MIT SoC Design Competition

ONE BOARD, ONE MONTH, UNLIMITED POSSIBILITIES....

ARM® AMBA®3 AHB-Lite*

Karthik Shivashankar

ARM LTD., Cambridge, UK

About Me

- Sr. Engineer @ ARM Research
- Based in Cambridge, UK
- 4+ years at ARM

Image Source: Google.com

Before we start

- Have you worked on AMBA before ?

Image Source: Google.com

Why we need Architecture Spec ?

Why we need Architecture Spec ?

Agenda

- AMBA Family (2)
- Introduction to AHB-Lite (6)
- AHB-Lite System (8)
- AHB-Lite Signals – Master & Slave (15)
- AHB-Lite Transactions (9)
- How to build a simple AHB-Lite Slave (4)
- AHB-Lite – Multilayer Design (3)

Agenda

- AMBA Family (2)
- Introduction to AHB-Lite (6)
- AHB-Lite System (8)
- AHB-Lite Signals – Master & Slave (15)
- AHB-Lite Transactions (9)
- How to build a simple AHB-Lite Slave (4)
- AHB-Lite – Multilayer Design (3)

AMBA Family

- **AMBA-1**
- **AMBA-2**
  - **AHB**
- **AMBA-3**
  - **AHB-Lite**
- **AMBA-4**

- Acronyms
  - AMBA® → **A**dvanced **M**icrocontroller **B**us **A**rchitectures
  - AHB® → **A**dvanced **H**igh-Performance **B**us

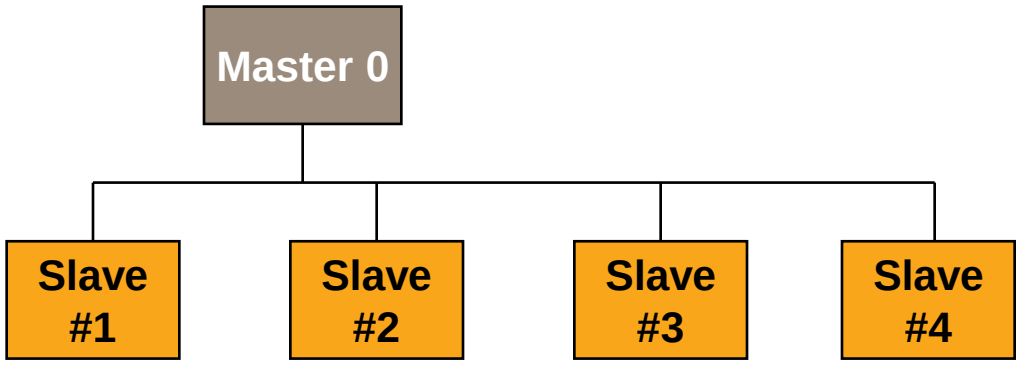
Image Source: Google.com

Agenda

- AMBA Family (2)
- Introduction to AHB-Lite (6)
- AHB-Lite System (8)
- AHB-Lite Signals – Master & Slave (15)
- AHB-Lite Transactions (9)
- How to build a simple AHB-Lite Slave (4)
- AHB-Lite – Multilayer Design (3)

Master-Slave Architecture

AHB-Lite



- Single Master
- Simple slaves
- Easier module design/debug
- No arbitration issues

Image Source: Walt Disney

AHB-Lite transactions

- Master
  - Register Read
  - Register Write
  - Burst Read
  - Burst Write
- Slave
  - Can make Master wait
  - Can give error response

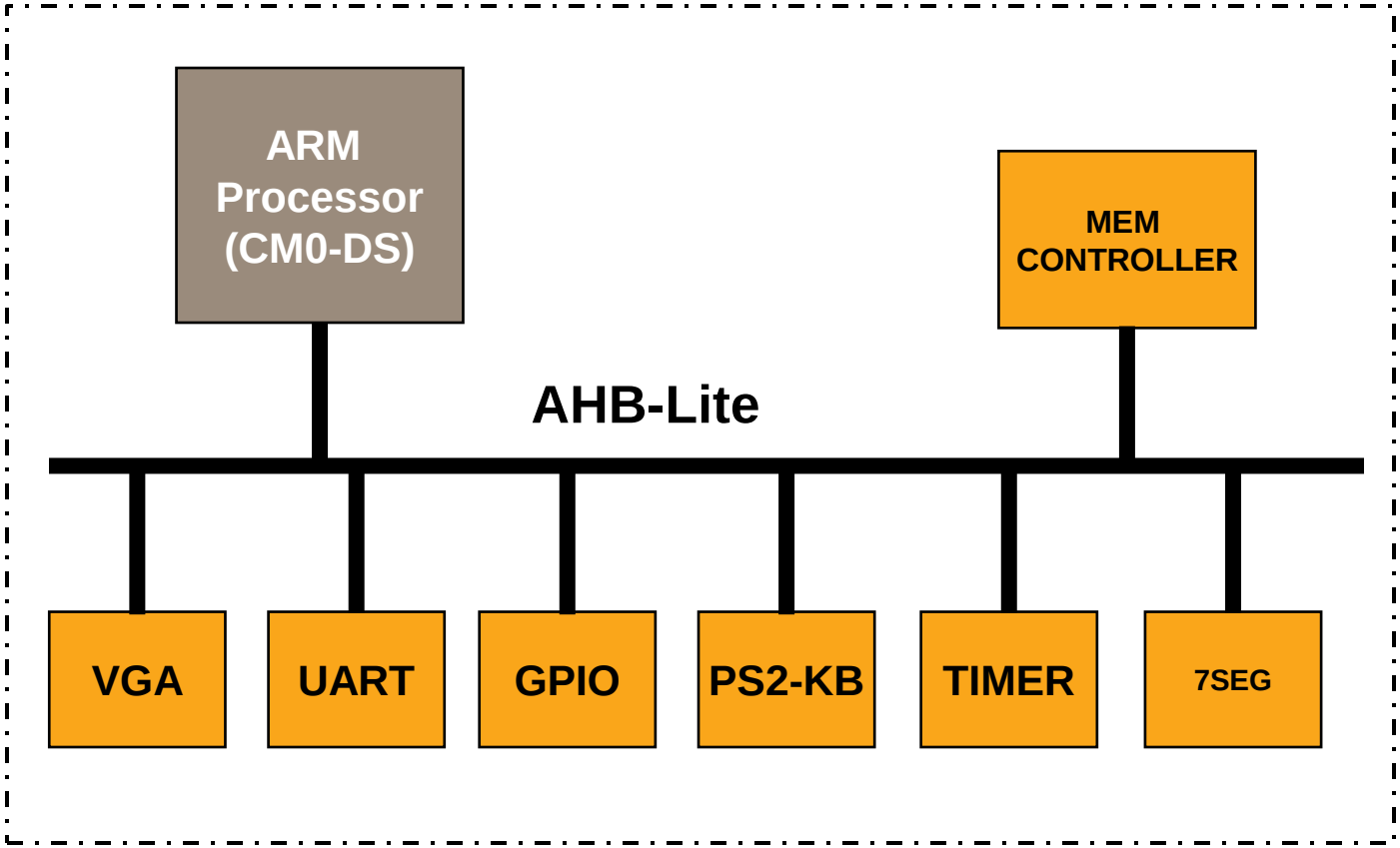
Image Source: Walt Disney

AHB-Lite Features

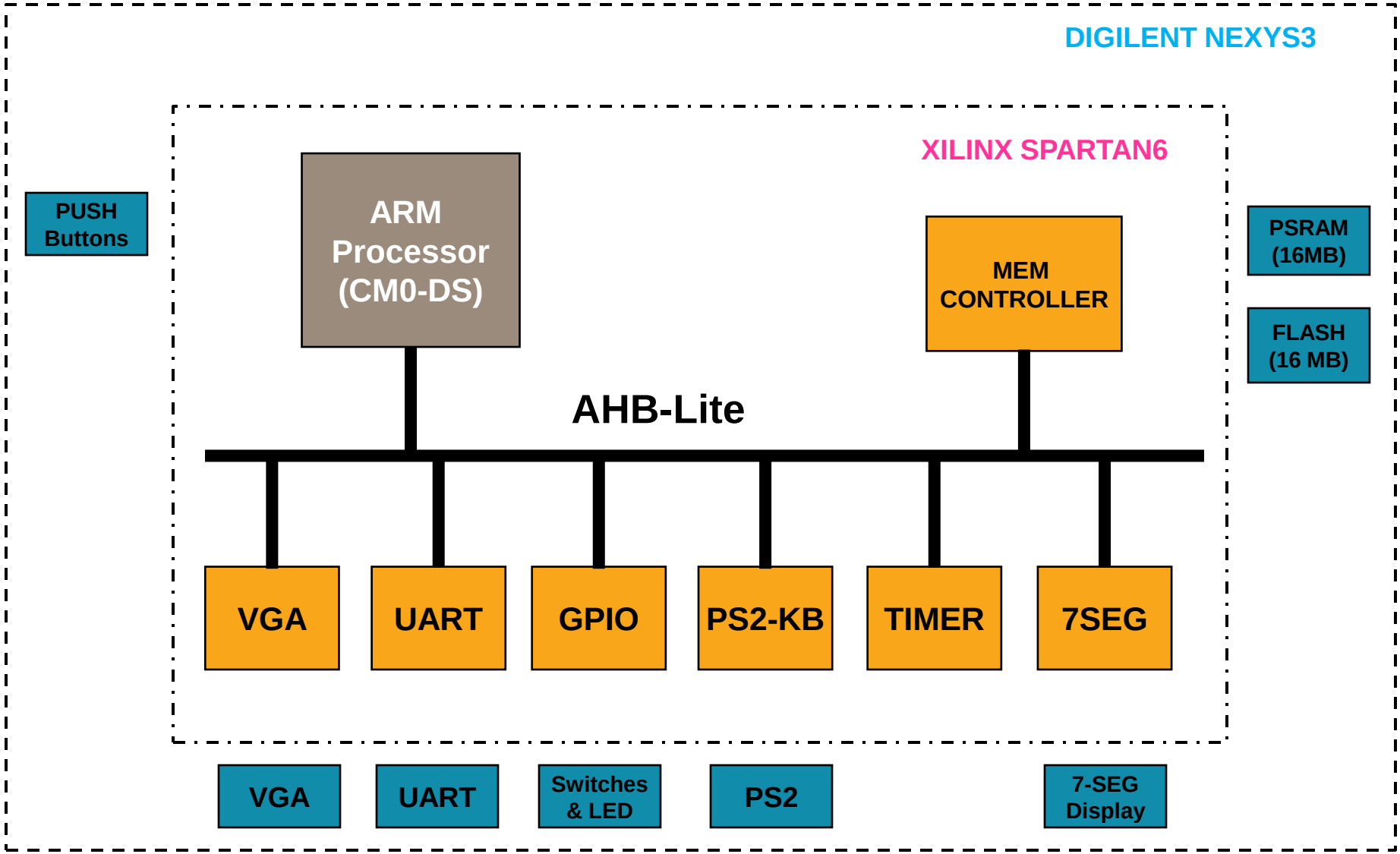
- Single Clock Edge operation
- Uni-directional busses
  - No tri-state signals
  - Good for synthesis
- Pipelined Operation

Image Source:: Walt Disney

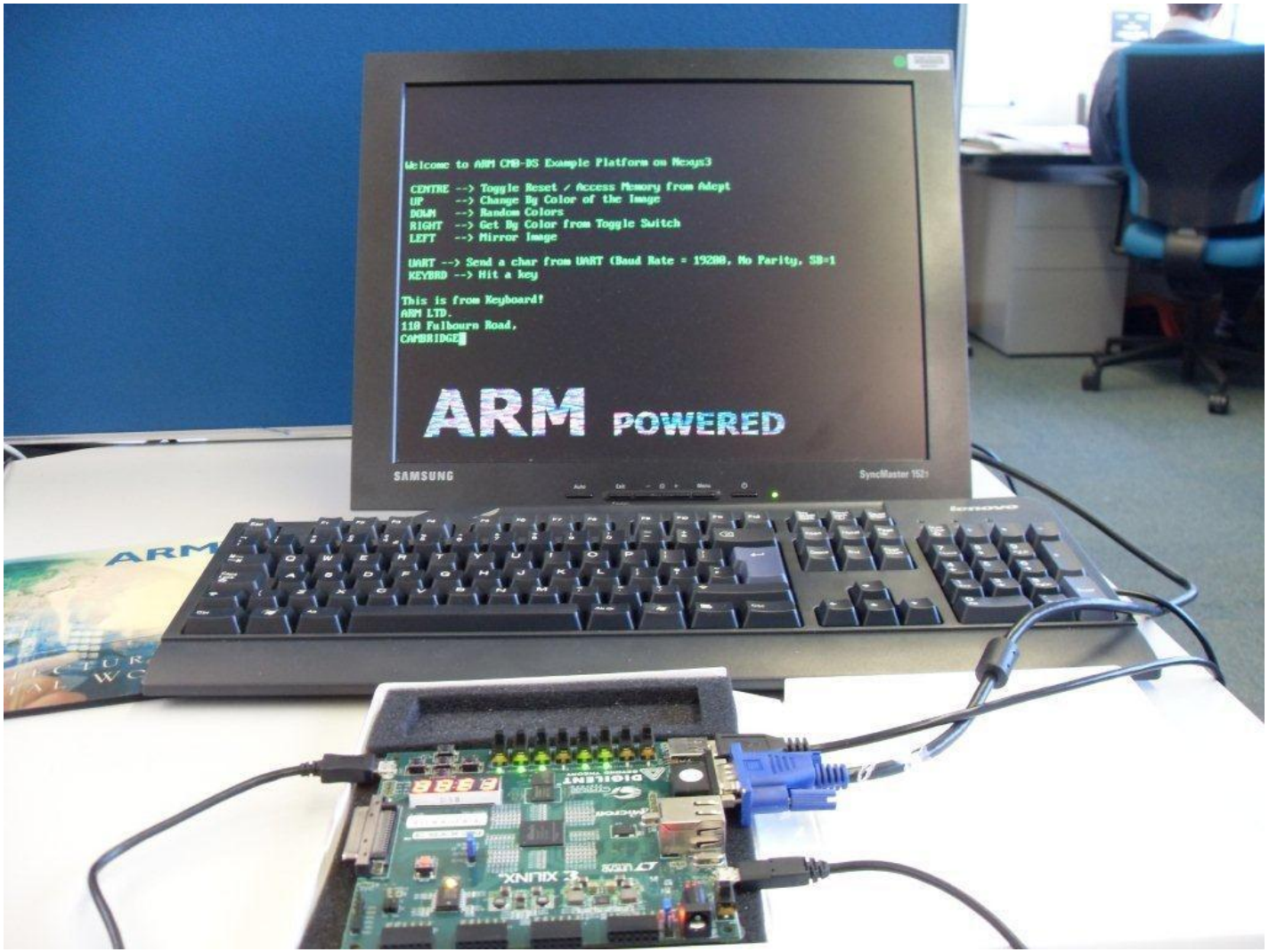
An Example AMBA AHB-Lite System



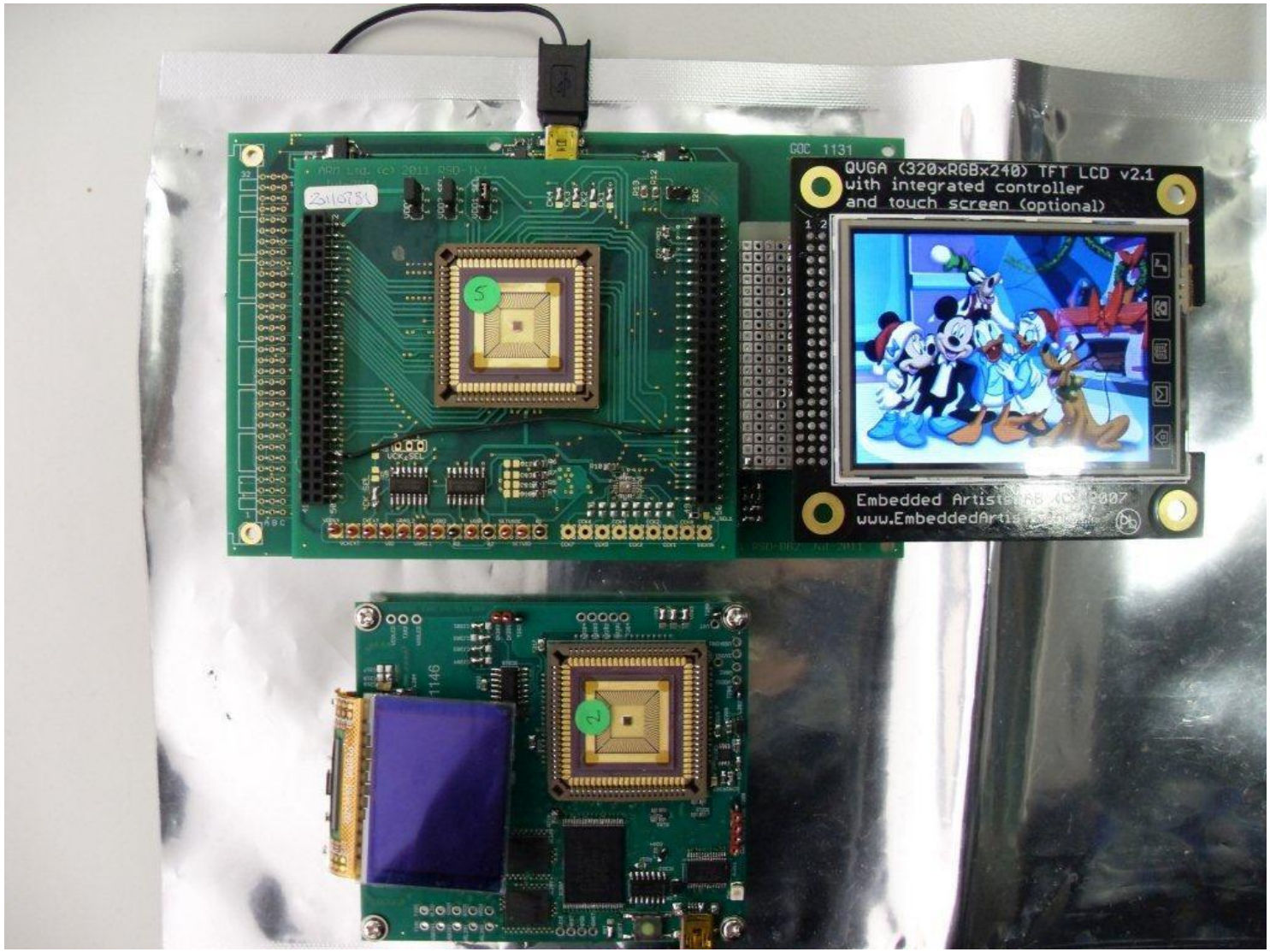
An Example AMBA AHB-Lite System



An Example AMBA AHB-Lite System



ARM R&D Testchip using CM0-DS



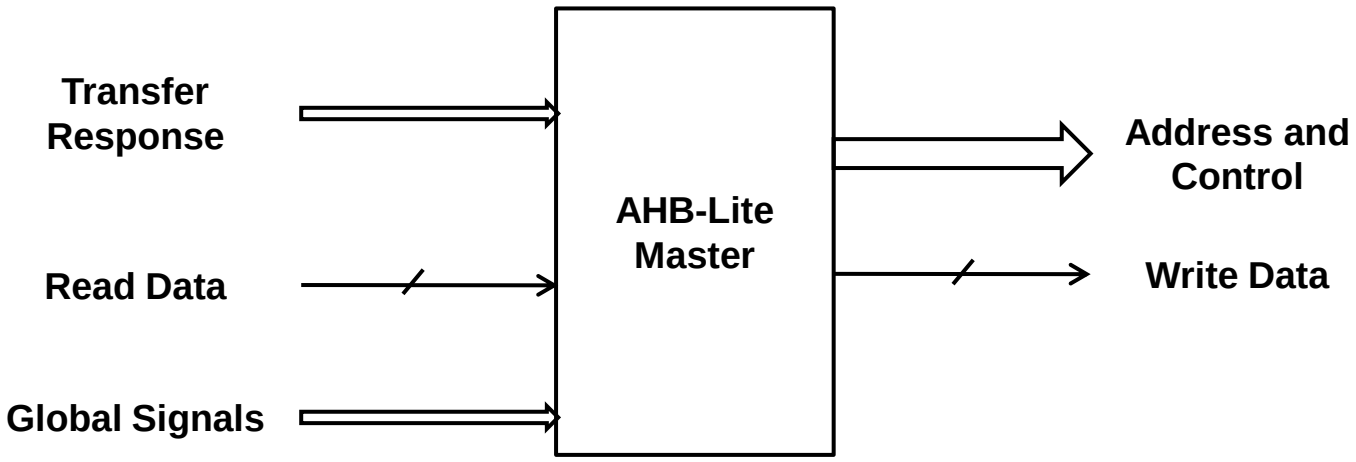
Agenda

- AMBA Family (2)
- Introduction to AHB-Lite (6)
- AHB-Lite System (8)
- AHB-Lite Signals – Master & Slave (15)
- AHB-Lite Transactions (9)
- How to build a simple AHB-Lite Slave (4)
- AHB-Lite – Multilayer Design (3)

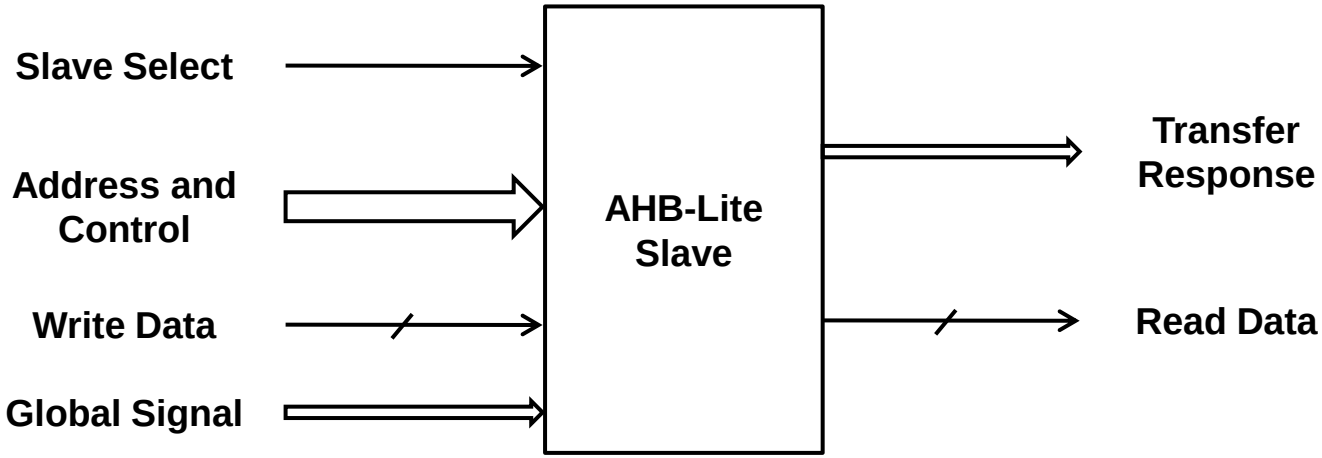
Main Components of AHB Lite System

- Master
- Slaves
- Address Decoder
- Multiplexor

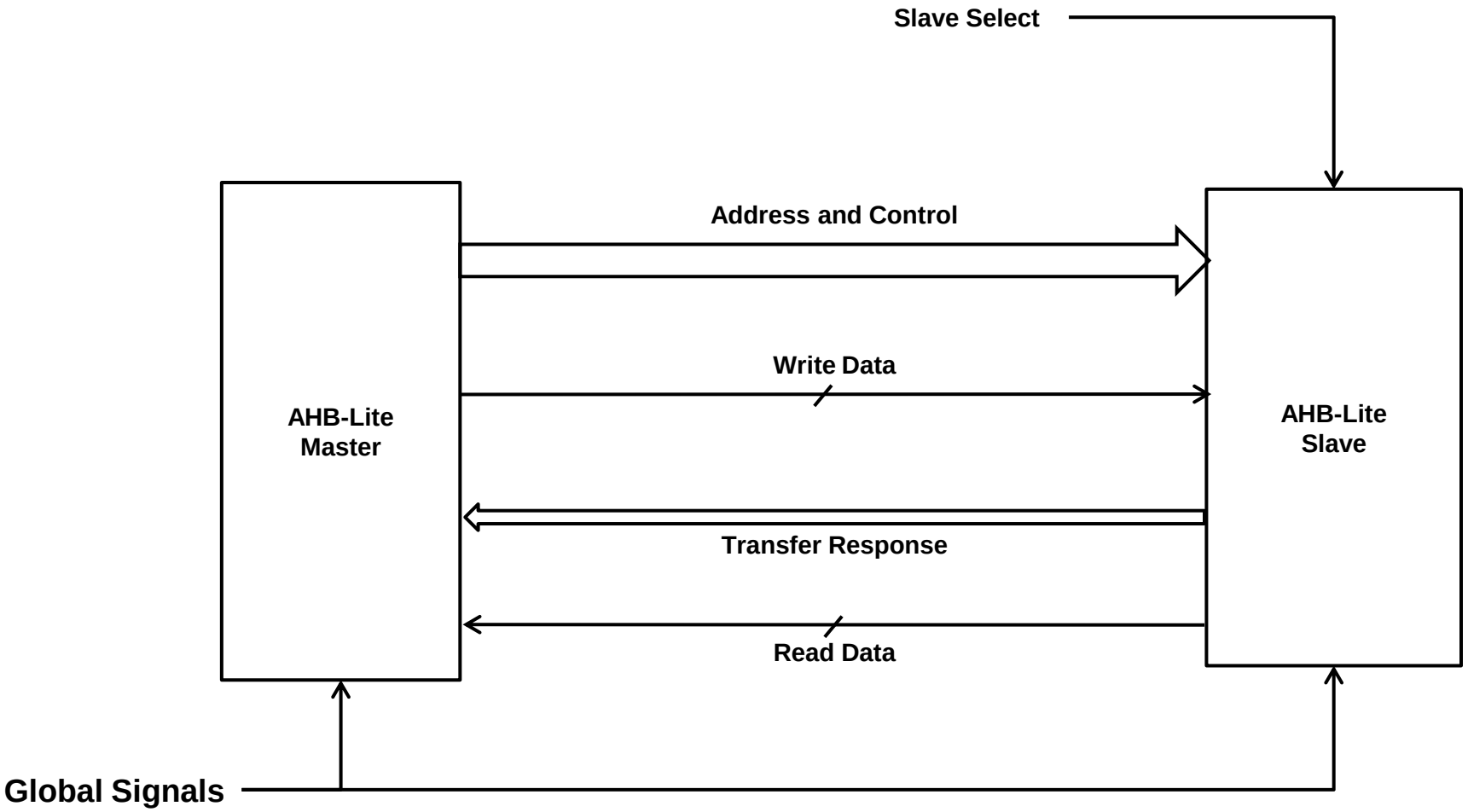
AHB-Lite Master



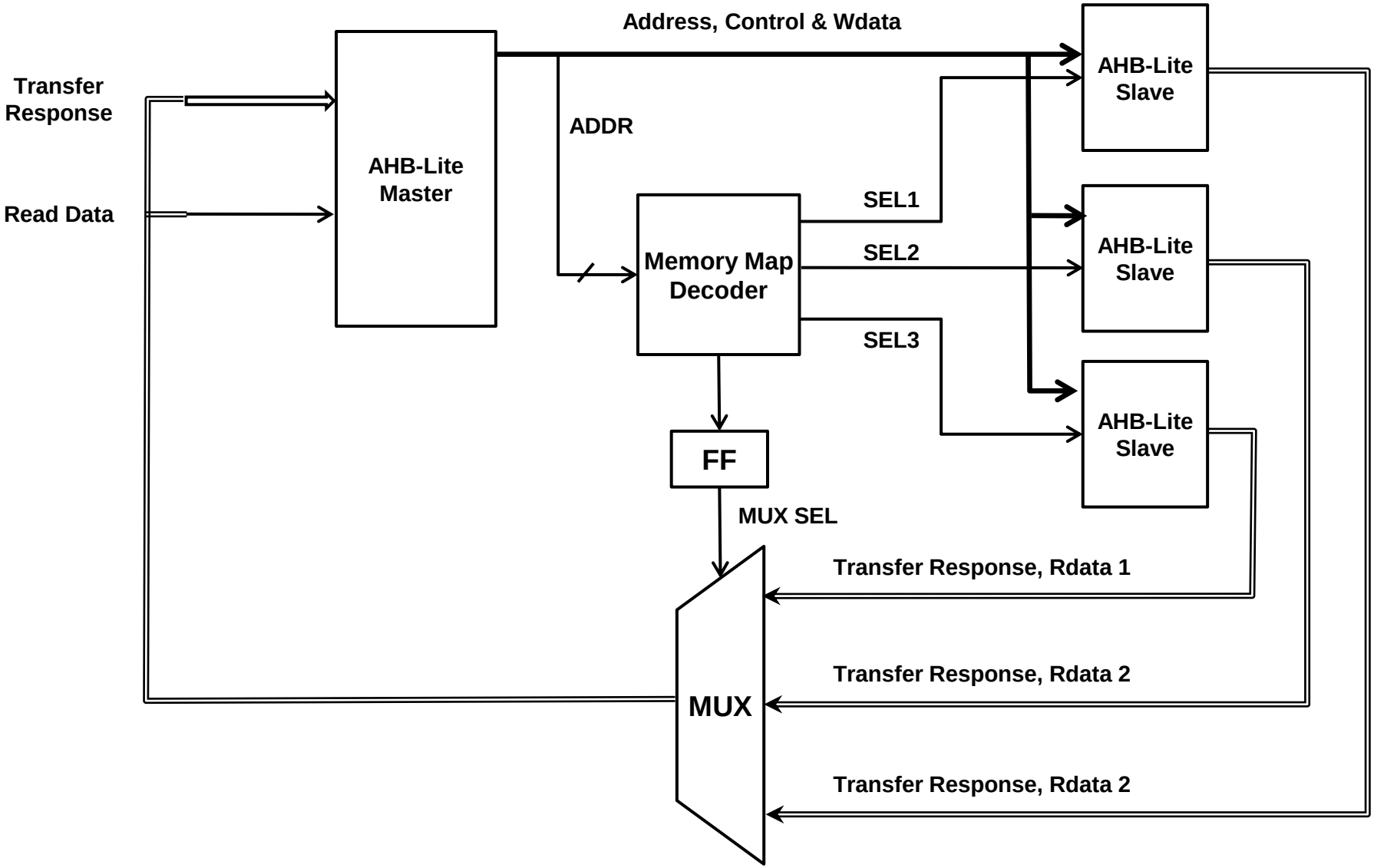
AHB-Lite Slave



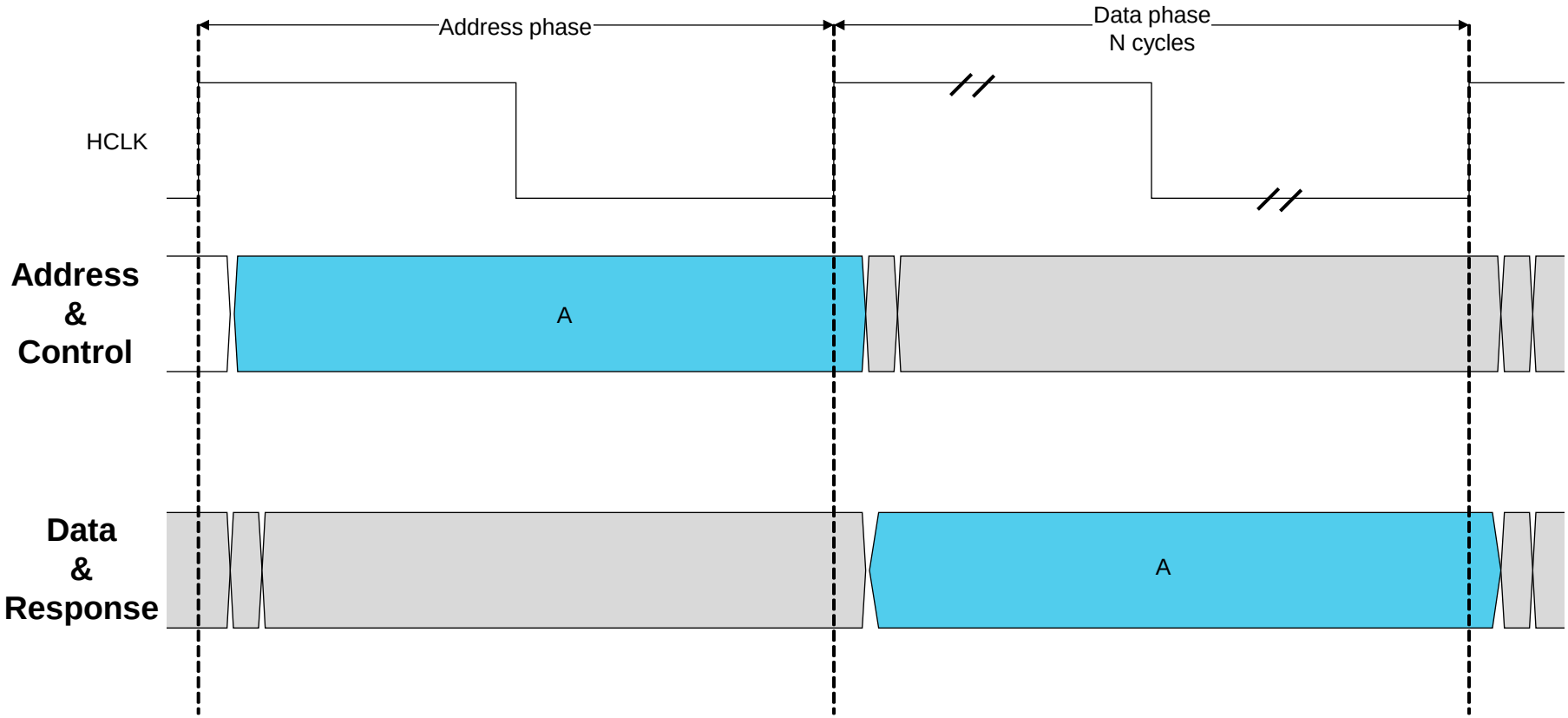
AHB-Lite Master & Slave



Memory Map Decoder & MUX



Pipelined Transactions (Conceptual Level)

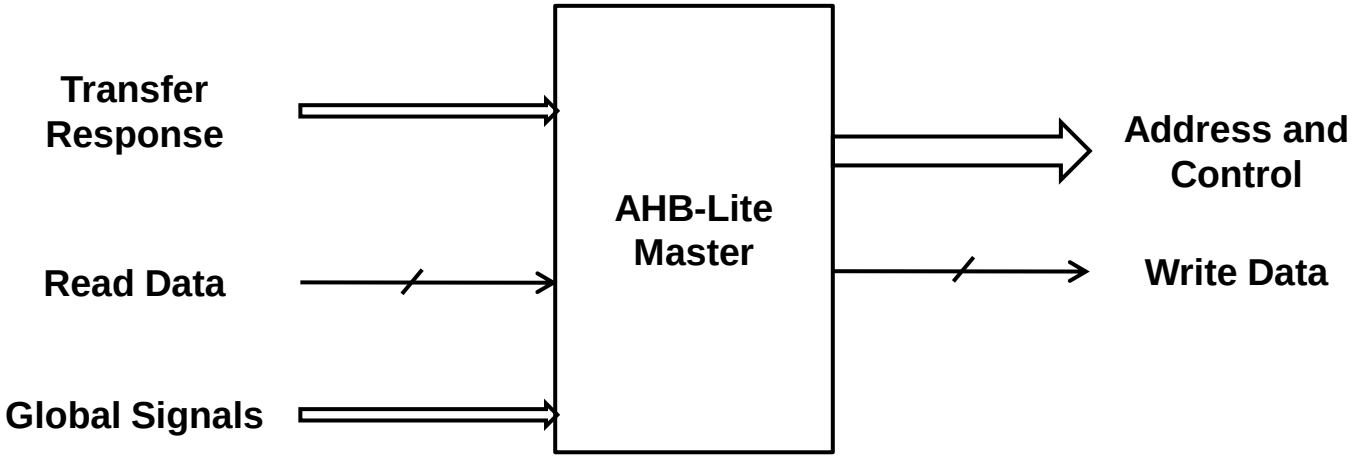


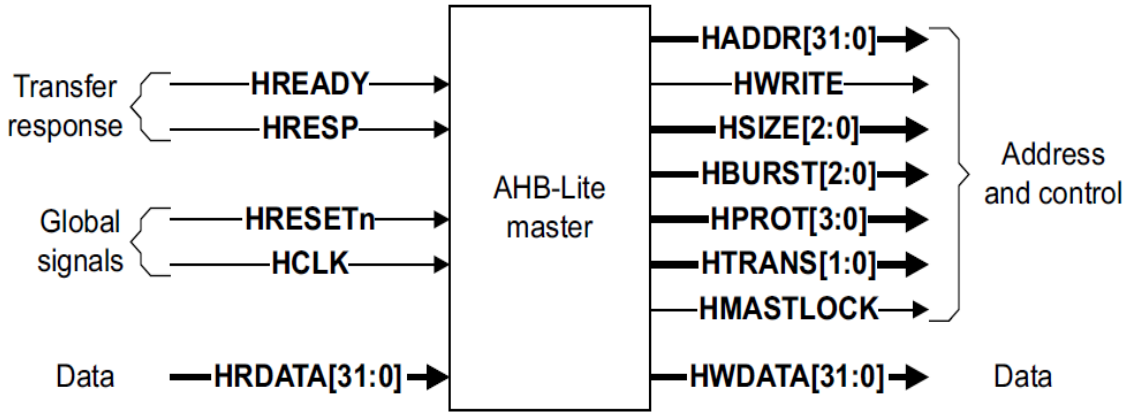
Memory Mapped Transactions: READ & WRITE

Agenda

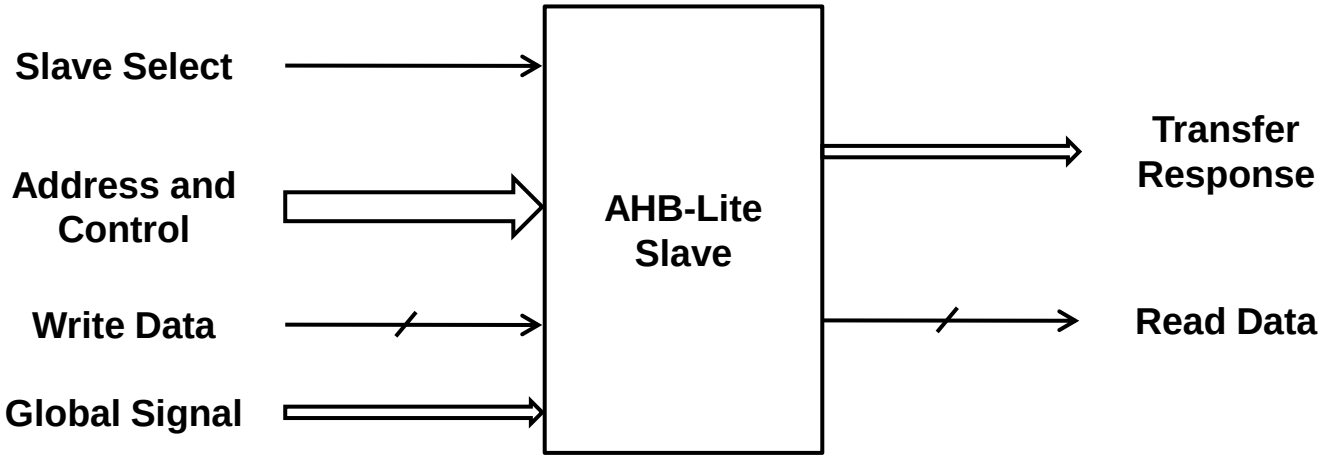
- AMBA Family (2)
- Introduction to AHB-Lite (6)
- AHB-Lite System (8)
- AHB-Lite Signals – Master & Slave (15)
- AHB-Lite Transactions (9)
- How to build a simple AHB-Lite Slave (4)
- AHB-Lite – Multilayer Design (3)

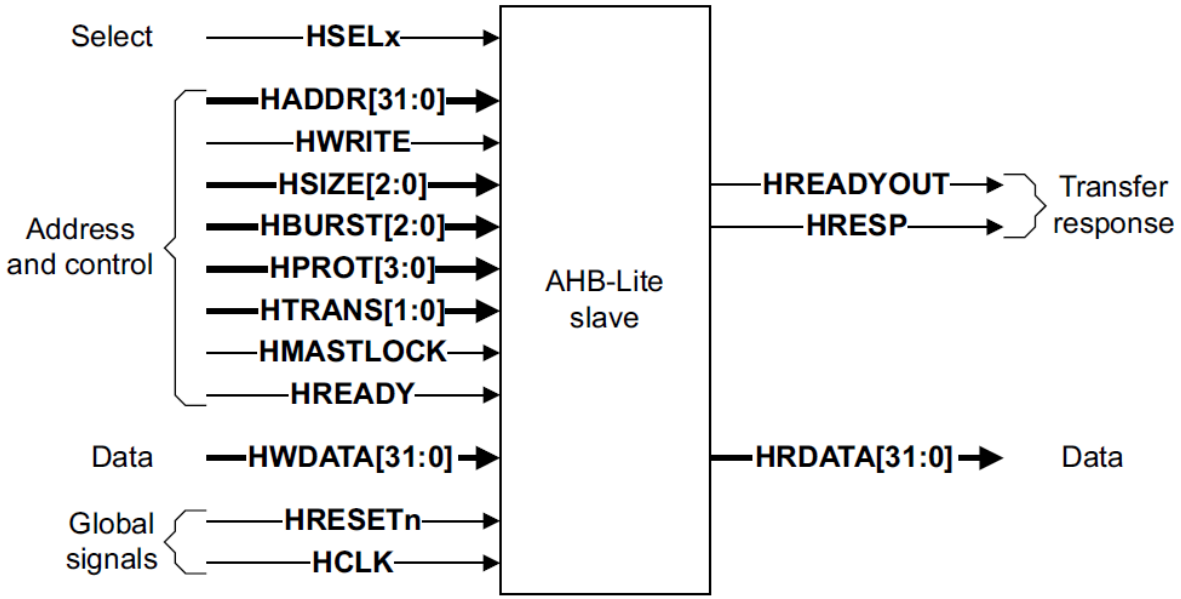
AHB-Lite Master Signals



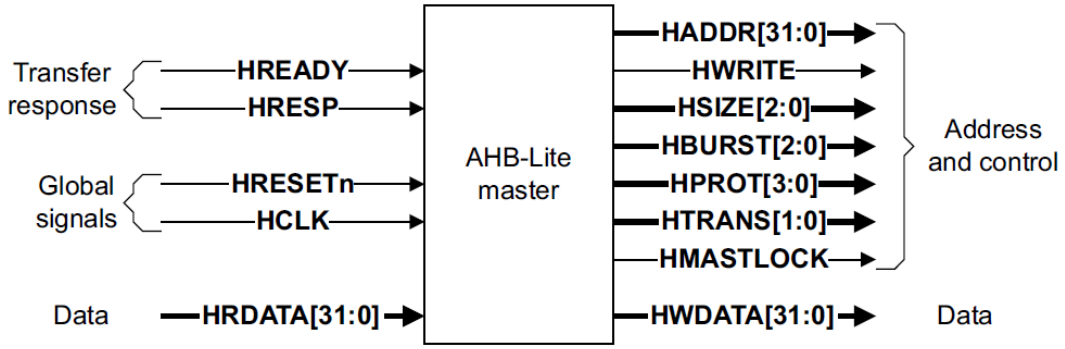


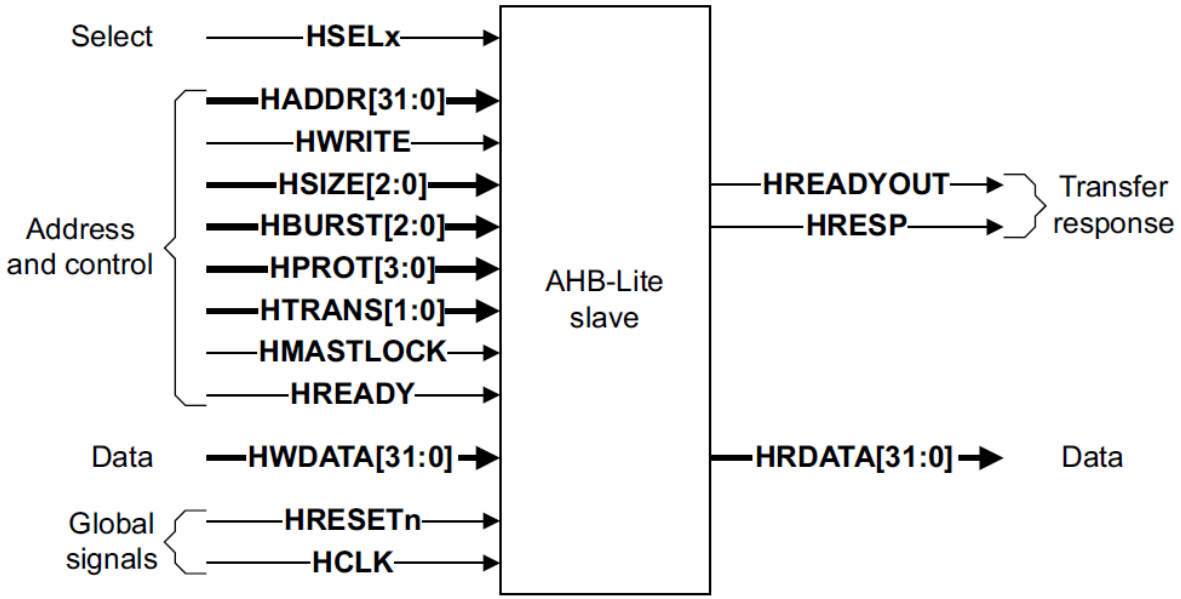
AHB-Lite Slave Signals



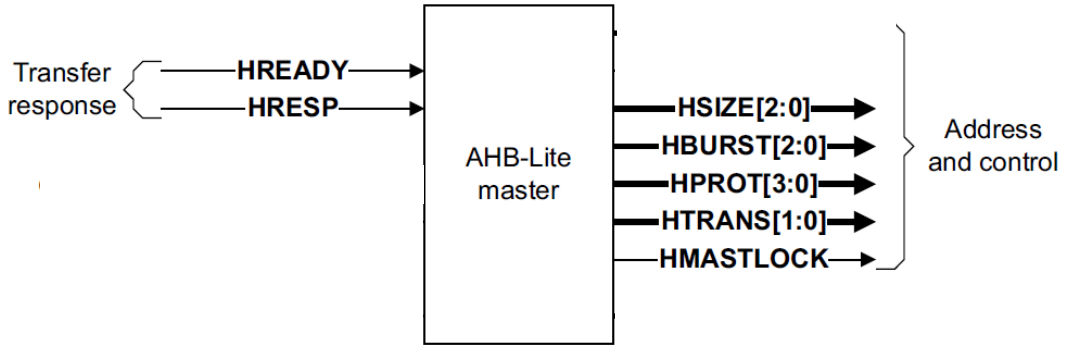


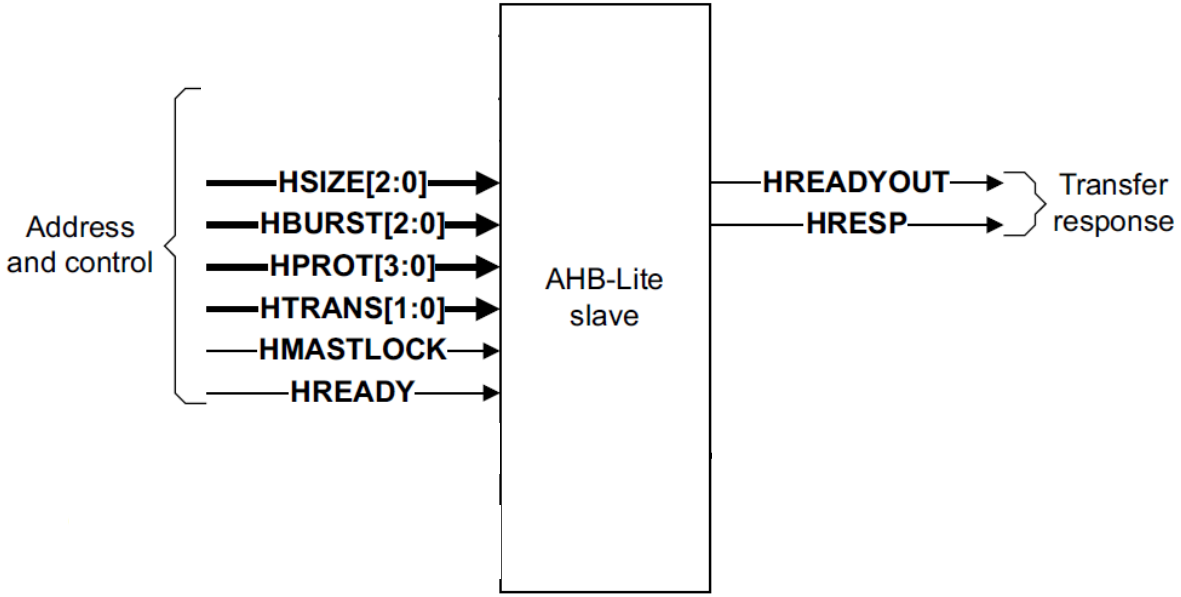
AHB-Lite Master & Slave





AHB-Lite Master & Slave

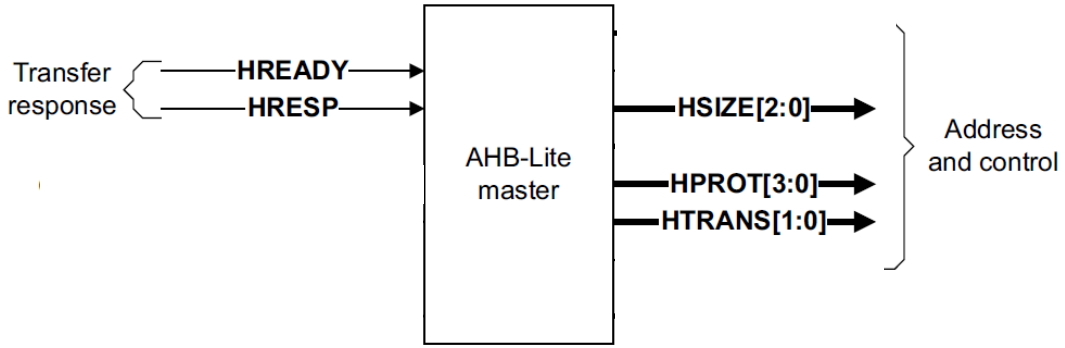


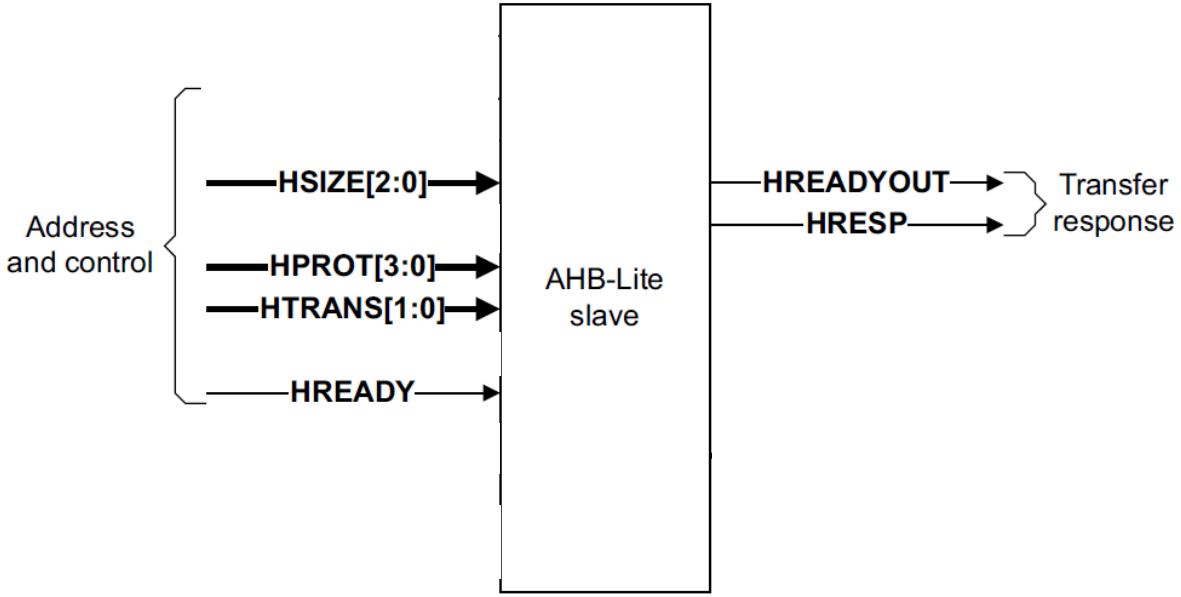


CM0-DS Doesn’t speak the entire language !

- CM0-DS do not generate BURST transaction
  - HBURST[2:0] is always 3’b000
- CM0-DS never generates locked transactions
  - HMASTLOCK is always 1’b0
- All transactions issued are non-sequential transfers
  - HTRANS[1:0] is either 2’b00 (IDLE) or 2’b10 (Non Sequential)

AHB-Lite Master & Slave





HSIZE[1:0]

Address-phase:		Data-phase:			
HSIZE [1:0]	**HADDR [1:0]**	**HxDATA [31:24]**	**HxDATA [23:16]**	**HxDATA [15:8]**	**HxDATA [7:0]**
00	00	-	-	-	Rd[7:0]
00	01	-	-	Rd[7:0]	-
00	10	-	Rd[7:0]	-	-
00	11	Rd[7:0]	-	-	-
01	00	-	-	Rd[15:8]	Rd[7:0]
01	10	Rd[15:8]	Rd[7:0]	-	-
10	00	Rd[31:24]	Rd[23:16]	Rd[15:8]	Rd[7:0]

HPROT[3:0] Protection Signal Encoding

HPROT[3] Cacheable	HPROT[2] Bufferable	HPROT[1] Privileged	HPROT[0] Data/Opcode	Description
-	-	-	0	Opcode fetch
-	-	-	1	Data access
-	-	0	-	User access
-	-	1	-	Privileged access
-	0	-	-	Non-bufferable
-	1	-	-	Bufferable
0	-	-	-	Non-cacheable
1	-	-	-	Cacheable

Master Generates these signals !
Slaves have the freedom to ignore !!

HPROT[3:2] For CM0-DS

HADDR[31:0]	Type	HPROT[3:2]	Recommended usage
32'hF0000000 – 32'hFFFFFFFF	Device	01	None.
32'hE0000000 – 32'hEFFFFFFF	Reserved	-	Maps to the processor's internal peripherals, for example NVIC.
32'hA0000000 – 32'hDFFFFFFF	Device	01	Peripherals.
32'h80000000 – 32'h9FFFFFFF	Normal (write-through)	10	Off chip RAM.
32'h60000000 – 32'h7FFFFFFF	Normal (write-back write-allocate)	11	Off chip RAM.
32'h40000000 – 32'h5FFFFFFF	Device	01	Peripherals.
32'h20000000 – 32'h3FFFFFFF	Normal (write-back write-allocate)	11	On chip RAM.
32'h00000000 – 32'h1FFFFFFF	Normal (write-through)	10	Program code.

HTRANS[1:0]

HTRANS	Type	Description
00	IDLE	Master does not wish to perform a transfer
01	BUSY	Bus Master is in the middle of a burst but cannot immediately continue with the next transfer
10	NON-SEQ	Indicates the first transfer of a burst or a single transfer
11	SEQ	The remaining transfers in the burst are sequential address steps from the previous transfer. Step size is that of data width of transfer (which is shown by HSIZE)

CM0-DS Always generates NON-SEQ Transactions

Transactions

Transaction	Access
HTRANS[1:0] = 2'b00	IDLE
HTRANS[1:0] = 2'b10 HPROT[0] = 1'b0 HSIZE[1:0] = 2'b10 HWRITE = 1'b0	FETCH

Transaction	Access
HTRANS[1:0] = 2'b10 HPROT[0] = 1'b1 HSIZE[1:0] = 2'b00	BYTE
HTRANS[1:0] = 2'b10 HPROT[0] = 1'b1 HSIZE[1:0] = 2'b01	HALF-WORD
HTRANS[1:0] = 2'b10 HPROT[0] = 1'b1 HSIZE[1:0] = 2'b10	WORD

Control Signals Recap

HTRANS[1:0]
- IDLE
- BUSY
- NONSEQ
- SEQ

HBURST[2:0]
- SINGLE
- INCR
- WRAP[4|8|16]
- INCR[4|8|16]

HMASTLOCK
- UNLOCKED
- LOCKED

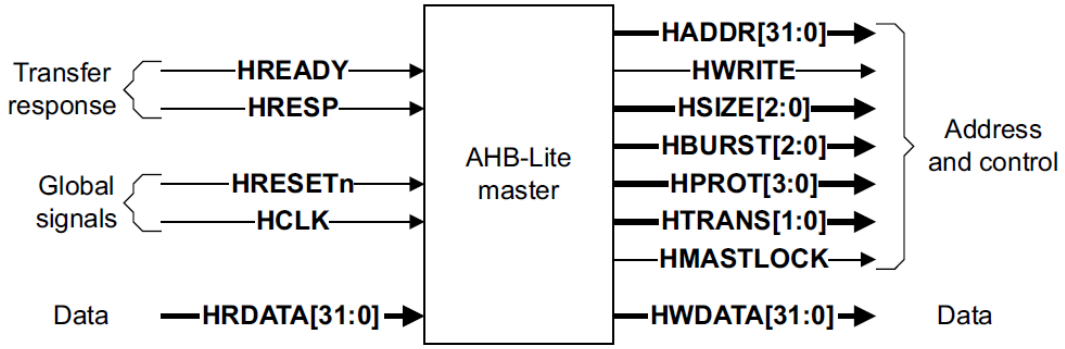
HSIZE[2:0]
- Byte
- Halfword
- Word
- Doubleword
- ...

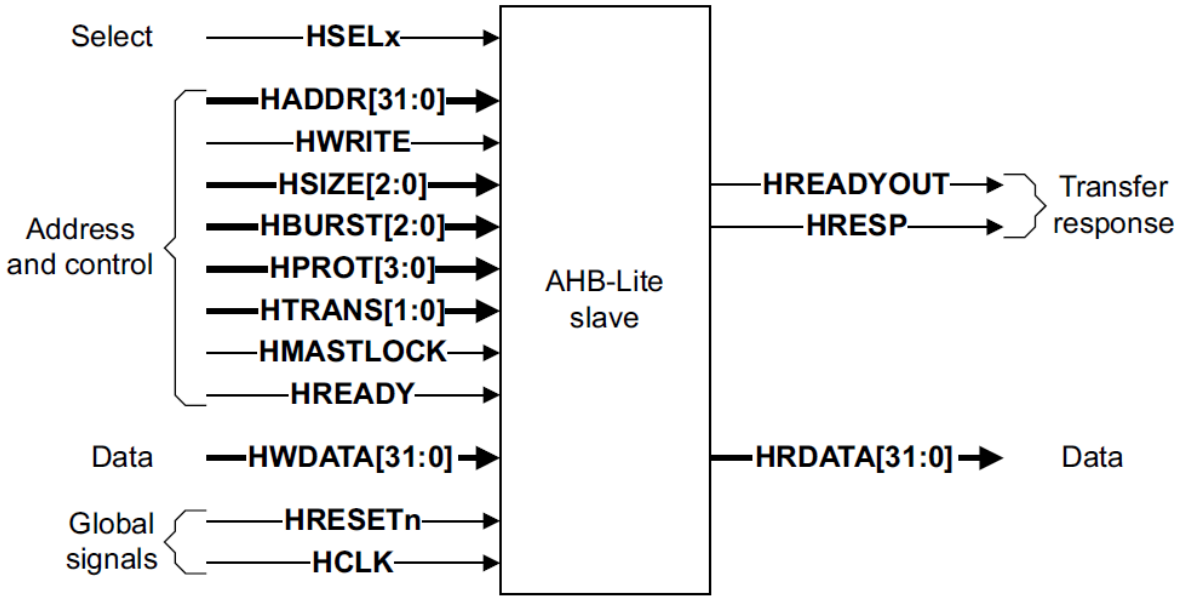
HPROT[3:0]
- Data/Opcode
- Privileged/user
- Bufferable
- Cacheable

Transfer Response Signals

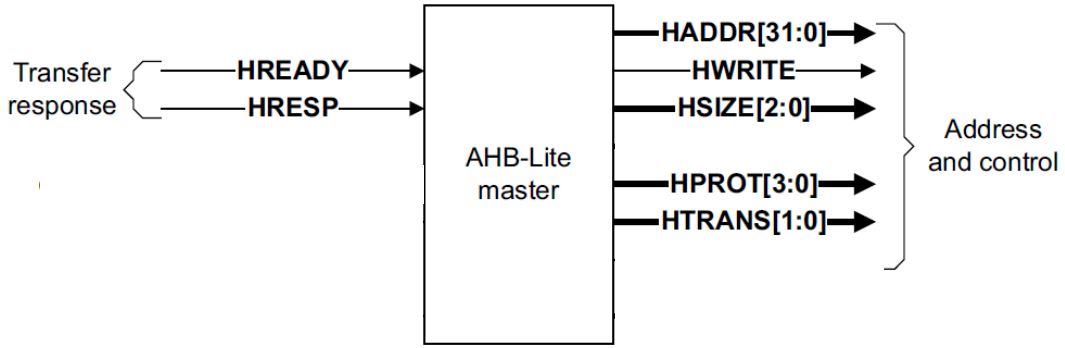
HREADYOUT	Multiplexor	When HIGH, the **HREADYOUT** signal indicates that a transfer has finished on the bus. This signal can be driven LOW to extend a transfer.
HRESP	Multiplexor	The transfer response, after passing through the multiplexor, provides the master with additional information on the status of a transfer. When LOW, the **HRESP** signal indicates that the transfer status is OKAY. When HIGH, the **HRESP** signal indicates that the transfer status is ERROR.

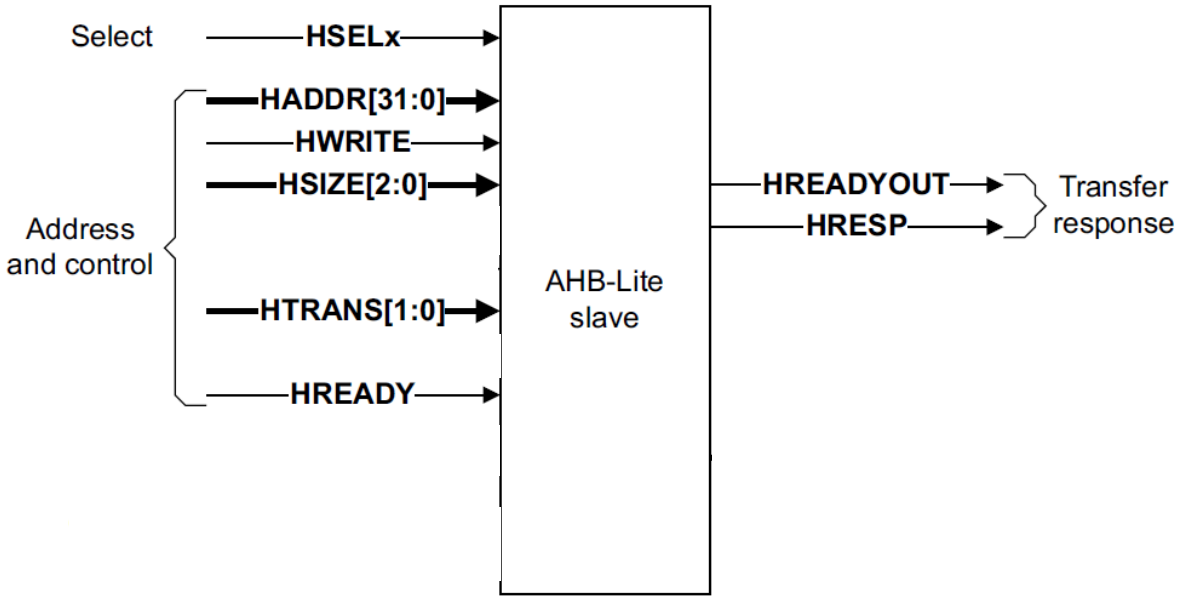
AHB-Lite Master & Slave





Control Signals to Care about

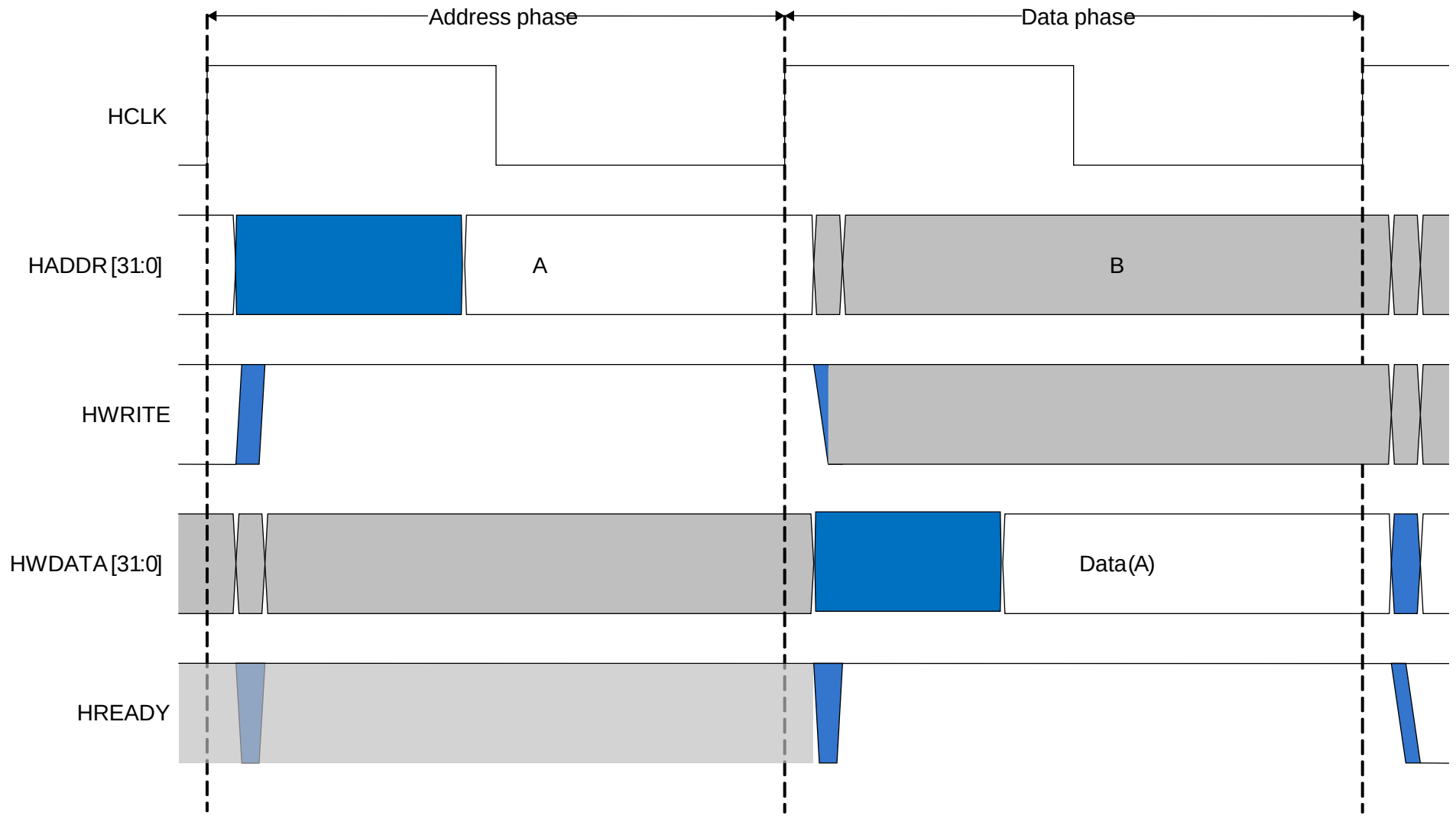




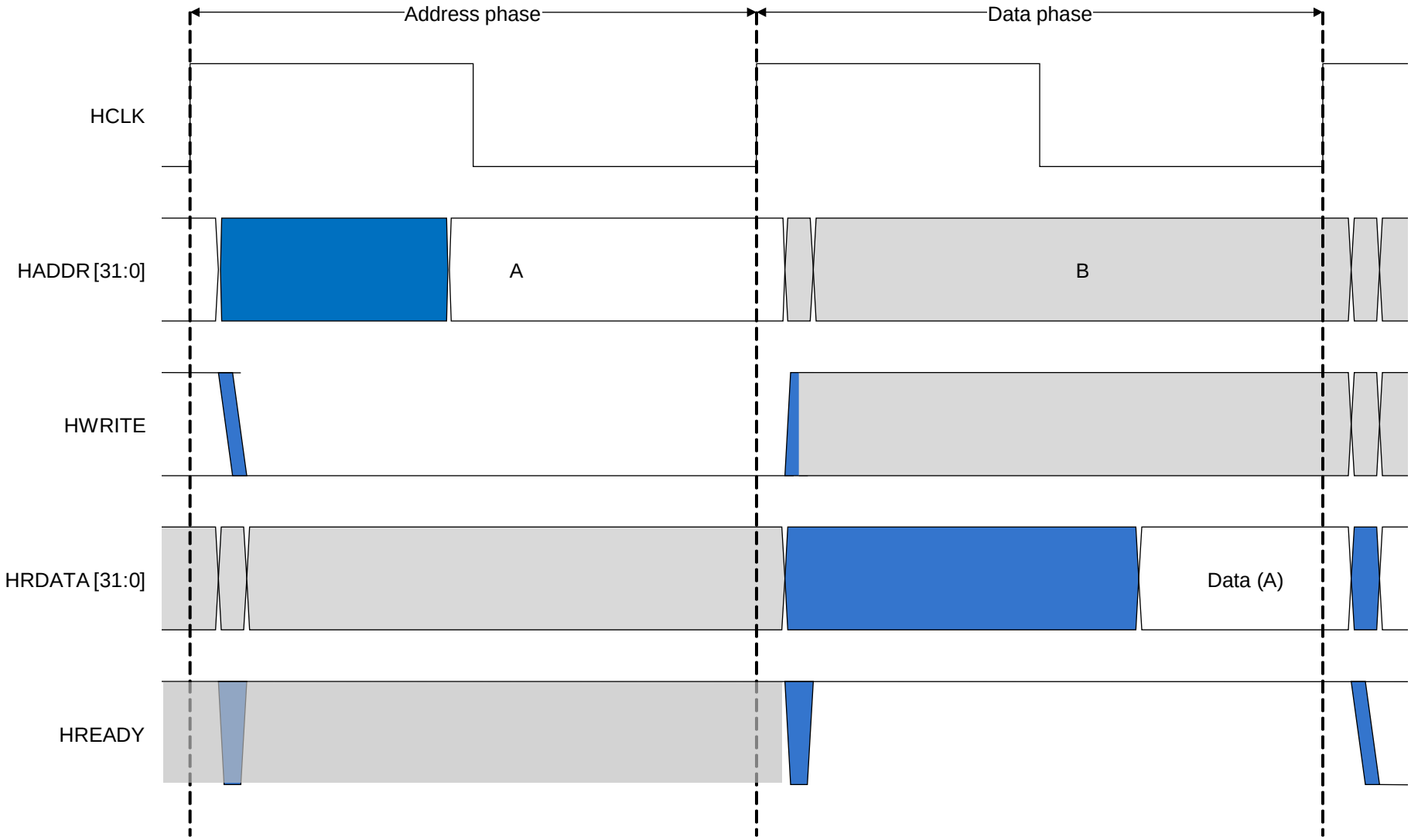
Agenda

- AMBA Family (2)
- Introduction to AHB-Lite (6)
- AHB-Lite System (8)
- AHB-Lite Signals – Master & Slave (15)
- AHB-Lite Transactions (9)
- How to build a simple AHB-Lite Slave (4)
- AHB-Lite – Multilayer Design (3)

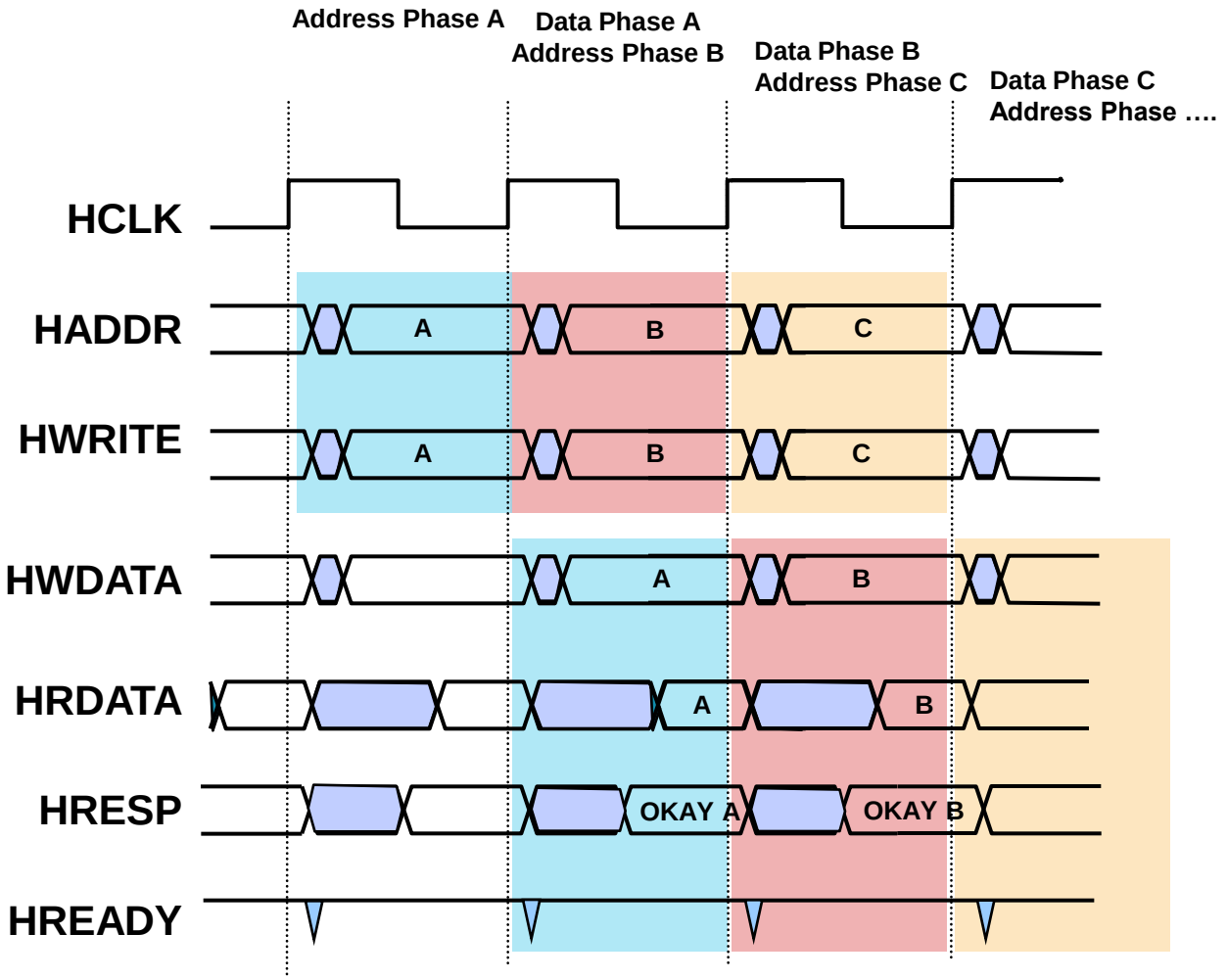
Basic transfer - Write



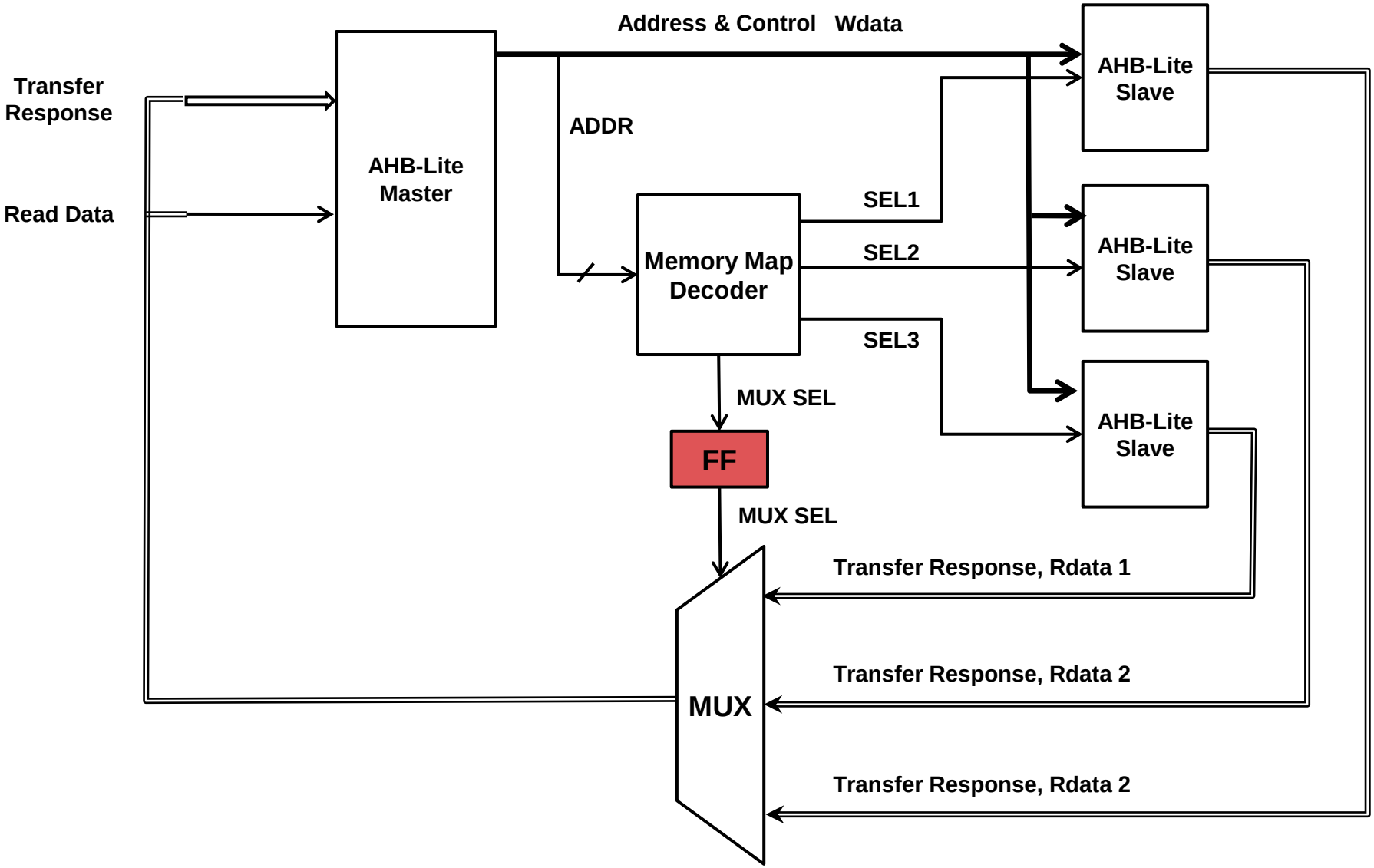
Basic transfer - Read



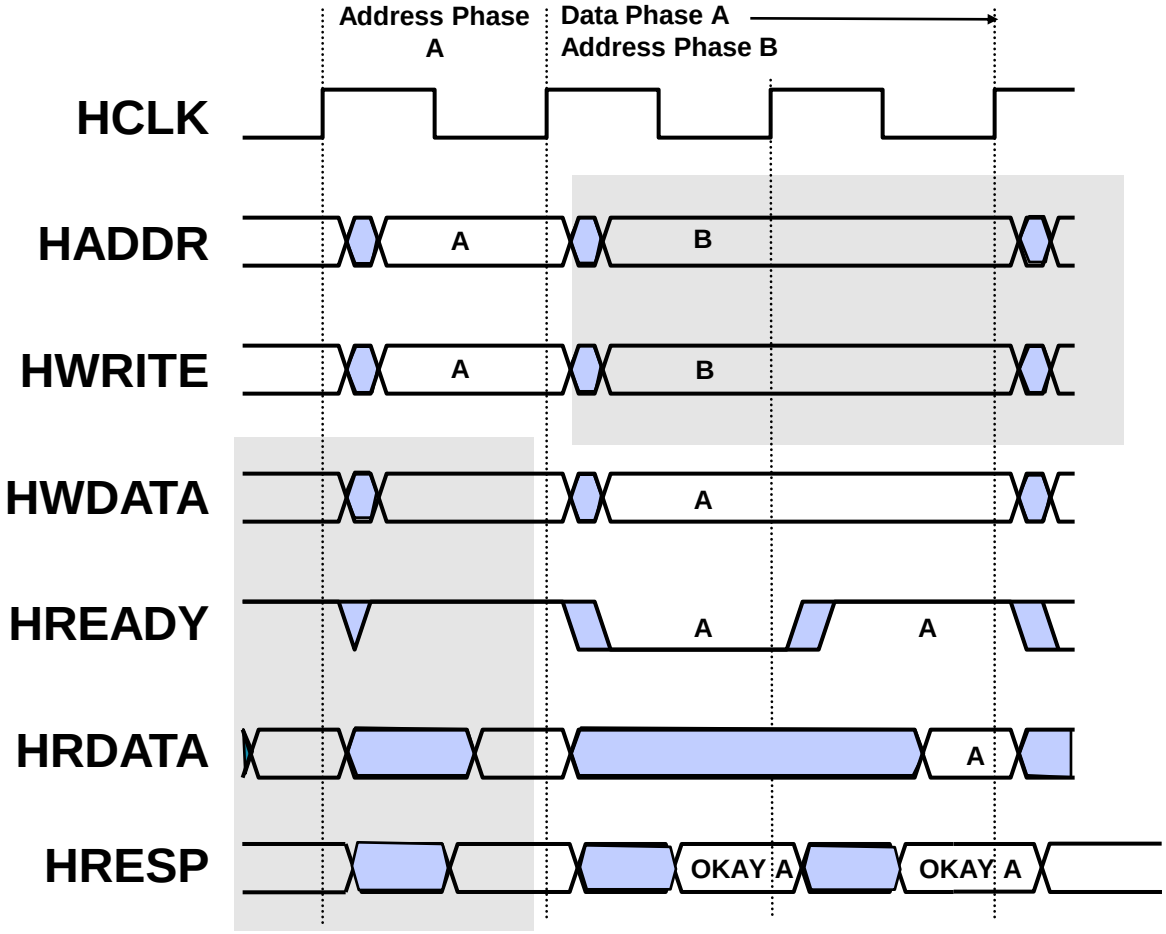
AHB Pipelined Transaction



Pipelined Operation

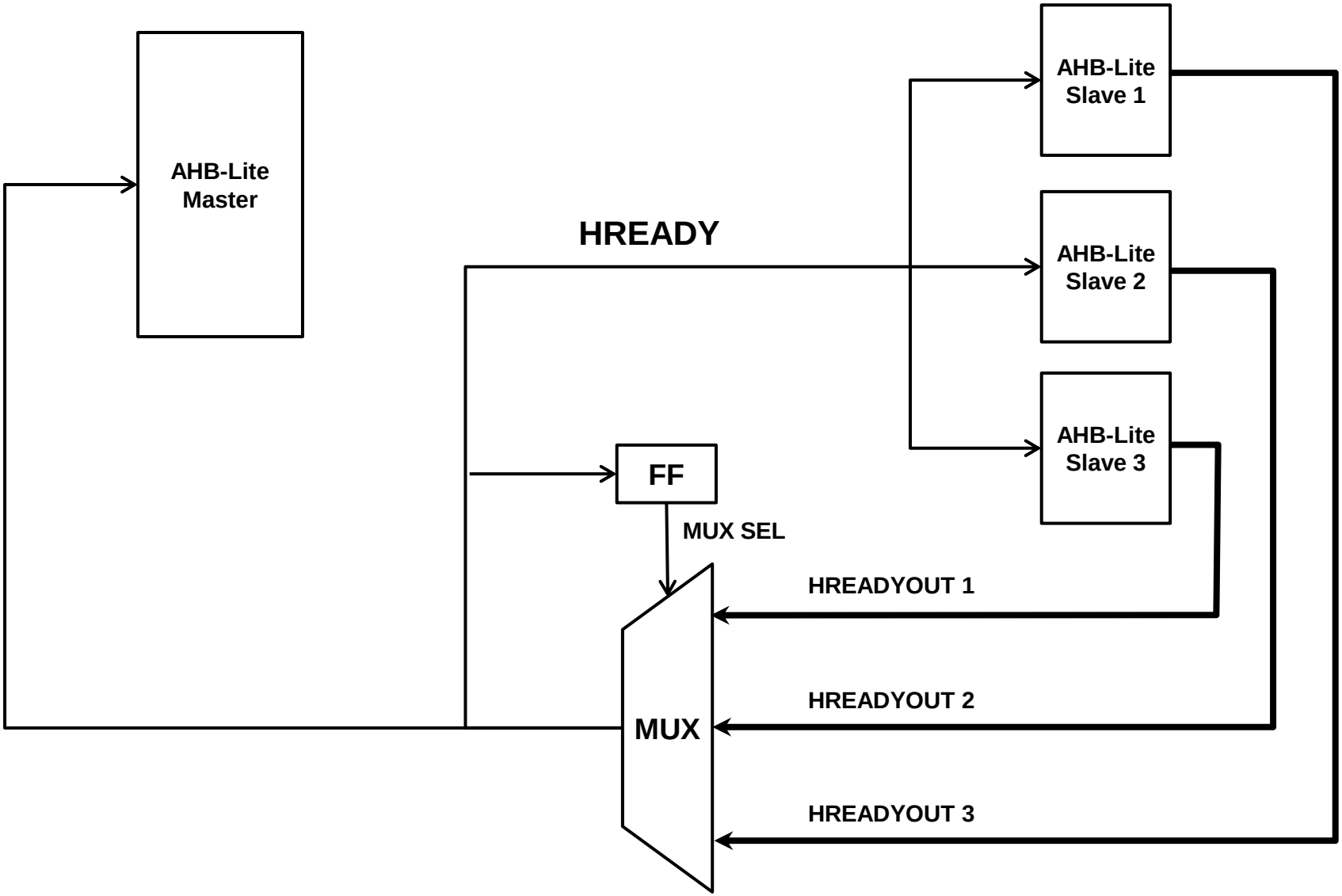


AHB basic signal timing – Adding wait states



Master will extend Address Phase B

HREADY (Inform all)

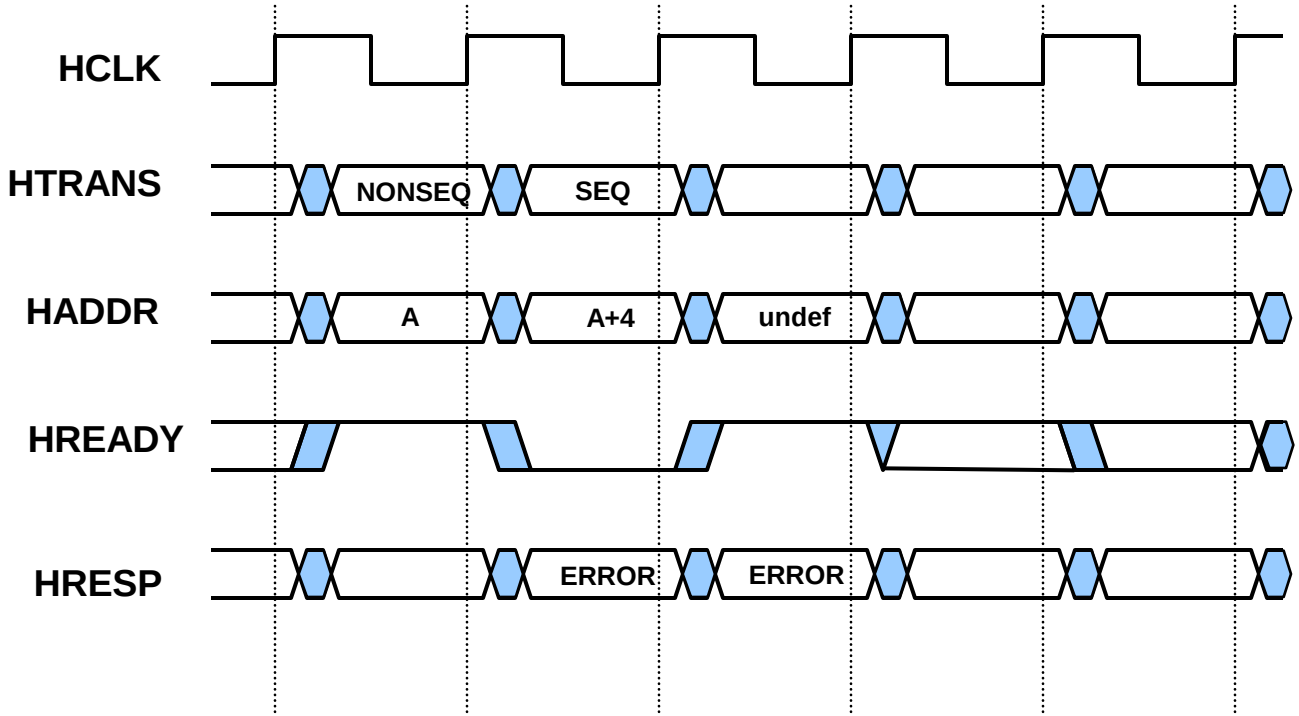


HRESP – Slave Response

HRESP	Event	Bus Master operation
OKAY	Access completed normally	
ERROR	Slave aborts access, (2 cycle response)	Master has option of continuing or terminating a burst containing an ERROR

It is permissible to continuously drive HRESP Low in a system which does not wish to generate any errors.

ERROR Response

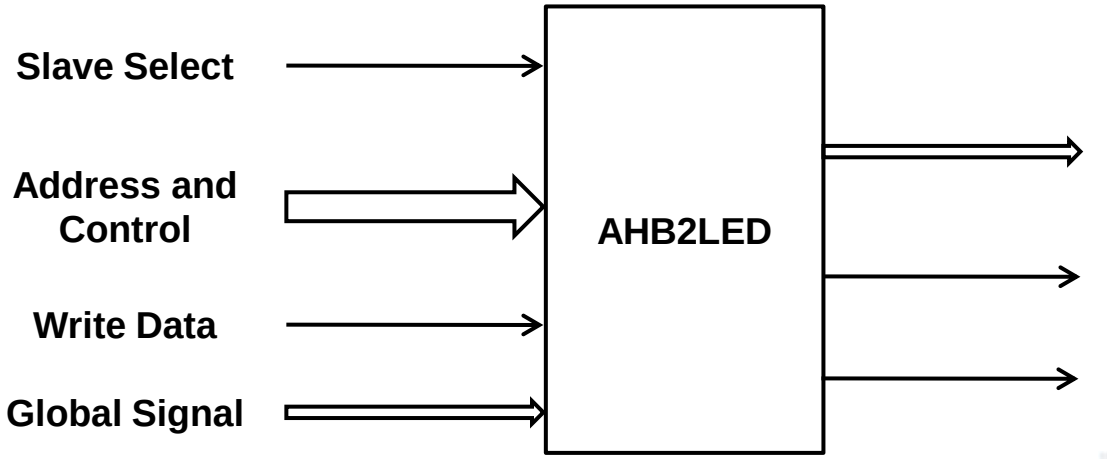


- If HRESP = ERROR, CM0-DS takes an exception and you should implement appropriate exception handler to catch the error

Agenda

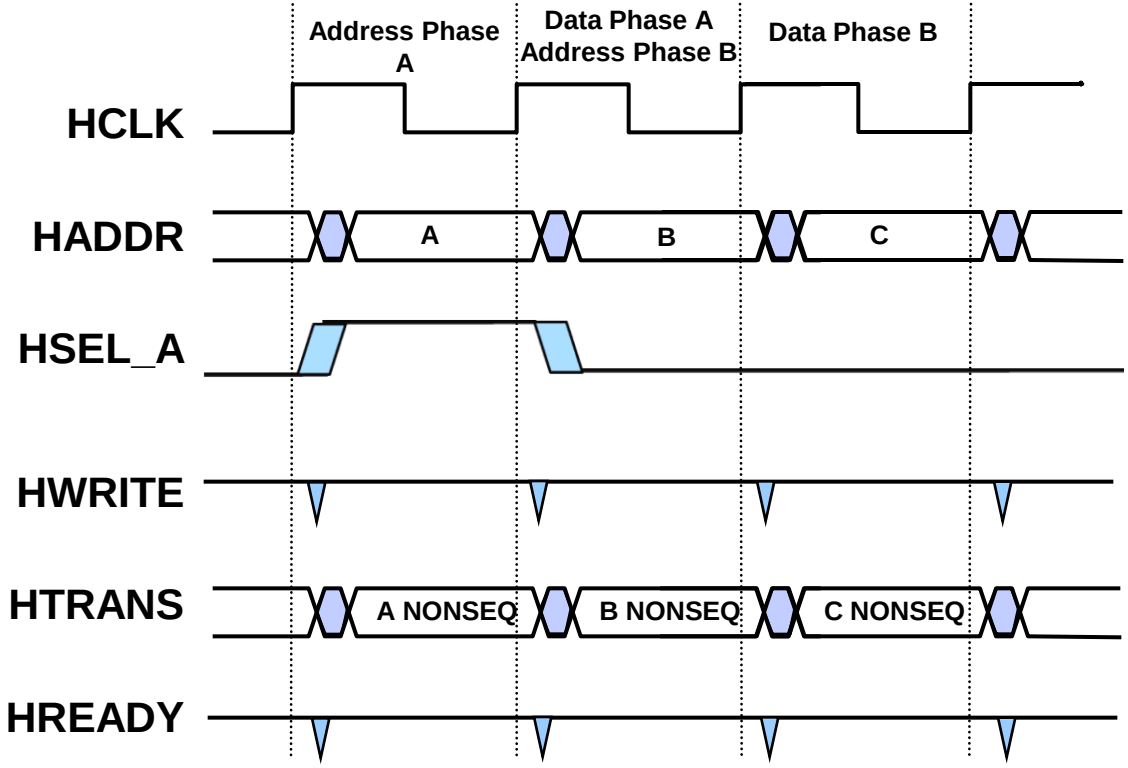
- AMBA Family (2)
- Introduction to AHB-Lite (6)
- AHB-Lite System (8)
- AHB-Lite Signals – Master & Slave (15)
- AHB-Lite Transactions (9)
- How to build a simple AHB-Lite Slave (4)
- AHB-Lite – Multilayer Design (3)

AHB2LED TOP LEVEL



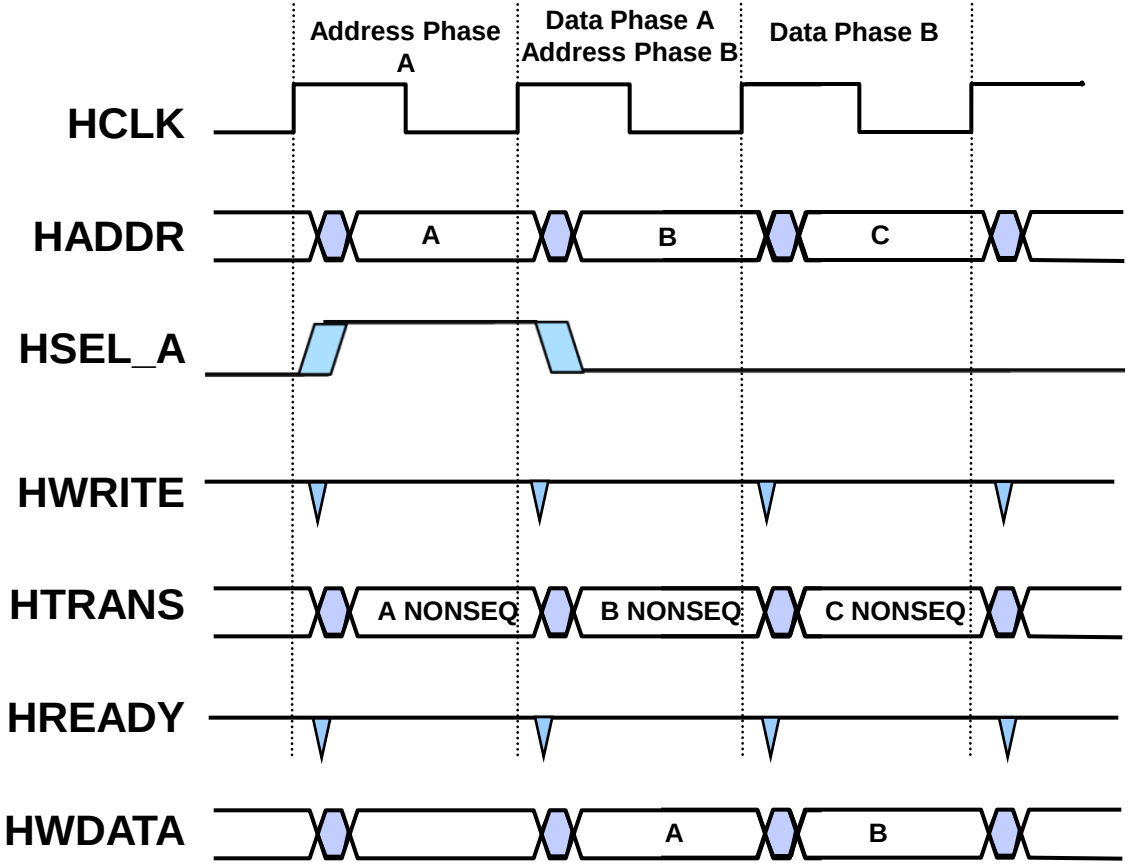

```
12  module AHB2LED(
13     //Inputs
14     input wire HSEL,
15     input wire HCLK,
16     input wire HRESETn,
17     input wire [1:0] HTRANS,
18     input wire [7:0] HWDATA,
19     input wire HWRITE,
20    input wire HREADY,
21
22
23     //Output
24     output wire HREADYOUT,
25     output wire [31:0] HRDATA,
26
27     //LED Output
28     output reg [7:0] LED
29  );
```

Sampling Address & Control




```
32  assign HREADYOUT = 1'b1; // Always ready
33
34  reg APhase_HSEL;
35  reg APhase_HWRITE;
36  reg [1:0] APhase_HTRANS;
37
38
39  always @(posedge HCLK)
40  begin
41    if(HREADY)
42    begin
43      APhase_HSEL <= HSEL;
44      APhase_HWRITE <= HWRITE;
45      APhase_HTRANS <= HTRANS;
46    end
47  end
```

Sampling Address & Control




```
32  assign HREADYOUT = 1'b1; // Always ready
33
34  reg APhase_HSEL;
35  reg APhase_HWRITE;
36  reg [1:0] APhase_HTRANS;
37
38
39  always @(posedge HCLK)
40  begin
41    if(HREADY)
42    begin
43      APhase_HSEL <= HSEL;
44      APhase_HWRITE <= HWRITE;
45      APhase_HTRANS <= HTRANS;
46    end
47  end
```

```
49  always @(posedge HCLK or negedge HRESETn)
50  begin
51    if(!HRESETn)
52      LED <= 8'b0000_0000;
53    else if(APhase_HSEL & APhase_HWRITE & APhase_HTRANS[1])
54      LED <= HWDATA[7:0];
55  end
```

AHB2LED Verilog Module

```
12  module AHB2LED(
13     //Inputs
14     input wire HSEL,
15     input wire HCLK,
16     input wire HRESETn,
17     input wire [1:0] HTRANS,
18     input wire [7:0] HWDATA,
19     input wire HWRITE,
20    input wire HREADY,
21
22
23     //Output
24     output wire HREADYOUT,
25     output wire [31:0] HRDATA,
26
27     //LED Output
28     output reg [7:0] LED
29  );
30
31
32    assign HREADYOUT = 1'b1; // Always ready
33
34    reg APhase_HSEL;
35    reg APhase_HWRITE;
36    reg [1:0] APhase_HTRANS;
37
38
39    always @(posedge HCLK)
40    begin
41      if(HREADY)
42      begin
43        APhase_HSEL <= HSEL;
44        APhase_HWRITE <= HWRITE;
45        APhase_HTRANS <= HTRANS;
46      end
47    end
48
49    always @(posedge HCLK or negedge HRESETn)
50    begin
51      if(!HRESETn)
52        LED <= 8'b0000_0000;
53      else if(APhase_HSEL & APhase_HWRITE & APhase_HTRANS[1])
54        LED <= HWDATA[7:0];
55    end
56
57    assign HRDATA = {24'h0000_00,LED};
58
59
60  endmodule
```

One way of implementing

Address and Control Sampling Phase

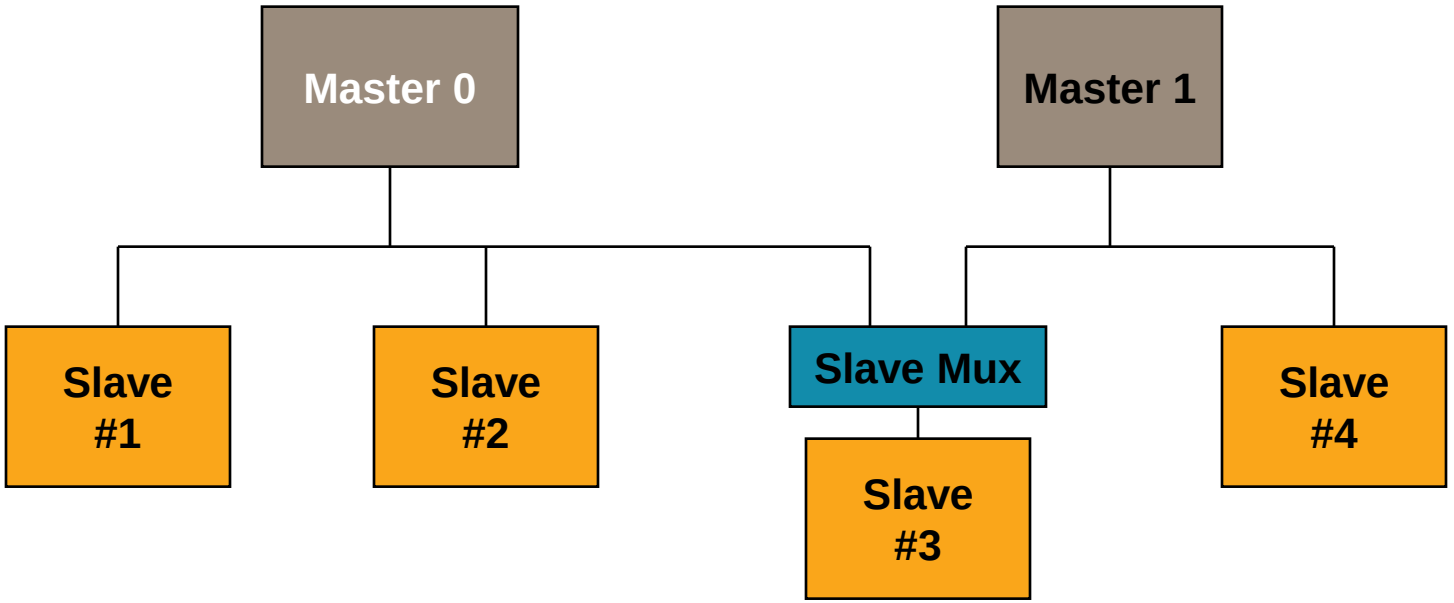
Data Sampling Phase

BUS MATRIX - ISE

Agenda

- AMBA Family
- Introduction to AHB-Lite
- AHB-Lite System
- AHB-Lite Signals – Master & Slave
- AHB-Lite Transactions
- How to build a simple AHB-Lite Slave
- AHB-Lite – Advanced

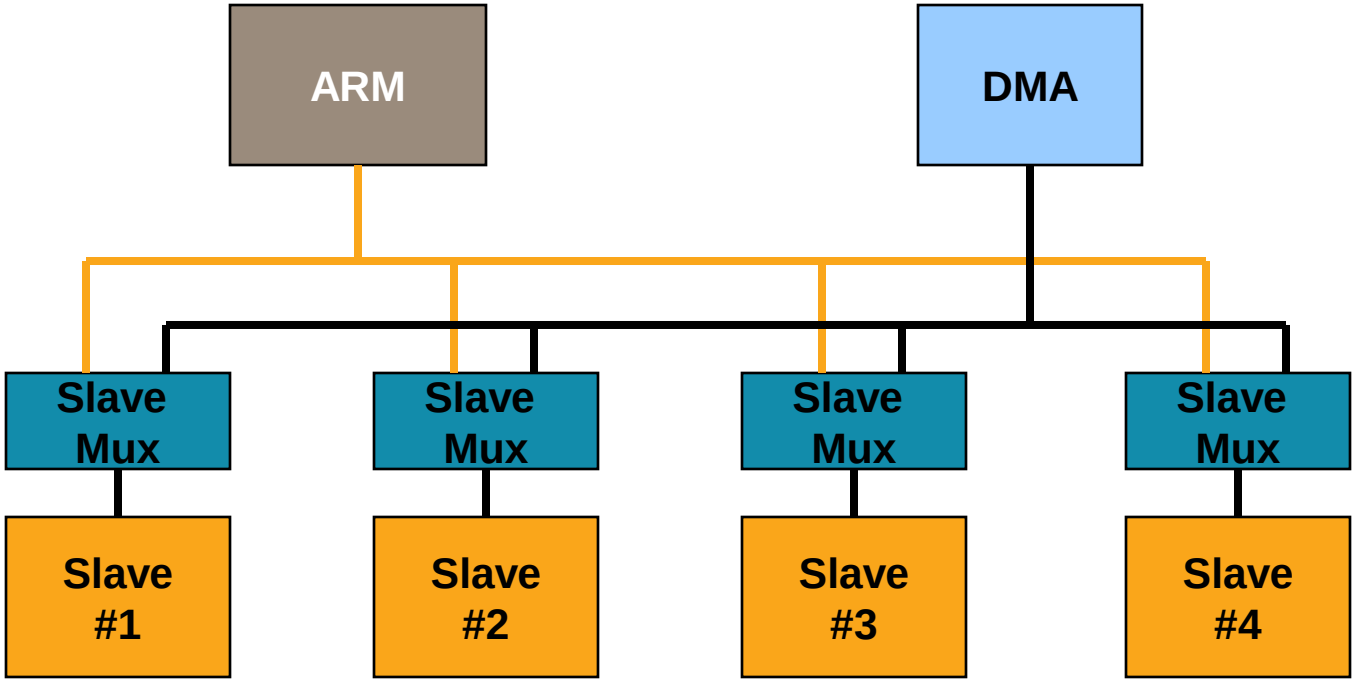
Shared Slave



- Master 0 can access slaves #1, #2 & #3
- Master 1 can access slaves #3 & #4
- Contention occurs only if Master 0 & Master 1 try to access slave #3 sametime

Arbitration is at the slave level

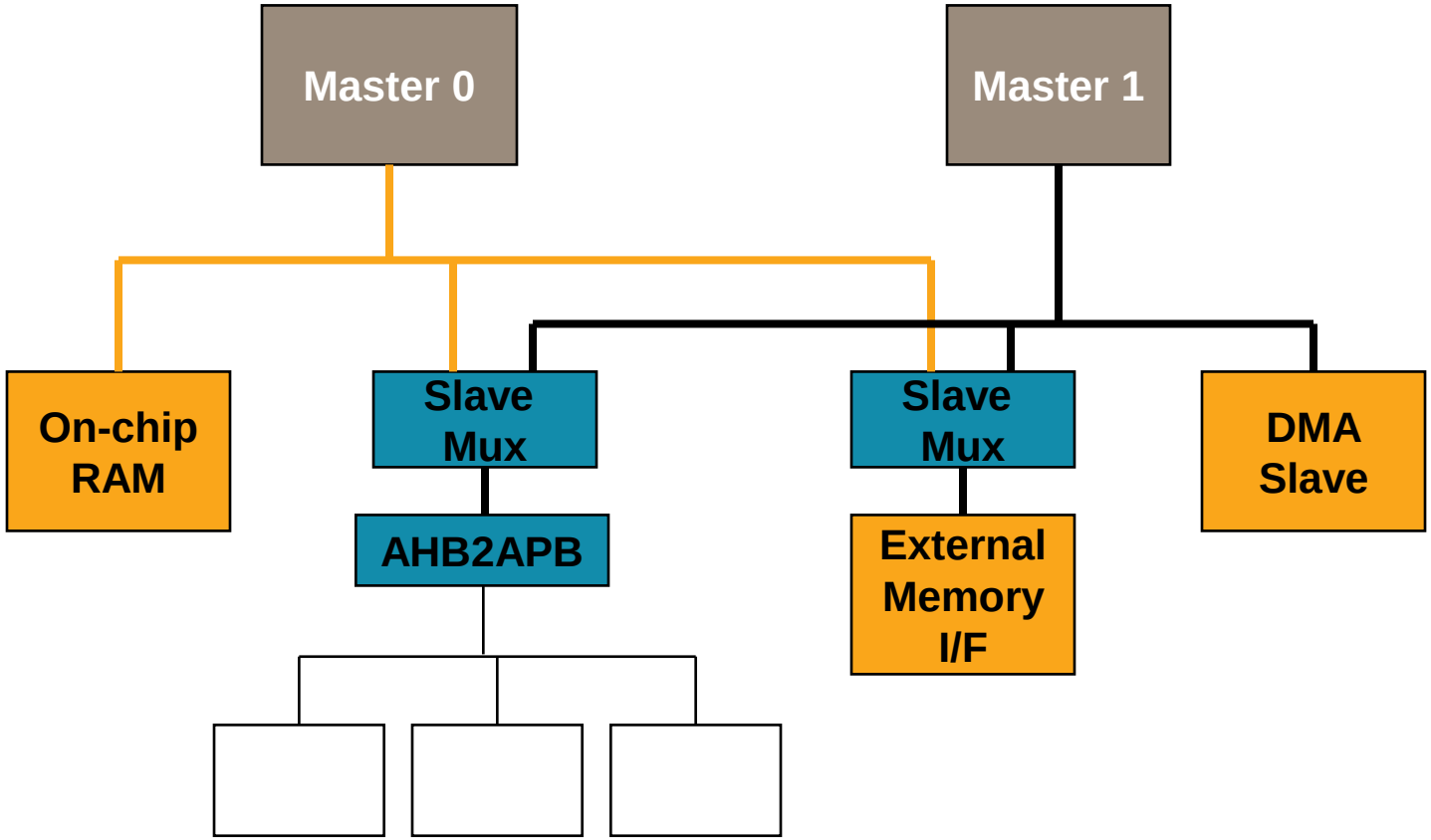
Multi-layer



- Generalizing on previous slide
- Contention occurs only if Master 0 & Master 1 try to access same slave at same time

Arbitration is at the slave level

Typical Multi-layer example



- Master 0 can access private RAM, APB and external interface
- Master 1 can access DMA slave, APB and external interface

Further Information

- ARM IHI 0033 - AMBA 3 AHB-Lite Protocol Specification
  - http://infocenter.arm.com/

THE END