08: High-Performance Buses

Engr 315: Hardware / Software Codesign

Andrew Lukefahr

Indiana University

Announcements

- P3 is out
- P4 coming "soon."

ProTip:

- If you can't connect Vivado to your Pynq

1. Restart your machine (for ILAs)

   -- OR --

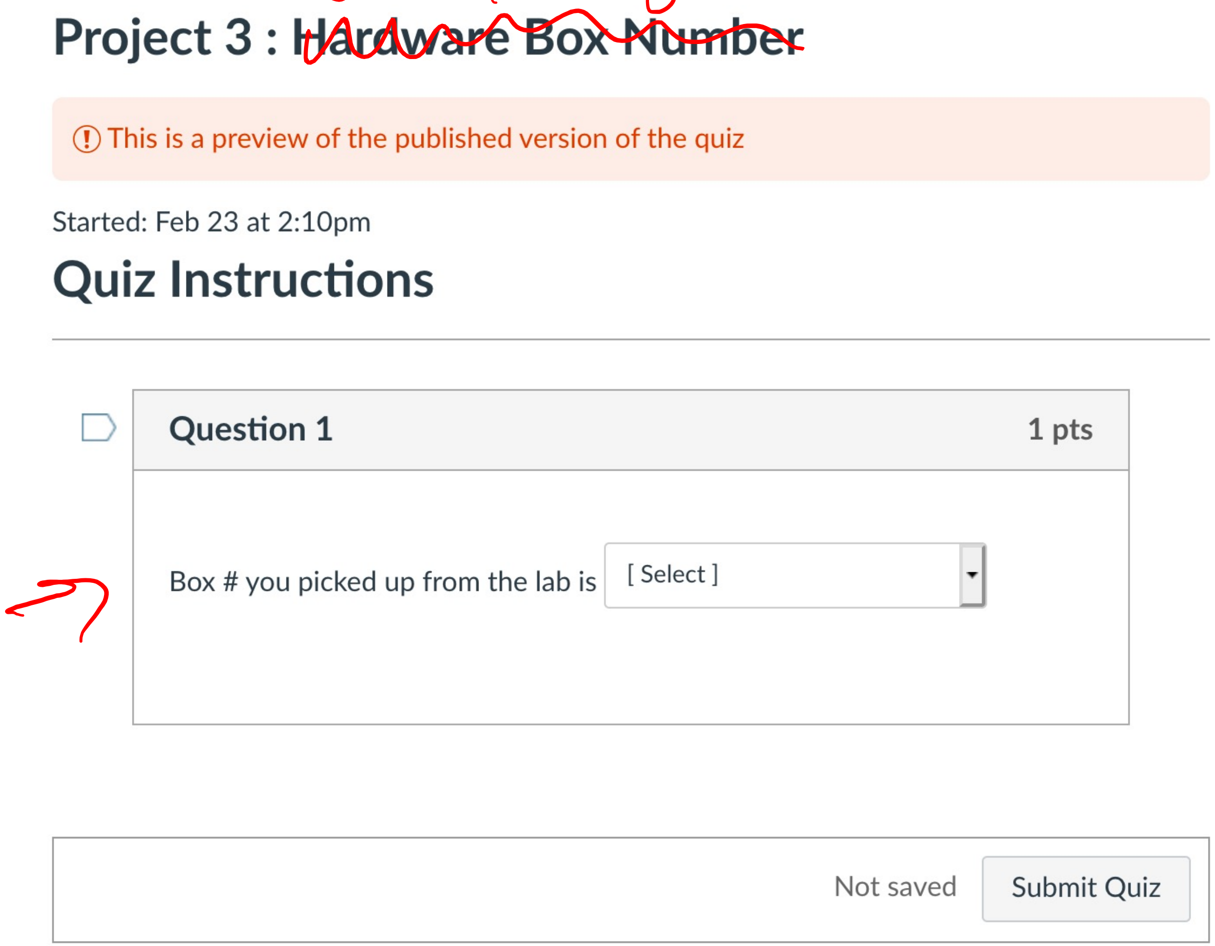
2. Ignore and proceed (for Pynq)

It's dumb. I know...

Register your Hardware

- Go to Canvas
- Select “P3 Box Number”
- Enter your box number

What's ~~y.~~ UR box?



Project 3 : ~~Hardware Box Number~~

ⓘ This is a preview of the published version of the quiz

Started: Feb 23 at 2:10pm

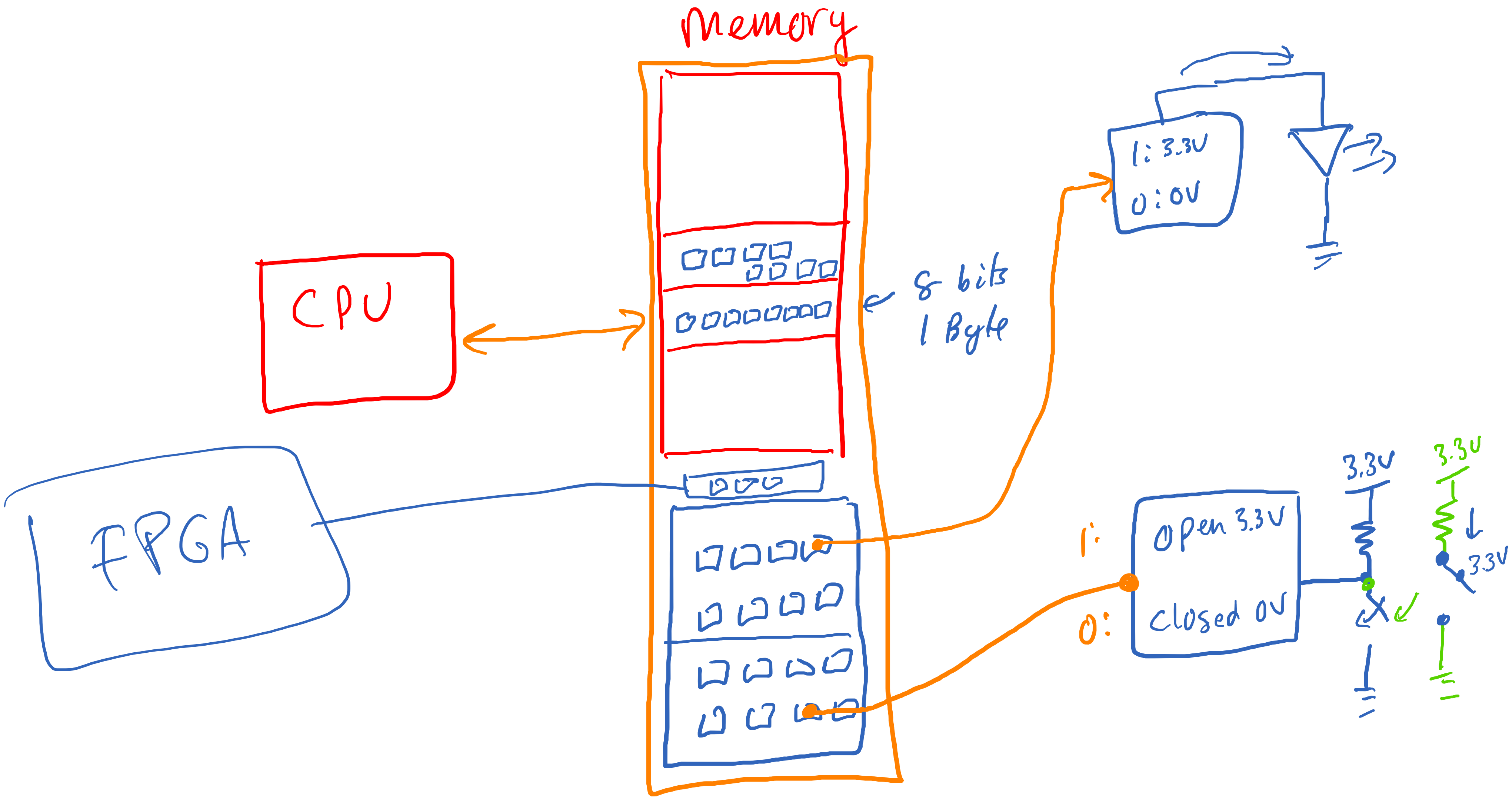
Quiz Instructions

Question 1 — 1 pts

Box # you picked up from the lab is [Select]

Not saved

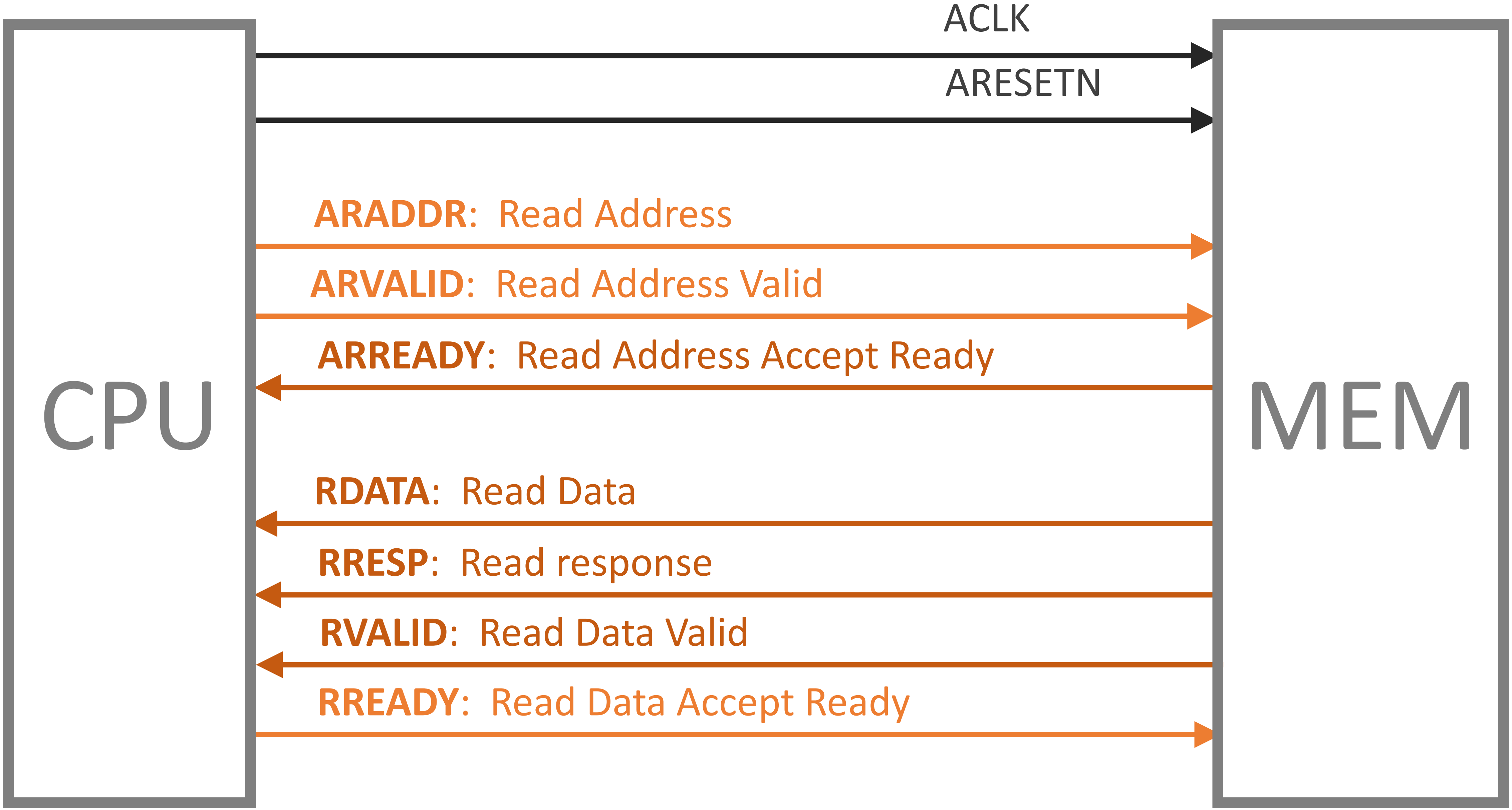
Review: Memory-Mapped I/O



ARM AXI Bus

- “Advanced eXtensible Interface” Bus Version 4, “AXI4”

- Three Variants
  - AXI4: Fast but complicated; Memory-mapped
  - AXI4 Lite: Slow but simple; Memory-mapped
  - AXI4 Stream: Fast and simple; Not memory-mapped



CPU

MEM

ACLK

ARESETN

ARADDR: Read Address

ARVALID: Read Address Valid

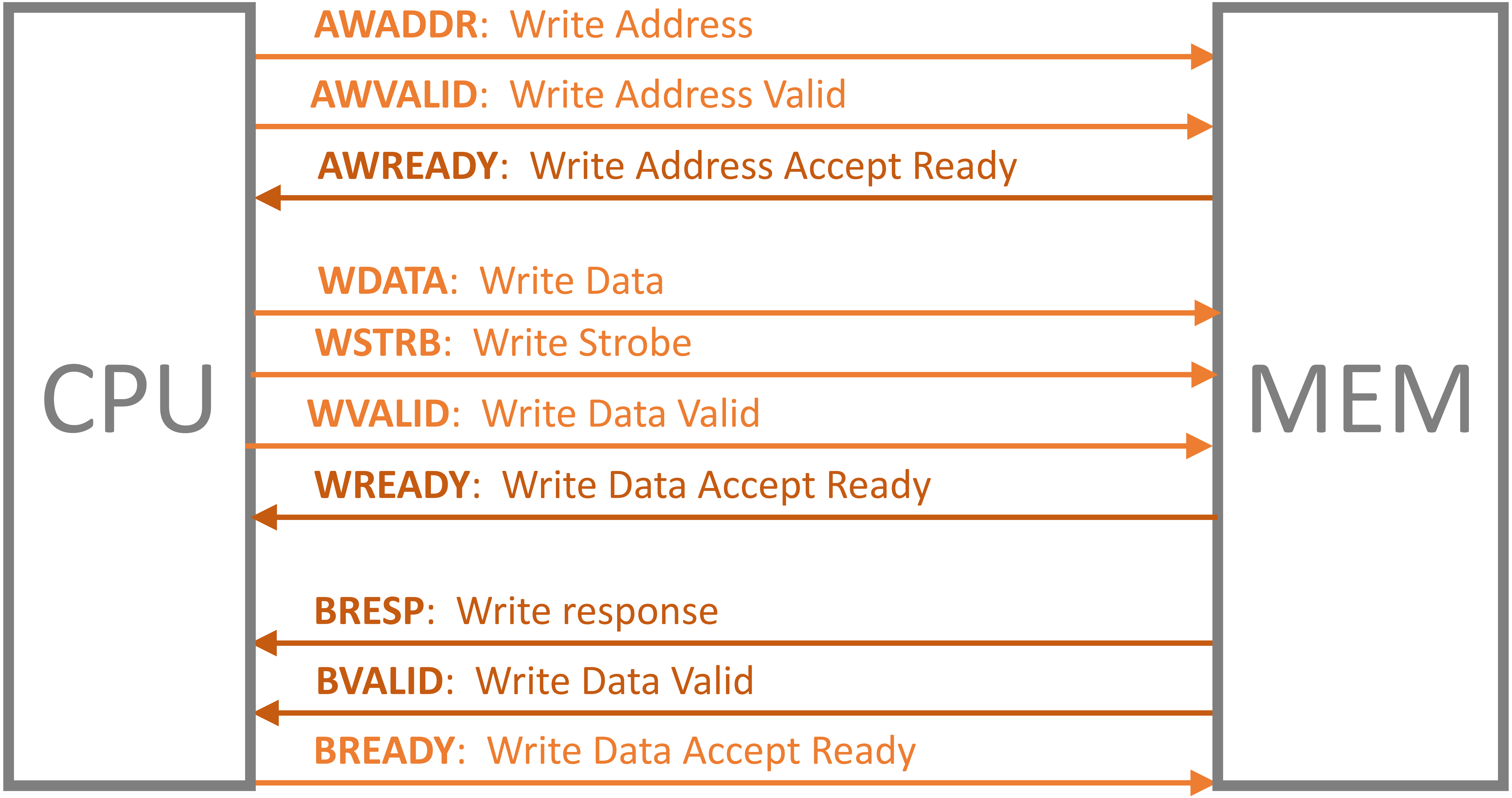
ARREADY: Read Address Accept Ready

RDATA: Read Data

RRESP: Read response

RVALID: Read Data Valid

RREADY: Read Data Accept Ready



CPU

MEM

AWADDR: Write Address

AWVALID: Write Address Valid

AWREADY: Write Address Accept Ready

WDATA: Write Data

WSTRB: Write Strobe

WVALID: Write Data Valid

WREADY: Write Data Accept Ready

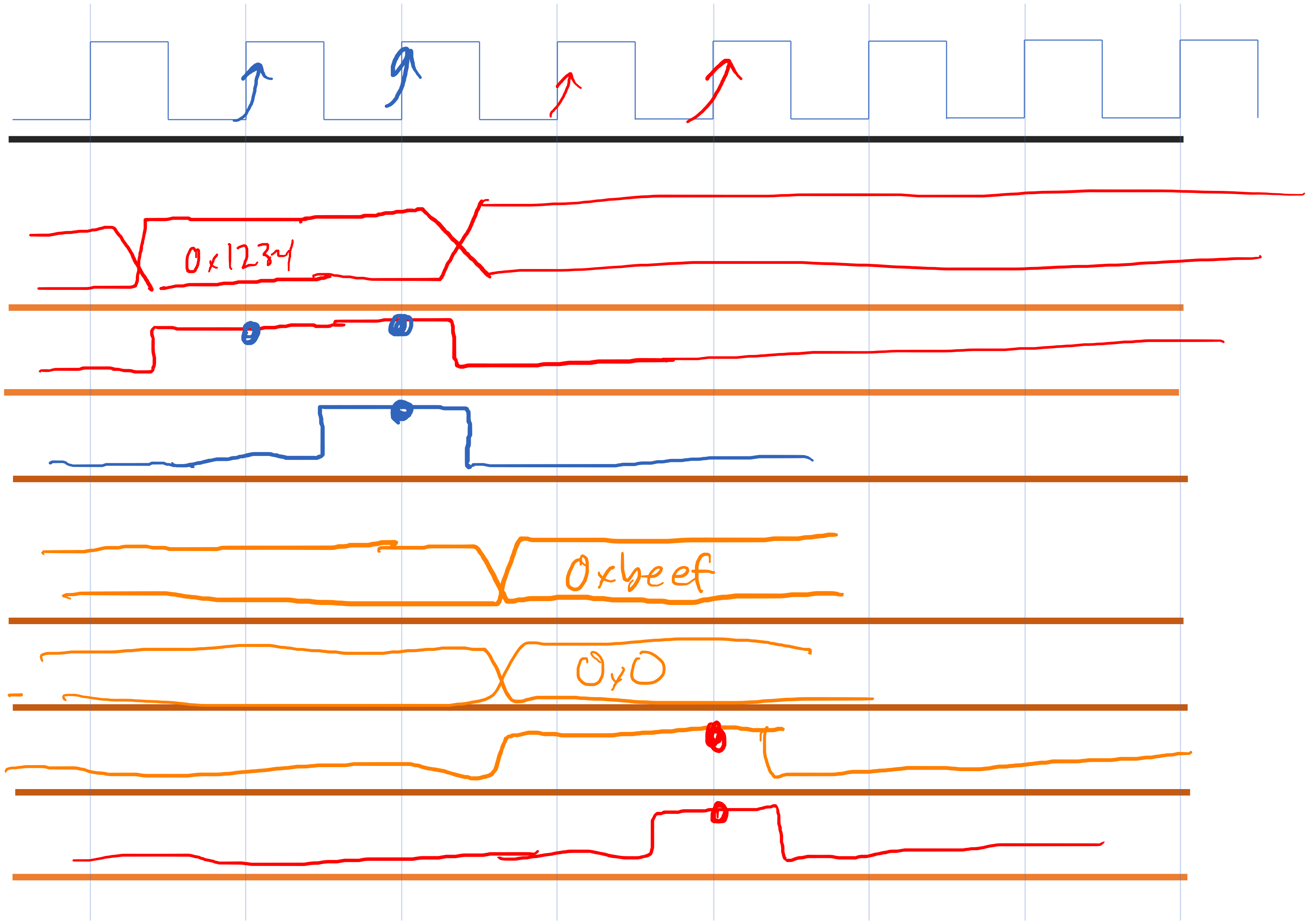
BRESP: Write response

BVALID: Write Data Valid

BREADY: Write Data Accept Ready

ACLK and ARESETN not shown

How long does a read(load) take?



ACLK:

ARADDR:

ARVALID:

ARREADY:

RDATA:

RRESP:

RVALID:

RREADY:

when is store (write)

High-Performance Bus Ideas

- Make single transaction faster

AXI Handshake Speedup

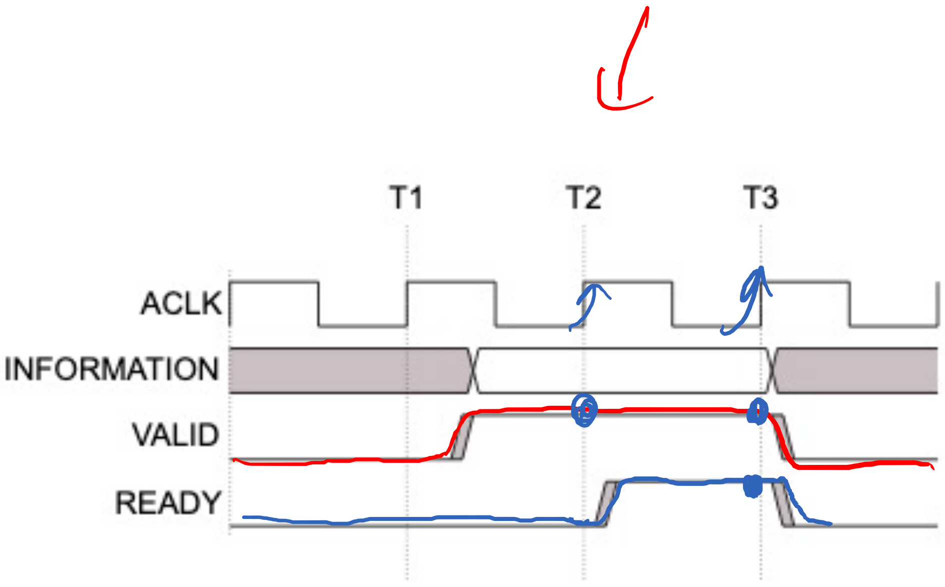


Figure A3-2 VALID before READY handshake

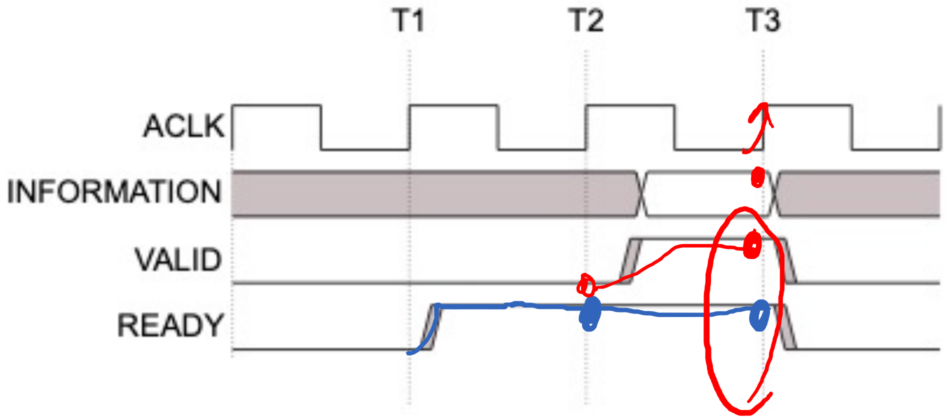


Figure A3-3 READY before VALID handshake

- Both are valid
- Figure A3-3 is faster

What happens here?

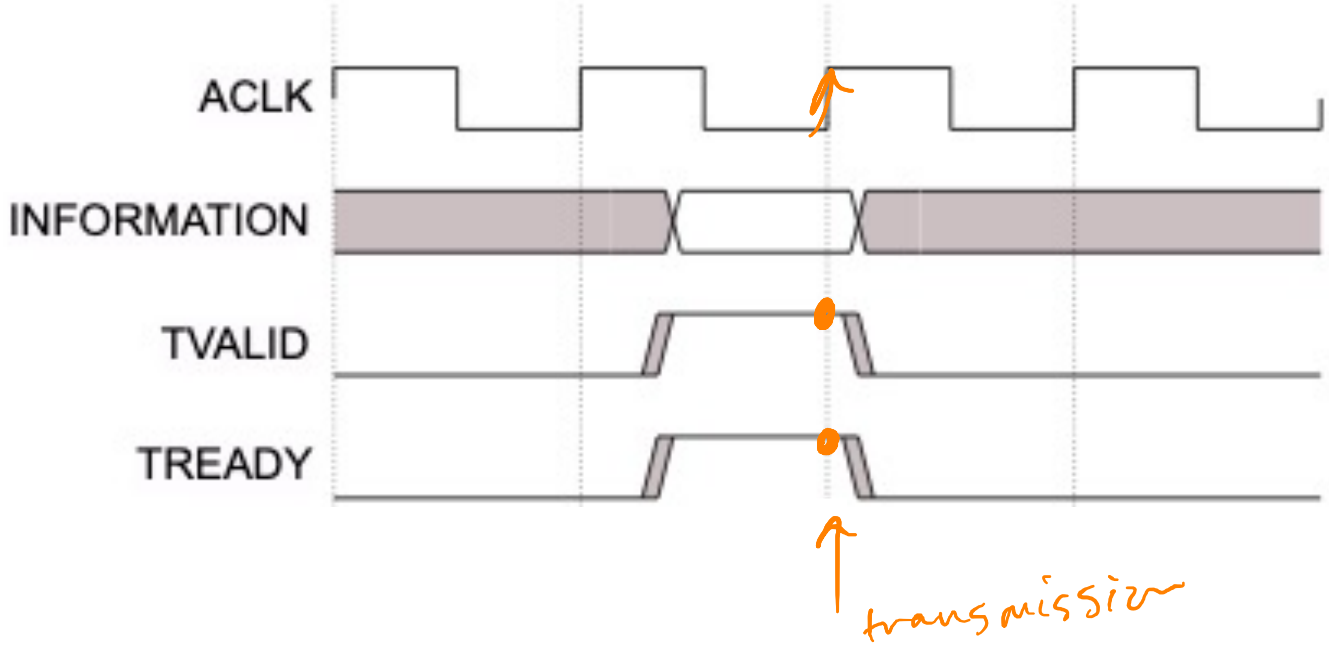
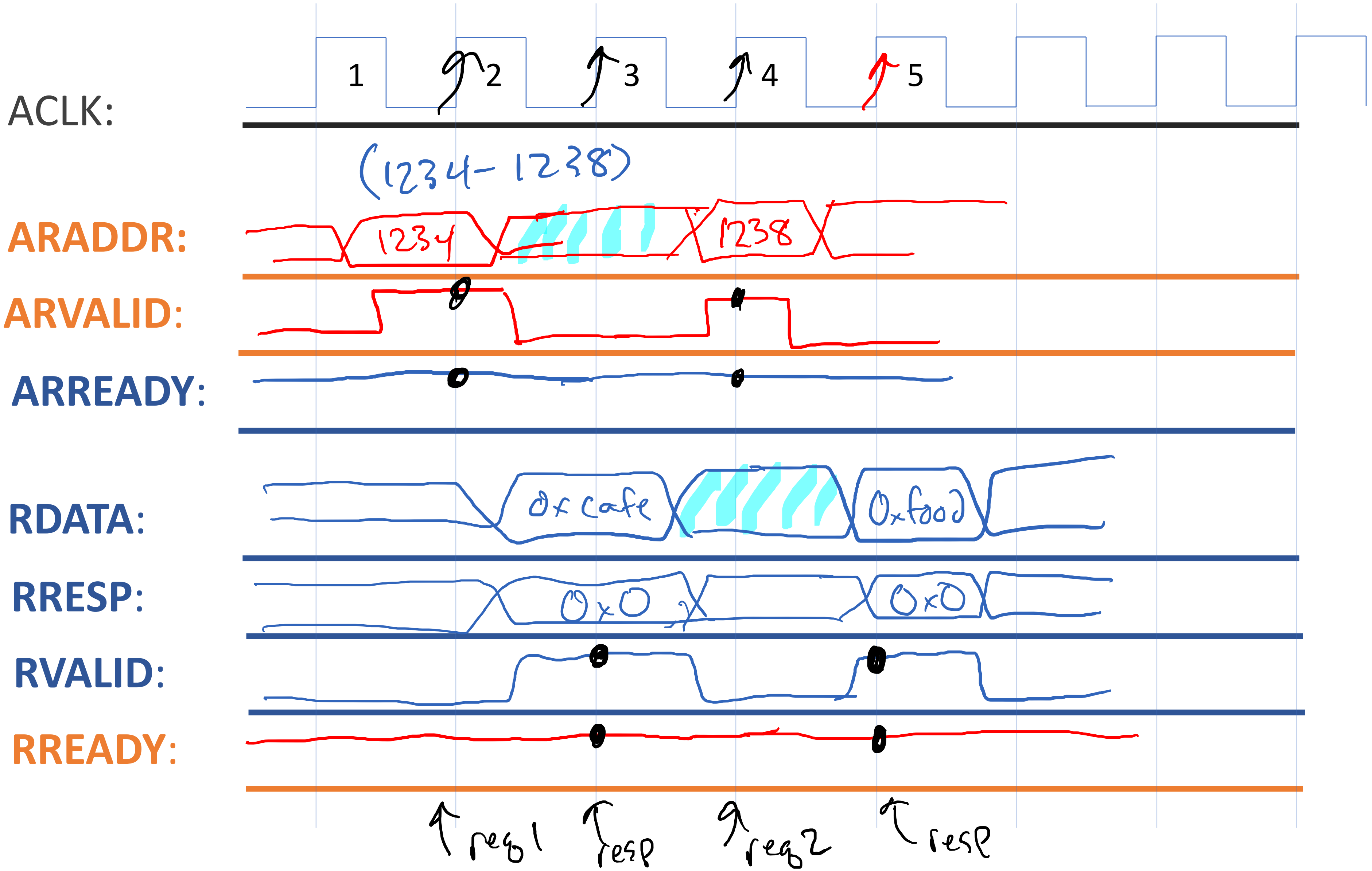


Figure 2-3 TVALID with TREADY handshake

What can we do to make this faster?



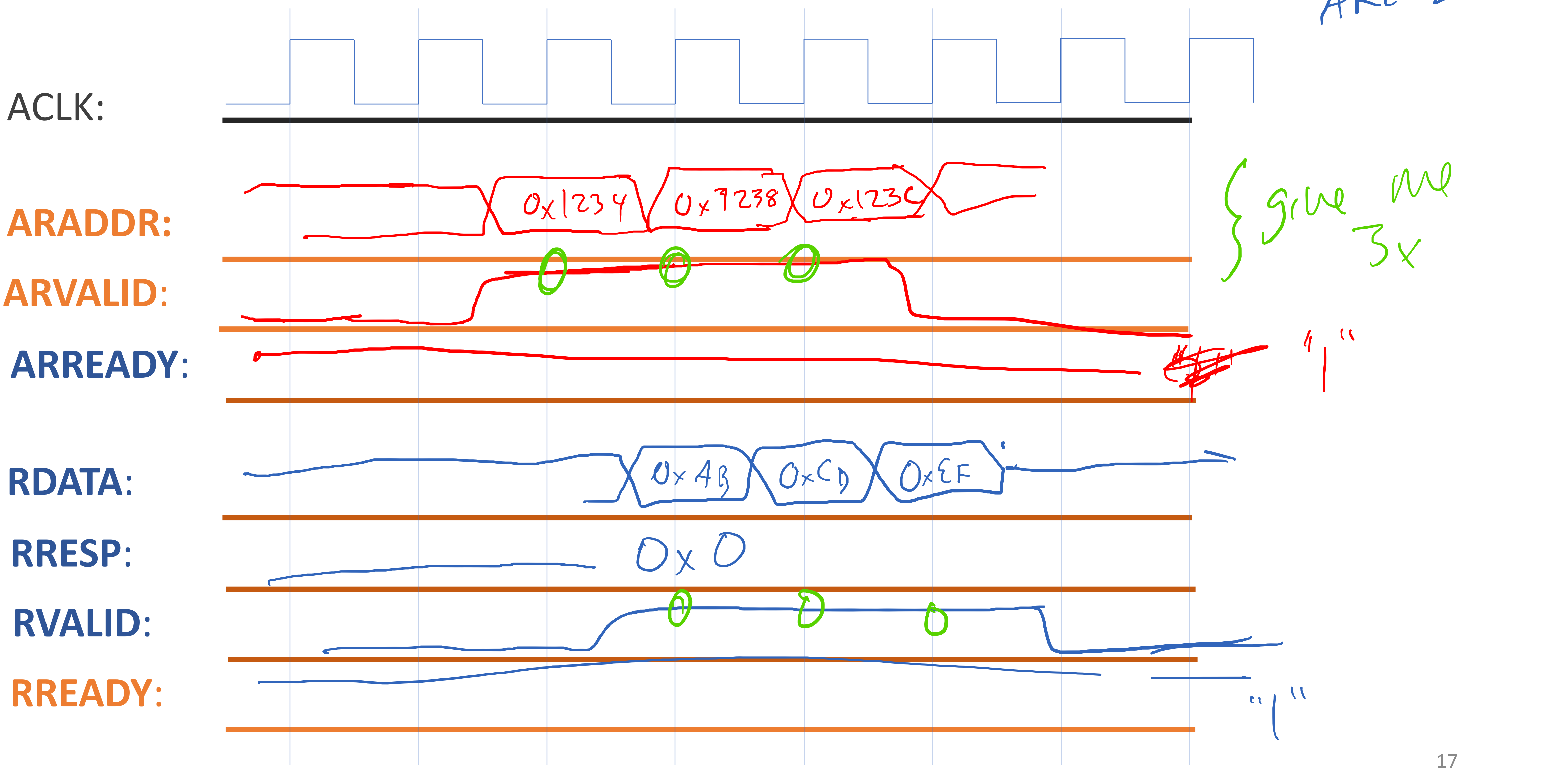
High-Performance Bus Ideas

- Make single transaction faster

- Overlap multiple transactions

Can we load 0x1234 and 0x1238?

assume ARESETN=1



ACLK:

ARADDR:

ARVALID:

ARREADY:

RDATA:

RRESP:

RVALID:

RREADY:

Burst Transactions

- When a device is transmitting data repeatedly without going through all the steps required to transmit each piece of data in a separate transaction

Burst Transaction

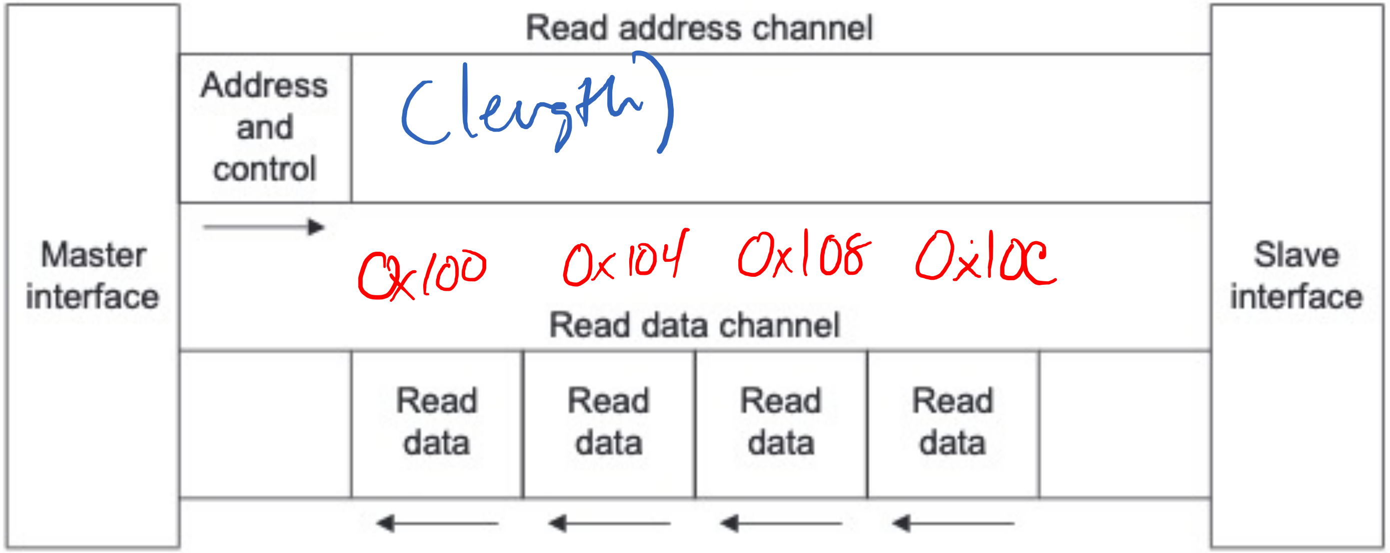
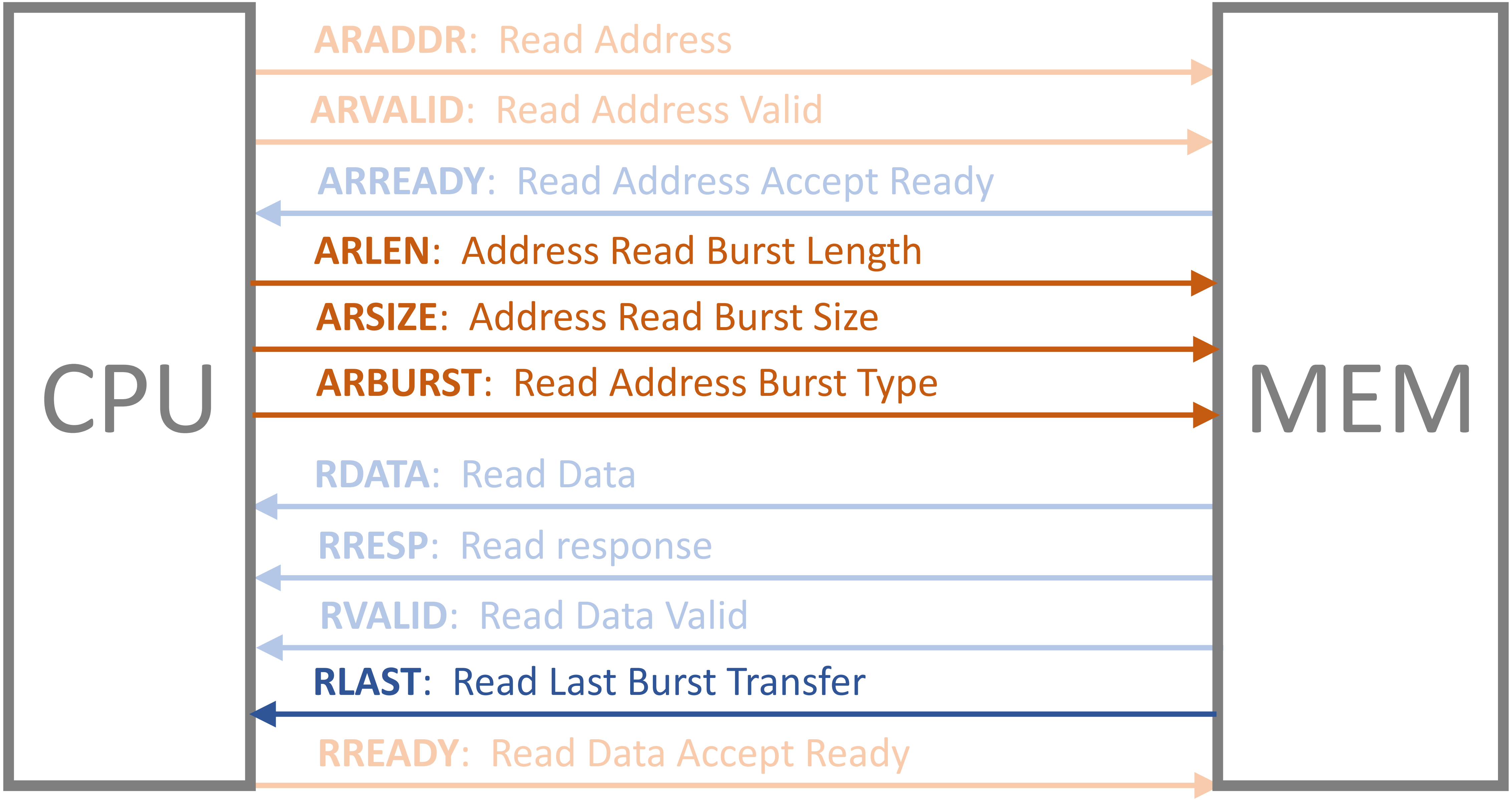


Figure 1-1 Channel architecture of reads



CPU

ARADDR: Read Address

ARVALID: Read Address Valid

ARREADY: Read Address Accept Ready

ARLEN: Address Read Burst Length

ARSIZE: Address Read Burst Size

ARBURST: Read Address Burst Type

RDATA: Read Data

RRESP: Read response

RVALID: Read Data Valid

RLAST: Read Last Burst Transfer

RREADY: Read Data Accept Ready

MEM

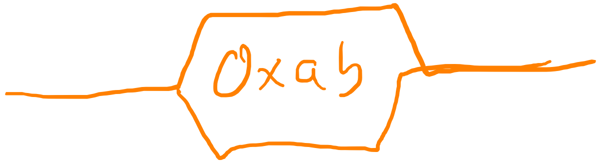
What do the new signals do?

ARLEN: Address Read Burst Length

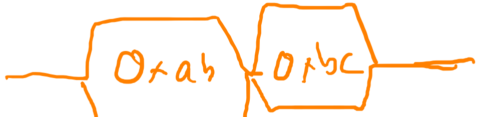
How many bursts should occur? (+1)

Burst_Length = ARLEN[7:0] + 1

ARLEN = 0 ⇒ Burst Length = 1



ARLEN = 1 ⇒ Burst = 2



What do the new signals do?

ARLEN: Address Read Burst Length

How many bursts should occur? (+1)

ARSIZE: Address Read Burst Size

How many bytes should be in each burst?

2^n Bytes / Transfer

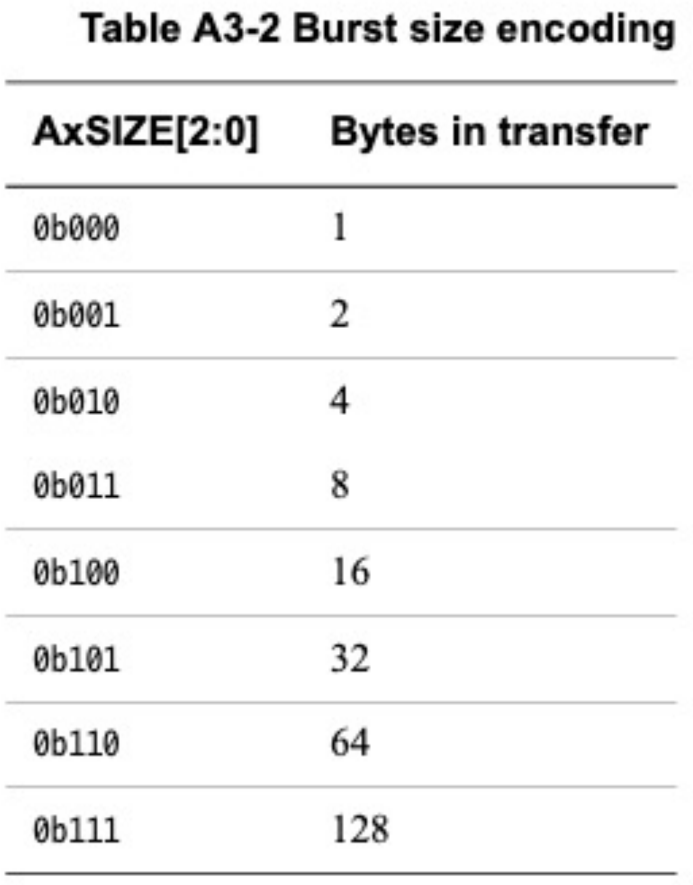


Table A3-2 Burst size encoding

AxSIZE[2:0]	Bytes in transfer
0b000	1
0b001	2
0b010	4
0b011	8
0b100	16
0b101	32
0b110	64
0b111	128

What do the new signals do?

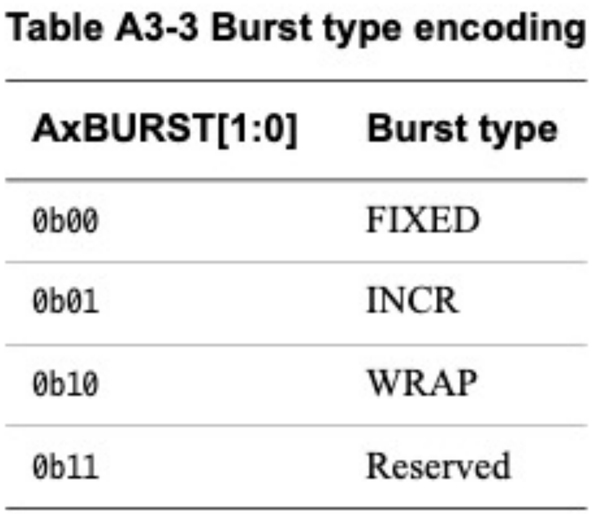


Table A3-3 Burst type encoding

AxBURST[1:0]	Burst type
0b00	FIXED
0b01	INCR
0b10	WRAP
0b11	Reserved

ARLEN: Address Read Burst Length

How many bursts should occur? (+1)

ARSIZE: Address Read Burst Size

How many bytes should be in each burst?

ARBURST: Read Address Burst Type

Are the addresses incrementing, or repeating?

FIXED: The address is the same for every transfer (Next Address = Address)

INCR: The address for each transfer is an increment of previous transfer
(Next Address = Address + 0x4)

What do the new signals do?

ARLEN: Address Read Burst Length

How many bursts should occur? (+1)

ARSIZE: Address Read Burst Size

How many bytes should be in each burst?

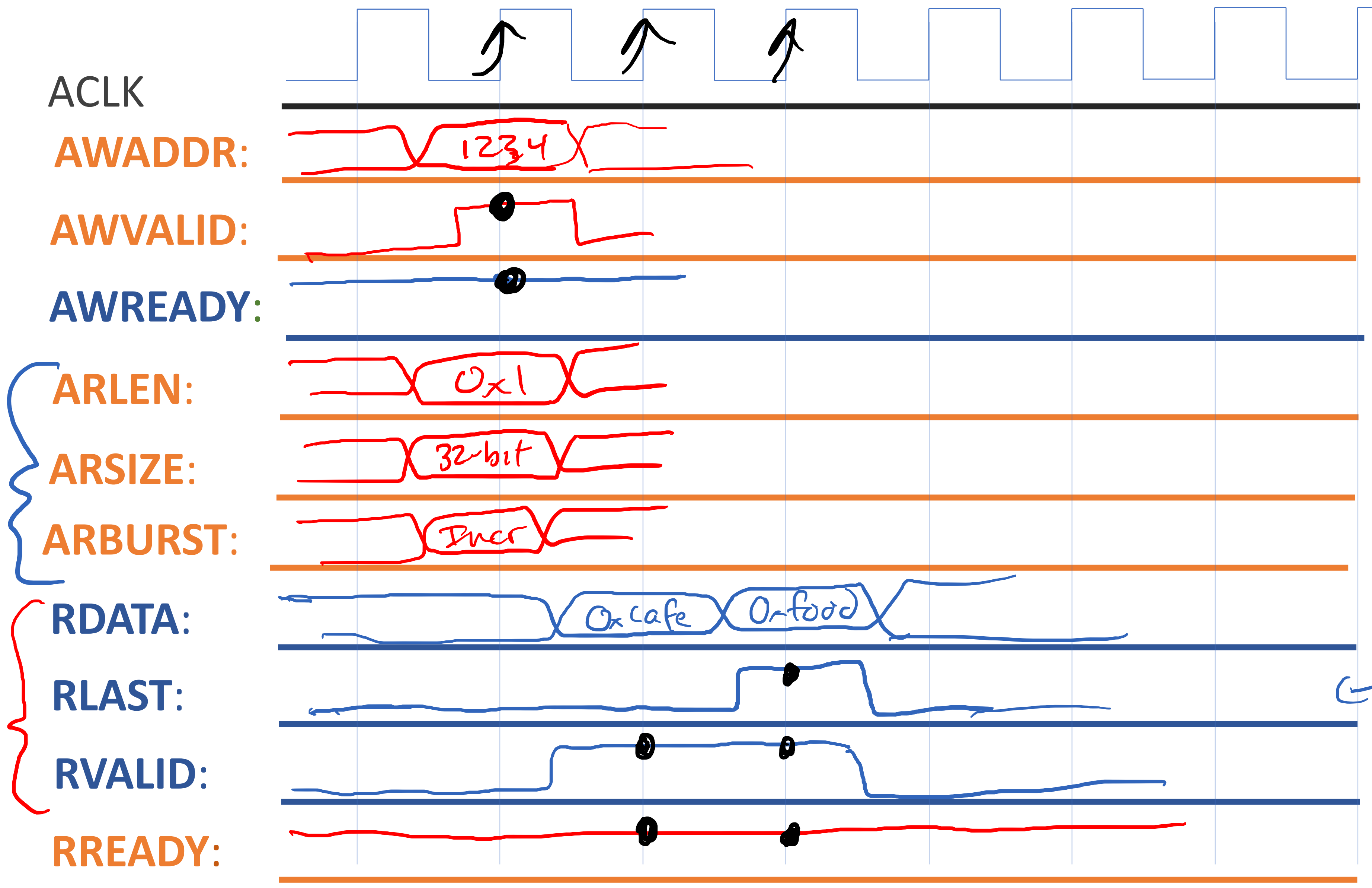
ARBURST: Read Address Burst Type

Are the addresses incrementing, or repeating?

RLAST: Read Last Burst Transfer (Blue)

Are we done yet?

Reading 0x1234 and 0x1238



Addr	Value
0x1234 ->	0xcafe
0x1238 ->	0xfood

(+1)

old 2·128 = 256 cycles

New: 1 + 128 = 129 cycles

indicates done

ARESETN and RRESP omitted

Read Burst Example

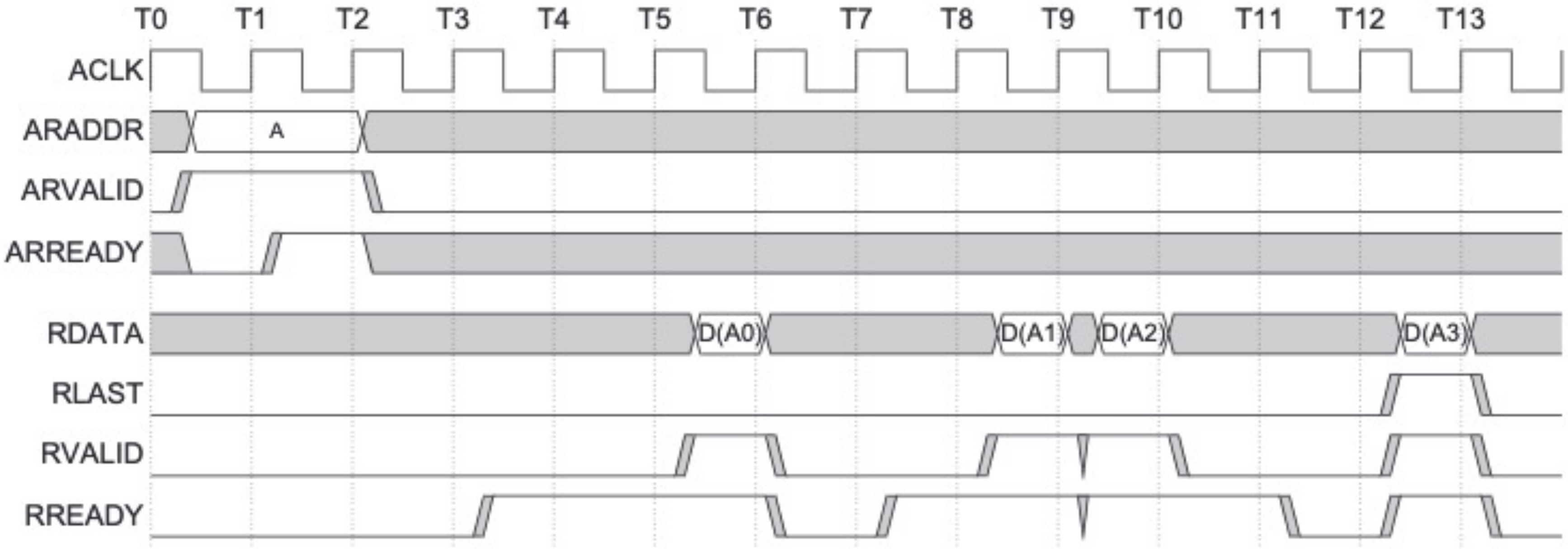


Figure 1-4 Read burst

Note

The master also drives a set of control signals showing the length and type of the burst, but these signals are omitted from the figure for clarity.

fixed = 0xa 0xa 0xa 0xa (Address)
Incr = 0xa 0xb 0xc 0xd

Overlapping Read Burst Transactions

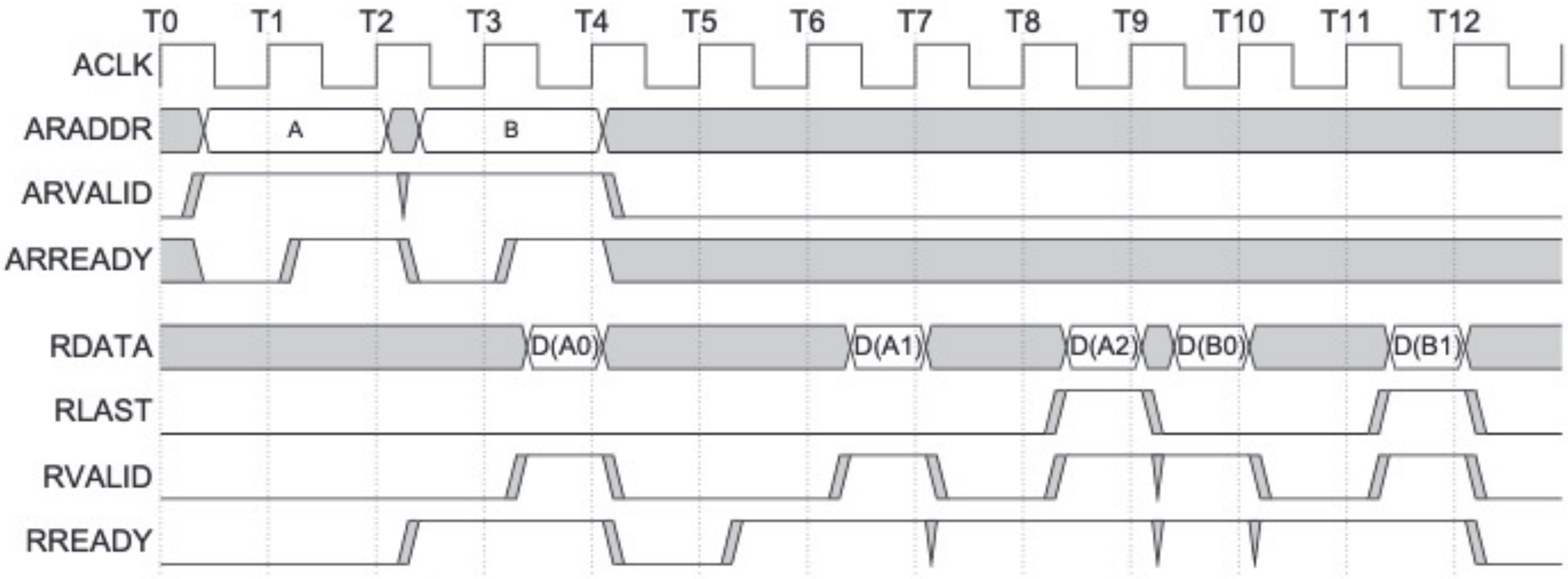


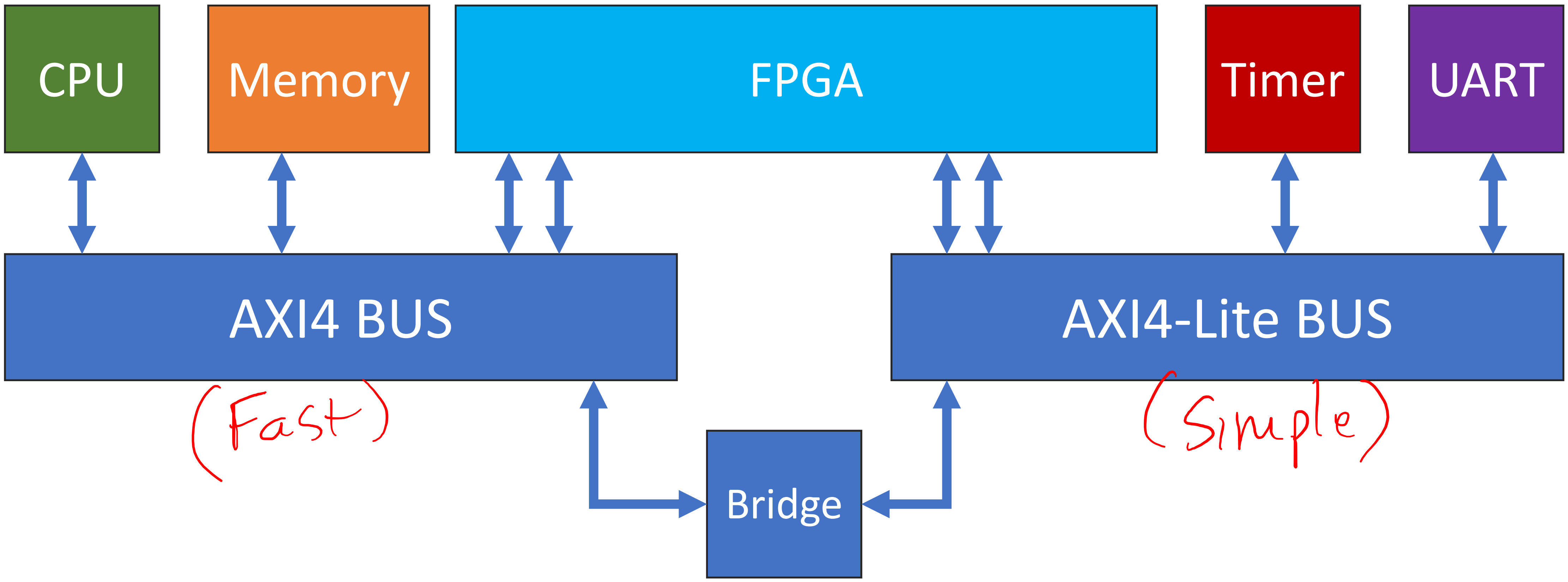
Figure 1-5 Overlapping read bursts

System’s Architecture

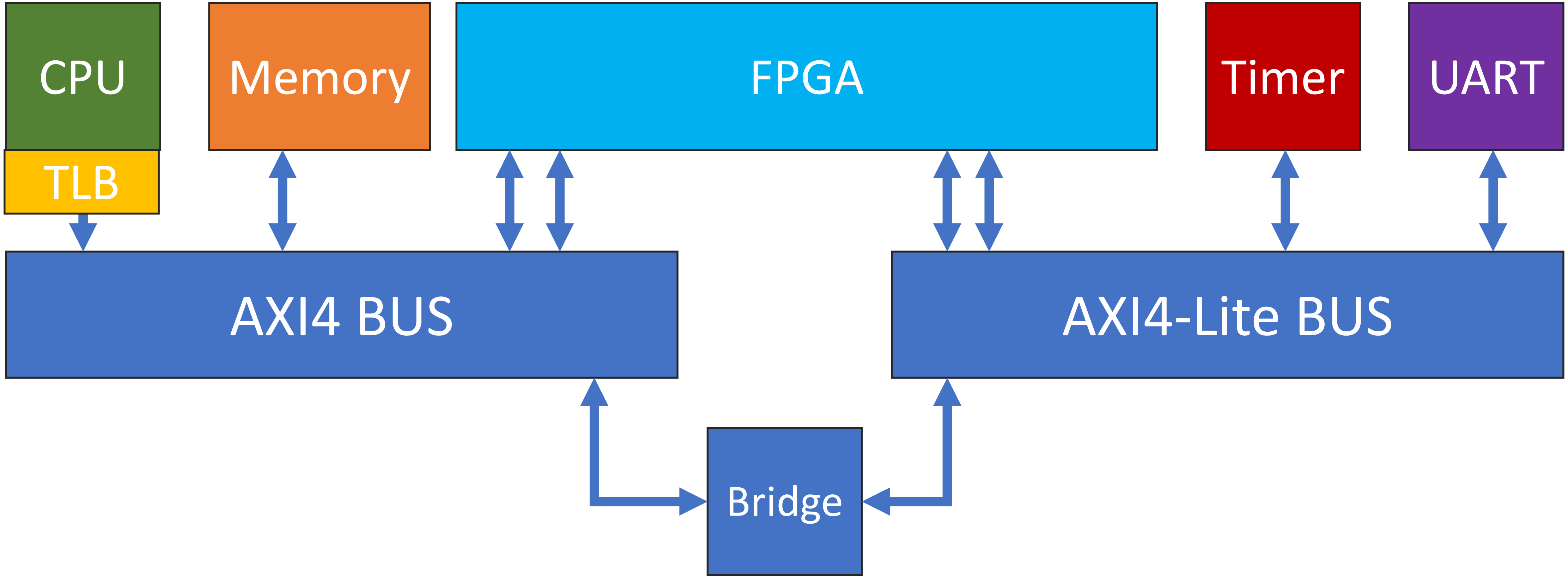
AXI4 BUS

AXI4-Lite BUS

System Architecture



Next Time: TLBs (CPU Memory Games!)



P3 Vivado Debug Time

References

- Zynq Book, Chapter 19 “AXI Interfacing”

- Practical Introduction to Hardware/Software Codesign
  - Chapter 10

- AMBA AXI Protocol v1.0
  - http://mazsola.iit.uni-miskolc.hu/~drdani/docs_arm/AMBAaxi.pdf

08: High-Performance Buses

Engr 315: Hardware / Software Codesign

Andrew Lukefahr

Indiana University