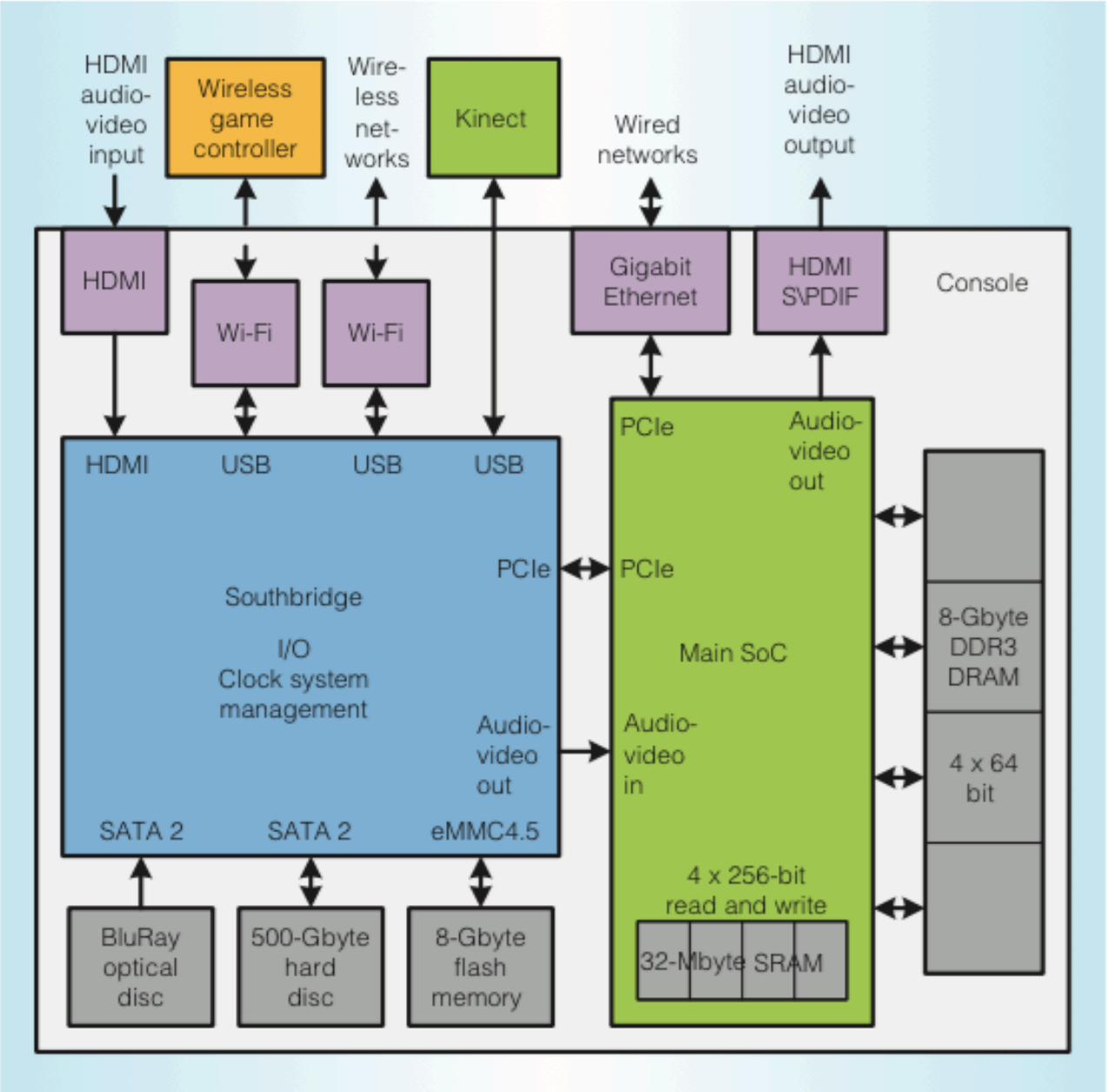
CSEP 548: Computer Systems Architecture

Xbox One, Hololens, Raspberry Pi3

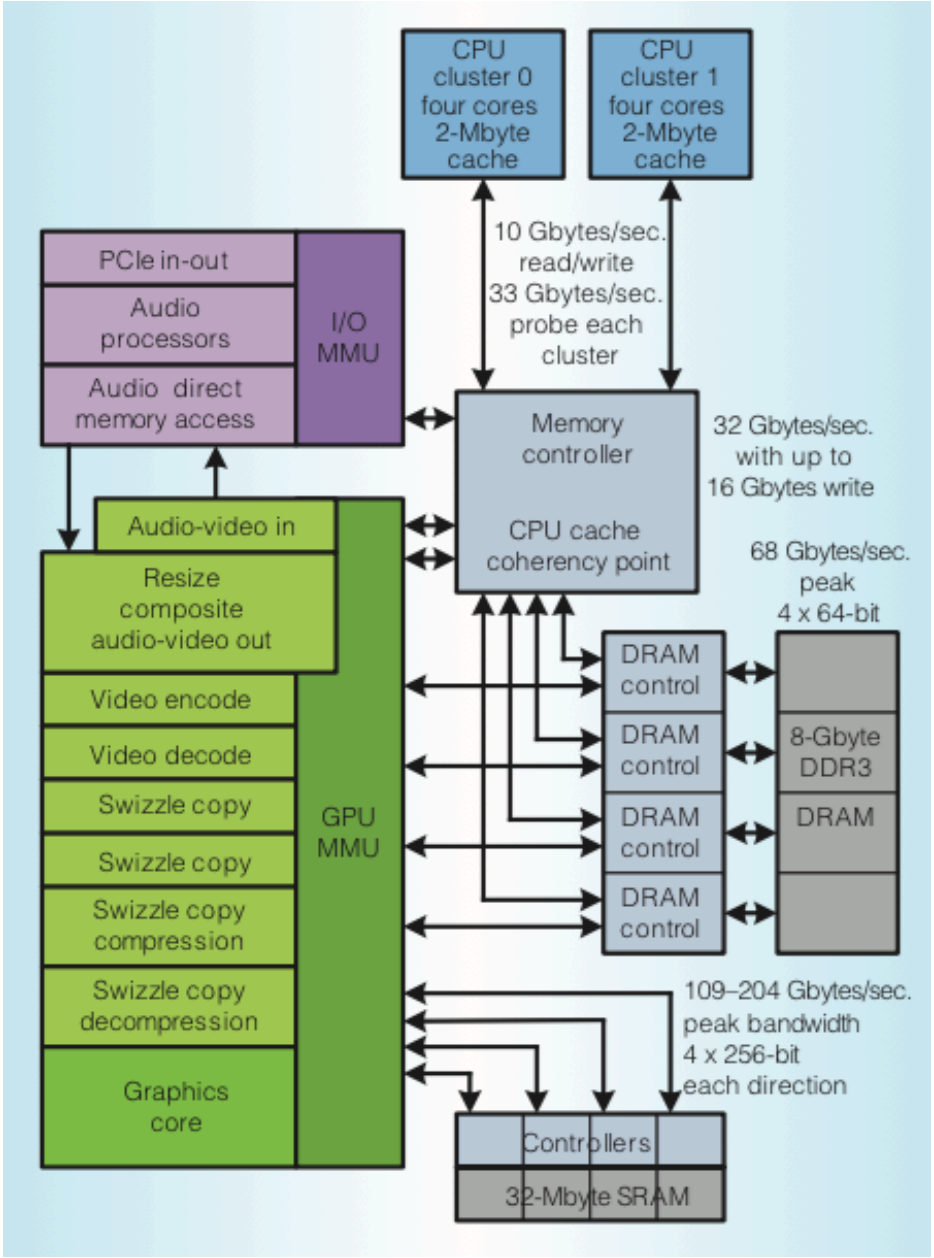
Spring 2017

Luis Ceze (Instructor)

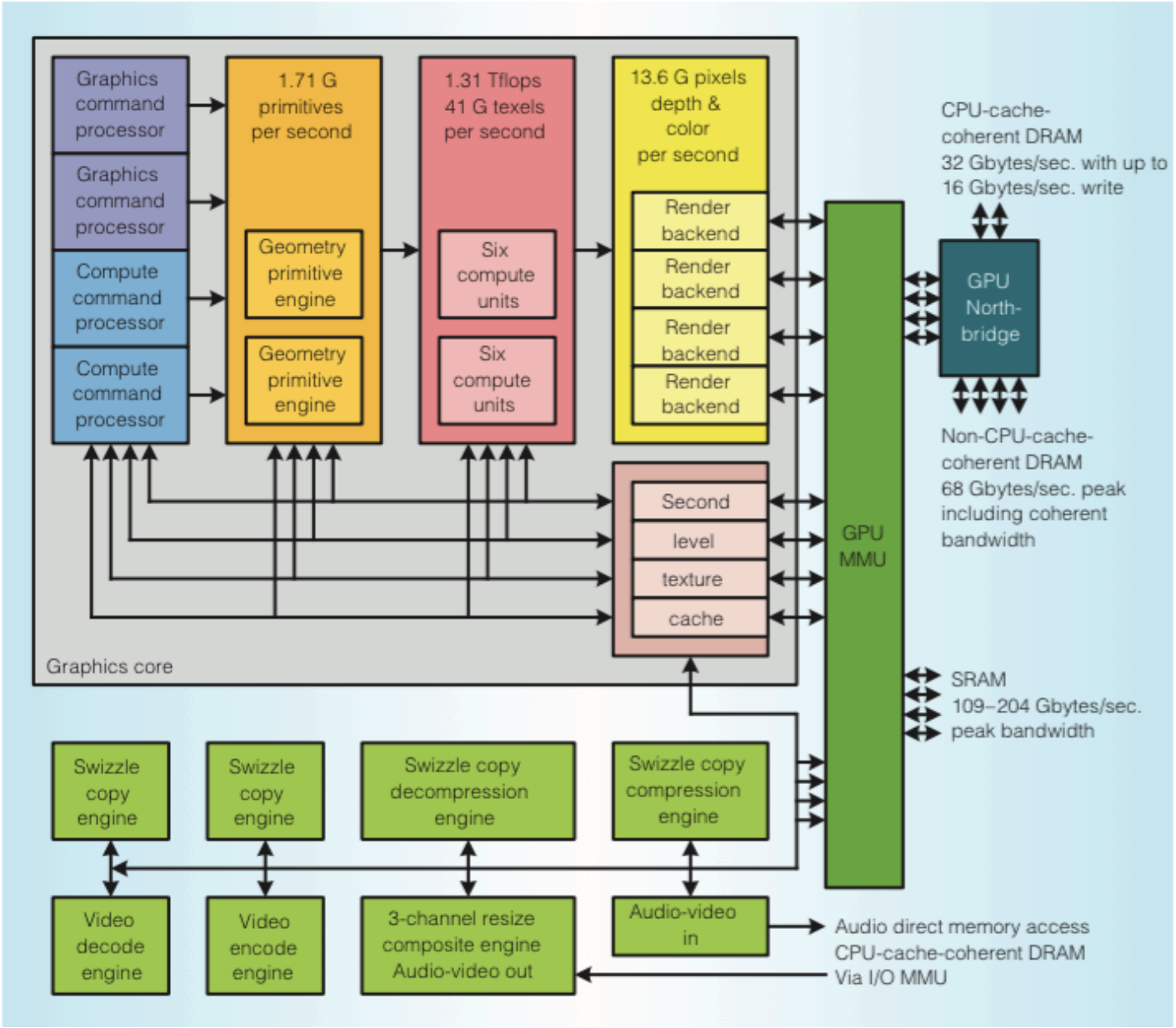
XBox One


HDMI audio-video input
Wireless game controller
Wire-less net-works
Kinect
Wired networks
HDMI audio-video output
HDMI
Wi-Fi
Wi-Fi
Gigabit Ethernet
HDMI S\PDIF
Console
HDMI
USB
USB
USB
PCIe
Audio-video out
PCIe
PCIe
Southbridge
I/O
Clock system management
Main SoC
8-Gbyte DDR3 DRAM
Audio-video out
Audio-video in
4 x 64 bit
SATA 2
SATA 2
eMMC4.5
4 x 256-bit read and write
32-Mbyte SRAM
BluRay optical disc
500-Gbyte hard disc
8-Gbyte flash memory

The XBox-One SoC: 5B transistors!



XBox GPU


Graphics command processor
Graphics command processor
Compute command processor
Compute command processor
1.71 G primitives per second
Geometry primitive engine
Geometry primitive engine
1.31 Tflops 41 G texels per second
Six compute units
Six compute units
13.6 G pixels depth & color per second
Render backend
Render backend
Render backend
Render backend
Second level texture cache
Graphics core
GPU MMU
CPU-cache-coherent DRAM
32 Gbytes/sec. with up to 16 Gbytes/sec. write
GPU North-bridge
Non-CPU-cache-coherent DRAM
68 Gbytes/sec. peak including coherent bandwidth
SRAM
109–204 Gbytes/sec. peak bandwidth
Swizzle copy engine
Swizzle copy engine
Swizzle copy decompression engine
Swizzle copy compression engine
Video decode engine
Video encode engine
3-channel resize composite engine Audio-video out
Audio-video in
Audio direct memory access
CPU-cache-coherent DRAM
Via I/O MMU

Audio and RGBD Camera System

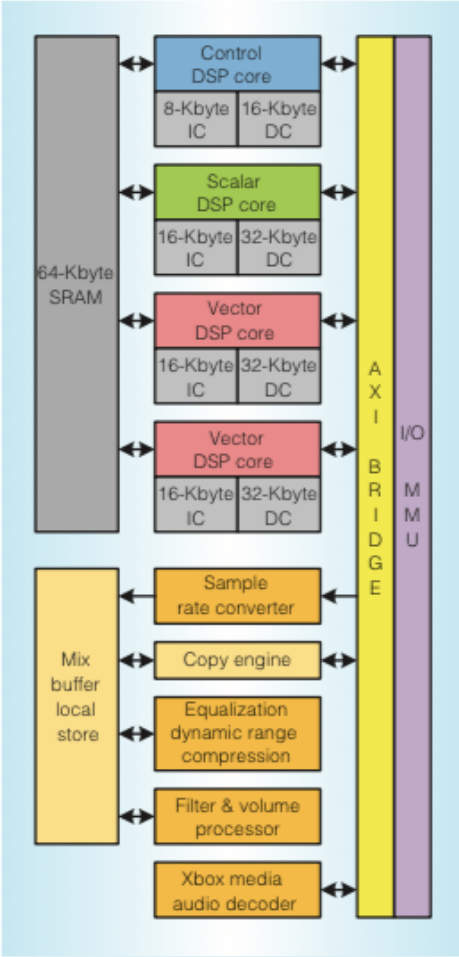
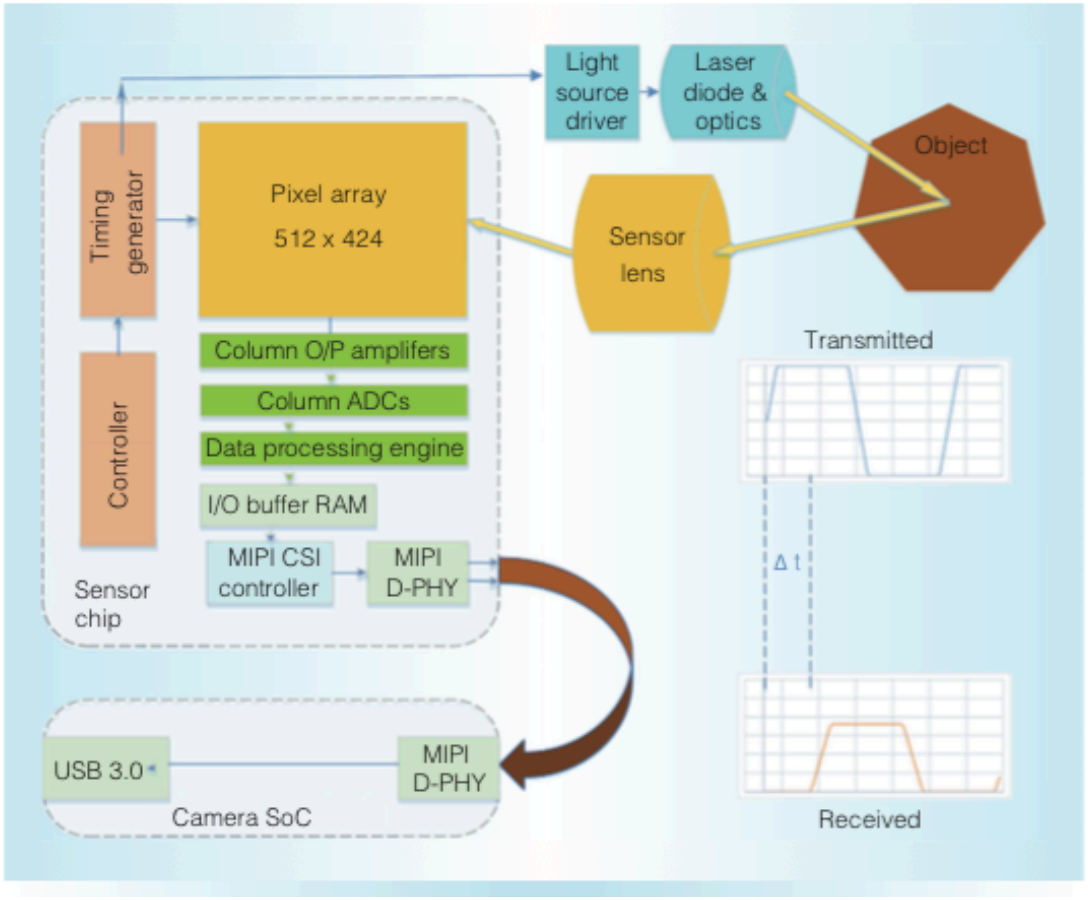
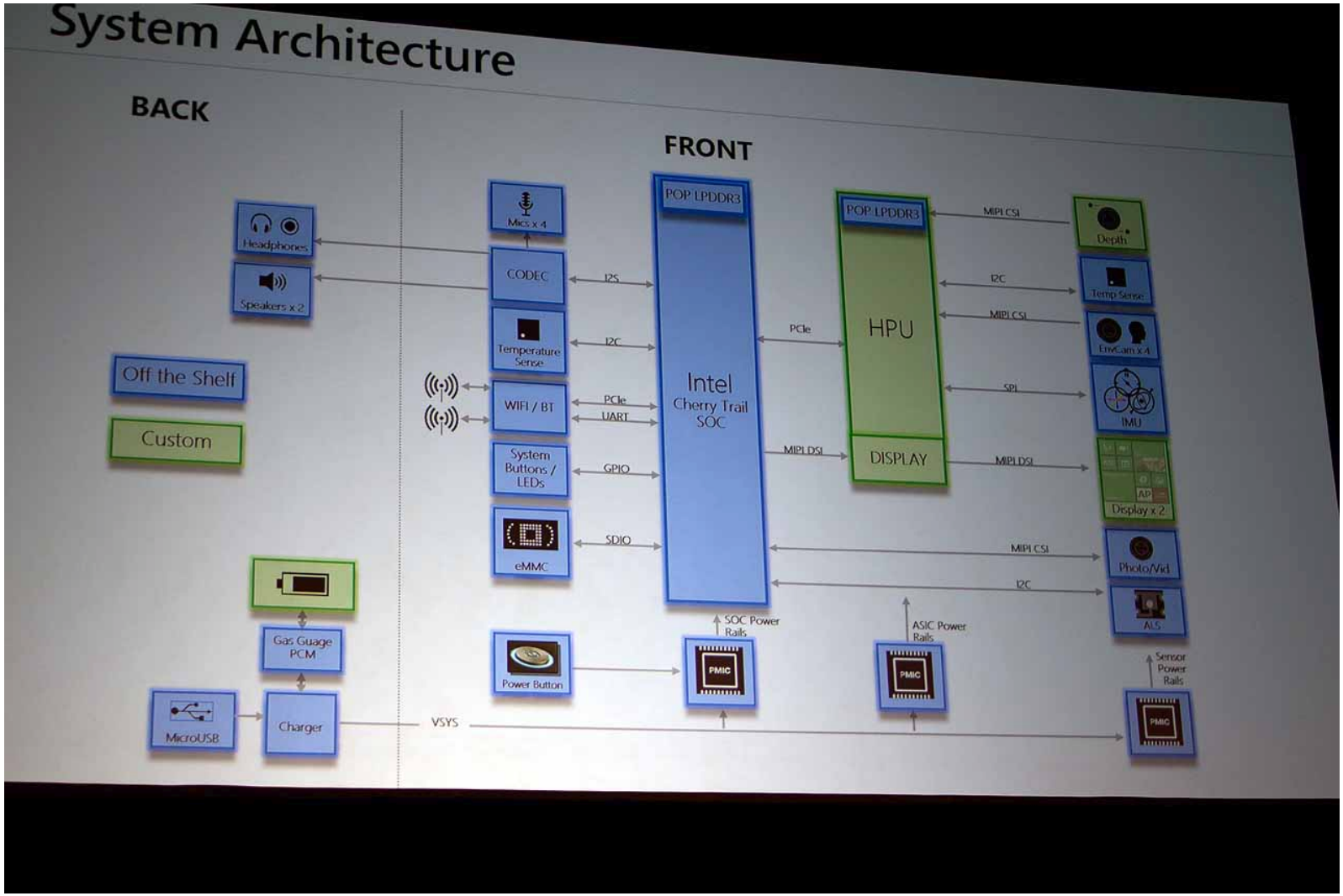


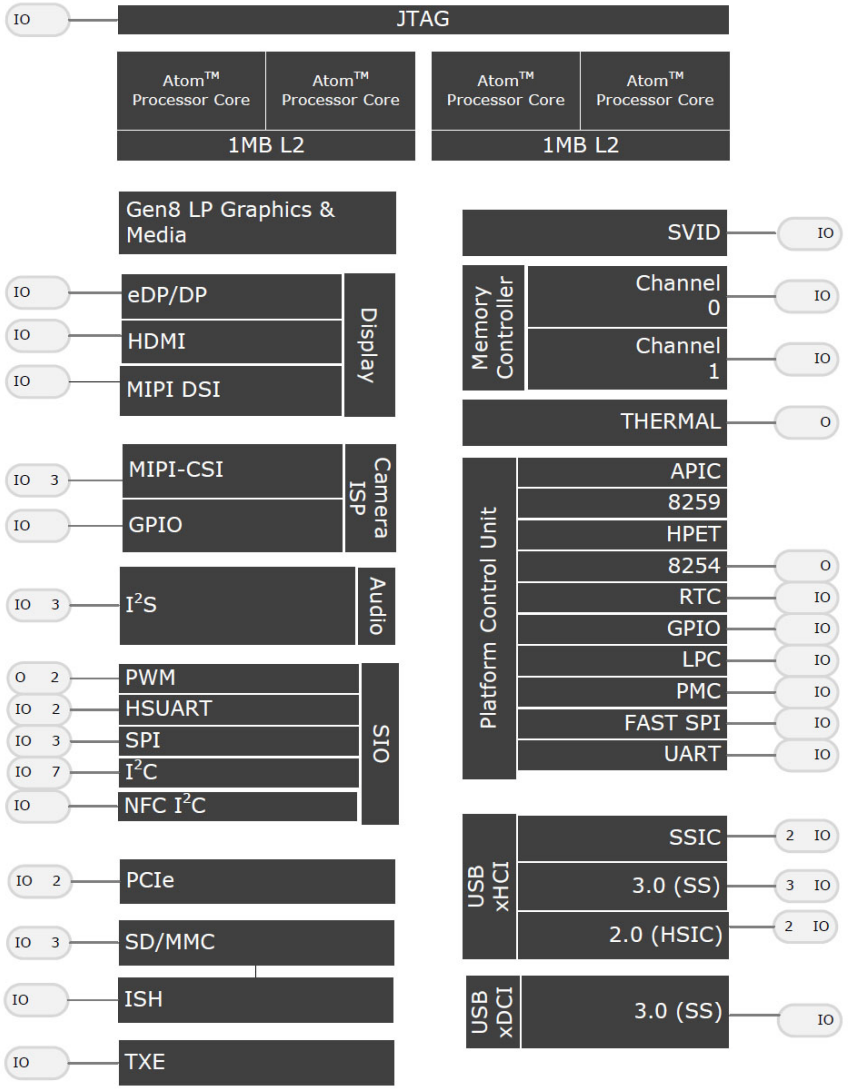
Figure 5. Audio processors. These processors support applications and system services with multiple work queues. Collectively, they are the equivalent of two CPU cores dedicated to audio processing.



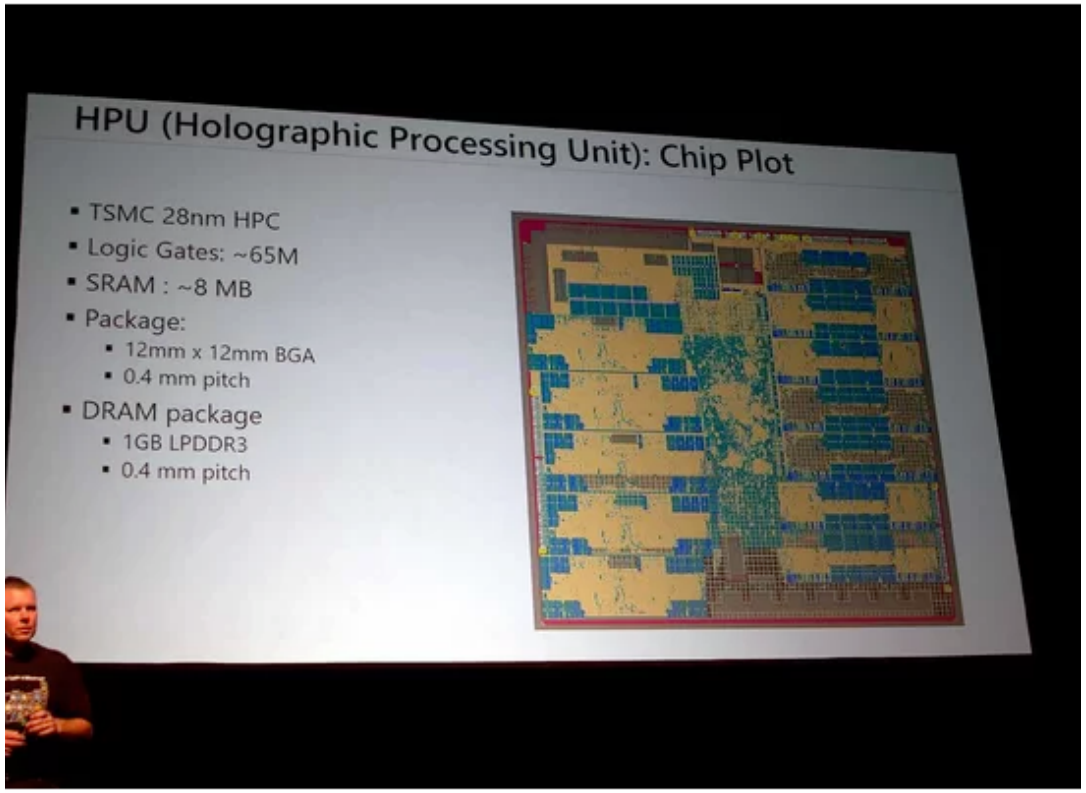
Hololens

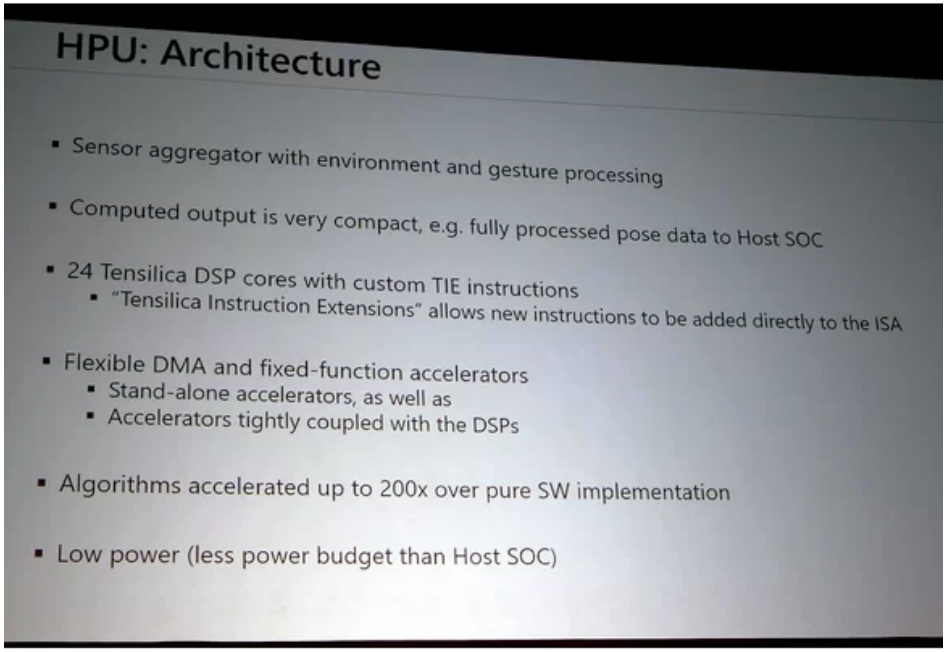

System Architecture
BACK
FRONT
Headphones
Speakers x 2
Off the Shelf
Custom
Mics x 4
CODEC
Temperature Sense
WIFI / BT
System Buttons / LEDs
eMMC
Power Button
Gas Guage PCM
MicroUSB
Charger
VSYS
I2S
I2C
PCIe
UART
GPIO
SDIO
POP LPDDR3
Intel Cherry Trail SOC
SOC Power Rails
PMIC
PCIe
MIPI DSI
POP LPDDR3
HPU
DISPLAY
ASIC Power Rails
PMIC
MIPI CSI
I2C
SPI
MIPI DSI
MIPI CSI
I2C
Depth
Temp Sense
EnvCam x 4
IMU
Display x 2
Photo/Vid
ALS
Sensor Power Rails
PMIC

Hololens SoC

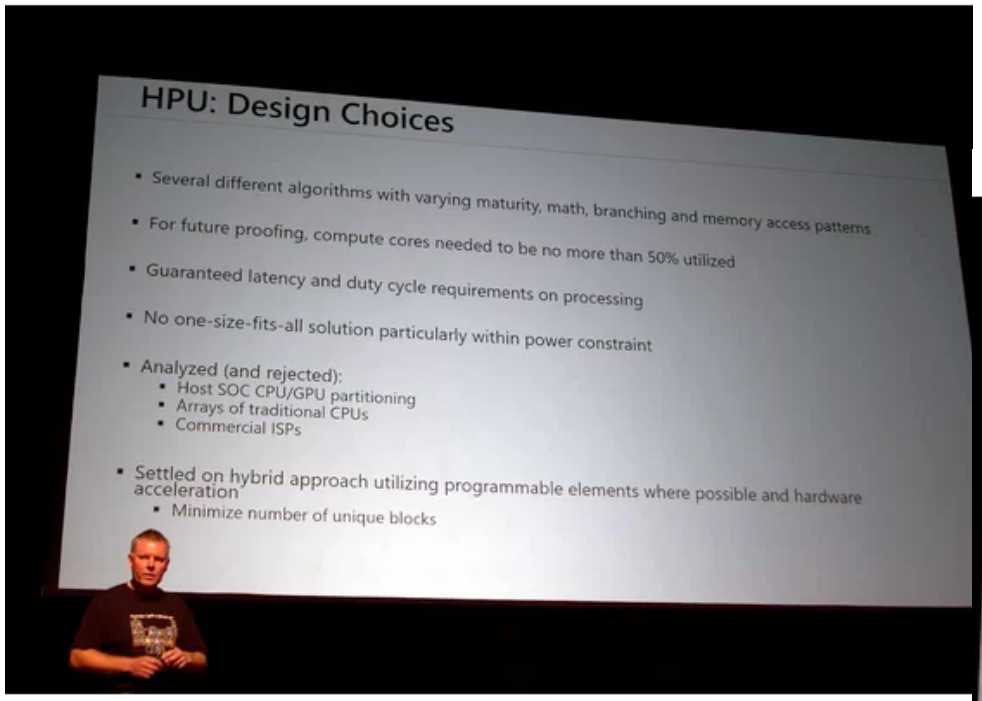


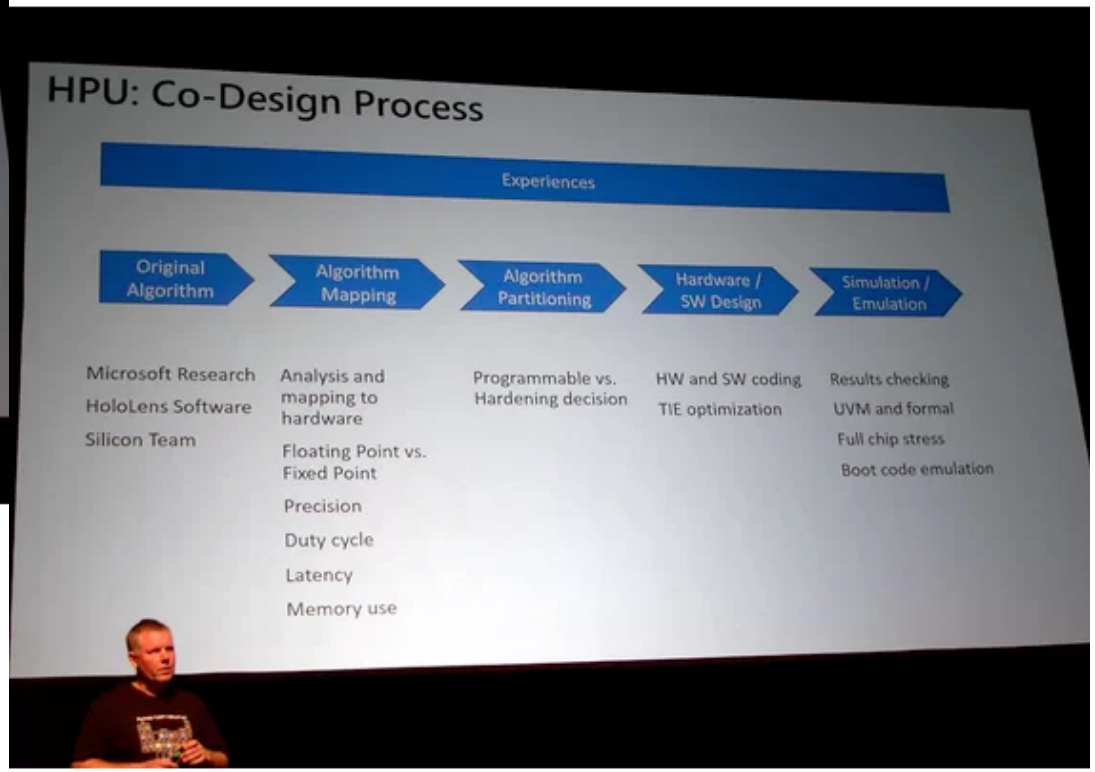
HPU


HPU (Holographic Processing Unit): Chip Plot
TSMC 28nm HPC
Logic Gates: ~65M
SRAM : ~8 MB
Package:
12mm x 12mm BGA
0.4 mm pitch
DRAM package
1GB LPDDR3
0.4 mm pitch

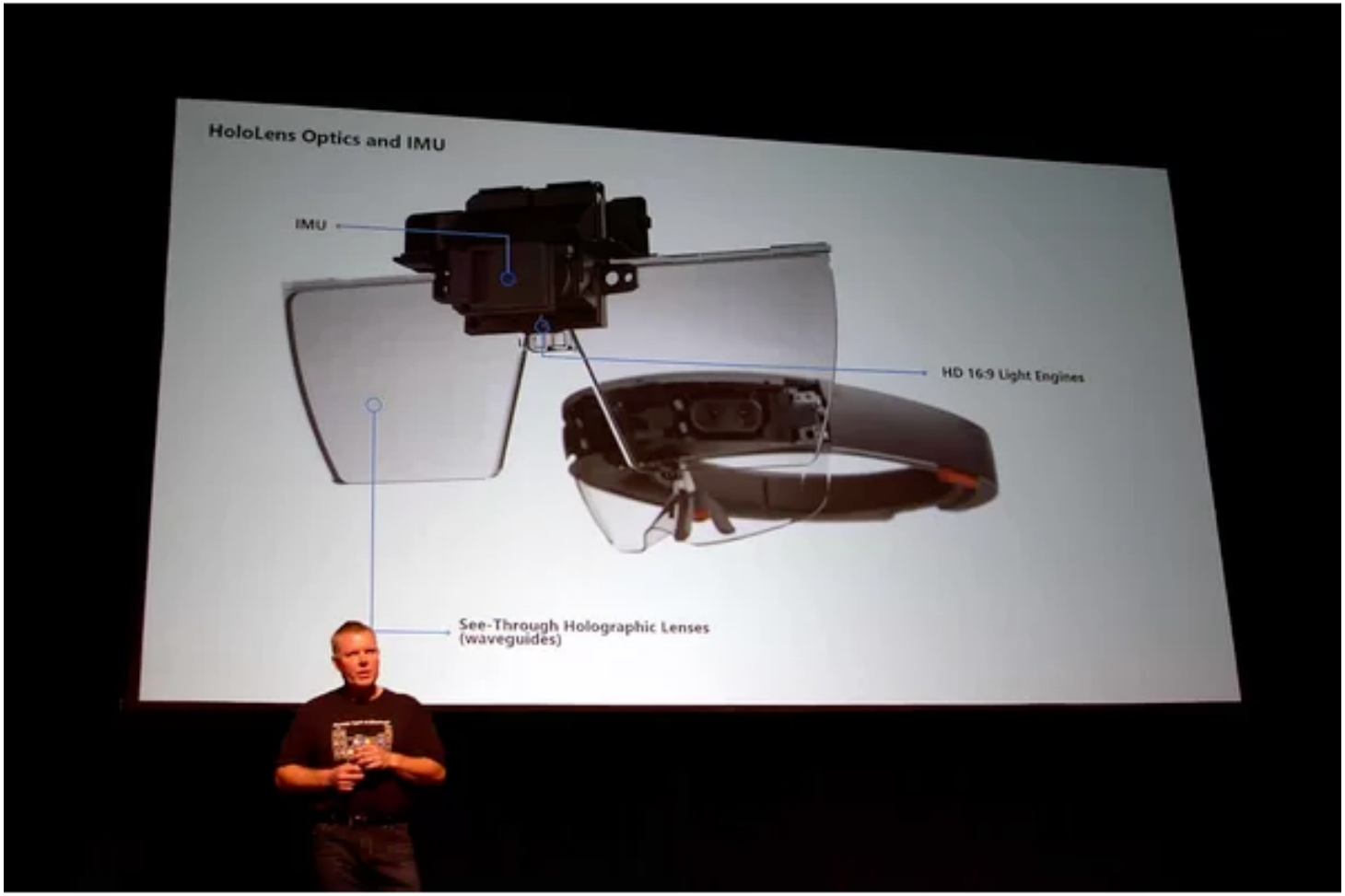

HPU: Architecture
Sensor aggregator with environment and gesture processing
Computed output is very compact, e.g. fully processed pose data to Host SOC
24 Tensilica DSP cores with custom TIE instructions
"Tensilica Instruction Extensions" allows new instructions to be added directly to the ISA
Flexible DMA and fixed-function accelerators
Stand-alone accelerators, as well as
Accelerators tightly coupled with the DSPs
Algorithms accelerated up to 200x over pure SW implementation
Low power (less power budget than Host SOC)

HPU Design

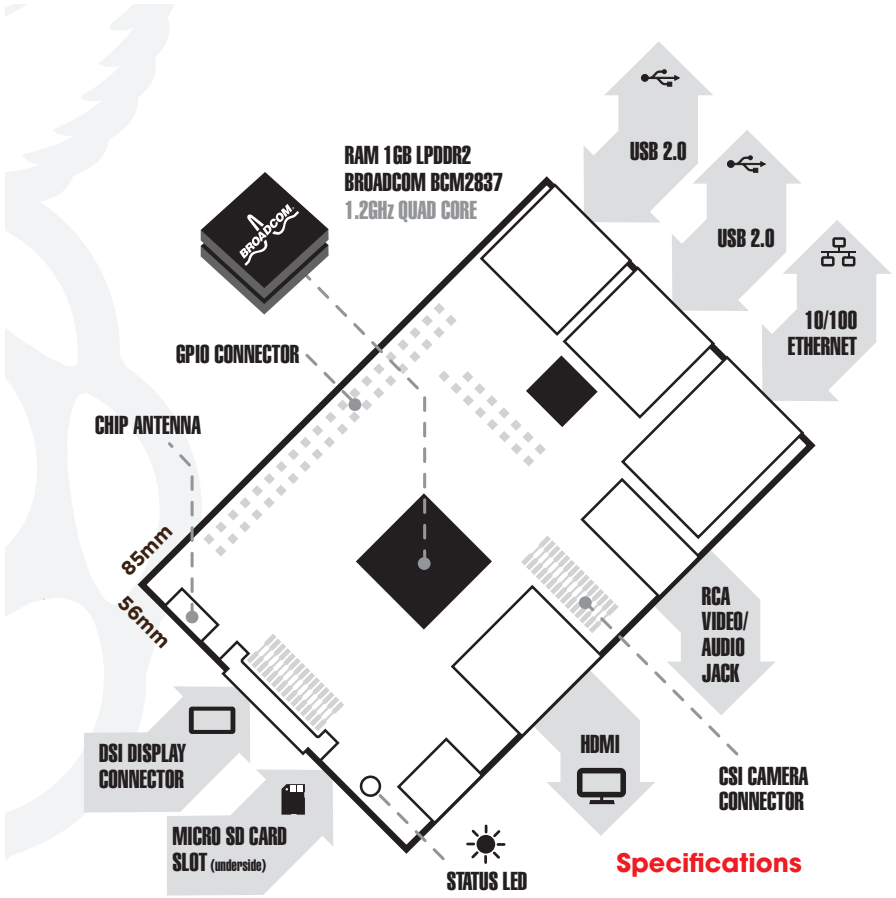

HPU: Design Choices
Several different algorithms with varying maturity, math, branching and memory access patterns
For future proofing, compute cores needed to be no more than 50% utilized
Guaranteed latency and duty cycle requirements on processing
No one-size-fits-all solution particularly within power constraint
Analyzed (and rejected):
Host SOC CPU/GPU partitioning
Arrays of traditional CPUs
Commercial ISPs
Settled on hybrid approach utilizing programmable elements where possible and hardware acceleration
Minimize number of unique blocks


HPU: Co-Design Process
Experiences
Original Algorithm
Algorithm Mapping
Algorithm Partitioning
Hardware / SW Design
Simulation / Emulation
Microsoft Research
HoloLens Software
Silicon Team
Analysis and mapping to hardware
Floating Point vs. Fixed Point
Precision
Duty cycle
Latency
Memory use
Programmable vs. Hardening decision
HW and SW coding
TIE optimization
Results checking
UVM and formal
Full chip stress
Boot code emulation

Display


HoloLens Optics and IMU
IMU
HD 16:9 Light Engines
See-Through Holographic Lenses
(waveguides)

Raspberry Pi 3



Specifications

Processor

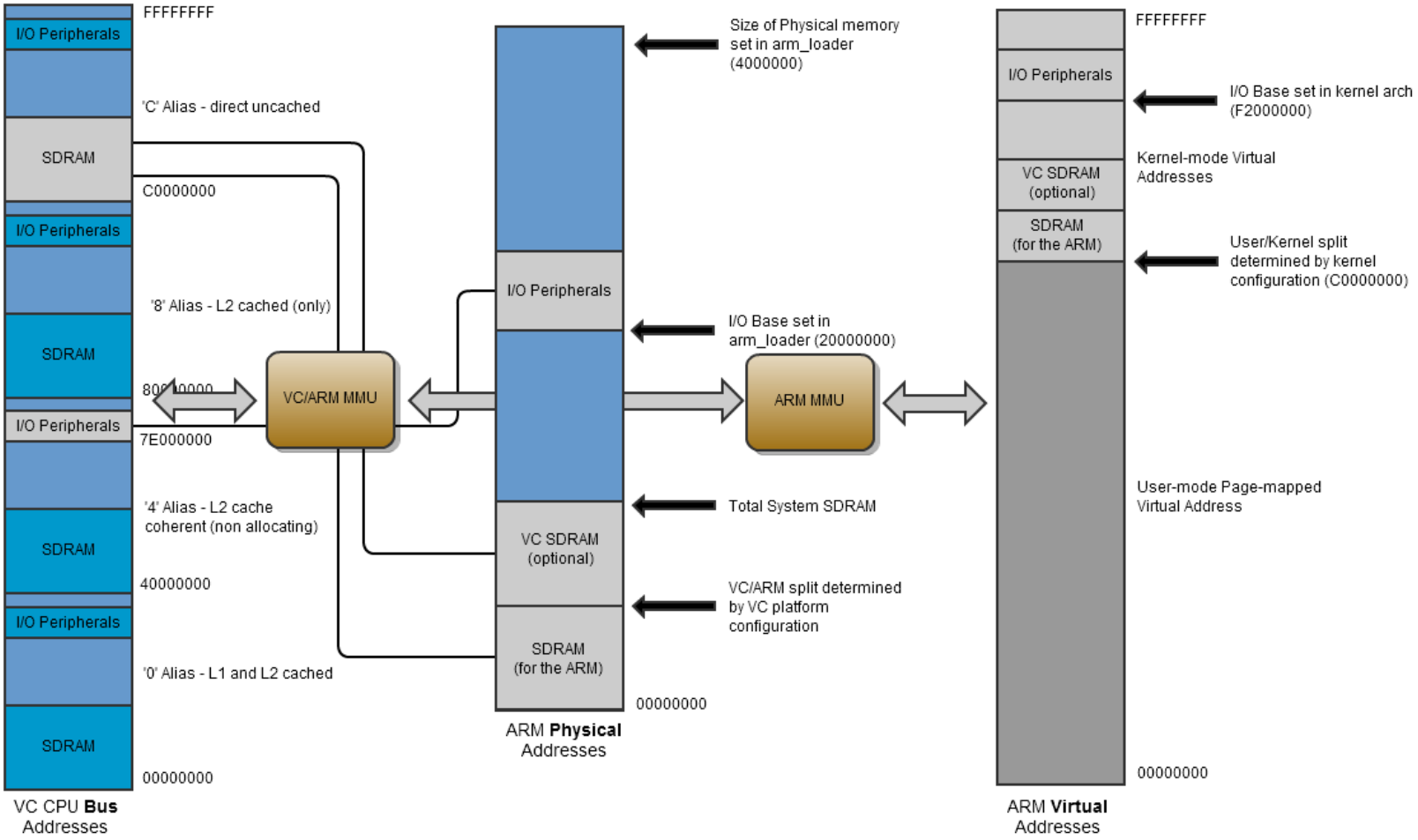
Broadcom BCM2387 chipset.
1.2GHz Quad-Core ARM Cortex-A53
802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)

GPU

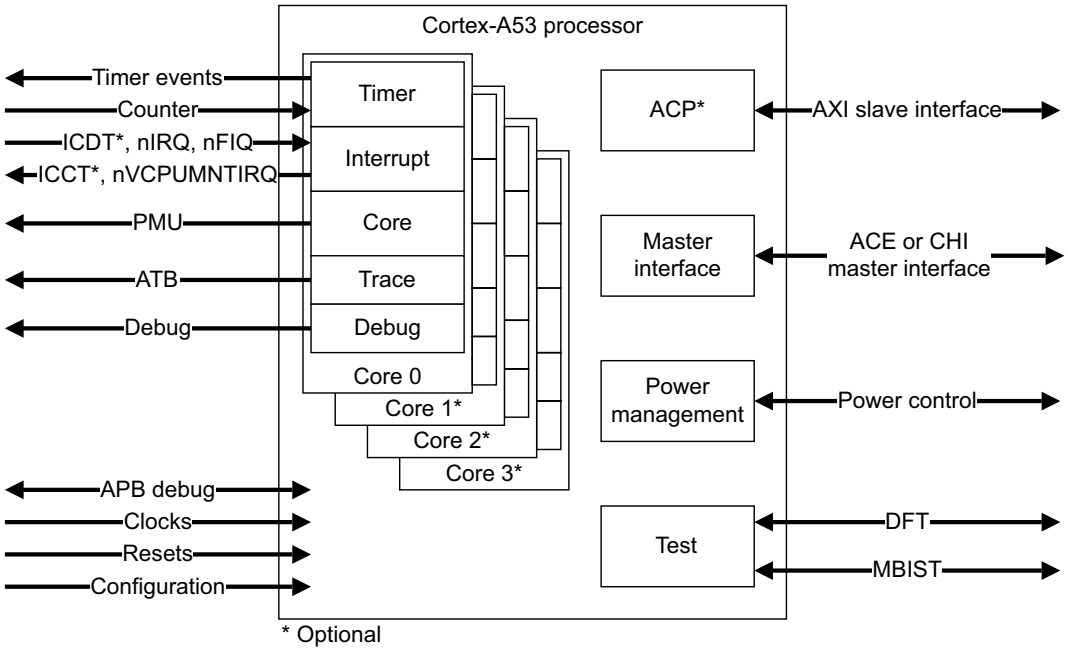
Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode.

Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure

Raspberry Pi SoC



ARM A-53



The Cortex-A53 processor implements the ARMv8-A architecture. This includes:

- Support for both AArch32 and AArch64 Execution states.
- Support for all exception levels, EL0, EL1, EL2, and EL3, in each execution state.
- The A32 instruction set, previously called the ARM instruction set.
- The T32 instruction set, previously called the Thumb instruction set.
- The A64 instruction set.

The Cortex-A53 processor supports the following architecture extensions:

- Optional Advanced SIMD and Floating-point Extension for integer and floating-point vector operations.

  Note

  — The Advanced SIMD architecture, its associated implementations, and supporting software, are commonly referred to as NEON technology.

  — To perform floating-point operations, you must implement the Advanced SIMD and Floating-point Extension. There is no software API library for floating-point in the ARMv8-A architecture.

  — You cannot implement floating-point without Advanced SIMD.

- Optional ARMv8 Cryptography Extensions.

  Note

  You cannot implement the Cryptography Extensions without Advanced SIMD and Floating-point.

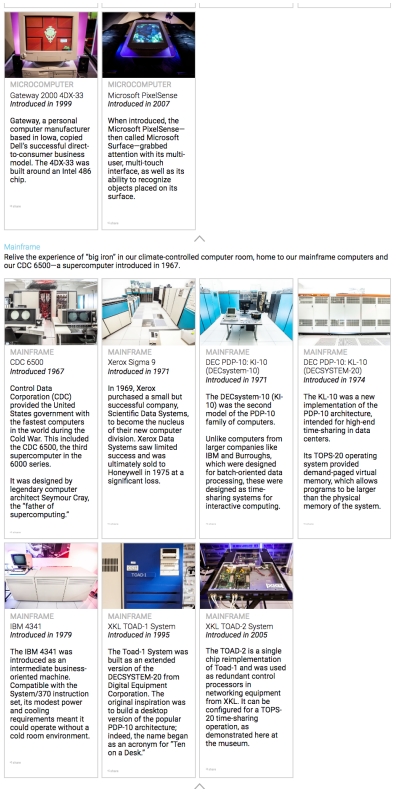
The Cortex-A53 processor includes the following features:

- Full implementation of the ARMv8-A architecture instruction set with the architecture options listed in *ARM architecture* on page 1-3.
- In-order pipeline with symmetric dual-issue of most instructions.
- Harvard *Level 1* (L1) memory system with a *Memory Management Unit* (MMU).
- *Level 2* (L2) memory system providing cluster memory coherency, optionally including an L2 cache.

Living Computer Museum

Vintage Computers

Microcomputer Minicomputer
Mainframe



Stored program
Caches
Multithreading
GUI/GPU
Separate I/D cache
Dynamic scheduling etc…
NVRAM (!)
Cooling?

Pick a machine, do a bit of research on key architecture/business innovation(s). Write and/or run a piece of code. Take picture of output ☺.