Graphics Processing Units

- Graphics Processing Units
  - GPU Overview
  - GPU Pipeline
  - Data-Parallel Architecture
  - GPGPU Computing
  - CUDA

GPGPU Computing (1)

- GPGPU – *General-Purpose Computing on Graphics Processing Unit*
  - Use of GPUs for computations commonly performed by CPUs
  - The shader cores of a GPU provide massive FP computational power
    - Example: an NVIDIA GB202 GPU (24,576 cores) achieves a performance of 104.8 TFLOPS (FP32)
  - The GPU's graphics pipelines can also be used for general-purpose applications
    - Performance can be orders of magnitude higher

GPGPU Computing (2)

- GPGPU started after the release of GPUs with programmable shader cores
  - Pixel shader: uses the position of a pixel and other information (e.g., input colors, texture coordinates) to compute a final color
  - Developers of scientific applications noticed that any data could be used as inputs
  - The final pixel color could be read back and used by the developers instead of being used for displaying an image

GPGPU Computing (3)

- Graphics APIs and GPUs impose constraints
  - Resource constraints: programs can receive input data only from a few input colors
  - Writing into memory: writing into arbitrary locations was not allowed by previous GPUs
  - Developers needed to be familiar with graphics APIs and the GPU architecture
  - Problems needed to be expressed in terms of coordinates, textures, and shader functions
  - Code had to be written in a special shading language (e.g., Cg, GLSL, HLSL)

GPGPU Computing (4)

- Programming languages developed for GPGPU computing
  - Brook: extension of the C language for stream processing (stream: array of records)
    - Abstracts GPUs as stream coprocessors
    - The GPU program is a collection of *kernels*
    - Kernel: function applied to each element in a set of input streams → produces output streams
  - Accelerator: based on the C# language
    - Provides a library that evaluates data-parallel operations using a GPU

GPGPU Computing (5)

- Parallel programming platforms developed for GPGPU computing
  - CUDA (*Compute Unified Device Architecture*)
    - Platform developed by NVIDIA
    - Enables to use high-level programming languages for applications accelerated by NVIDIA GPUs
  - OpenCL (*Open Computing Language*)
    - Framework developed by the Khronos Group
    - Enables to develop applications executed across heterogeneous systems: CPUs, GPUs, DSPs, and FPGAs

GPGPU Computing (6)

- Libraries developed for GPGPU computing
  - Enable to accelerate existing applications with minimal code changes
  - cuDNN (*CUDA Deep Neural Network*)
    - Contains primitives for deep neural networks
    - Used to accelerate deep learning frameworks (e.g., TensorFlow, PyTorch)
  - nvGRAPH (*NVIDIA Graph Analytics*)
    - Implements parallel algorithms for analytics applications (e.g., *Page Rank* algorithm)

GPGPU Computing (7)

- NVIDIA Video Codec SDK
  - Includes APIs for video encode and video decode acceleration
  - Use the NVENC and NVDEC hardware engines
- OpenCV (*Open Source Computer Vision Library*)
  - Library for computer vision, image processing
- FFmpeg
  - Open-source multimedia framework
  - Functions: decode, encode, transcode, multiplex, stream, filter, play various multimedia formats

Graphics Processing Units

- Graphics Processing Units
  - GPU Overview
  - GPU Pipeline
  - Data-Parallel Architecture
  - GPGPU Computing
  - CUDA

CUDA (1)

- Compute Unified Device Architecture
  - Parallel computing platform and programming model for NVIDIA GPUs
  - Enables to use the C, C++, and FORTRAN languages, the Java and Python languages (via wrappers), and the DirectCompute API
  - Allows to access directly the GPU resources for general-purpose computing
  - Exploits the GPU's capability to operate on large matrices in parallel
  - Current version: 12.9.0 (May 2025)

CUDA (2)

- Extends the C language with new features
  - New type qualifiers and API function calls
  - Specific C functions called kernels
    - Executed *N* times in parallel by *N* CUDA threads
    - At kernel definition, its execution configuration must also be specified
    - Each thread that executes the kernel is assigned a unique thread ID
- Threads are organized into a thread block
  - Each thread within a block executes one instance of the kernel

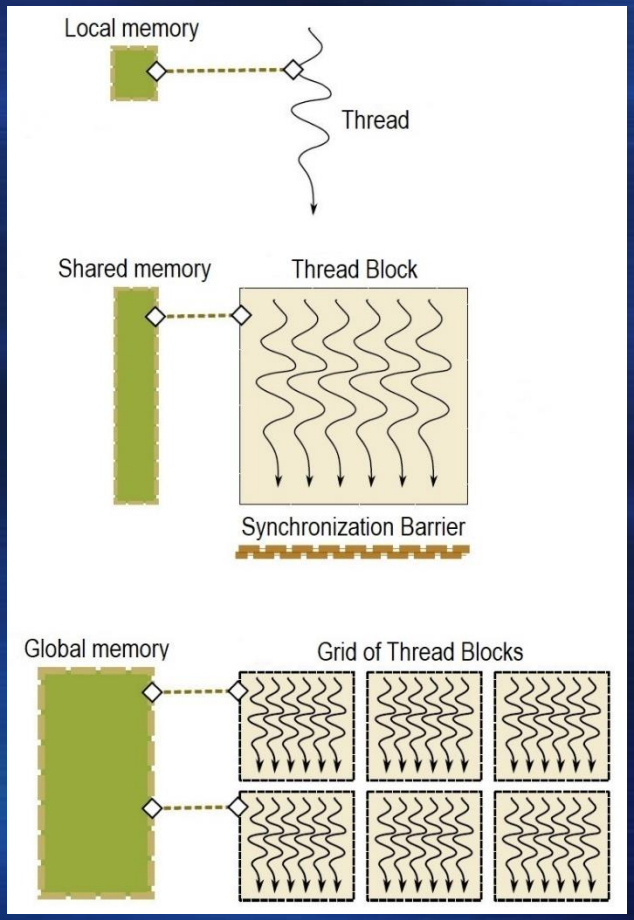
CUDA (3)

- Threads within a thread block can be organized into a 1D, 2D, or 3D structure
- The number of threads per block is limited
- A kernel can be executed by multiple blocks
- Thread blocks are required to execute independently, and in any order
- Threads of a block communicate via a shared memory and synchronization primitives
  - Each acts as a barrier at which all threads in a block must wait

CUDA (4)

- Thread blocks are organized into a grid of thread blocks
  - The execution configuration of a kernel includes the number of blocks per grid
  - Threads of a grid access a global memory
- Execution of the hierarchy of threads
  - Thread: executed on a CUDA core
  - Thread block: executed on a Streaming Multiprocessor (SM)
  - Grid of thread blocks: on an array of SMs

CUDA (5)



CUDA memory hierarchy

CUDA (6)

- Unified Memory
  - Part of the GPU memory is shared between CPUs and GPUs → single-pointer model
  - Memory is managed by the system, without the need for explicit memory copy calls
  - CUDA software migrates data allocated in the unified memory between GPUs and CPUs
  - The memory modified by a CPU should be synchronized with the GPU memory
  - Advantage: GPU programming is simplified

CUDA (7)

- Thread scheduling and execution
  - A CUDA SM uses a SIMT architecture
  - Threads are created, scheduled, and executed in groups of 32 threads → warps
  - Threads within a warp start execution at the same program address
    - They have their own program counter and state
  - Maximum efficiency is achieved if all threads of a warp agree on their execution paths
    - Branch divergence: the warp executes the branch path taken and disables threads that diverge

6. Graphics Adapters

- Structure of a Graphics Adapter
- Graphics Memory
- Graphics Processing Units
- Display Interfaces

Display Interfaces

- Display Interfaces
  - HDMI
  - DisplayPort

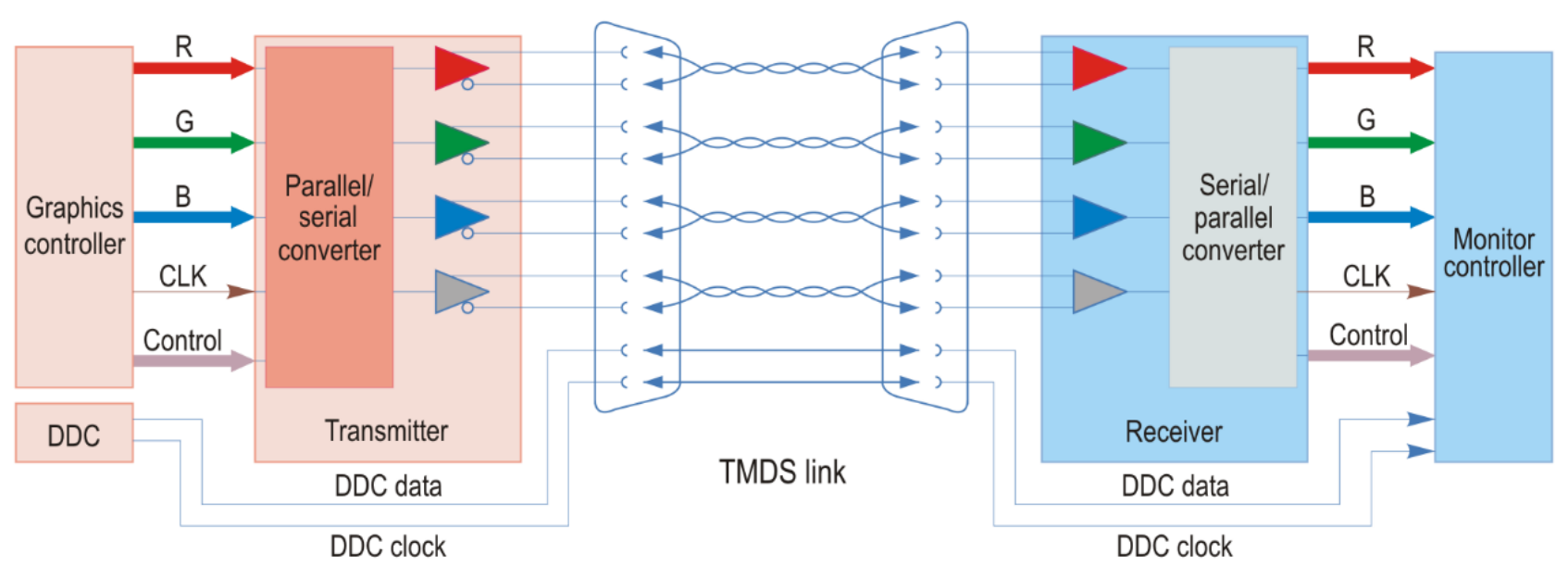
HDMI (1)

- HDMI – *High-Definition Multimedia Interface*
- Audio/video interface for uncompressed digital data
  - For connecting A/V sources to computer monitors, digital TVs, digital audio devices
- Enables to send on a single cable:
  - Video data in various formats
  - Up to 8 digital audio data streams
  - Auxiliary data and control information

HDMI (2)

- Uses the TMDS (*Transition Minimized Differential Signaling*) protocol
  - Developed by Silicon Image
  - Minimizes the number of 0-to-1 and 1-to-0 transitions for the signals
  - 8b/10b encoding (early versions)
  - Differential signaling is used on pairs of twisted wires
  - TMDS link: consists of a TMDS transmitter and a TMDS receiver

HDMI (3)


R
G
B
CLK
Control
Graphics controller
DDC
Parallel/ serial converter
Transmitter
DDC data
DDC clock
TMDS link
Serial/ parallel converter
Receiver
DDC data
DDC clock
R
G
B
CLK
Control
Monitor controller

HDMI (4)

- Contains three identical encoders
- The inputs of each encoder are 8 bits of pixel data, control bits, and auxiliary data
- In each clock cycle, the encoder generates a 10-bit character:
  - From 8 bits of video data, or
  - From 2 bits of control data, or
  - From 4 bits of auxiliary data
- Output of each encoder: a continuous stream of serialized TMDS characters

HDMI (5)

- Contains signals for a *Display Data Channel* (DDC) between the display and computer
  - Implemented with the ACCESS.bus serial bus (based on I^2C)
  - DDC2 provides bidirectional communication between the display and computer
  - Allows for automatic system configuration
  - Format of configuration data: defined by the EDID (*Extended Display Identification Data*) standard → EDID EPROM

HDMI (6)

- Version 1.0 (2002)
  - Maximum bandwidth of 4.95 Gbits/s (165 MHz) → resolution of 1920×1200 (WUXGA) at 60 Hz
- Version 1.1 (2004)
  - Supports the DVD Audio format
- Version 1.2 (2005)
  - Supports the SACD (*Super Audio CD*) format
  - Allows PC applications to only support the RGB color space

HDMI (7)

- Version 1.3 (2006)
  - Bandwidth of 10.2 Gbits/s (340 MHz) → resolution of 2560×1600 (WQXGA) at 60 Hz
  - Supports video images with more colors: 30, 36, or 48 bits/pixel (*Deep Color*, optional)
  - Supports the Dolby TrueHD and DTS-HD Master Audio formats (optional)
  - Two types of cables:
    - Category 1: up to 74.25 MHz (720p or 1080i)
    - Category 2: up to 340 MHz (1080p or more)
  - A smaller connector: Type C

HDMI (8)

- Version 1.4 (2009)
  - Same bandwidth
  - Resolutions of 4K×2K: 3840×2160p (Quad HD) at 24, 25, or 30 Hz; 4096×2160p at 24 Hz
  - HDMI Ethernet channel (100 Mbits/s)
  - Audio return channel (ARC)
  - Stereoscopic 3D formats
  - Micro HDMI connector (Type D)
  - Automotive connection system

HDMI (9)

- Version 1.4a (2010)
  - Specifies two new mandatory 3D formats
- Version 1.4b (2011)
  - Support for resolution of 1920×1080p, 120 Hz
- The HDMI Forum has been created in 2011
- Version 2.0 (2013)
  - Bandwidth has been increased to 6 Gbits/s per channel → 18 Gbits/s
  - Resolutions of 4K×2K at 60 Hz

HDMI (10)

- HDMI alternate mode for the USB Type-C connector (2016)
  - Enables to use a cable with a USB Type-C connector for connecting to a display with the HDMI interface
  - The alternate mode is compatible with HDMI version 1.4b: resolutions up to 4K×2K; ARC; Ethernet channel; 3D formats
  - Cable: USB Type-C (source) – HDMI (display)
  - Simple connection for personal computers

HDMI (11)

- Version 2.1 (2017)
  - Bandwidth increased up to 48 Gbits/s
    - *Ultra High Speed* HDMI cable → very low electro-magnetic interferences
  - Resolutions: 4K×2K (120 Hz); 8K×4K(60 Hz)
  - Resolution of 10K (10240×4320)
  - Variable refresh rate
  - *Quick Frame Transport* feature: reduces latency of frames → interactive virtual reality
- Versions: 2.1a (2022), 2.1b (2023), 2.2 (2025?)

HDMI (12)

- HDMI connections
  - Single-link
  - Dual-link: enables to double the pixel rate
- Audio formats
  - Uncompressed audio: PCM (*Pulse Code Modulation*)
  - Sampling rates: 32; 44.1; 48; 96; 192 KHz
  - Sample sizes: 16, 20, or 24 bits
  - Compressed audio: Dolby Digital, DTS
  - Lossless compressed audio: Dolby TrueHD, DTS-HD Master Audio

HDMI (13)

- Video formats
  - Color encodings: RGB, YC_BC_R, xvYCC (optional)
  - YC_BC_R: Y → luminance and synchronization; C_B and C_R → chroma (C_B = B – Y, C_R = R – Y)
  - xvYCC: chroma values may correspond to negative RGB values → more saturated colors
  - *Deep Color* option: 10 bits, 12 bits, or 16 bits per color component
    - 12 bits per color component: 68.7 billion colors

HDMI (14)

- CEC (*Consumer Electronics Control*)
  - One-wire bidirectional serial bus used to transfer remote control commands
    - *One Touch Play*, *System Standby*, *Tuner Control*
  - The user can control several devices connected through HDMI with a single remote control
  - Devices can command each other without user intervention
  - Alternative names: Anynet+ (Samsung), BRAVIA Link (Sony), EasyLink (Philips)

HDMI (15)

- Connectors
  - Type A: 19 pins, single-link connection
  - Type B: 29 pins, dual-link connection
  - Type C: mini-connector, 19 pins; can be connected to a Type A connector
  - Type D: micro-connector, 19 pins (similar to micro-USB)
  - Type E: for automobiles

Display Interfaces

- Display Interfaces
  - HDMI
  - DisplayPort

DisplayPort

- DisplayPort
  - DisplayPort Overview
  - DisplayPort Architecture
  - Embedded DisplayPort (eDP)

DisplayPort Overview (1)

- Developed by VESA (*Video Electronics Standards Association*)
- Intended to replace the DVI, VGA interfaces and the LVDS (*Low-Voltage Differential Signaling*) protocol
- DisplayPort and HDMI interfaces may coexist
- Versions of DisplayPort specifications
  - Version 1.0 (2006), 1.2 (2009)
  - Version 1.3 (2014)
  - Version 1.4 (2016)
  - Version 2.0 (2019)
  - Version 2.1 (2022), 2.1a (2024), 2.1b (2025)

DisplayPort Overview (2)

- Main link (unidirectional)
  - 1, 2, or 4 lanes
  - The transmission protocol is based on micro packets → pixel and audio data
  - 8b/10b encoding → the synchronization information is embedded into the data stream
- Auxiliary (AUX) channel (bidirectional)
  - For device control and auxiliary data
  - Default (standard) mode: Manchester encoding
  - Fast mode: 8b/10b encoding

DisplayPort Overview (3)

- Allows external and internal connections
  - For internal connections of portable computers: Embedded DisplayPort (eDP)
- Copper or fiber optic cables
- DisplayPort signals are not compatible with HDMI signals
  - Optional dual-mode: HDMI signals can be generated with a simple converter
  - The main link and AUX channel transmit 3 TMDS signals, a clock signal, and DDC data

DisplayPort Overview (4)

- Video data
  - 18, 24, 30, 36, or 48 bits per pixel (bpp)
  - High resolutions, refresh rates, and color depths
- Audio data
  - 1..8 channels, uncompressed data
  - Sampling rates: 48; 96; 192 KHz
  - Sample sizes: 16 or 24 bits
  - Maximum bit rate: 6.144 Mbits/s

DisplayPort Overview (5)

- Improvements in version 1.2
  - The bandwidth is doubled to 5.4 Gbits/s per lane $\rightarrow$ higher resolutions
  - Multiple independent audio/video streams
    - Up to 63 A/V streams across a single connection
  - Higher speed of the auxiliary channel
    - USB peripherals, video cameras, touch panel data
  - Support for stereoscopic 3D images
  - Addition of the Mini DisplayPort connector (Apple)

DisplayPort Overview (6)

- Improvements in versions 1.3, 1.4
  - Bandwidth has been increased to 8.1 Gbits/s per lane (with 4 lanes: 32.4 Gbits/s)
  - Example configurations:
    - Two 4K (4096×2160) displays, 60 Hz
    - One 4K stereo 3D display
    - Combination of one 4K display and USB 3.0
    - One 5K (5120×2880) display in RGB mode
    - One 8K (7680×4320) TV set, 30 Hz

DisplayPort Overview (7)

- Improvements in version 2.0
  - Bandwidth: up to 20 Gbits/s per lane
    - Effective bandwidth with 4 lanes: 77.4 Gbits/s
  - Encoding: 128b/132b
  - Thunderbolt 3 physical interface (Intel)
  - Display Stream Compression (DSC)
  - Example configurations:
    - One 16K (15360×8460) display, 60 Hz (with DSC)
    - One 10K (10240×4320) display, 60 Hz (no compress.)
    - Two 8K (7680×4320) displays, 120 Hz (with DSC)
    - Three 4K (3840×2160) displays, 90 Hz (no compress.)

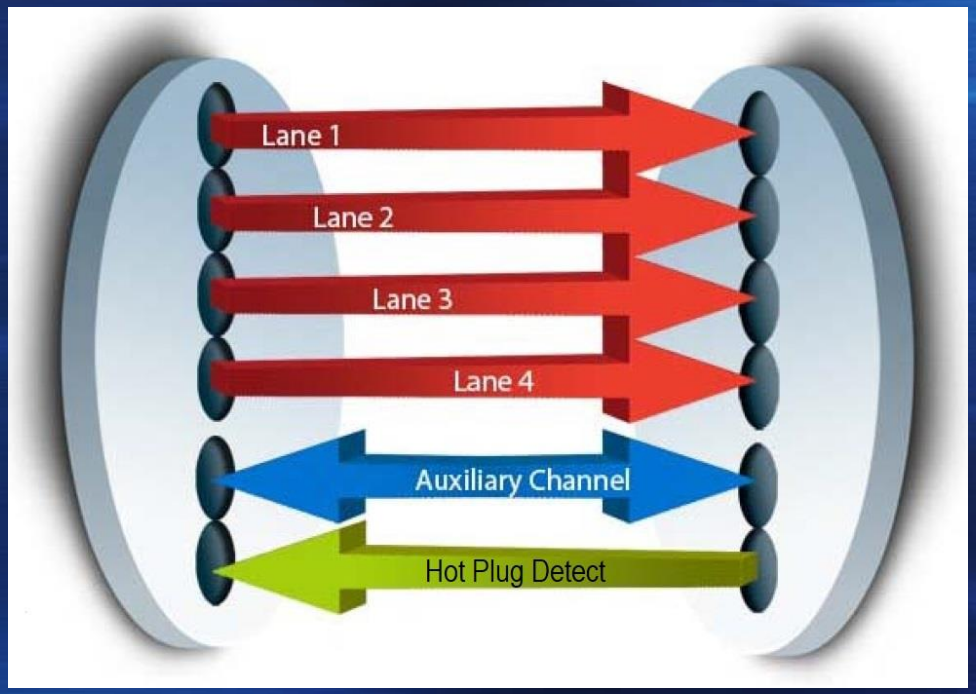
DisplayPort Overview (8)

- Connectors and cables
  - 20 pins for external connections
  - Powered connectors → 3.3 V, 500 mA (~1.5 W)
  - Cable length: up to 2 m for full bandwidth; 15 m for reduced bandwidth

DisplayPort

- DisplayPort
  - DisplayPort Overview
  - DisplayPort Architecture
  - Embedded DisplayPort (eDP)

DisplayPort Architecture (1)



- Hot Plug Detect signal: 0 V or 3.3 V
  - Indicates the presence or absence of a display
  - May signal an interrupt from the display

DisplayPort Architecture (2)

- Main link (versions 1.3, 1.4)
  - Clock signal frequency: 270; 540; 810 MHz
  - Raw bit rates: 2.7; 5.4; 8.1 Gbits/s per lane
  - Actual bit rates: 80% of raw bit rates (with 4 lanes: 8.64; 17.28; 25.92 Gbits/s)
  - No. of displays supported (24 bpp, 60 Hz):

Resolution	1920x1080	2560x1600	4096x216	7680x4320
Resolution name	HDTV	WQXGA	4K	8K
No. of displays	8	4	2	1

DisplayPort Architecture (3)

- Auxiliary channel
  - Default mode: 1 Mbit/s (200 Kbits/s duplex)
  - Fast mode: 720 Mbits/s (200 Mbits/s duplex)
  - Used by the video source (GPU) to identify the capabilities of the display
    - Rendering capabilities: reading the display's EDID memory
    - Support of video content protection: HDCP (*High-bandwidth Digital Content Protection*) key exchanges

DisplayPort Architecture (4)

- Allows to maintain link integrity
  - The display can notify the video source if data errors have occurred on the main link
- Can transport auxiliary data
  - Camera and microphone data, USB data
- Can be used to control monitor setting and operation
  - Supports the VESA MCCS (*Monitor Control Command Set*) standard: commands to control the properties of monitors $\rightarrow$ I^2C channel

DisplayPort

- DisplayPort
  - DisplayPort Overview
  - DisplayPort Architecture
  - Embedded DisplayPort (eDP)

Embedded DisplayPort (1)

- Interface for connecting video adapters to display panels of portable computers
  - Typically, an interface based on the LVDS electrical protocol is used (e.g., LDI)
- Based on the DisplayPort standard
  - Same electrical interface
  - Same basic digital protocol
- Can use the same GPU video port as external DisplayPort connections

Embedded DisplayPort (2)

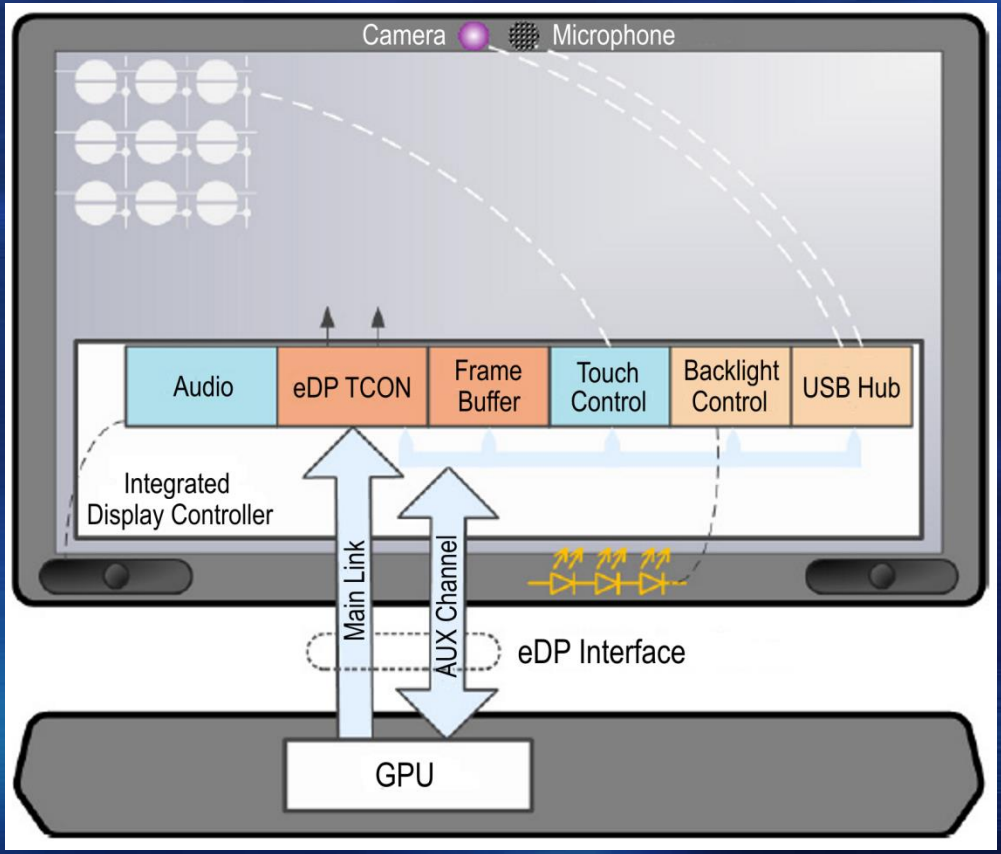
- Data transferred in main link:
  - Pixel data, audio data, and timing (e.g., pixel clock)
  - Video format information (e.g., color space, bpp)
  - ECC (*Error Correction Code*) for video data
- Data transferred in AUX channel:
  - EDID information
  - Display control: brightness control; dynamic backlight control; frame rate control (FRC)
  - Power state control

Embedded DisplayPort (3)

- Advantages compared to LVDS interfaces
  - A single connector (data, control, power)
  - Reduced wire count → simplified cable
    - Example for a resolution of 1920×1080, 24 bpp: 4 signal wires compared to 20 signal wires
  - Lower electromagnetic interference
  - Enables new display control capabilities
  - Reduced power consumption (e.g., display panel self-refresh feature)
  - The packet-based protocol is extensible

Embedded DisplayPort (4)

- Enables highly integrated display controllers



Summary (1)

- GPUs contain many processing cores, programmable for various operations
  - Can also be used for general-purpose applications
- CUDA enables to directly access the GPU resources for general-purpose computing
- The HDMI interface is intended especially for consumer electronics devices
  - Allows to send video data, audio data, and control information over a single cable
  - Uses a serial interface for uncompressed video data and the TMDS signaling protocol

Summary (2)

- The DisplayPort interface has been developed to replace the VGA and DVI interfaces
  - Uses a new protocol based on micro packets
  - Video and audio data are sent on a main link with up to 4 lanes
  - Control information is sent on an auxiliary channel
- The eDP interface is intended for internal display panels
  - Uses the same protocol as DisplayPort
  - Has several advantages compared to LVDS

Concepts, Knowledge

- Overview of CUDA architecture
- Memory hierarchy in the CUDA architecture
- Overview of HDMI interface
- The TMDS protocol and link
- The CEC (*Consumer Electronics Control*) bus
- Overview of DisplayPort interface
- Functions of DisplayPort auxiliary channel
- Overview of eDP interface
- Advantages of eDP interface