133
Fig. 4.17	GDC to IGC to IGP to iGPU (1980–2009)	136

Fig. 4.18	ArtX's Aladdin 7 3D integrated GPU controller .....	137
Fig. 4.19	SiS' 315 first integrated chipset GPU .....	143
Fig. 4.20	Nvidia's nForce IGP .....	145
Fig. 4.21	ATI/AMD's GPU developments over time .....	147
Fig. 4.22	Nvidia's GPU developments over time .....	148
Fig. 5.1	Significant developments in the GPU ecosphere .....	152
Fig. 5.2	The environment of the GPU .....	153
Fig. 5.3	From the ATI R520 of 2005 to the AMD Polaris architecture of 2016, the feature size was halved four times. Reproduced with permission from AMD .....	153
Fig. 5.4	In 2021, Intel announced it intended to build chips with a feature size of 20 angstroms in 2024. Reproduced with permission from Intel .....	154
Fig. 5.5	Intel's CEO, Pat Gelsinger, circa 2021. Reproduced with permission from Intel .....	154
Fig. 5.6	The road map for transistor design, from FinFET to CFET. Reproduced with permission from IMEC .....	155
Fig. 5.7	A typical computer architecture with a graphics AIB connected via PCI Express .....	159
Fig. 5.8	Typical integrated GPU with allocated system memory .....	159
Fig. 5.9	Types of GPU memory .....	160
Fig. 5.10	Multi-chip module alternatives. <i>Source</i> OCP ODSA .....	161
Fig. 5.11	MCM-GPU: aggregating GPU modules and DRAM on a single package. <i>Source</i> ISCA/Nvidia .....	162
Fig. 5.12	Intel's die-to-die stacking is like the interposer technology in EMIB. <i>Source</i> Intel .....	162
Fig. 5.13	AMD's The die-to-die interface uses a direct copper-to-copper bond with no solder bumps. <i>Source</i> AMD ....	163
Fig. 5.14	Heterogeneous integration enabled by an open chiplet ecosystem. <i>Source</i> UCle .....	164
Fig. 5.15	Packaging options using UCle. <i>Source</i> UCle Consortium .....	165
Fig. 5.16	The development of graphics interconnect systems over time .....	166
Fig. 5.17	The physical evolution of graphics AIBs over time. <i>Source</i> Free Online Dictionary of Computing .....	168
Fig. 5.18	A typical VL AIB. <i>Source</i> Wikipedia .....	170
Fig. 5.19	A riser card with a black slot for ISA and a white slot for PCI AIBs .....	171
Fig. 5.20	A typical PCI AIB. <i>Source</i> Wikipedia .....	171
Fig. 5.21	AGP signaling and power conventions. Reproduced with permission from JigPu at English Wikipedia .....	173
Fig. 5.22	Jim Pappas. <i>Source</i> Intel .....	174
Fig. 5.23	Evolution of PCIe technology. Reproduced with permission from Intel .....	175

Fig. 5.24	PCIe results and road map. Reproduced with permission from PCI SIG	175
Fig. 5.25	PCIe form factors. Reproduced with permission from Intel	176
Fig. 5.26	The first eGPU: ATI's XGA (2008)	177
Fig. 5.27	BT.709, sRGB, SMPTE 1886 (Gamma 2.4) = today's digital content, BT.2020, SMPTE 2084 (PQ) = HDR Content's color container. <i>Source</i> SMPTE	178
Fig. 5.28	A virtual reality system	184
Fig. 5.29	An augmented reality system	185
Fig. 5.30	AR tracking and other components	186
Fig. 5.31	An MR system	187
Fig. 5.32	A simplified representation of a metaverse continuum (Milgram, 1994)	187
Fig. 5.33	Examples of telepresence robots. Reproduced with permission from Projexive	188
Fig. 5.34	Tearing when V-Sync was switched off. Reproduced with permission from Nvidia	190
Fig. 5.35	Disabling V-Sync allows the GPU frame to be displayed when ready, which causes tearing on the screen as two or more frames were displayed in each refresh cycle	190
Fig. 5.36	Frames from the GPU with V-Sync: if a particular frame is early or late, the result is stutter and input delay	190
Fig. 5.37	If the G-Sync monitor was capable of a variable refresh rate, the GPU determines that rate	190
Fig. 5.38	FreeSync could sometimes cause brightness flickering with FPS fluctuations. Reproduced with permission from Display Ninja	194
Fig. 5.39	Voodoo2 in SLI configuration. Reproduced with permission from imgur.com	197
Fig. 5.40	AMD's XDMA multi-AIB. Reproduced with permission from AMD	198
Fig. 6.1	EDSAC was the world's first stored-program computer. It began operation on May 6, 1949, 3,000 vacuum tubes and 12KW. <i>Source</i> Wikipedia Copyright Computer Laboratory, University of Cambridge. Reproduced by permission	202
Fig. 6.2	Block diagram of the API and its relationship to the other components in a computer	203
Fig. 6.3	The history of APIs	204
Fig. 6.4	Mobile graphics evolution through OpenGL ES to Vulkan. <i>Source</i> Khronos	208
Fig. 6.5	A retained-mode API	213
Fig. 6.6	An immediate-mode API	213

Fig. 6.7	Timeline of the evolution from IRIS GL to OpenGL evolution. Reproduced with permission from SIGGRAPH Asia 2008 .....	215
Fig. 6.8	The history of OpenGL to 2008. Reproduced with permission from SIGGRAPH Asia 2008 .....	216
Fig. 6.9	The Fahrenheit project was a clever idea that did not work out (image used with permission from Microsoft) .....	216
Fig. 6.10	AMD's Mantle program. Reproduced with permission from AMD .....	218
Fig. 6.11	Apple's depiction of a thin API between the GPU and application. Reproduced with permission from Apple .....	219
Fig. 6.12	A family tree of graphics APIs. Reproduced with permission from Neil Trevett .....	220
Fig. 6.13	The Vulkan development team, circa 2017. Reproduced with permission from Khronos .....	221
Fig. 6.14	API layering. Reproduced with permission from Khronos .....	221
Fig. 6.15	Microsoft's DirectX 12 Ultimate logo (Used with permission from Microsoft) .....	224
Fig. 6.16	The shading rate is dynamically adjusted across the image, meaning that each $16 \times 16$ -pixel region of the screen could have a different shading rate. Reproduced with permission from Nvidia .....	226
Fig. 6.17	Clipping the portion of a model that is not visible .....	227
Fig. 6.18	DirectX 8 introduced The programmable vertex shader in 2000 .....	228
Fig. 6.19	Expanded view of the geometry pipeline .....	228
Fig. 6.20	George Lucas (AP Wirephoto—eBay, public domain, Wikipedia) .....	229
Fig. 6.21	Increasing the level of detail of a model through tessellation. Reproduced with permission from GDC .....	230
Fig. 6.22	The geometry pipeline as of 2007—long and complicated .....	232
Fig. 6.23	AMD's vertex processor can be a domain shader or a vertex shader .....	233
Fig. 6.24	Evolution of the GPU pipeline .....	236
Fig. 6.25	The traditional pipeline .....	237
Fig. 6.26	The three stages of a mesh shader (where the choice of data was left to the developer) .....	237
Fig. 6.27	Breaking down a model into sections using meshlets .....	239
Fig. 6.28	Triangles and vertices in a strip .....	239
Fig. 6.29	Triangles and vertices in an array .....	240
Fig. 6.30	Injection of the amplification shader .....	240
Fig. 6.31	Dataflow in the mesh shader pipeline .....	241
Fig. 6.32	The iconic 3DMark mesh shader test image .....	242

Fig. 6.33	A screenshot from the interactive mode, in which the meshlets in the scene were visualized. Notice that the foreground occludes the faces behind . . . . .	243
Fig. 6.34	An improvement of 730% in the FPS was achieved using mesh shaders . . . . .	243
Fig. 6.35	This entire image was calculated—it is not a photograph or texture map. Reproduced with permission from Epic Games . . . . .	244
Fig. 6.36	This image is what 20-million triangles look like at 4 k—each color represents a different triangle. Reproduced with permission from Epic Games . . . . .	245
Fig. 6.37	A soldier consisting of 30 million triangles, rendered in real time with global illumination. Reproduced with permission from Epic Games . . . . .	246
Fig. 6.38	There are extensive models, tremendous depth of field, real-time physics, character animation, motion blur, and more. Reproduced with permission from Epic Games . . . . .	247
Fig. 7.1	Data flow for Nvidia’s DLSS 2.0 process . . . . .	254
Fig. 7.2	Amazon’s Bistro demo . . . . .	256
Fig. 7.3	Anton Kaplanyan, developer of denoising. <i>Source</i> Intel . . . . .	257
Fig. 7.4	AMD’s super-resolution gaming patent. <i>Source</i> U.S. Patent and Trademark Office . . . . .	258
Fig. 7.5	Alexander Potapov, machine learning graphics specialist at AMD . . . . .	258
Fig. 7.6	Comparison of high resolution, XeSS scaled image, and low resolution. <i>Courtesy</i> Intel . . . . .	260
Fig. 7.7	SYCL was a cross-platform and OS model. <i>Source</i> Khronos . . . . .	262
Fig. 7.8	Getting from an image to maps and back again. <i>Source</i> Khronos . . . . .	265
Fig. 7.9	KTX was a lightweight container format for consistent, cross-vendor distribution of GPU textures and contained all the parameters necessary for efficient texture loading and handling. <i>Source</i> Khronos . . . . .	265
Fig. 7.10	AMD tried to convince game developers to develop for both platforms, circa 2013. <i>Source</i> AMD . . . . .	267
Fig. 7.11	Nvidia launched its GameWorks suite in 2014 to persuade developers to take advantage of Nvidia’s special features. <i>Source</i> Nvidia . . . . .	268
Fig. 7.12	Percentage closer soft shadows softens shadows, making them more realistic. <i>Source</i> Nvidia . . . . .	269
Fig. 7.13	Comparison of AMD’s and Nvidia’s revenue and graphics market share over time . . . . .	270
Fig. 7.14	Richard Huddy being interviewed about GameWorks. <i>Source</i> PC perspective . . . . .	271
Fig. 7.15	AMD’s end-to-end open-source compute software stack . . . . .	273



Fig. 7.16	Dennis “Thresh” Fong in the Ferrari he won from John Carmack (right). <i>Source</i> Heresy22, CC BY-SA 3.0 Wikipedia . . . . .	275
Fig. 7.17	UI for AMD’s Adrenalin . . . . .	276
Fig. 7.18	Screenshot of the control panel for Nvidia’s GeForce experience . . . . .	277
Fig. 7.19	UI for Nvidia’s GeForce experience . . . . .	278
Fig. 7.20	Ray marching takes a different approach to the ray-object intersection problem ray marching does not calculate an intersection analytically. Instead, it marches a point along the ray until it finds a point that intersects an object . . . . .	279

# List of Tables

Table 1.1	Graphics chip supplier from 1980 to 2023	16
Table 2.1	Shading rates and coarse pixel size	52
Table 3.1	The evolution of the APIs and OS relative to the eras of GPUs	67
Table 4.1	First-generation GPUs	109
Table 4.2	Early GPUs from ATI and Nvidia	122
Table 4.3	ATI's R100 had three texture units per graphics pipeline	127
Table 4.4	Northbridge comparisons	139
Table 5.1	Types and generations of memory	157
Table 5.2	Comparison of GDDR5, GDDR5X, HBM, and HBM2 memory	158
Table 5.3	PC bus standards	167
Table 5.4	AGP and PCI: 32-bit buses operated at 66 and 33 MHz, respectively	173
Table 5.5	AGP power provisioning	174
Table 5.6	Formats and versions of HDMI video (Wikipedia)	180
Table 5.7	Comparison of AR and VR characteristics. <i>Source</i> Phillip Rauschnabel	189
Table 5.8	G-Sync modes and results	191
Table 6.1	History of modern APIs for all platforms	205
Table 6.2	DirectX 9 shader version	208
Table 6.3	Comparison of DirectX vertex shaders	209
Table 6.4	Compatibility of APIs with OSs and their deployment	210
Table 6.5	New programmable features introduced to the API in DirectX 10 and 11	230
Table 6.6	Comparison of the characteristics of the DirectX 8, DirectX 9, and DirectX 10 shaders	234
Table 7.1	The five modes of AMD's FidelityFX FSR	259
Table 7.2	Quality modes offered by Intel's XeSS scaling options	260
Table 7.3	Intel's XeSS data-flow diagram	260
Table 7.4	GPU-compressed texture format fragmentation	263

Table 7.5	Comparison of the two modes of Basis Universal, UASTC, and ETCIS .....	265
Table 7.6	AMD's initial GPUOpen resources from 2015 .....	272

# Chapter 1

## Introduction



By the mid-nineties, the forerunners of the fully integrated single-chip GPU were appearing with a range of functionality. What companies built often reflected their origin of their fonder, and target markets.

LSI graphics accelerator chips appeared in the early 1980s as 2D drawing engines, with CAD being the primary application. In 1981, SGI introduced the first 3D transform engine, the Geometry Engine [1]. The first company to offer a 3D AIB was Matrox in 1987 with the SM 640 that used the SGI Geometry Engine [2].

Semiconductor companies saw the need to decouple the geometry processing from the CPU and make it more tightly coupled with the pixel pipeline. Several companies developed floating-point processors (FPUs) to do the job.

Weitek introduced its floating-point processor (FFP) in the early 1980s and was successful with it as a coprocessor to the CPU. Some groups tried using it for geometry processing. The Pixel Planes system used it and it was used in the Sun workstation, and even by SGI. However, it never was used in any PC graphics AIBs. It was not a geometry processor per se but a general-purpose floating-point coprocessor. In the early 1990s, the company introduced a VGA clone, the Power 9100. The company did not do well, and in late 1996, Rockwell's Semiconductor Systems bought the assets.

In May 1994, 3Dlabs announced its first consumer 3D graphics chip, Gigi (which stood for *Game Glint*). It was a scaled-down version of the GLINT 300TX. Creative Labs built an AIB with it, the 3D Blaster VLB, in early 1995. However, Windows did not have an API that exposed T&L functions in the graphics hardware. Therefore, 3Dlabs chips came with 3Dlabs drivers for Windows 95/NT (accelerated 2D), accelerated D3D drivers, and accelerated OpenGL drivers. The 3D drivers would use whatever acceleration was enabled by the API and available on the current hardware.

The first consumer grade Geometry Engine chip was the Fujitsu (MB86242) Pinolite FXG-1, revealed at the Hot Chips conference at Stanford University in 1995 [3] and formally announced in July 1997 [4]. Rendition was the first company to use (and helped Hercules develop) an AIB with it and Rendition's 3D chip in 1998. However, the AIB's release got canceled because 3Dfx came out with a faster product.

Also, DirectX did not expose the geometry engine to applications so custom drivers had to be written.

Also, in 1997, 3Dlabs announced their Glint Gamma (G1) stand-alone geometry processor for the professional graphics market. Then they adapted it to a Creative Labs consumer AIB in 1988.

Being first is always a tricky thing to identify. First to announce or first to ship? Or, if you could find out, first start a design? However, the one thing that can be said is that 3D consumer chips appeared in 1995.

Those early 3D chips did not include hardware transform and lighting (T&L) capabilities. The CPU or a separate coprocessor was used for those operations. But Moore's law was on the march, and it would be only a few years until the first fully integrated single-chip graphics processing unit (GPU) was introduced, marking the beginning of the GPU era.

## 1.1 Nvidia's NV10—First Integrated PC GPU (September 1999)

The term *GPU* has been in use since at least the 1980s. Nvidia popularized it in 1999 by marketing the GeForce 256 AIB as “the world's first GPU.” It offered integrated transform, lighting, triangle setup/clipping, and rendering engines as a single-chip processor.

It had all the features for a truly integrated GPU.

Very large-scale integrated circuitry (VLSI) started taking hold in the early 1990s. As the number of transistors that engineers could incorporate on a single chip increased almost exponentially, the number of functions in the CPU and the graphics processor increased. Among the biggest consumers of the CPU were graphics transformation compute elements into graphics processors. Architects from various graphics chip companies decided that transform and lighting (T&L) was a function that should be in the graphics processor. A T&L engine is a vertex shader and a geometry translator—many names for the little FFP.

In 1997, 3Dlabs (in the UK) developed its Glint Gamma processor, the first programmable transform and lighting engine, as part of its Glint workstation graphics chips. They introduced the term *GPU*—geometry processor unit. 3Dlabs' GPU was a separate chip named Delta and was known as the DMX. 3Dlabs' GMX was a coprocessor to the Glint rasterizer.

Then in September 1999, Nvidia introduced the NV10 GPU with an integrated T&L engine for their consumer graphics chip. ATI quickly followed with their Radeon graphics chip and called it a *VPU*—visual processing unit. But Nvidia popularized the term GPU. It has forever since been associated with the GPU and credited with inventing it (Fig. 1.1).

Built on TSMC's 220 nm process, the 120 MHz NV10 had 23 million (incorrectly listed as 17 on some websites) transistors in a 139 mm<sup>2</sup> die and used DirectX 7.0.

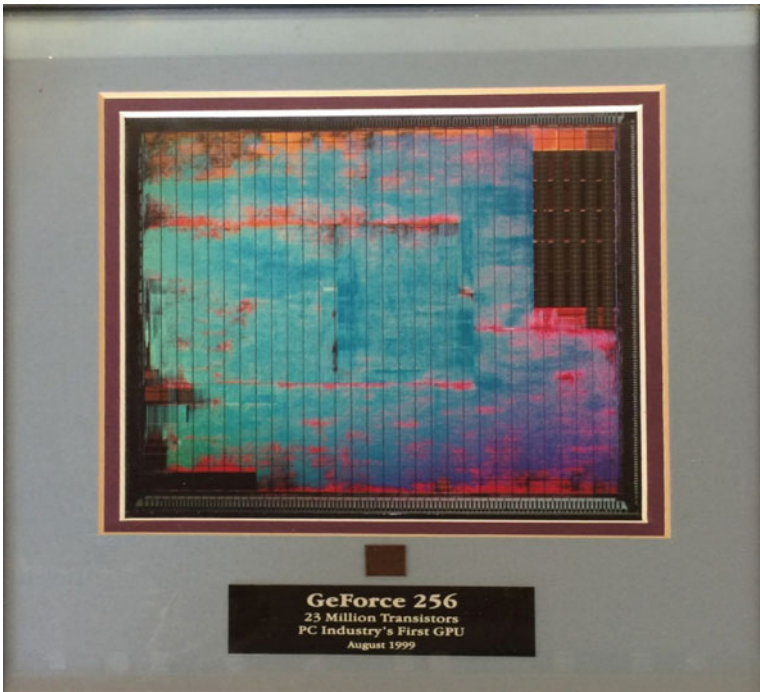


Fig. 1.1 Die shot of Nvidia's first GPU, the NV10. Reproduced with permission from Curtis Priem

The GeForce 256 AIB employed the NV10 with SDR memory. Refer to the block diagram in Fig. 1.2.

The chip had many advanced features, including four independent pipelined engines that ran at 120 MHz. That allowed the GPU to produce a 480 Mpix/s fill rate.

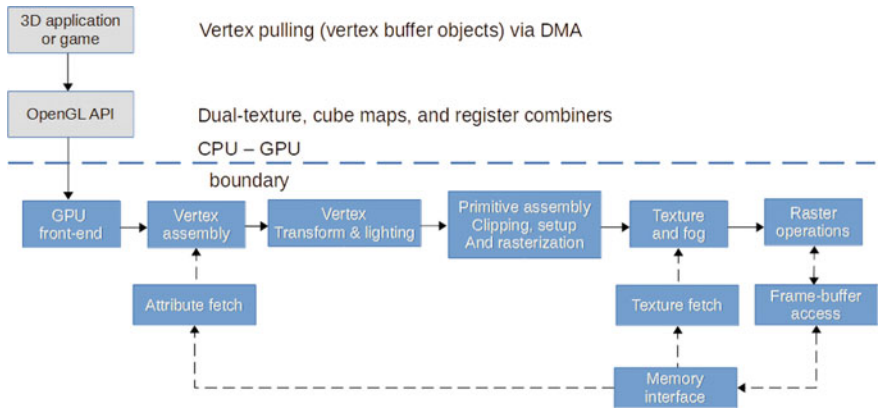


Fig. 1.2 Nvidia's GeForce 256 (NV10) block diagram with OpenGL