AARHUS
UNIVERSITY
SCHOOL OF ENGINEERING

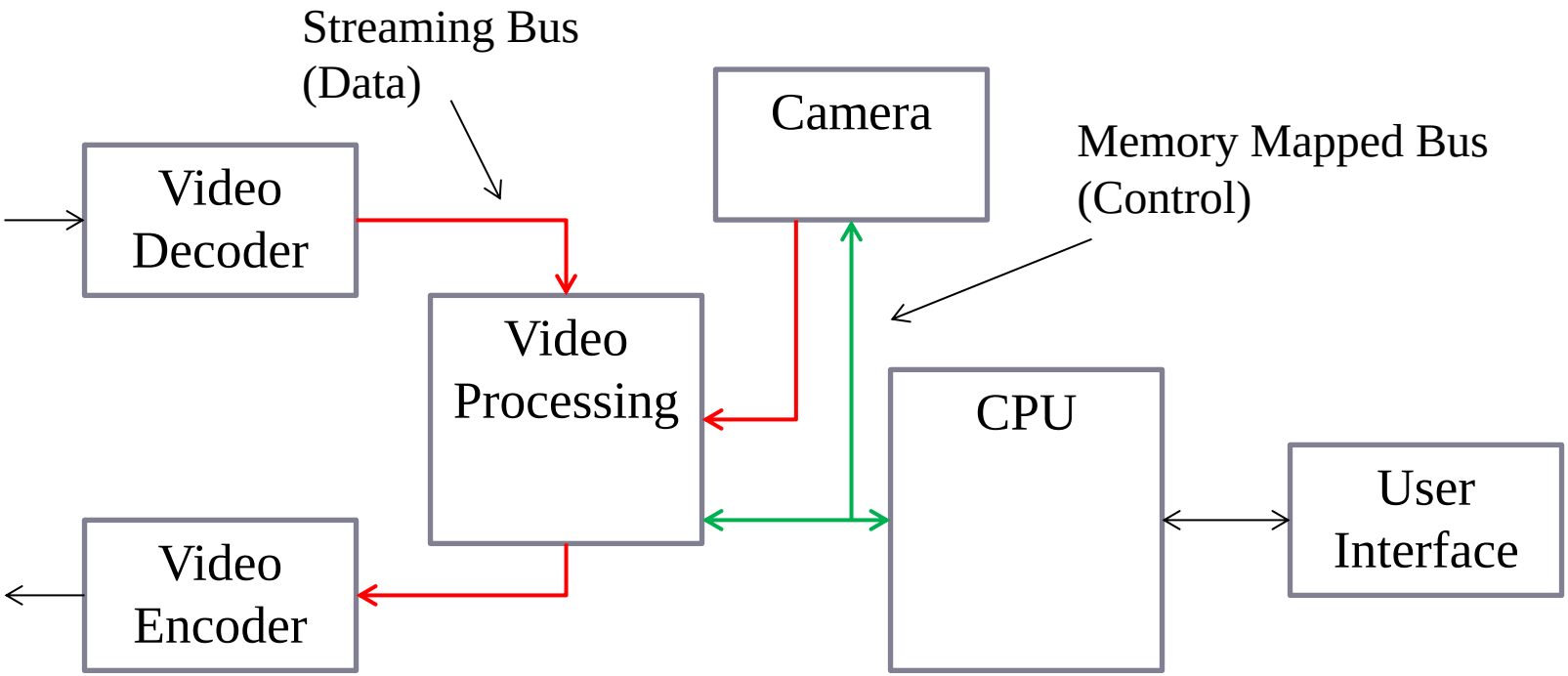
PETER HØGH MIKKELSEN
16. FEBRUARY, 2015

SOPC STREAMING BUSSES

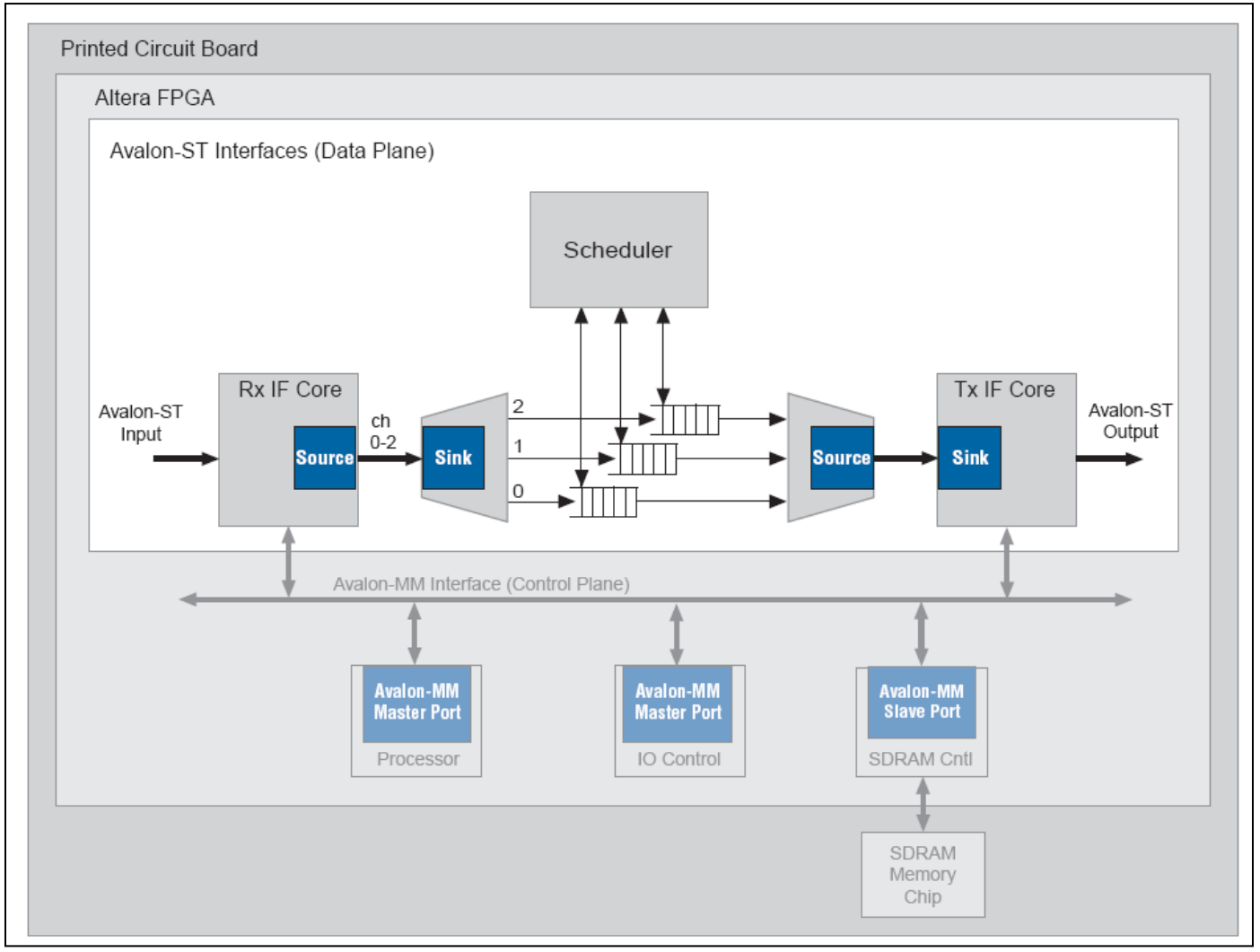
AGENDA

- Avalon Streaming Bus
- When and how to use it

VIDEO PHONE


Streaming Bus
(Data)
Camera
Memory Mapped Bus
(Control)
Video
Decoder
Video
Processing
CPU
User
Interface
Video
Encoder

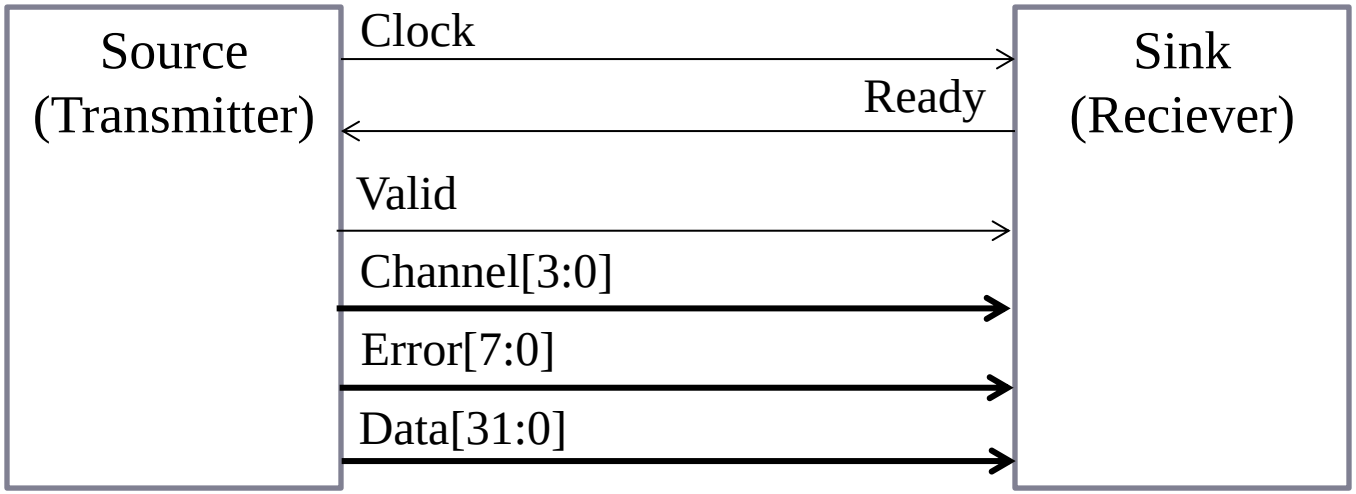
AVALON ST BUS



AVALON ST BUS

- Point-to-Point connections (Source-Sink)
- Packet Oriented
- For components that drive high bandwidth, low latency, unidirectional data.
- Typical applications include multiplexed streams, packets, and DSP data.
- The Avalon-ST interface signals can describe traditional streaming interfaces supporting a single stream of data without knowledge of channels or packet boundaries.

ST BUS TERMS (1:3)

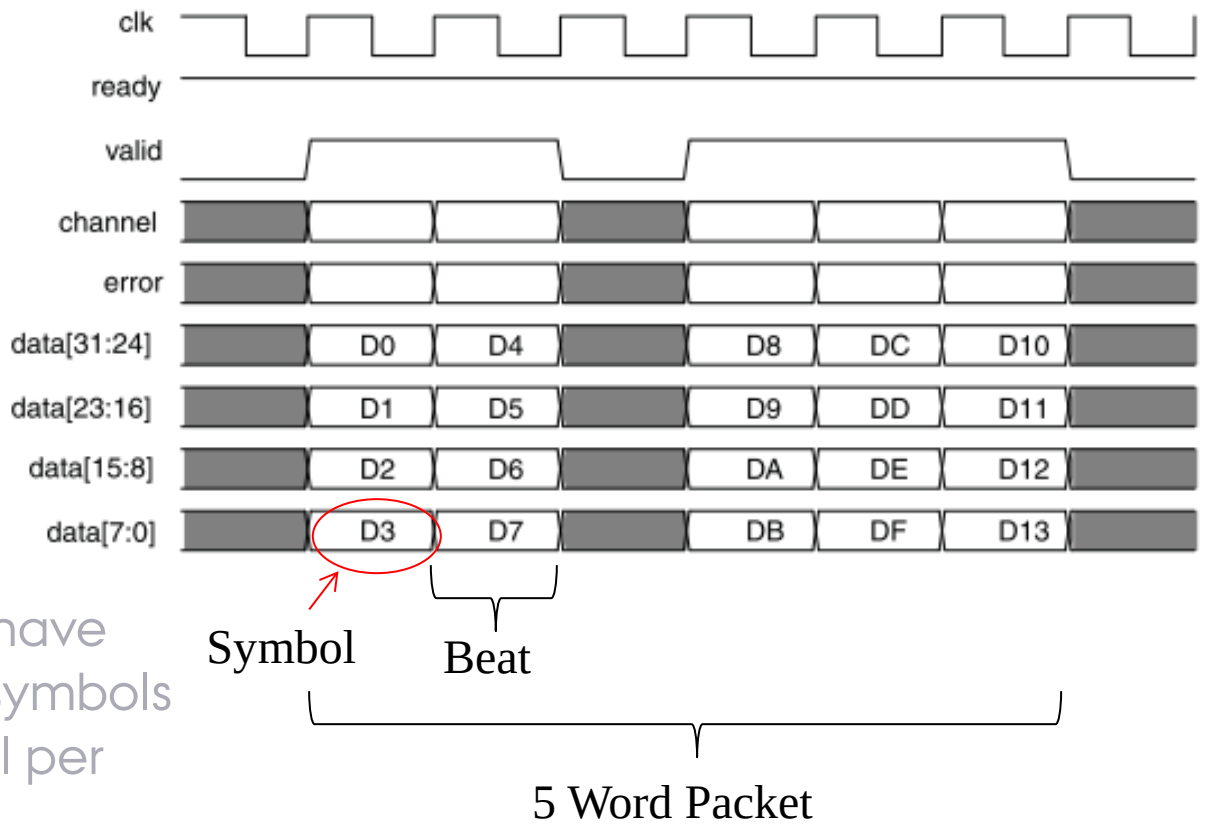


ST BUS TERMS (2:3)

- Source – Emits data
- Sink – Recieves data
- Backpressure – Sink can hold-off data until ready
- Symbol – Smallest unit of data to be transferred (# parallel data bits)
- Transfer Cycle – Transfer of a given number of symbols
- Channel – Physical or logical path between source and destination (”similar” to ports in TCP)
- Beat – A single cycle transfer (# of symbols)
- Packet – An aggregation of data and control (beats)

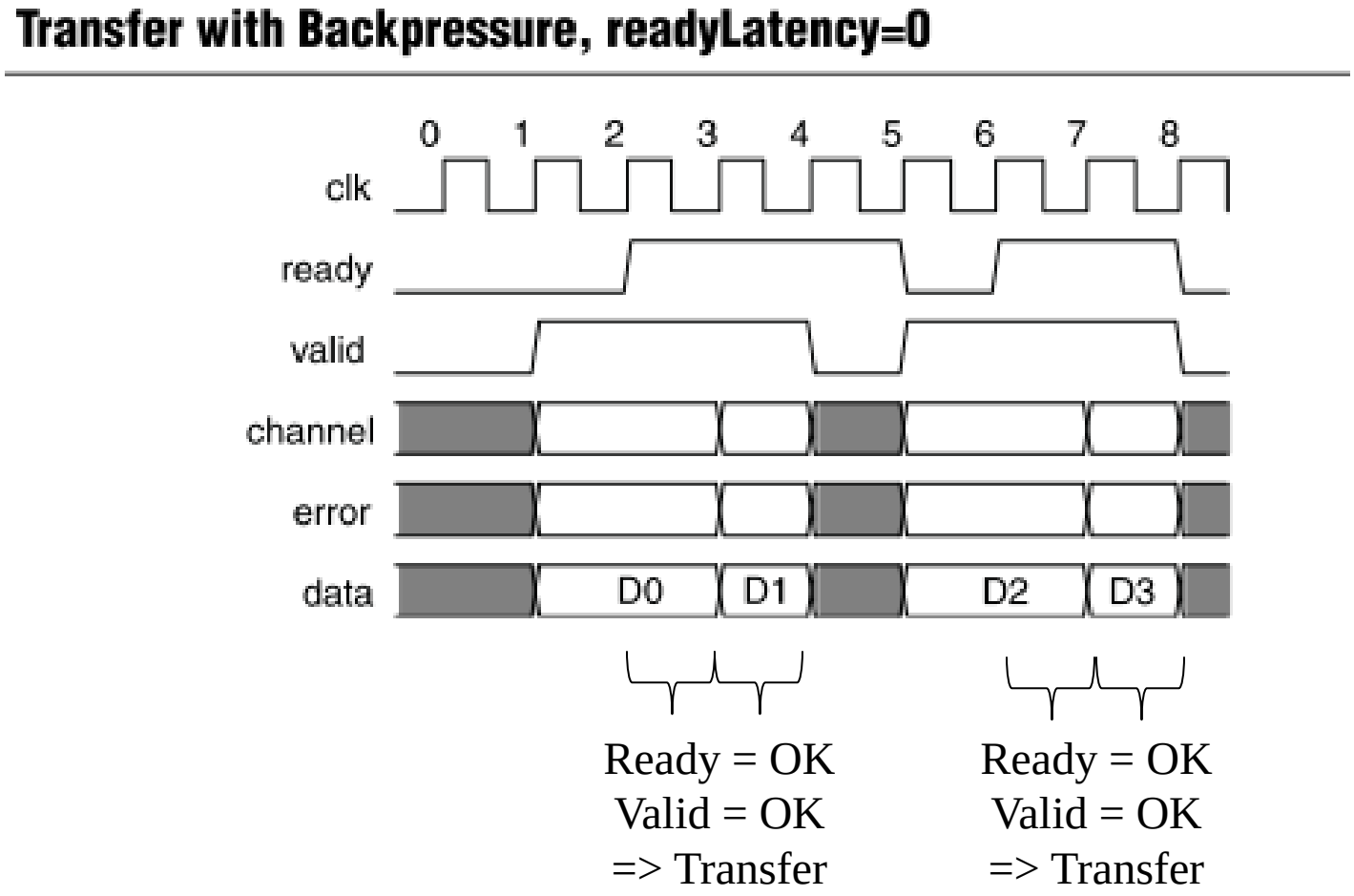
ST BUS TERMS (3:3)

Setting: 32-bit data, 8-bits per symbol, 4 symbols per beat



It could also have been 32-bit symbols and 1 symbol per beat!

BACKPRESSURE



SOPC ST EXAMPLE (1:2)

```
entity st_bus is
port (
    -- Common --
    clk             : in  std_logic;
    reset_n         : in  std_logic;
    -- ST Bus --
    ast_sink_data    : in  std_logic_vector(23 downto 0);
    ast_sink_ready   : out std_logic                    := '0';  -- Value at startup
    ast_sink_valid   : in  std_logic;
    ast_sink_error   : in  std_logic_vector(1 downto 0);
    ast_source_data  : out std_logic_vector(23 downto 0) := (others => '0');
    ast_source_ready : in  std_logic;
    ast_source_valid : out std_logic                    := '0';
    ast_source_error : out std_logic_vector(1 downto 0)  := (others => '0');

    );
end entity st_bus;
```

SOPC ST EXAMPLE (2:2)