GPEA

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Logistics**
- **Direct Memory Access (DMA) (Lab 7)**
- **Encryption (Lab 8)**

Logistics

TECHNISCHE UNIVERSITÄT DARMSTADT

- Please submit bitfile for Lab 5 (Buzzer lab)
- Be sure to save **backups** of your work
- You must **register for the Prüfung** in TUCaN to receive a grade in the course (but there is no Klausur)
- **Final two labs:**
  - Lab 7: DMA
  - Lab 8: DES Encryption

Logistics

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Course Evaluations** – please complete before July 14 (info via Moodle about how you will receive the forms)
- **Board & parts return:**
  - Week of July 11th – details will be sent over Moodle
  - Frau Reimund, Piloty (S2/02) E103

Lab Statistics

TECHNISCHE UNIVERSITÄT DARMSTADT

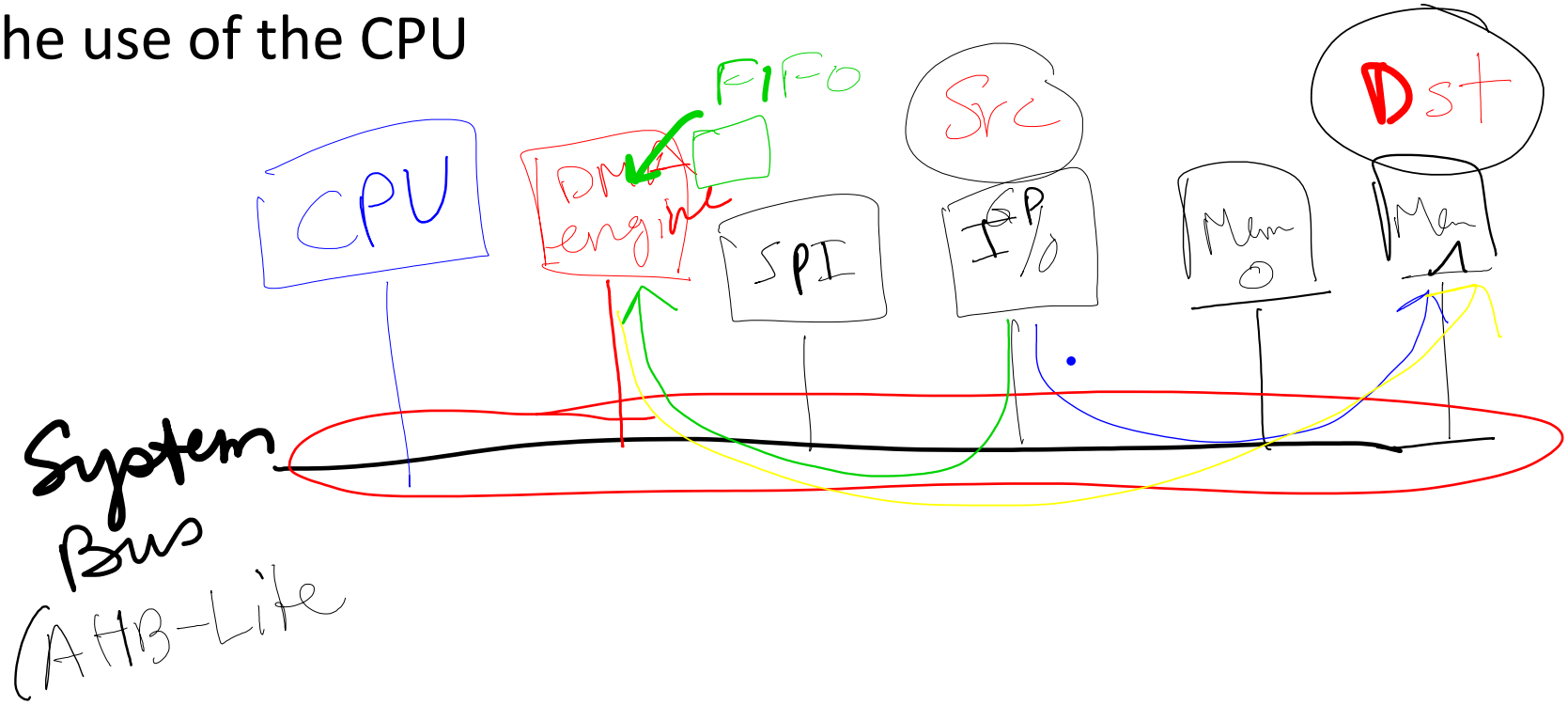
Lab #	Average Time (Hours)
1	5
2	9.5
3	7.5 (15 over 2 weeks)
4	3

Lab 7: DMA

TECHNISCHE UNIVERSITÄT DARMSTADT

Direct Memory Access (DMA):

- Enables transfers between peripherals/memory without the use of the CPU

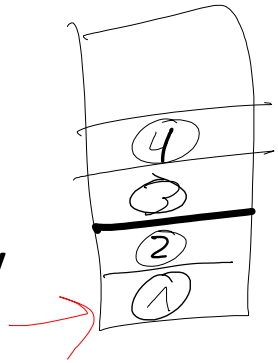


DMA Engine

TECHNISCHE UNIVERSITÄT DARMSTADT

Hardware Module that transfers data from one peripheral/memory to another:

- Reads a block of data from a peripheral/memory
- Writes that block of data to another peripheral/memory



Requirements:

- internal FIFO memory (destination for read data, source for data it writes)
- Memory-mapped I/O registers (written by CPU):
  - **Source** address
  - **Destination** address
  - **Size** (number of words to be transferred)
  - **Start** (to initiate DMA transfer)

DMA Engine

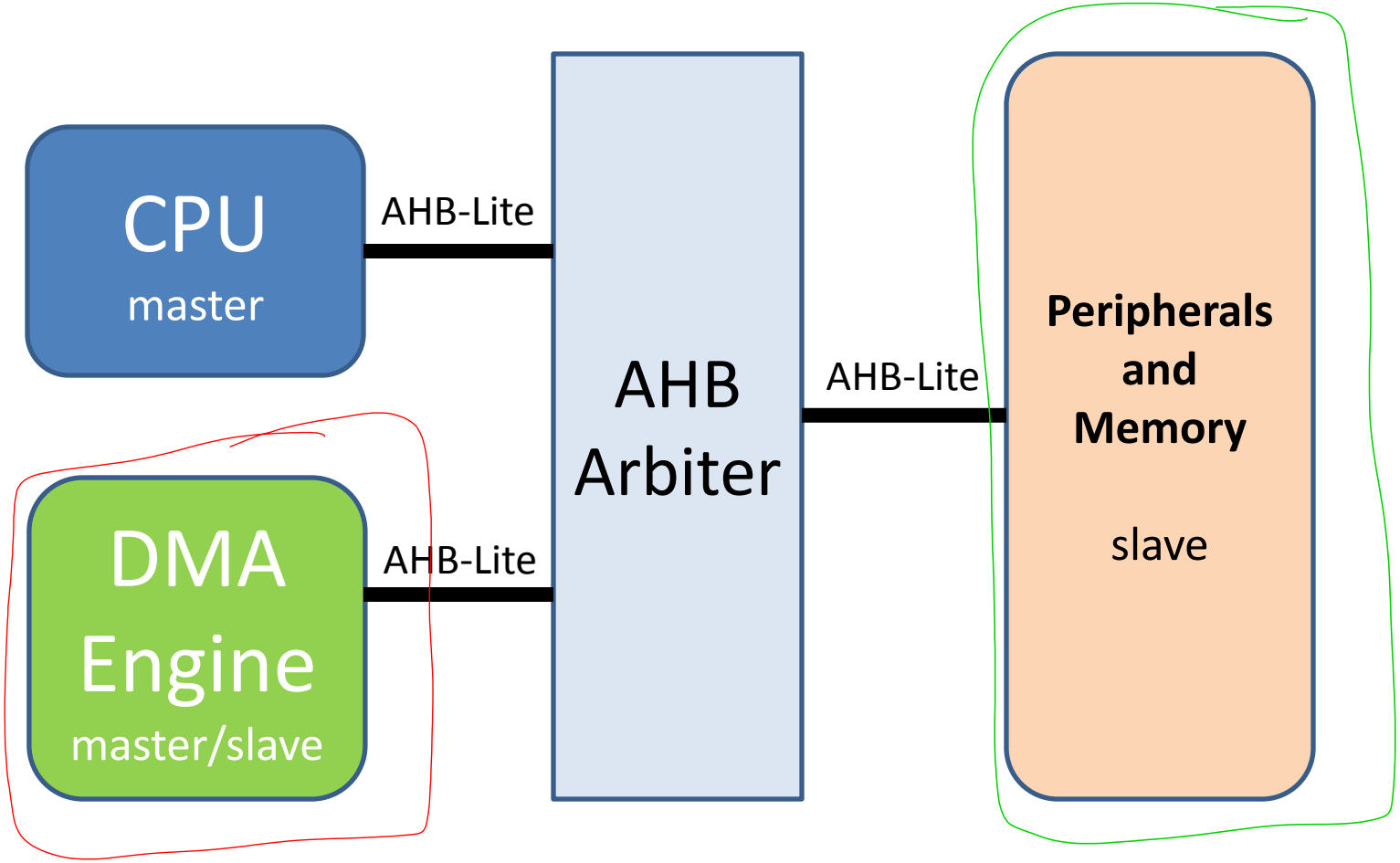
TECHNISCHE UNIVERSITÄT DARMSTADT

To transfer data, the DMA Engine must become the **master** of the system bus, in this case the AHB-Lite bus.

- slave: when I/O registers (src, dst, size, start) being written
- master: when transferring data between peripherals

AHB Arbiter

TECHNISCHE UNIVERSITÄT DARMSTADT



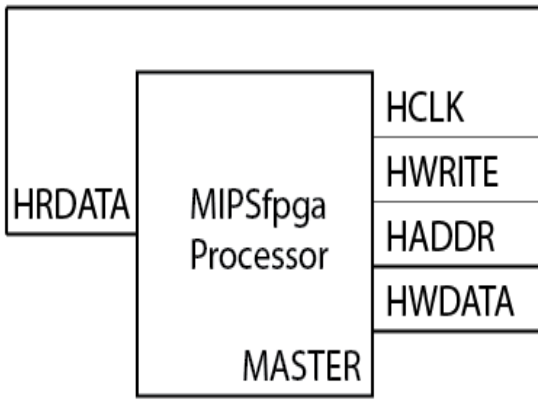
Review: AHB-Lite

TECHNISCHE UNIVERSITÄT DARMSTADT

Signal Name	Description
HADDR[31:0]	Address bus
HRDATA[31:0]	Read data bus
HWDATA[31:0]	Write data bus
HWRITE	Write enable
HCLK	Clock

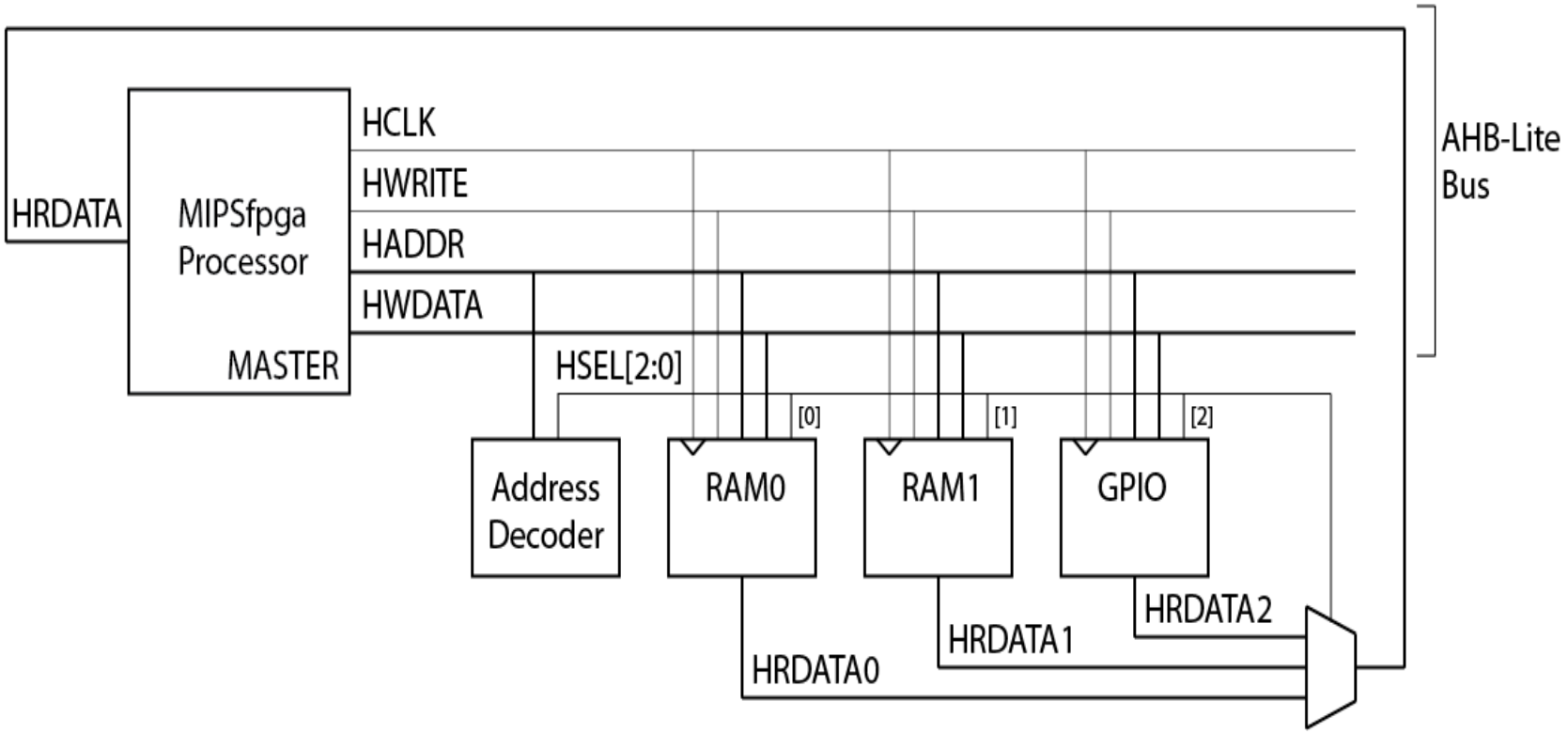
H: prefix for A**H**B-Lite Interface signals in Verilog files

MIPSfpga Interfaces: AHB-Lit



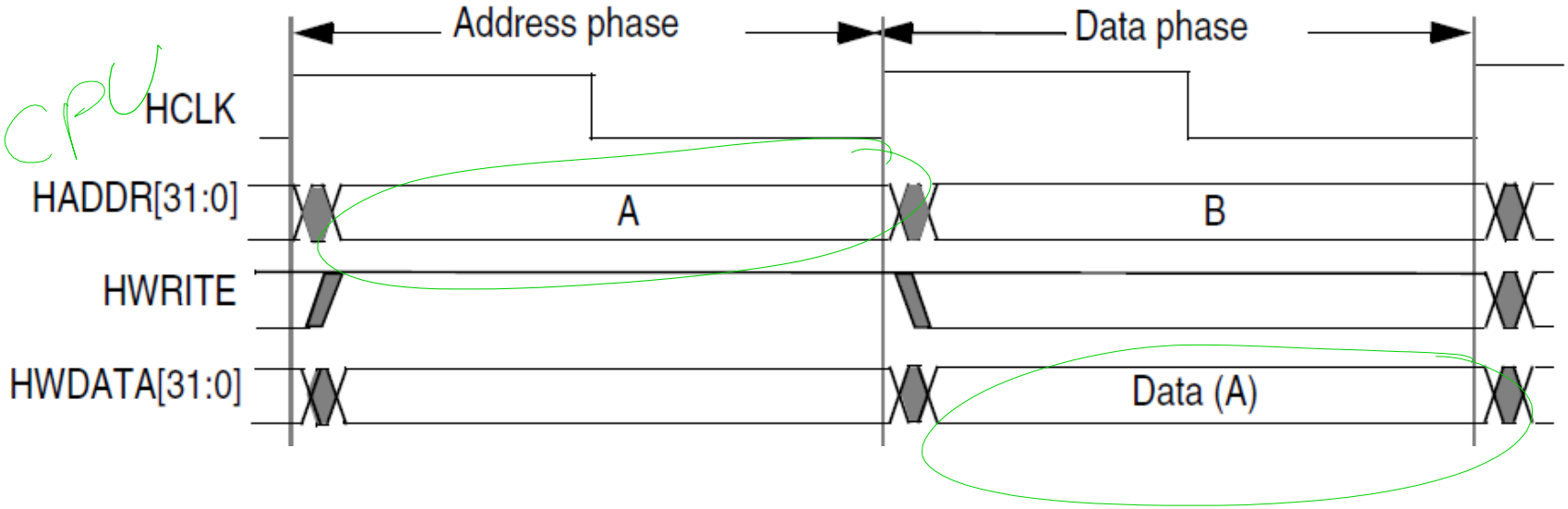
AHB-Lite Memory / Peripher

TECHNISCHE UNIVERSITÄT DARMSTADT



AHB-Lite Write Timing

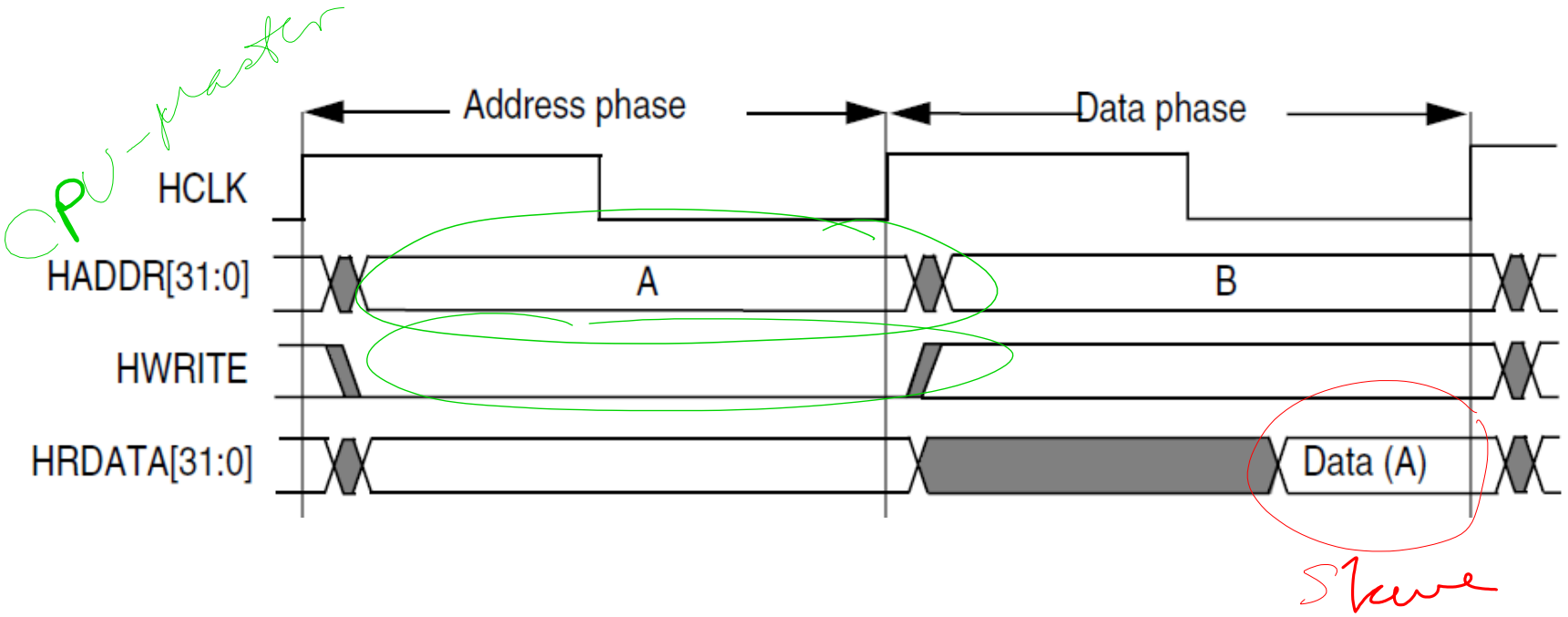
TECHNISCHE UNIVERSITÄT DARMSTADT



Cycle 1: Address and Write Enable (HADDR & HWRITE)

Cycle 2: Write Data (HWDATA)

AHB-Lite Read Timing



Cycle 1: **Address** (HADDR) (also, HWRITE = 0)

Cycle 2: **Read Data** (HRDATA)

Additional Signals: AHB-Lite

TECHNISCHE UNIVERSITÄT DARMSTADT

Signal Name	Description
HADDR[31:0]	Address bus
HRDATA[31:0]	Read data bus
HWDATA[31:0]	Write data bus
HWRITE	Write enable
HCLK	Clock
HREADY	Data ready
HTRANS[1:0]	Transaction

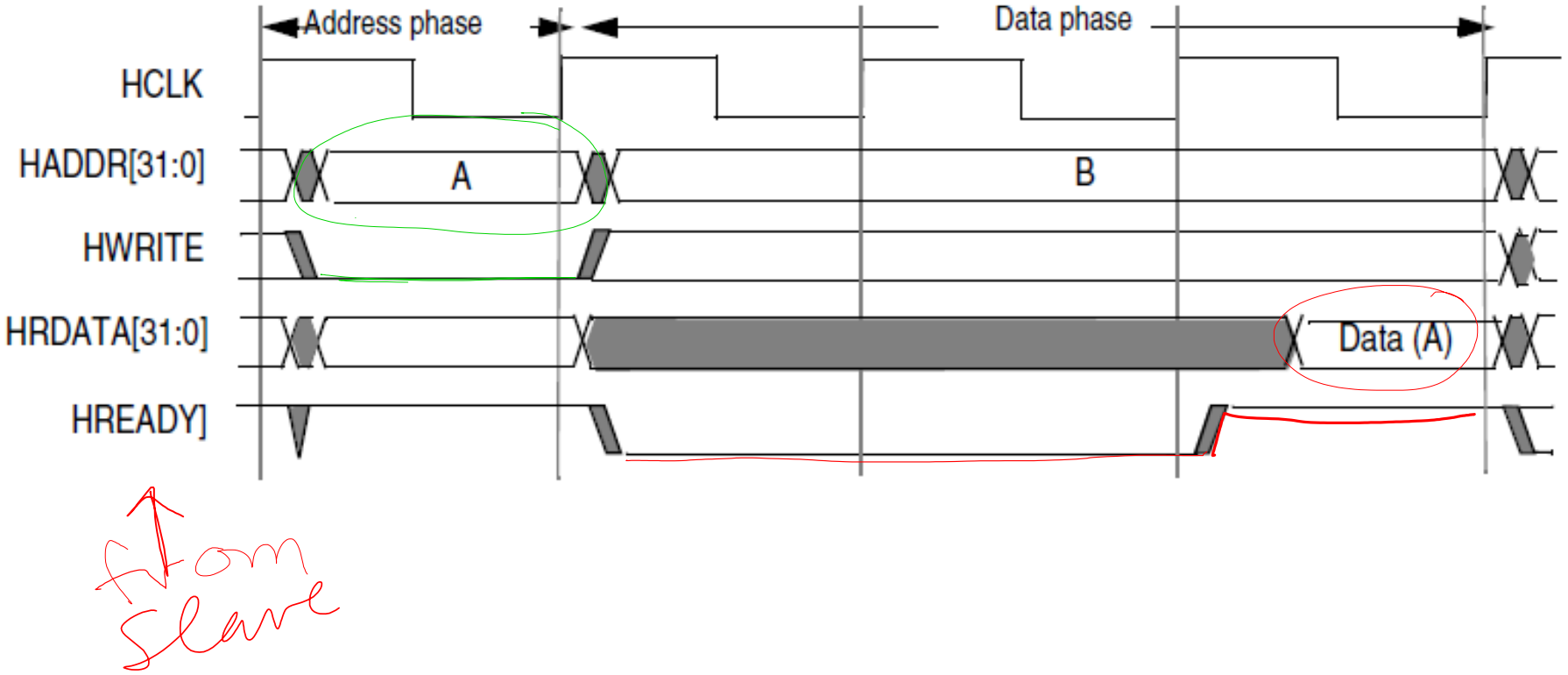
Additional Signals: AHB-Lite

HREADY: 1 = ready, 0 = wait

- sent from slave to master
- halts the master when HREADY = 0

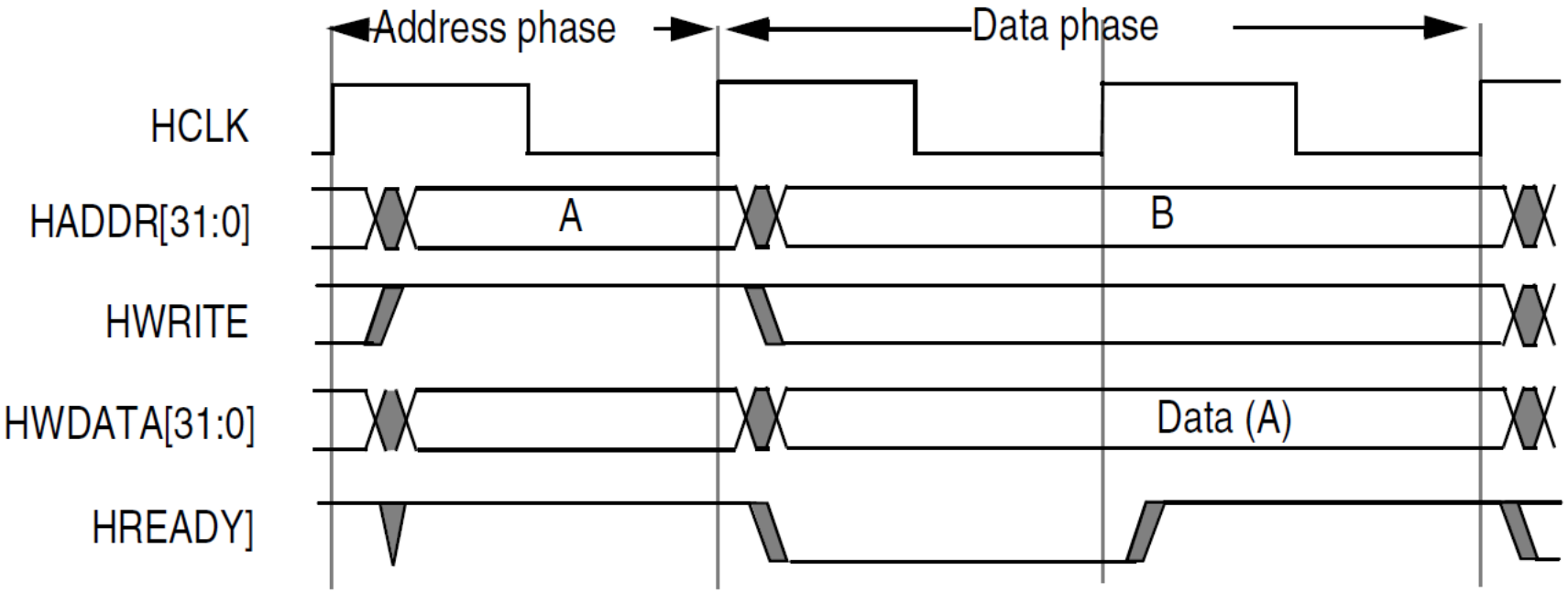
AHB-Lite Read w/ Wait

TECHNISCHE UNIVERSITÄT DARMSTADT



AHB-Lite Write w/ Wait

TECHNISCHE UNIVERSITÄT DARMSTADT



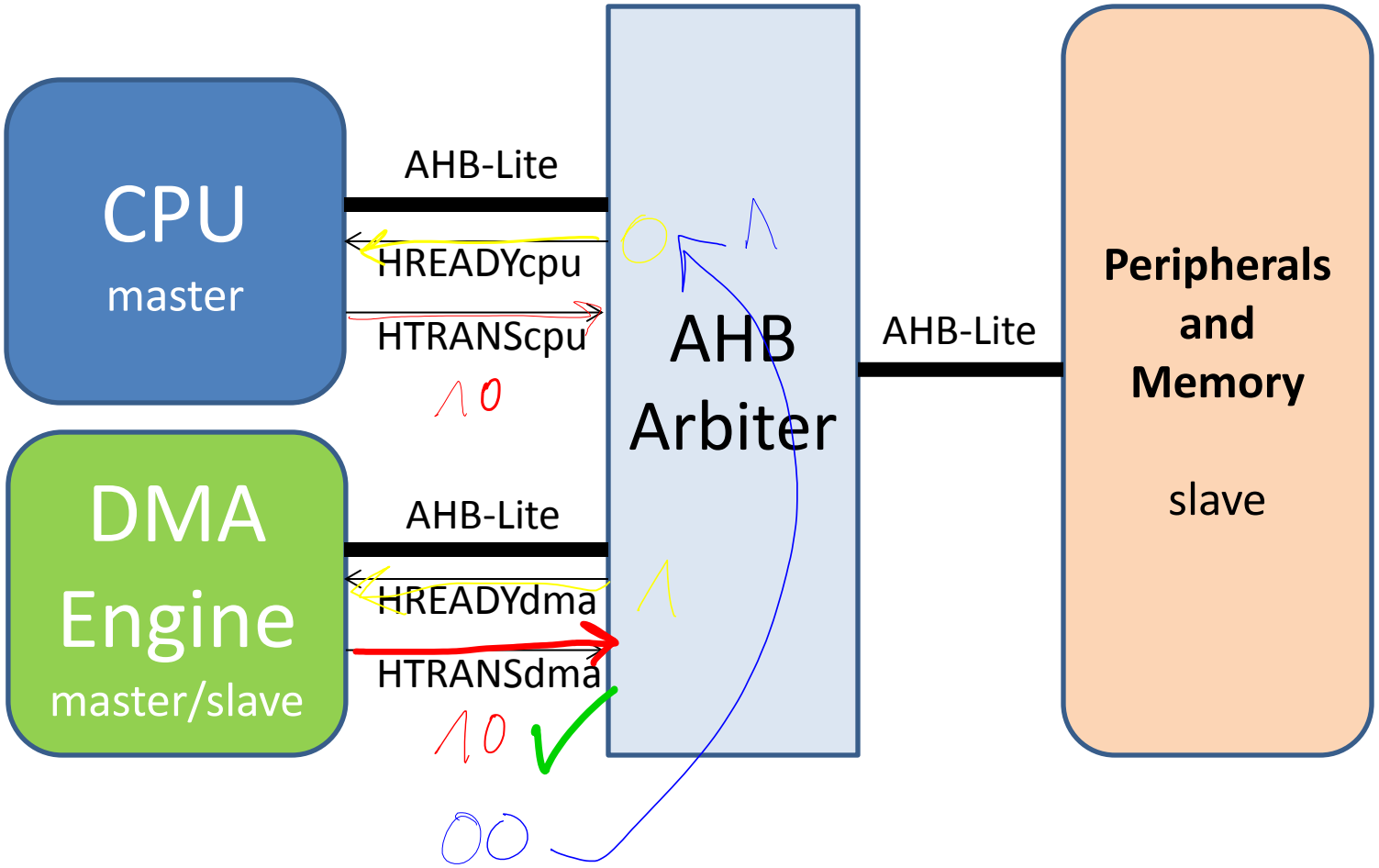
Additional Signals: AHB-Lite

HTRANS: 10 = transferring data, 00 = idle

- sent from master to slave

AHB Arbiter

TECHNISCHE UNIVERSITÄT DARMSTADT



Lab 8: DES Encryption

TECHNISCHE UNIVERSITÄT DARMSTADT

You will build a DES Encryption Engine (hardware module), and integrate it into your MIPSfpga system.

- As always, modularity is your friend:
  - Build and test DES Encryption Engine first
  - Then integrate it into MIPSfpga