Create Your Own GPU From Scratch - Complete Book Plan

Book Structure

This book is organized into 5 major parts with 25 comprehensive chapters, each 15-30 pages of detailed content.

PART I: GPU FUNDAMENTALS (Chapters 1-5)

Chapter 1: Big Picture - What is a GPU?

Length: ~20 pages
Topics:

- Introduction to parallel computing
- CPU vs GPU philosophy
- Real-world GPU applications (graphics, AI, scientific computing)
- History of GPU evolution
- Why build your own GPU?
- Case studies: NVIDIA, AMD, Intel Arc
- What you'll learn in this book

Chapter 2: GPU vs CPU - Architecture Deep Dive

Length: ~25 pages
Topics:

- CPU microarchitecture review (pipelines, caches, branch prediction)
- GPU microarchitecture fundamentals
- Transistor budget allocation
- Control logic vs compute logic

- Memory hierarchy differences
- Power and thermal considerations
- Performance metrics comparison
- When to use GPU vs CPU

Chapter 3: Parallel Execution Models

Length: ~30 pages
Topics:

- SIMD (Single Instruction Multiple Data)
- SIMT (Single Instruction Multiple Thread)
- Threads, warps/wavefronts, blocks, grids
- Thread divergence and convergence
- Programming model abstractions
- CUDA/OpenCL/HIP overview
- Kernel launch mechanics
- Thread synchronization primitives
- Practical examples with code

Chapter 4: Core GPU Components

Length: ~28 pages
Topics:

- Arithmetic Logic Unit (ALU) design
- Register files and organization
- Program Counter (PC) and instruction fetch
- Scheduler architectures
- Load/Store Unit (LSU) design
- Execution pipelines
- Datapath design
- Control signals and FSMs
- Block diagrams and schematics

Chapter 5: Instruction Set Architecture (ISA)

Length: ~25 pages
Topics:

- ISA design principles
- Instruction encoding formats
- Register allocation strategies
- Arithmetic instructions
- Memory instructions
- Control flow instructions
- Special registers (%threadIdx, %blockIdx, etc.)
- Instruction latency and throughput
- ISA evolution and extensions
- Example ISA specification

PART II: EXECUTION & SCHEDULING (Chapters 6-11)

Chapter 6: Pipeline Design and Control Flow

Length: ~30 pages
Topics:

- Fetch-Decode-Execute cycle
- Multi-stage pipeline design
- Hazard detection and forwarding
- Stall conditions
- State machine design
- Branch handling
- Pipeline diagrams
- Timing diagrams
- Performance analysis

Chapter 7: Memory Hierarchy

Length: ~35 pages
Topics:

- Memory pyramid (registers → cache → DRAM)
- Cache design (L1, L2, L3)
- Cache coherence protocols
- Shared memory / scratchpad
- Global memory organization
- Memory addressing modes
- Bank conflicts
- Memory latency hiding
- Bandwidth optimization

Chapter 8: Memory Coalescing

Length: ~22 pages
Topics:

- Why coalescing matters
- Access pattern detection
- Coalescing logic design
- Transaction merging
- Performance impact
- Software optimization techniques
- Hardware implementation
- Case studies

Chapter 9: Thread Scheduling

Length: ~28 pages
Topics:

- Warp/wavefront scheduling
- Round-robin schedulers
- Scoreboarding

- Dependency tracking
- Latency hiding strategies
- Occupancy optimization
- Scheduler FSM design
- Performance modeling

Chapter 10: Advanced Execution Units

Length: ~30 pages
Topics:

- Tensor cores / matrix engines
- Ray tracing acceleration
- Texture sampling units
- Special function units (SFU)
- Fixed-function pipelines
- Custom accelerators
- Dataflow architectures
- Performance tradeoffs

Chapter 11: Graphics vs Compute Pipelines

Length: ~25 pages
Topics:

- Rasterization pipeline
- Vertex shaders
- Fragment shaders
- Compute shaders
- Unified shader architecture
- Graphics-specific features
- Compute-focused design
- Hybrid approaches

PART III: SOFTWARE & TOOLING (Chapters 12-14)

Chapter 12: Software Stack & Programming Model

Length: ~32 pages
Topics:

- Driver architecture
- Runtime system design
- Kernel launch APIs
- Memory management APIs
- CUDA/OpenCL/Vulkan comparison
- DSL design for custom GPUs
- Host-device communication
- Debugging and profiling tools

Chapter 13: Compiler & Code Generation

Length: ~35 pages
Topics:

- Compiler frontend (parsing, AST)
- Intermediate representations (LLVM IR, SPIR-V)
- Optimization passes
- Register allocation
- Instruction scheduling
- Code generation
- Kernel compilation flow
- JIT compilation
- Example compiler implementation

Chapter 14: Testing & Verification

Length: ~30 pages
Topics:

- Unit testing strategies
- Integration testing
- Kernel testbenches
- Simulation frameworks (cocotb, VCS, etc.)
- Waveform analysis
- Coverage metrics
- Formal verification basics
- Continuous integration
- Bug tracking and triage

PART IV: HARDWARE IMPLEMENTATION (Chapters 15-21)

Chapter 15: Microarchitecture Specification

Length: ~28 pages
Topics:

- Requirements gathering
- Performance targets
- Power budgets
- Area constraints
- Block diagrams
- Interface specifications
- Timing budgets
- Microarchitecture document template
- Review and iteration

Chapter 16: RTL Design Fundamentals

Length: ~35 pages
Topics:

- Verilog vs SystemVerilog
- Module hierarchy
- Coding style guide
- Synchronous design principles
- Reset strategies
- Clock domain crossing
- Parameterization
- Generate blocks
- Arrays of instances
- Common pitfalls

Chapter 17: Building Your First GPU Core

Length: ~40 pages
Topics:

- Top-level architecture
- ALU implementation
- Register file design
- Decoder implementation
- PC and branch logic
- LSU with async memory
- Scheduler FSM
- Integration and wiring
- Full RTL walkthrough
- Simulation and debug

Chapter 18: Multi-Core GPU Design

Length: ~30 pages
Topics:

- Replication strategies
- Dispatcher design
- Memory controller arbitration

- Interconnect fabrics
- Scalability considerations
- Resource sharing
- Load balancing
- Performance scaling

Chapter 19: Advanced RTL Topics

Length: ~32 pages
Topics:

- Pipelining for performance
- Warp scheduling in hardware
- Branch divergence handling
- Cache implementation
- Coalescing unit design
- Power gating
- Clock gating
- Area optimization techniques

Chapter 20: Simulation & Debug

Length: ~28 pages
Topics:

- Testbench architecture
- Stimulus generation
- Memory models
- Assertion-based verification
- Waveform debugging
- Performance analysis
- Regression testing
- CI/CD for hardware

Chapter 21: Synthesis & Timing

Length: ~30 pages
Topics:

- Synthesis tool flow (Synopsys, Cadence, Yosys)
- Timing constraints (SDC)
- Clock definitions
- Input/output delays
- False paths
- Multi-cycle paths
- Area vs speed tradeoffs
- Interpreting reports
- Fixing timing violations

PART V: PHYSICAL DESIGN & MANUFACTURING (Chapters 22-25)

Chapter 22: Physical Design Flow

Length: ~35 pages
Topics:

- Floorplanning
- Power planning
- Placement strategies
- Clock tree synthesis
- Routing (global and detailed)
- Timing closure techniques
- IR drop analysis
- EM analysis
- Tool flows (Innovus, ICC2, OpenLane)

Chapter 23: Signoff & Validation

Length: ~30 pages
Topics:

- DRC (Design Rule Check)
- LVS (Layout vs Schematic)
- STA (Static Timing Analysis)
- Power analysis
- Signal integrity
- Crosstalk analysis
- Antenna checks
- Formal equivalence checking
- Signoff checklist

Chapter 24: Tape-out Process

Length: ~28 pages
Topics:

- Foundry selection
- PDK (Process Design Kit)
- GDS generation
- Mask preparation
- Reticle layout
- OPC (Optical Proximity Correction)
- Packaging options (BGA, flip-chip, chiplets)
- Pinout planning
- Documentation requirements
- Cost estimation

Chapter 25: Manufacturing & Bring-up

Length: ~32 pages
Topics:

- Wafer fabrication overview
- Die testing

- Packaging and assembly
- Board design
- First silicon bring-up
- Debug strategies
- Validation testing
- Performance characterization
- Yield analysis
- Production ramp
- Lessons learned

APPENDICES

Appendix A: ISA Reference

- Complete instruction set
- Encoding tables
- Pseudocode

Appendix B: RTL Code Listings

- Full Verilog/SystemVerilog source
- Testbench examples

Appendix C: Tool Setup Guides

- Installing Icarus Verilog, Verilator, Yosys
- cocotb setup
- OpenLane flow
- Open-source PDKs

Appendix D: Glossary

- GPU terminology

- Hardware design terms
- Acronyms

Appendix E: Further Reading

- Academic papers
- Industry white papers
- Open-source projects
- Online courses

Total Book Statistics

- **Total Chapters:** 25
- **Total Pages:** ~750-800 pages
- **Reading Time:** 40-50 hours
- **Code Examples:** 150+
- **Diagrams:** 200+
- **Exercises:** 100+