Video Coding Unit (VCU)

Hot Chips 2021

Aki Kuusela, Google Devices & Services
Clint Smullen, Google Systems Infrastructure
(on behalf of the larger VCU team)

Video: top contributor to Internet traffic

Video is > 60% of the global Internet traffic

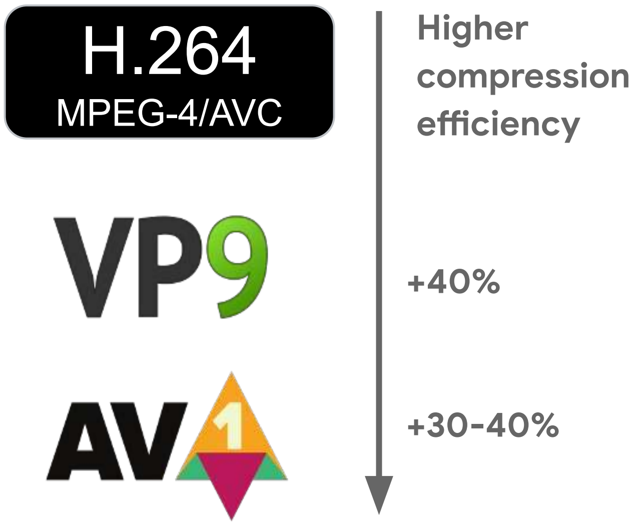
- growth accelerated by Covid

Resolutions & frame rates growing

Complexity of the compression formats growing

Video is getting harder to compress

Year	Video Format	SW encoding time at best quality
2003	AVC/H.264	1x
2013	VP9	10x
2018	AV1	200x
202x	AV2	?



Higher compression efficiency

+40%

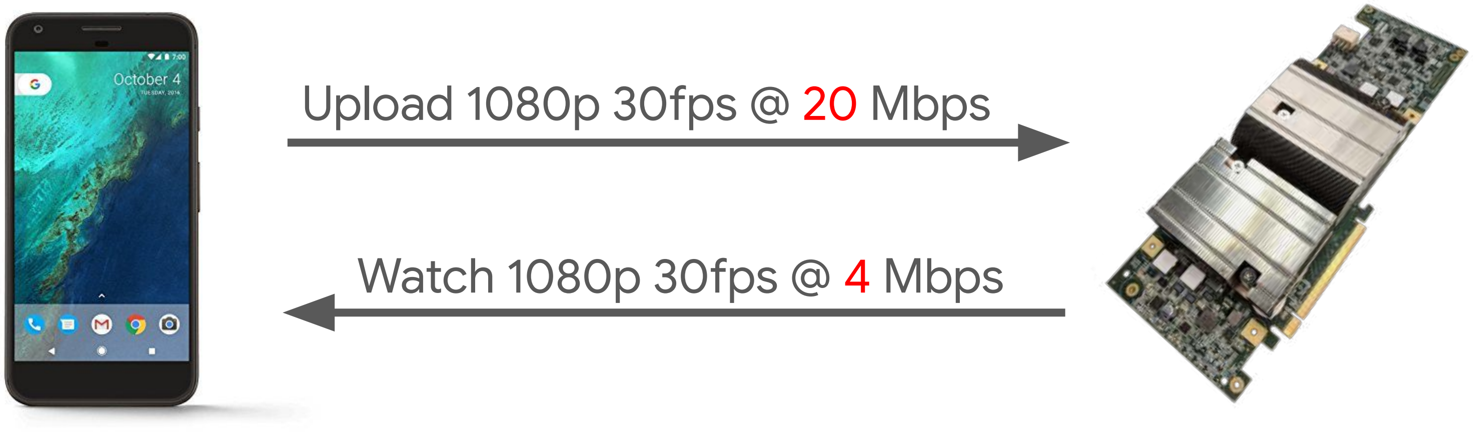
+30-40%

Video is getting harder to compress

Year	Video Format	SW encoding time at best quality	Times pixels/second increase
2003	AVC/H.264	1x	1x (1080p 24 fps)
2013	VP9	10x	**100x** (4k 60 fps)
2018	AV1	200x	**8000x** (8k 60 fps)
202x	AV2	?	?

Why develop our own video chips

Existing HW encoders needed up to 5x more bits at equal quality



Upload 1080p 30fps @ 20 Mbps

Watch 1080p 30fps @ 4 Mbps

Typical cell phone H.264 encoder

YouTube's H.264 using Google VCU

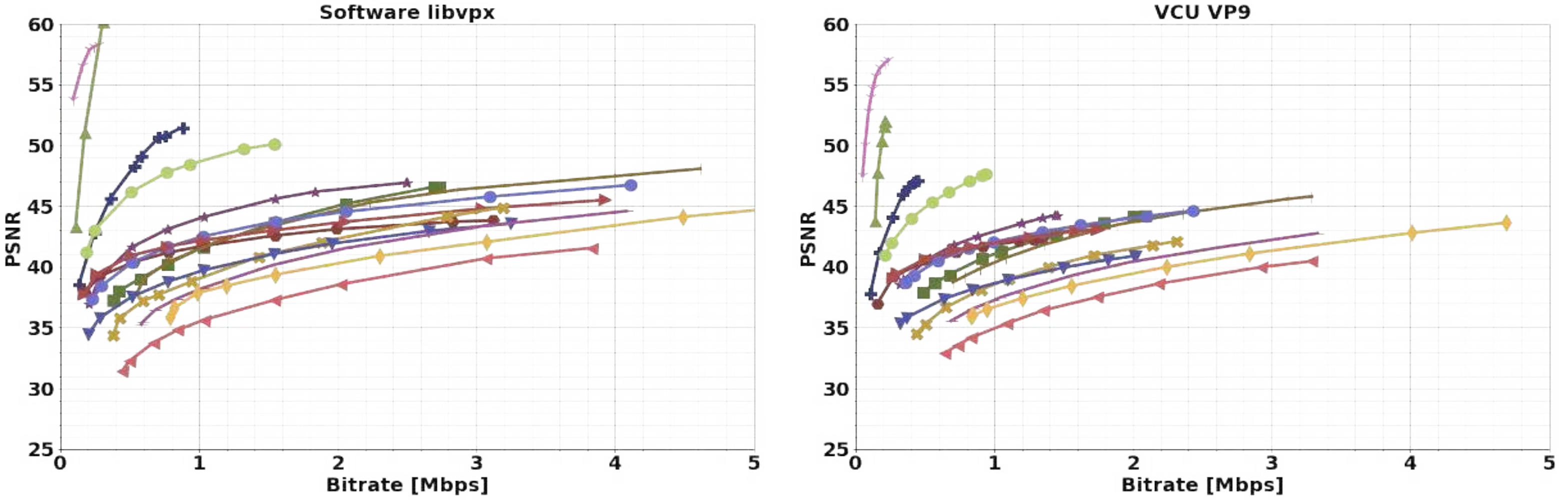
Why develop our own video chips

Things we wanted but were not available:

- Full implementation for H.264 and VP9
- Single and Multi-Output Transcoding (SOT, MOT)
- Speed vs. quality tuning, live streaming and offline transcoding
- Full access to SW control algorithms (rate control, group-of-pictures selection etc.)

Why develop our own video chips

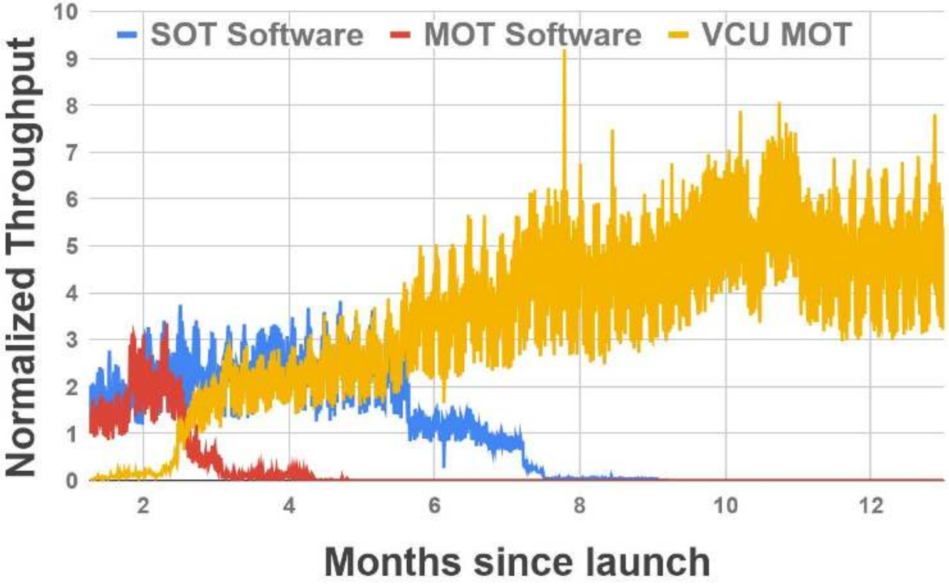
Resulting in near-parity to software encoding quality



Google

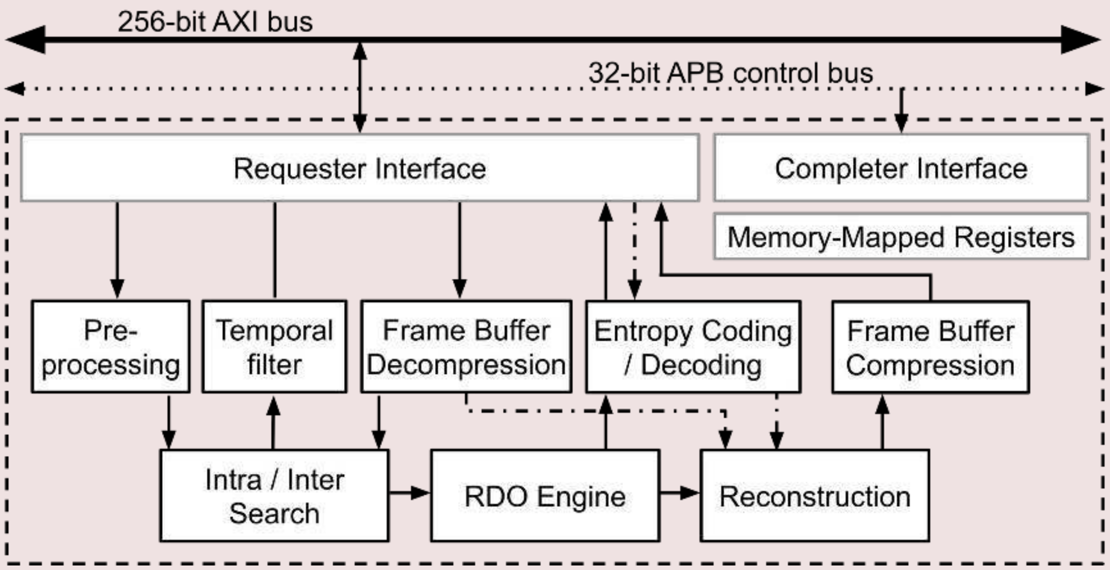
Turning the VCU fleet on ...

...cut down YouTube’s computing cycles dramatically



The Video Encoder Core

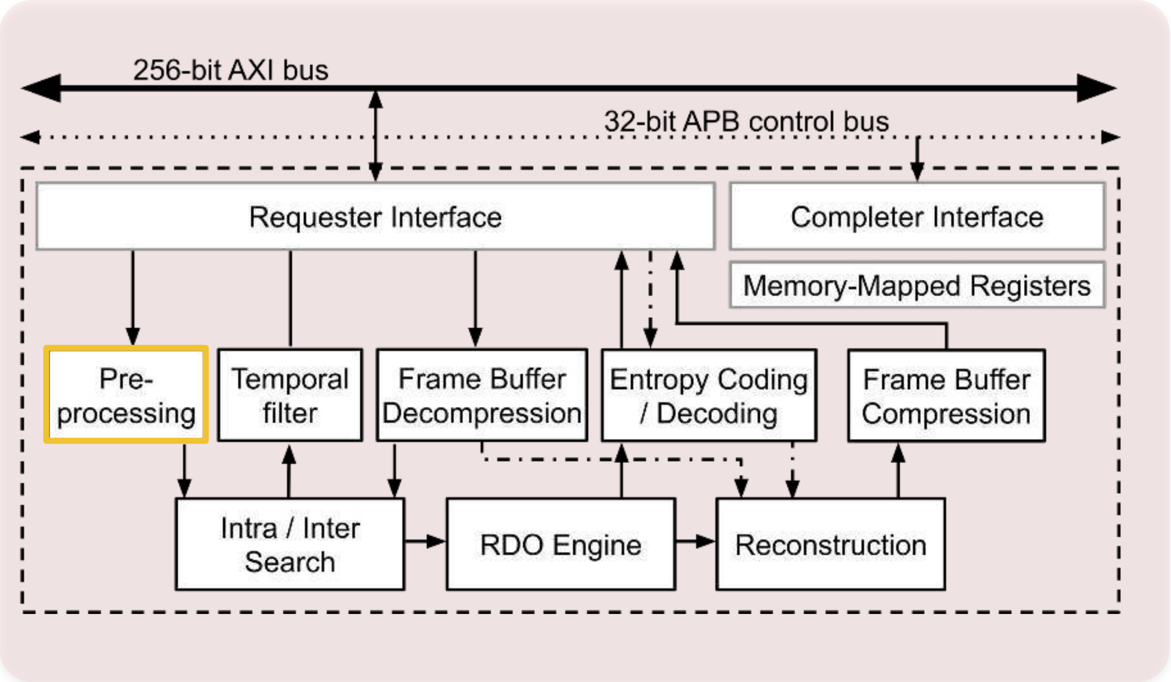
- Full hardware acceleration of H.264 or VP9 encoding at up to 4k 60 fps
- All coding tools of both formats included
- Sits on a standard AXI / APB bus



The Video Encoder Core

- High-quality inline pre-processing engine
    - Color space conversions
    - Cropping
    - Scaling
    - Rotation

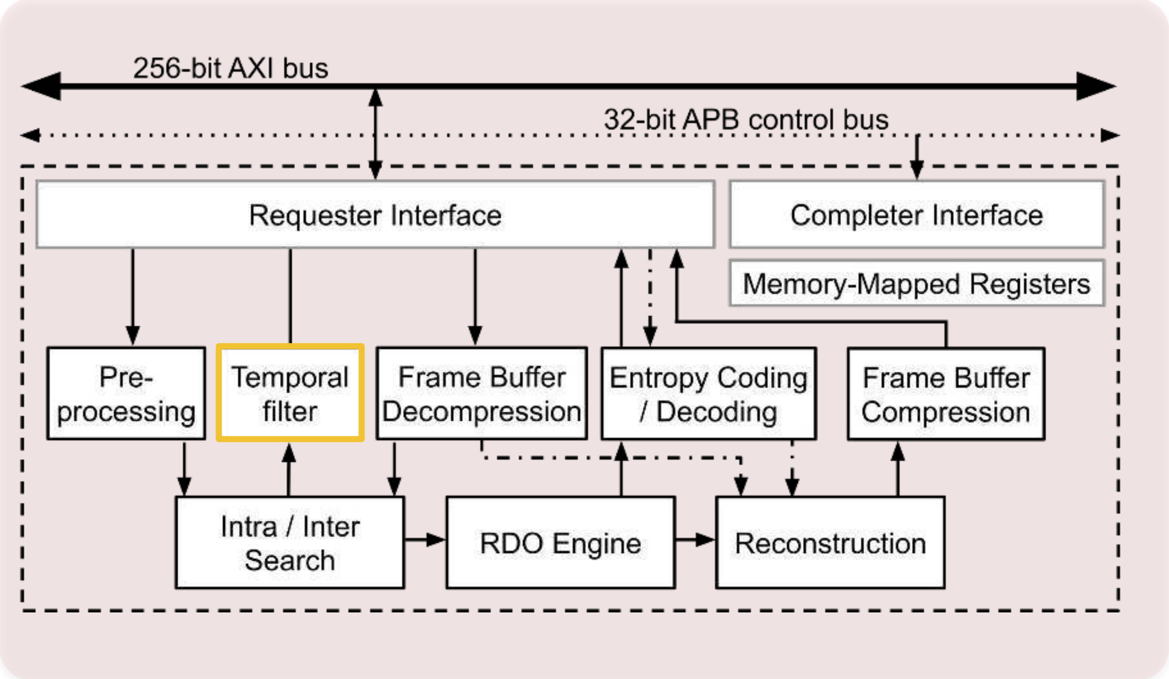
- Also acts as a standalone path



Google

The Video Encoder Core

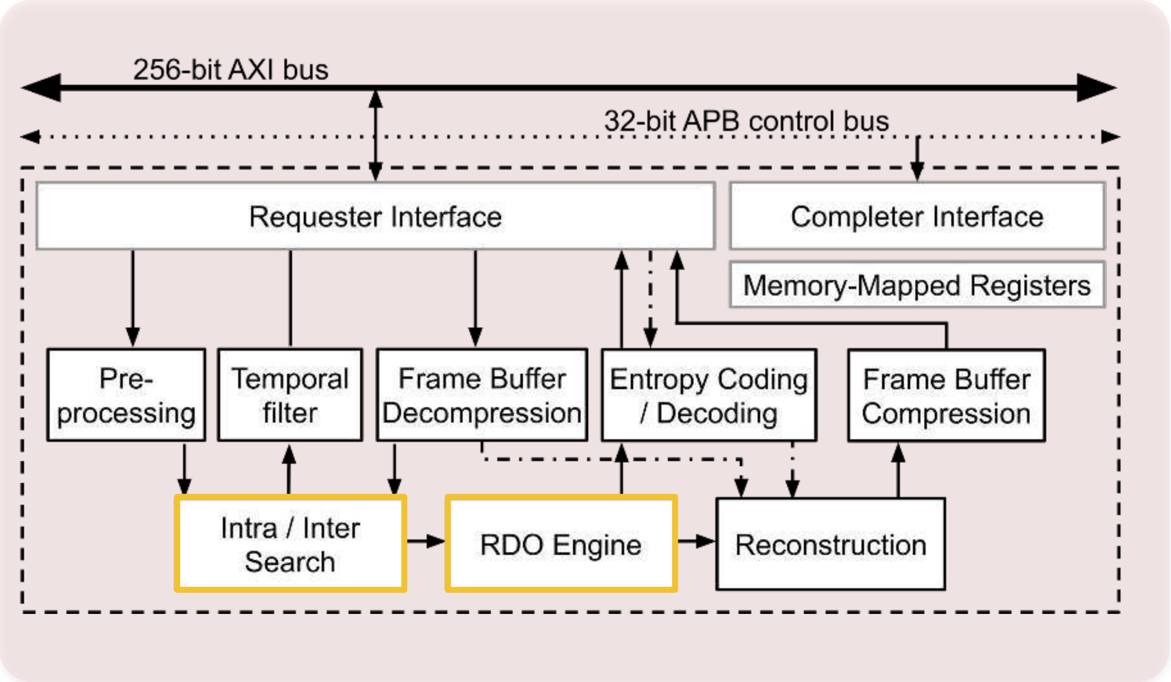
- Temporal denoiser for VP9 alternate reference frame generation
- Separate operation from encoding utilizing the encoder resources



Google

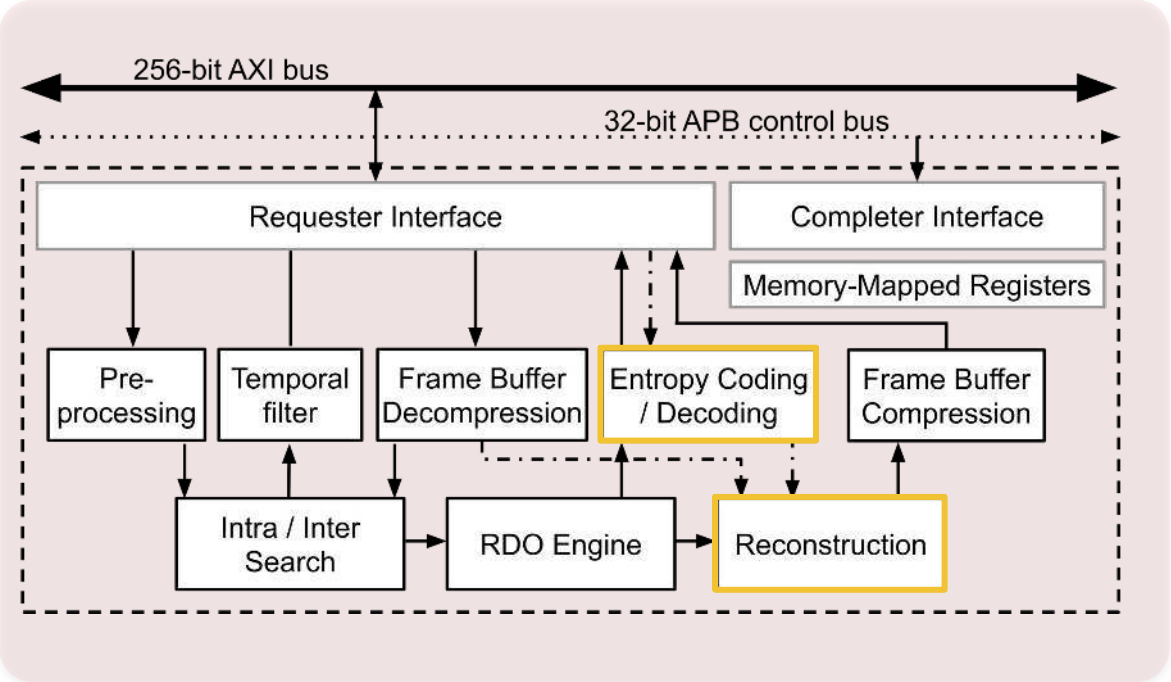
The Video Encoder Core

- Motion search and rate-distortion optimization engine
  - Trade-off speed and quality
  - Adjustable motion search window
  - Adjustable number of RDO candidates



The Video Encoder Core

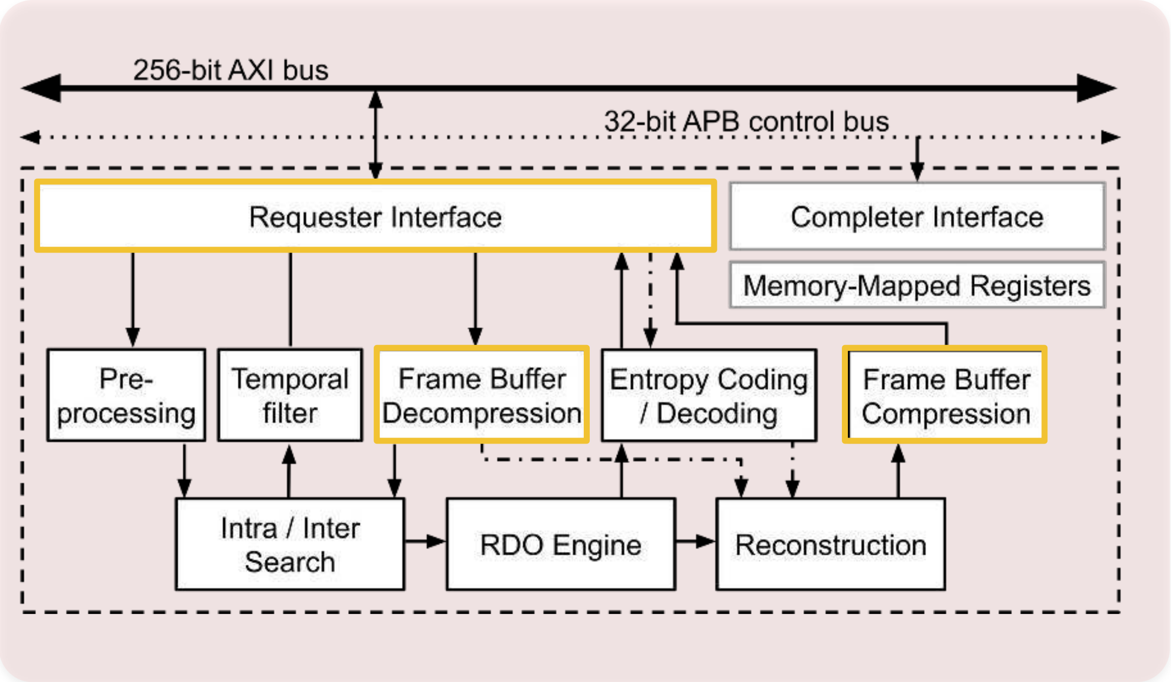
- Reconstruction and entropy coding
    - RD-optimal quantization
    - PSNR calculation
    - First pass statistics collection



The Video Encoder Core

- Each encoder core reads up to 4 frames and writes one

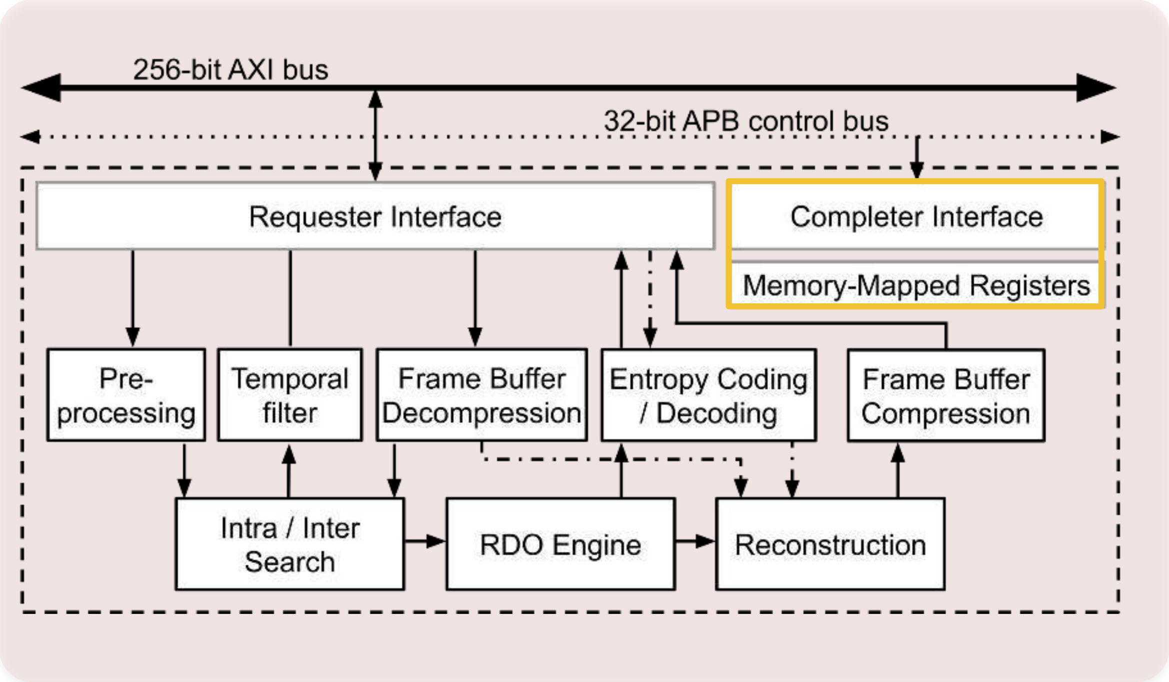
- Frame buffer compression allows more cores per chip



The Video Encoder Core

- Hundreds of programmable registers allow coding quality fine-tuning

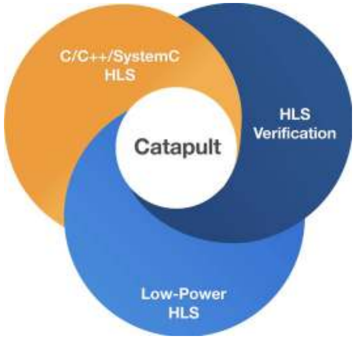
- Controlled by software algorithms like rate control



Google

A high-level design flow

- Google codec team has been using high-level synthesis design flows for almost 10 years
- The VCU video core was designed with Catapult, a C++ HLS flow from Siemens
- HLS was instrumental in VCU development enabling SW/HW co-design and allowing very fast design iteration



A high-level design flow - the benefits of C++

- No separate algorithmic model needed, single source of truth
- Always bit-exact results between model and RTL
- 5-10x less code to write, review, and maintain vs. RTL

- Software development tools
  - Address/MemorySanitizer
  - Distributed computing
- Testing throughput 7-8 orders of magnitude higher vs. RTL
- 99% of the functional bugs found in C++ before running any RTL simulation

A high-level design flow - more time for a better product

- Team working on high-value problems
    - Leave cycle-by-cycle design for the compiler
    - No debugging of block internal timing bugs
- Design space exploration
    - Try out high number of algorithms / architectures
- Feature creep, please!
    - Able to keep adding features & improvements very late in the process
- Technology scaling is trivial
    - Compiler creates new data path / FSM for a new clock target & technology from the same C++ source

VCU ASIC and System

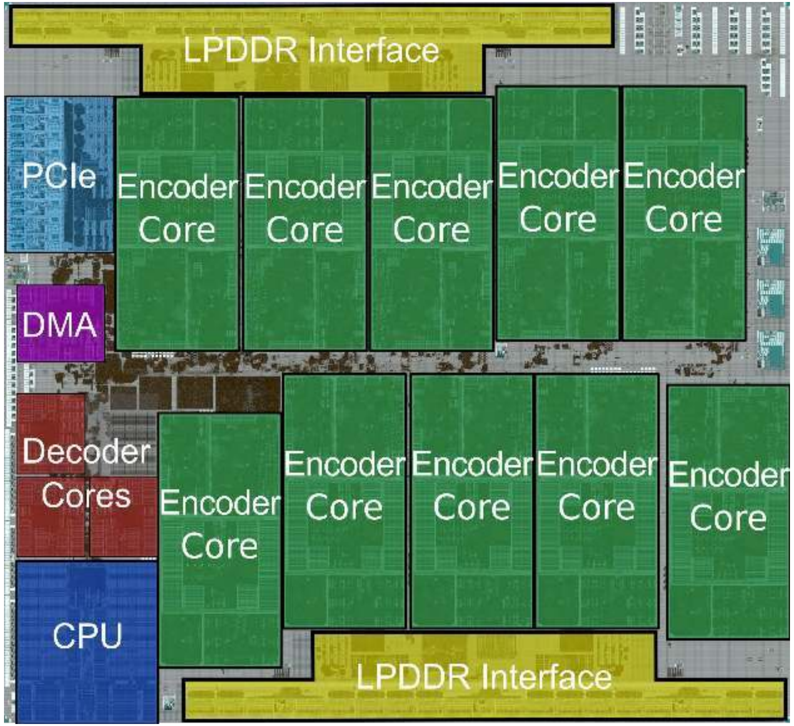
Warehouse-scale Approach to ASICs

- End of Moore’s Law:
  - Accelerators are needed to address cost/performance gap
- Design for the datacenter
  - Only deployed at cluster scale, heterogeneously mixed with CPU machines

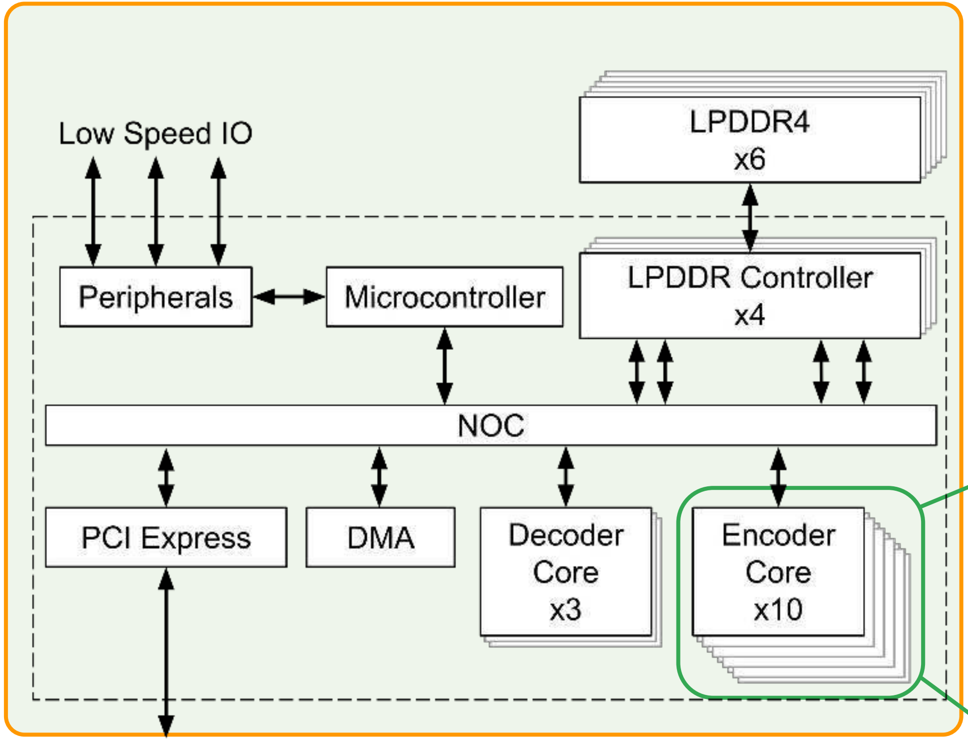
- Globally maximize utilization:
  - Diverse use-cases spread across many regions → Support fungible workloads
- Optimize for deployment at scale:
  - Reduce disruption from changes and failures → Tolerate chip- and core-level errors
- Design for agility and adaptability:
  - Neither use-cases nor usage patterns are fixed → HLS + Software for flexibility

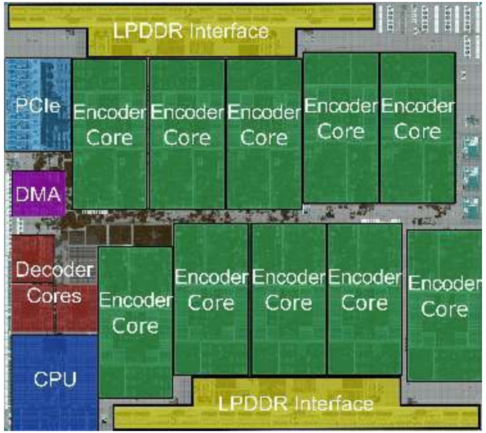
Chip Design Goals

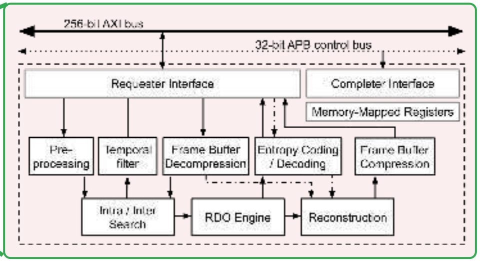
- Maximize utilization
  - Few jobs can use an entire chip
  - Isolated userspace queues
- Maximize userspace control
  - Video rate control, quality, or performance not determined by hardware, firmware, or kernel driver
  - Simple firmware work items (DMA data, run-on-core, etc.)
- Optimize to serve the encoder cores
  - DRAM latency, average/peak bandwidth



VCU ASIC

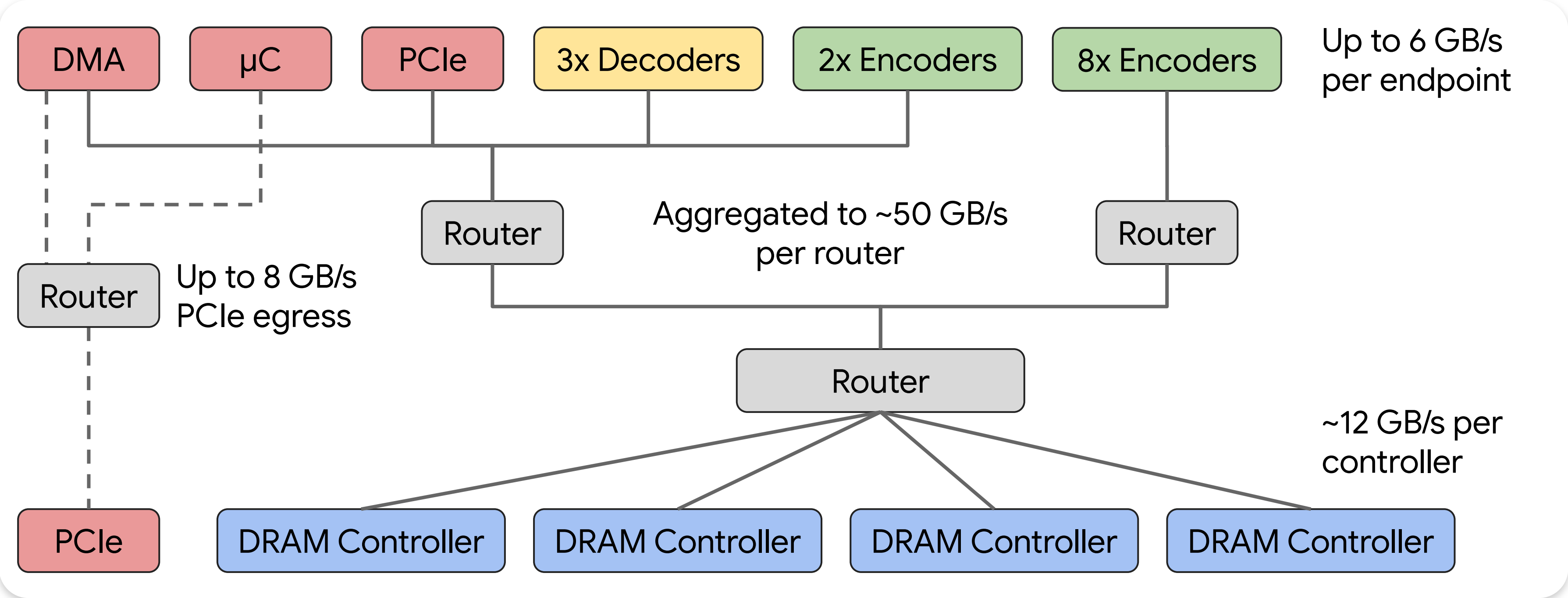






Match core counts and DRAM to handle multi-output transcoding use-cases

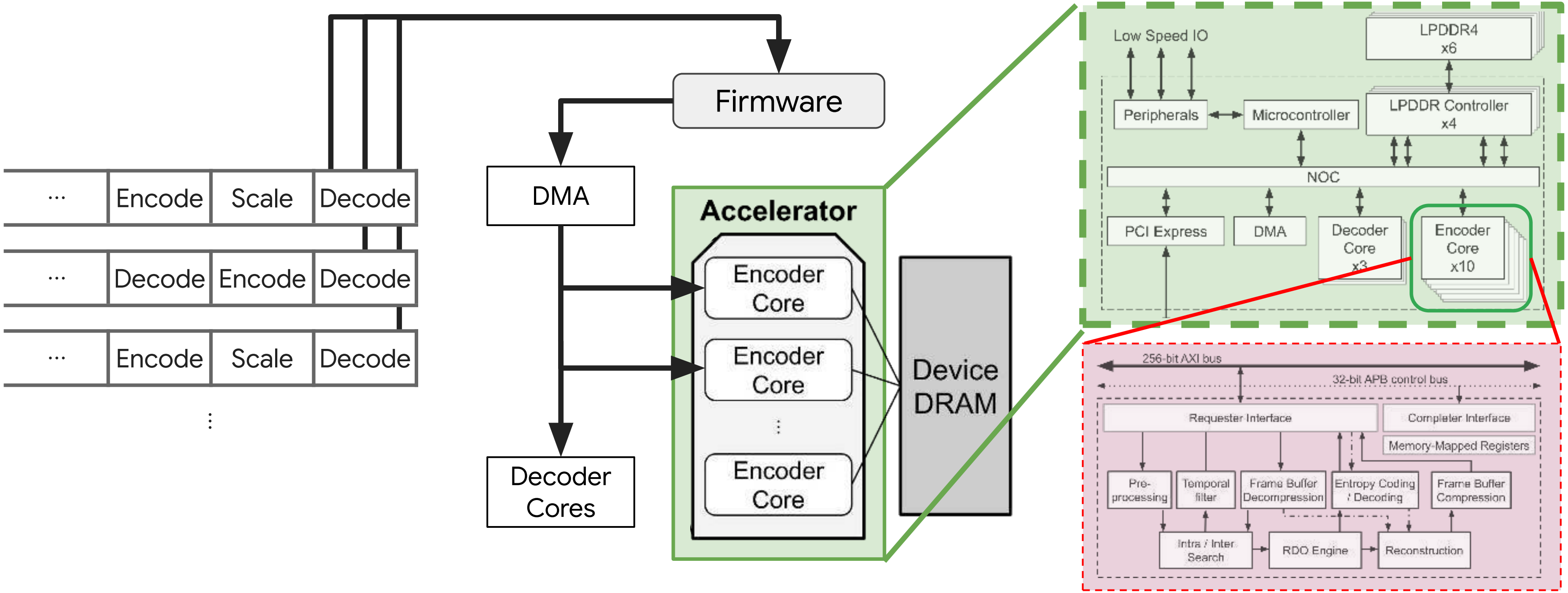
NoC Topology



Support bursty traffic and uniform access to DRAM for software simplicity

Google

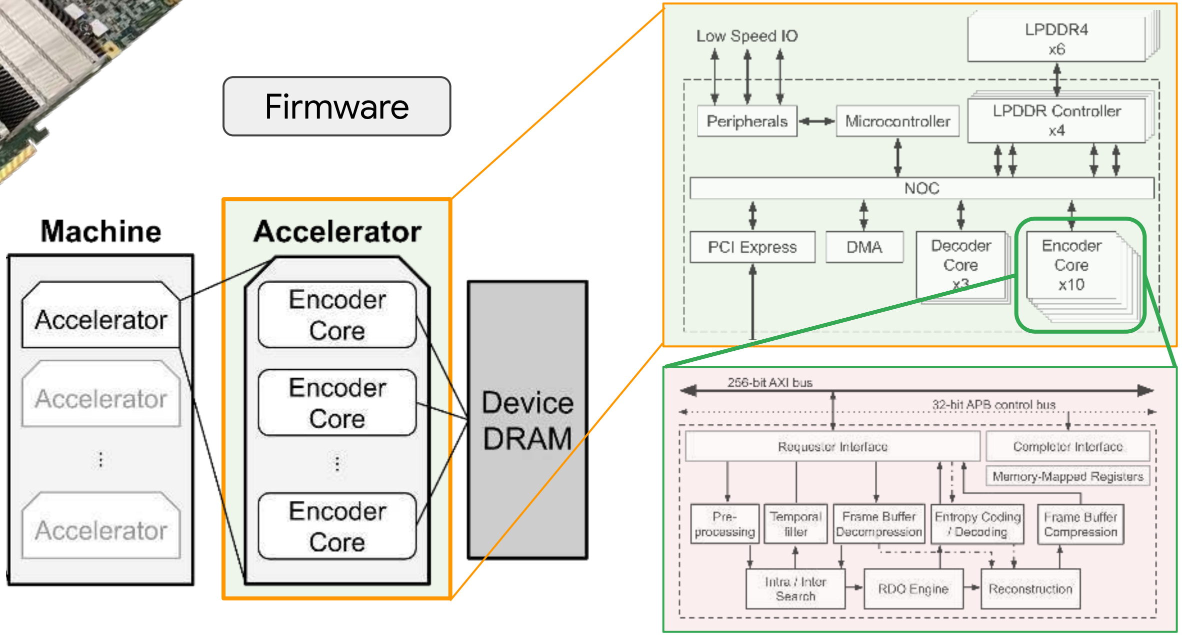
VCU Firmware



Userspace control of codec choices, parameters, and dependencies

Firmware controls work dispatch and isolation

System & Rack



Firmware

Machine

Accelerator

Accelerator

⋮

Accelerator

Accelerator

Encoder Core

Encoder Core

⋮

Encoder Core

Device DRAM

Low Speed IO

LPDDR4 x6

Peripherals

Microcontroller

LPDDR Controller x4

NOC

PCI Express

DMA

Decoder Core x3

Encoder Core x10

256-bit AXI bus

32-bit APB control bus

Requestor Interface

Completer Interface

Memory-Mapped Registers

Pre-processing

Temporal filter

Frame Buffer Decompression

Entropy Coding / Decoding

Frame Buffer Compression

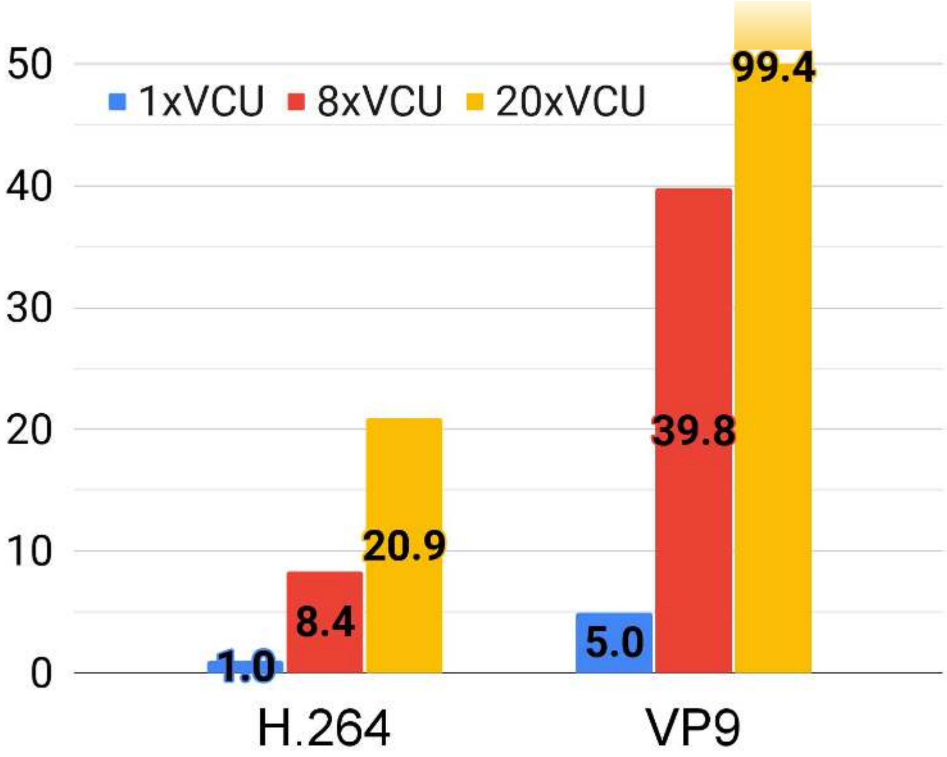
Intra / Inter Search

RDO Engine

Reconstruction

Maximize VCUs per board and per system for Perf/TCO$: 20 VCUs per system

Performance Comparison

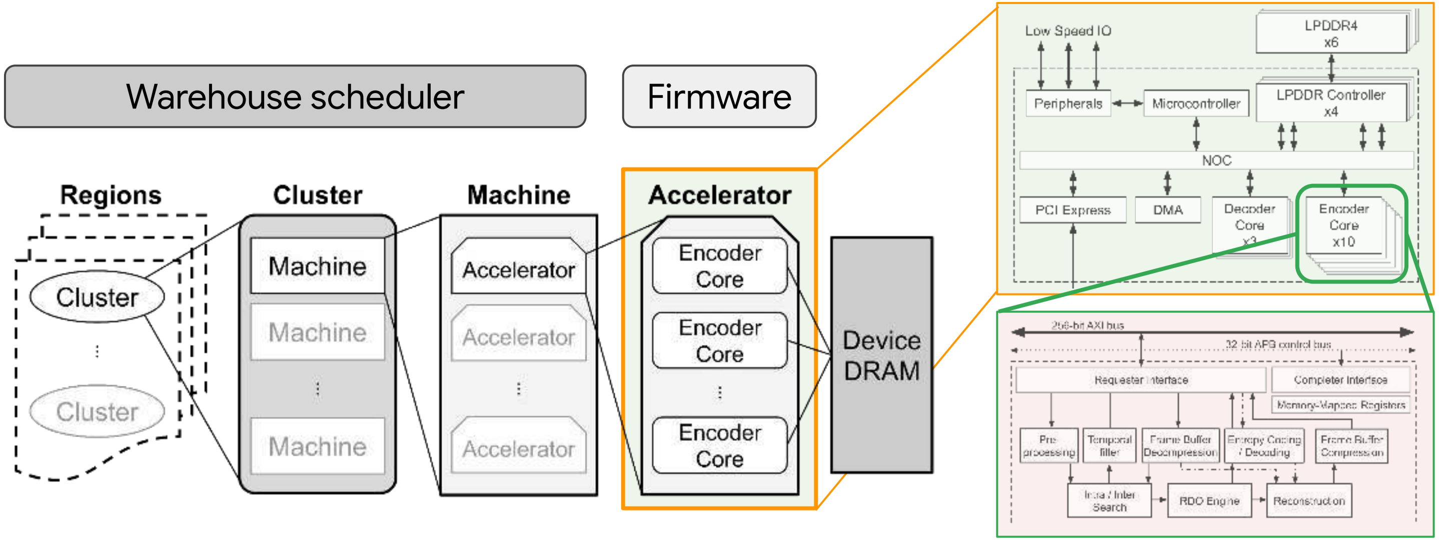


Normalized Throughput vs. Software

- Compared for single-output transcoding at production quality
  - Hardware decoding limits SOT speed
  - MOT speed is 1.2-1.3x faster
- One VCU matches a two-socket Intel Skylake for H.264 speed
  - Uses much less power
  - As fast as five machines on VP9
- For VP9, one 20x VCU machine replaces multiple racks of CPUs

Google

Cluster & Beyond



Warehouse scheduler

Firmware

Regions

Cluster

Cluster

Machine

Machine

Machine

Machine

Accelerator

Accelerator

Accelerator

Accelerator

Encoder Core

Encoder Core

Encoder Core

Device DRAM

Leverage heterogeneous clusters of CPU and VCU machines

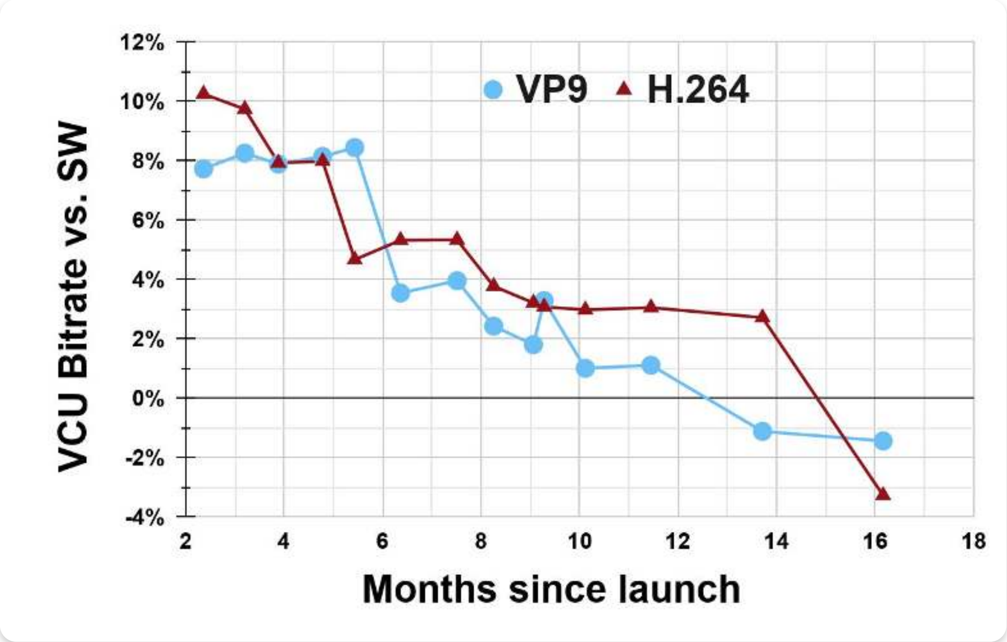
Hardware/Software Co-design

Post-deployment tuning

- Quality improvement over time by parameter and rate-control tuning
  - No required changes to firmware or kernel driver
- Opportunistic software decoding to reduce hardware decode contention

Failure management & recovery

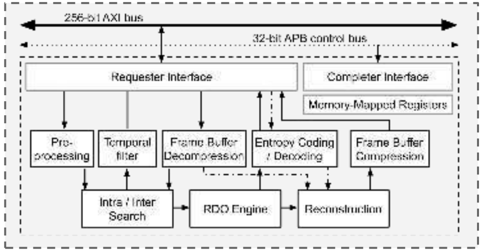
- Most hardware errors do not persist
- Core errors can be retried by software
- Queue errors can be tried at the datacenter level

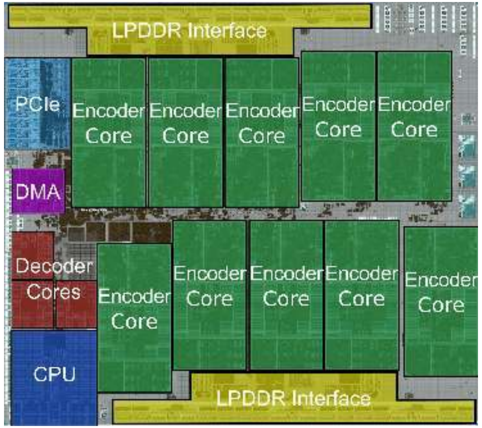


Post-deployment Quality Improvements

Conclusion

- Hardware/software co-design provides many benefits
  - HLS provides software-like velocity during hardware design
  - Design ASIC to maximize utilization while keeping hardware/firmware from getting in the way
- Designing for warehouse scale changes priorities
  - Highly dense servers to maximize cost savings
  - Handle reliability at the cluster-level
- Addressed an unmet need
  - Balance of quality, performance, flexibility, and cost





Google

Acknowledgements: **It takes a village!**