Lab5: On-chip bus: AHB-Lite

Speaker: Nelson Chang

Directed by Prof. Tian-Sheuan Chang

March, 2004, NCTU

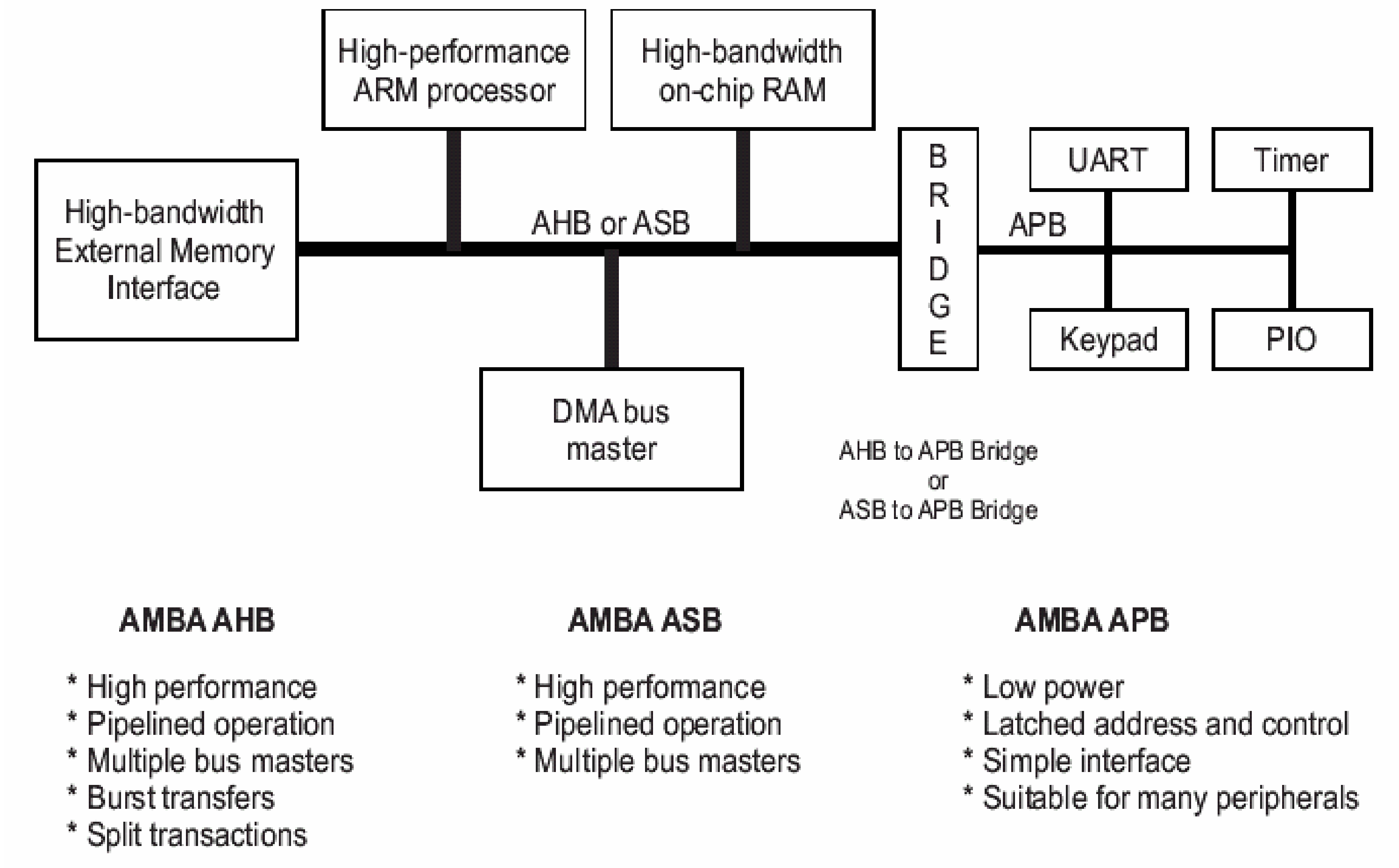
Goal of This Lab

- ❑ Understand how AHB-Lite works.
- ❑ Learn how to add new slaves.
- ❑ Learn to verify AHB-Lite compliance of a slave.

Outline

- ❑ ***AMBA AHB system***
- ❑ AHB-Lite system
- ❑ AHB compliance verification
- ❑ Lab5 – On-chip bus: AHB-Lite

Typical AMBA system


High-performance ARM processor
High-bandwidth on-chip RAM
High-bandwidth External Memory Interface
AHB or ASB
B R I D G E
UART
Timer
APB
Keypad
PIO
DMA bus master

AHB to APB Bridge
or
ASB to APB Bridge

AMBA AHB

* High performance
* Pipelined operation
* Multiple bus masters
* Burst transfers
* Split transactions

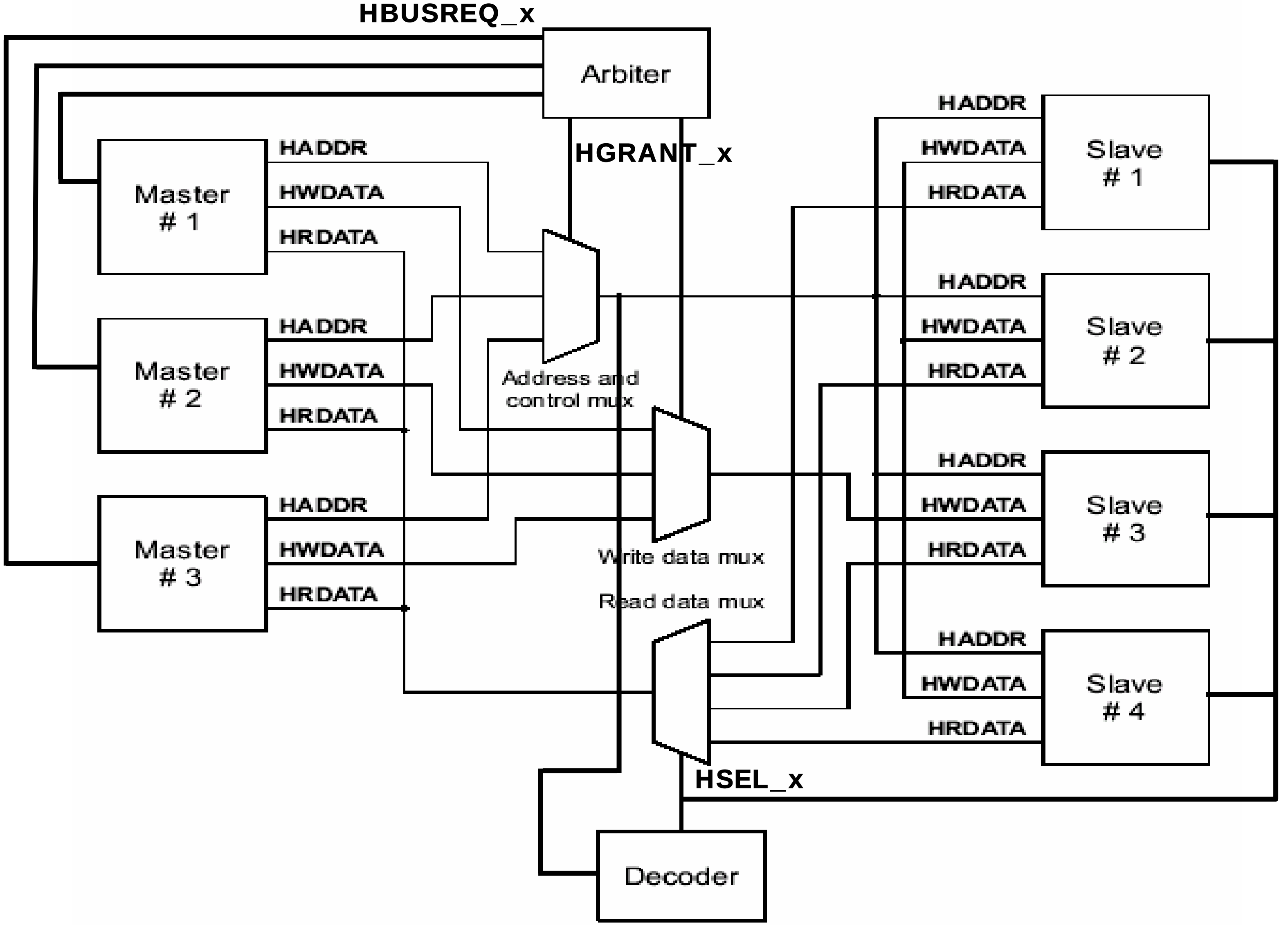
AMBA ASB

* High performance
* Pipelined operation
* Multiple bus masters

AMBA APB

* Low power
* Latched address and control
* Simple interface
* Suitable for many peripherals

An AHB System



Components in AHB (1/2)

- ❑ Master
  - – AHB master is able to initiate read and write operations by providing an address and control information. Only one bus master is allowed to actively use the bus at any one time.(max. 16)
- ❑ Slave
  - – AHB slave responds to a read or write operation within a given address-space range. The bus slave signals back to the active master the success, failure or waiting of the data transfer.

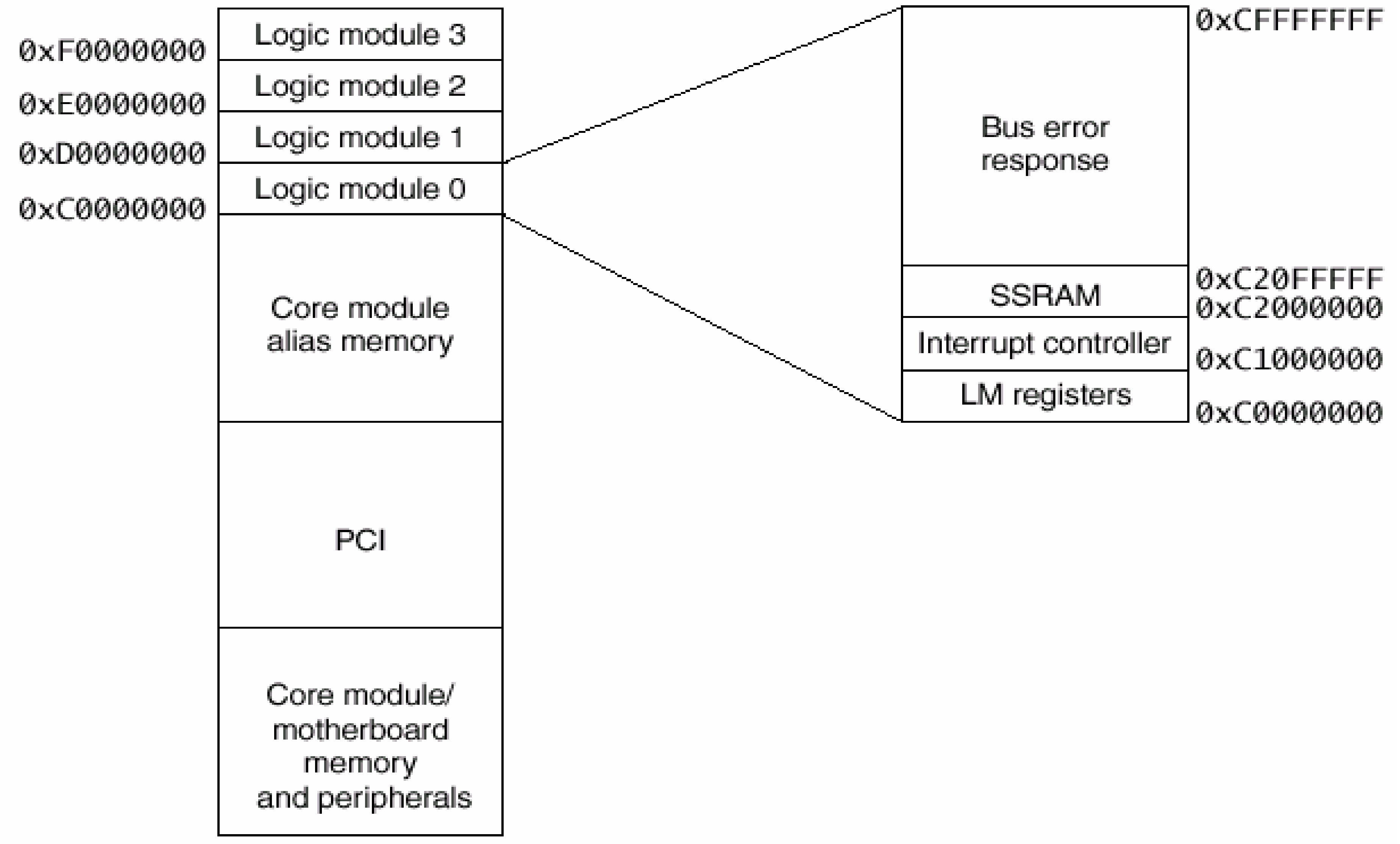
Components in AHB (2/2)

- ❑ Arbiter
  - – AHB arbiter ensures that only one bus master at a time is allowed to initiate data transfers.
- ❑ Decoder
  - – AHB decoder is used to decode the address of each transfer and provide a select signal for the slave that is involved in the transfer. A single centralized decoder is required in all AHB implementations.

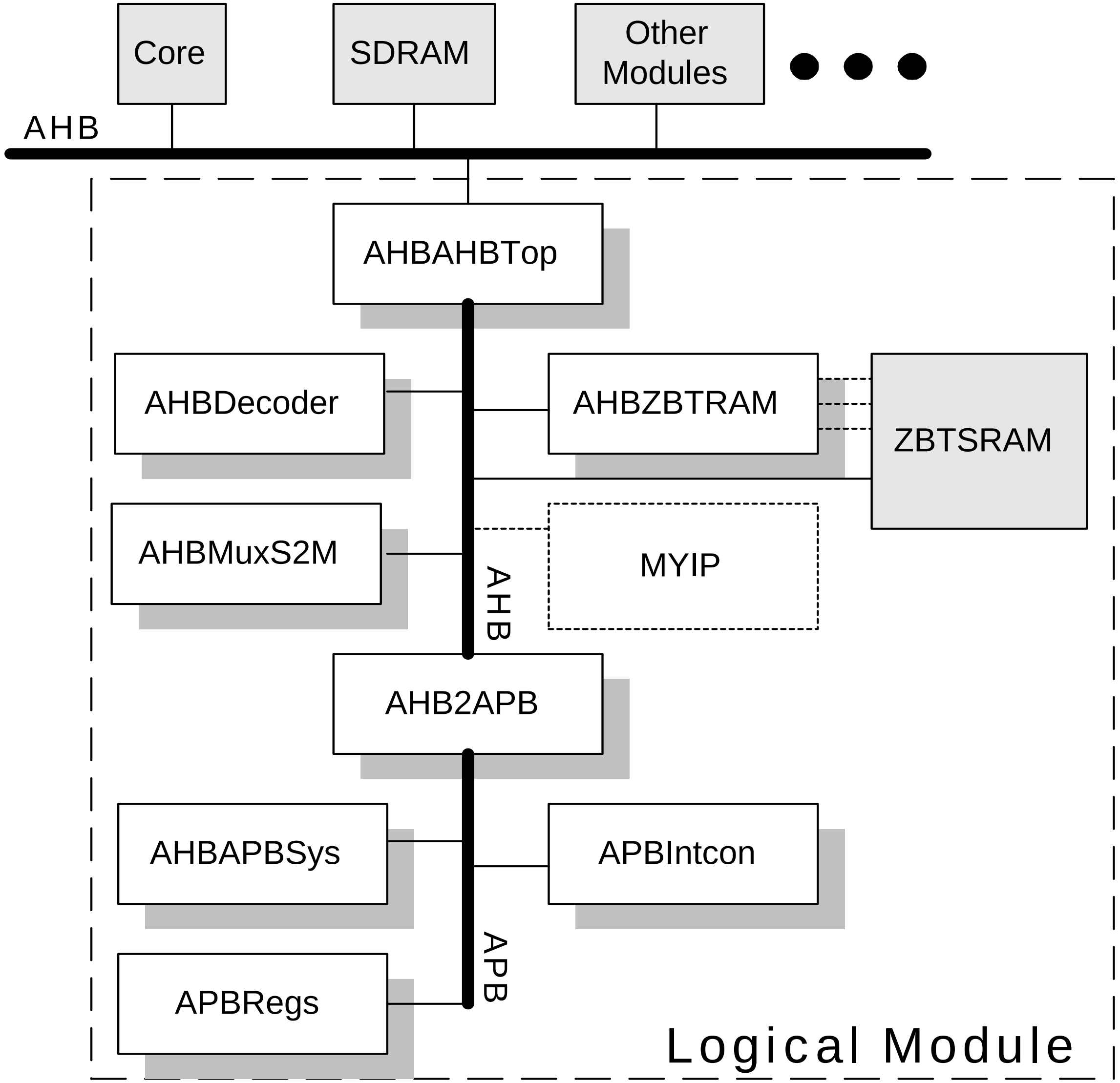
Outline

- ❑ AMBA AHB system
- ❑ ***AHB-Lite system***
- ❑ AHB compliance verification
- ❑ Lab5 – On-chip bus: AHB-Lite

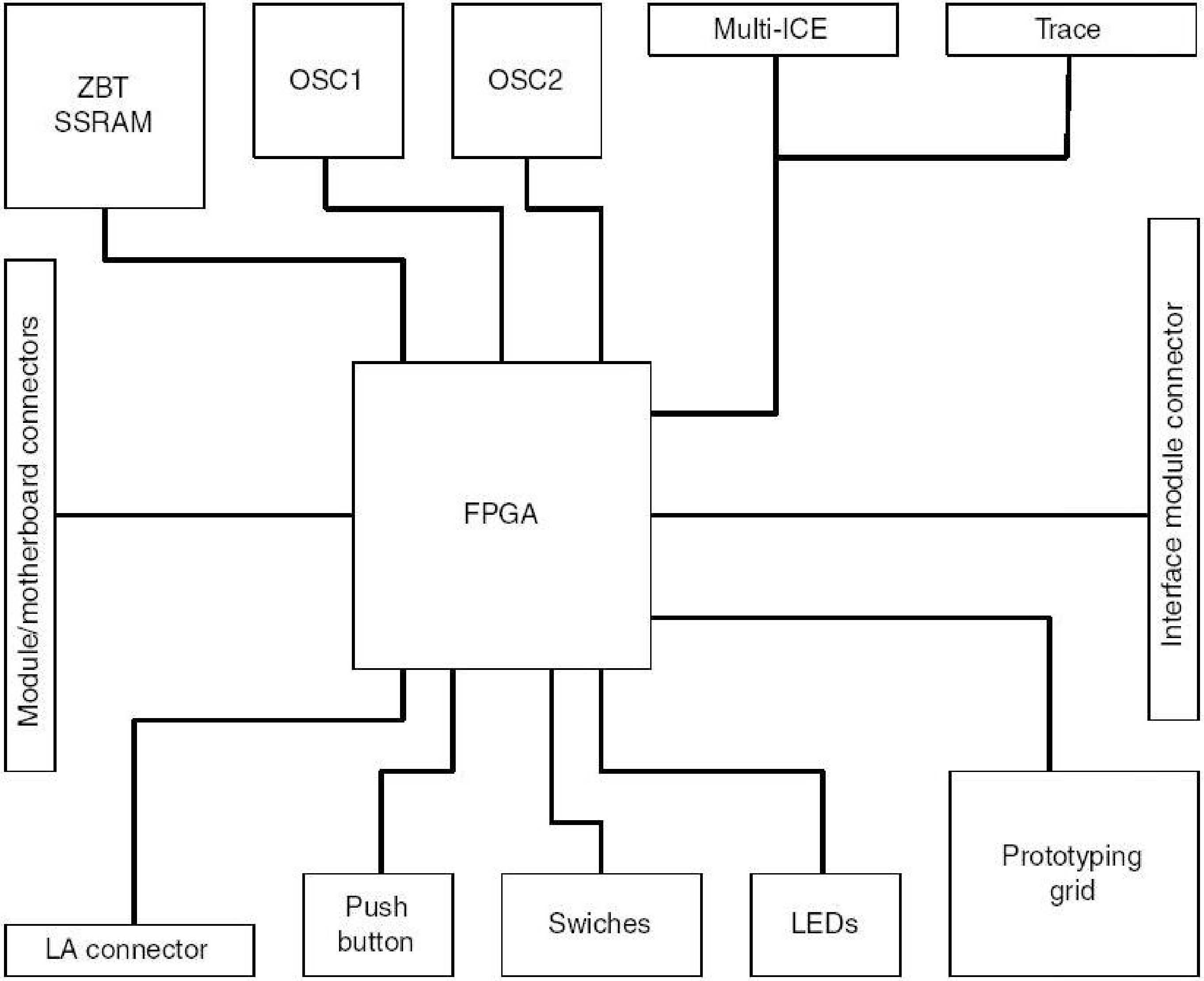
Logic Module Memory Map



LM AHB-Lite Block Diagram



Integrator LM Block Diagram



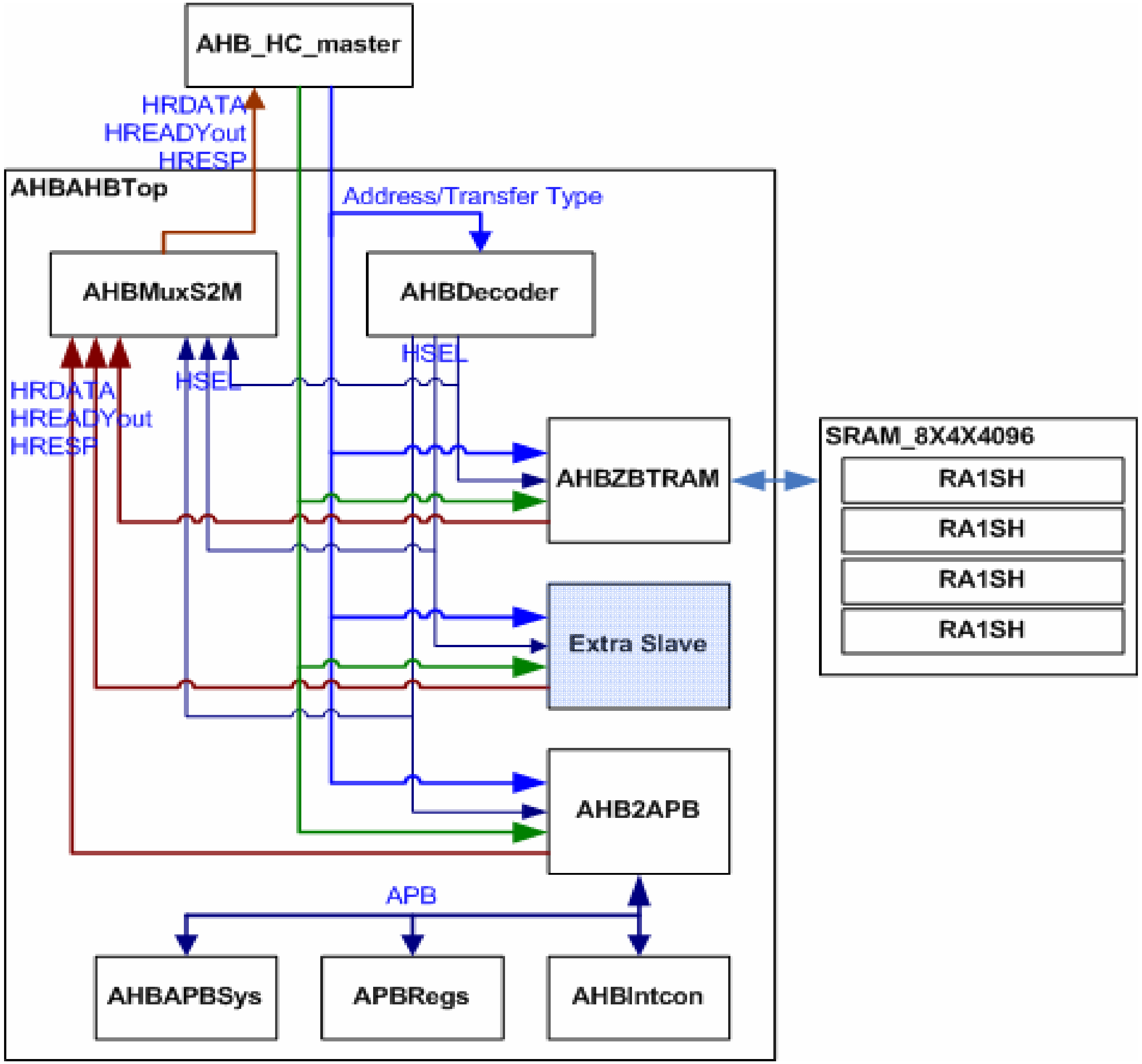
Notes about LM AHB-Lite system

- LM AHB-Lite is a slave in Integrator system
- Uses bi-directional tri-state signals
  - HDATA
  - SDATA
  - HREADY
- AHB ZBT SRAM controller needs ZBT to be able to read/write without dead cycle.
  - HREADY always HIGH → No wait-state access
  - ZBT→Zero Bus Turn-around

Modified AHB-Lite system

- ❑ Modified from LM AHB-Lite system
  - – Replaced bi-directional signal with input and output signals
    - HDATA → HWDATA/HRDATA
    - SDATA → SWDATA/SRDATA
    - HREADY → HREADYout/HREADY (internal)
- ❑ Added 2 more modules
  - – AHB_HC_master
    - Master module to drive test pattern to AHB-Lite
  - – SRAM_8X8X4096
    - Connected to ZBT SRAM controller
    - Limitation:
      - – 16 KB (originally was 1MB on LM)
      - – Cannot perform read after write immediately due to AHB pipeline access characteristic.
- ❑ Removed some signals (pins)
  - – Flash related signals
  - – TDI, TDO, RTCK, etc.

AHB-Lite system diagram



AHB-lite Access

- ❑ Master checks HREADYin
  - – HREADYin LO: hold control
  - – HREADYin HI: initiates access request
    - Read/write
    - Transfer type
    - Transfer size
- ❑ Decoder decodes access address
  - – HDRID=0xC → First LM
  - – HADDR[27:20]= 0x20 → ZBT SRAM controller
  - – HSEL_zbtssram=1’b1 → send slave select signal
- ❑ ZBT SRAM controller
  - – Write: master send HWDATA to slave in data phase
  - – Read: slave returns HRDATA to MUX
- ❑ MUX slave to master read data:
  - – HSEL_zbtssram=1’b1→ select HRDATA from ZBT SRAM controller to send to master
    - Ready signals
    - Response signals

Outline

- ❑ AMBA AHB system
- ❑ AHB-Lite system
- ❑ ***AHB compliance verification***
- ❑ Lab5 – On-chip bus: AHB-Lite

Compliance check method

- ❑ Simulation check (coverage driven)
  - – Create check list according to specification
  - – Create test pattern to hit all the cases in check list
    - Synopsys DesignWare Verification IP (VIP) + Vera
    - ARM AMBA Compliance Test-bench (ACT)
    - Manually check (most error prone)
  - – May not cover some corner case
    - 100% coverage (of check list) does not suggest 100% proven
- ❑ Formal techniques
  - – Property and rules extraction
  - – Model & property checking and state space exploration
  - – Averants Solidify SolidAHB
    - If rules are 100% proven, the interface will not violate the rules

Example of check list from DesignWare VIP

❑ State cases

```
State Name # defined values # hits
---------- ---------------- ------
cv_nseq_rd                1       26
cv_seq_rd                 1       14
cv_busy_rd                1       3
cv_nseq_wr                1       9
cv_seq_wr                 1       8
cv_busy_wr                1       2
```

❑ Transition cases

```
Transition Name # defined transitions # hits
--------------- --------------------- ------
cv_nseq2nseq                       1       22
cv_nseq2seq                        1       5
cv_nseq2idle                       1       3
cv_nseq2busy                       1       3
cv_seq2seq                         1       13
```

Outline

- ❑ AMBA AHB system
- ❑ AHB-Lite system
- ❑ AHB compliance verification
- ❑ ***Lab5 – On-chip bus: AHB-Lite***

Lab 5: On-chip bus: AHB-Lite

- Goal
  - Familiarize AHB using AHB-Lite
  - Learn how to add new slave into AHB-Lite
  - Practice checking the compliance of an AHB-Lite slave
- Principles
  - AMBA Protocols
- Guidance
  - Observer the AHB read/write
  - Identify which module defines the memory map
- Steps
  - Run the example AHB-Lite
  - Observe signals
- Requirements and Exercises
  - Add a new slave
  - Check AHB-Lite compliance of the new slave
- Discussion
  - Disadvantage of using AMBA AHB (Overhead)

Files Descriptions

- Modify **AHB_HC_master.v** to modify test pattern
- Modify **AHB_Testbench.v** to modify simulation cycles
- Modify **AHB_Testbench.f** to add new files
- **MyIP.v** is the new slave to be added into AHB-Lite

Files	Path	Description
AMBA_declare.v	/beh	AMBA related predefined keywords
AHB_HC_master.v	/beh	AHB_HC_master behavioral module
SRAM_8X4X4096.v	/beh	16KB SRAM module
RA1SH.v	/beh	SRAM behavioral model
AHB_Testbench.v	/beh	AHB_Testbench top module including AHB_HC_master, SRAM_8X4X4096, and RA1SH
AHBAHBTop.v	/rtl	AHBAHBTop modified module
AHBDecoder.v	/rtl	AHBDecoder module
AHBMuxS2M.v	/rtl	AHBMuxS2M module
AHBZBTRAM.v	/rtl	AHBZBTRAM module
AHB2APB.v	/rtl	AHB2APB module, it is also a slave
AHBAPBSys.v	/rtl	AHBAPBSys module
APBRegs.v	/rtl	APBRegs module
APBIntcon.v	/rtl	APBIntcon module
MyIP.v	/rtl	Slave to be added into AHB-Lite
AHB_Testbench.f	/verif	File lists of the whole test bench
LM_AHBAPB.f	/verif	File lists of AHBAHBTop and its sub-modules

Accesses made by AHB_HC_master

- 7 Accesses
- Data and control are not within the same cycle

No.	HWRITE	HSIZE	HADDR	HRDATA/ HWDATA	Remarks
1	Read	32-bit	`0xc2000000`	`0xXXXXXXXX`	
2	Read	32-bit	`0xc2000010`	`0xXXXXXXXX`	
3	Write	32-bit	`0xc2000000`	`0x00001234`	
4	Read	32-bit	`0xc2000010`	`0x00001234`	SRAM cannot read immediately after write using ZBT SRAM controller.
5	Read	32-bit	`0xc2000010`	`0xXXXXXXXX`	
6	Read	32-bit	`0xc2000000`	`0x00001234`	
7	Read	32-bit	`0xc2000010`	`0xXXXXXXXX`	

References

[1] AMBA Specification, Rev. May, 2.0, 1999.

[2] High-Speed Single-Port SRAM (HS-SRAM-SP) Generator User Manual, Artisan Components Inc., Release 4.0, Aug. 2000.

[3] Debussy User Guide and Tutorial, NOVAS Software Inc., Sept. 2002.

[4] Compatibility of Network SRAM and ZBT SRAM, Mitsubishi LSIs Application Note (AP-S001E), Rev. C, Renesas Tech. Corp., Sept. 2002.

[5] DesignWare AHB Verification IP Databook, ver. 2.0a, Synopsys Inc., July 2002.

[6] VMT User Manual, Release 2.0a, Synopsys Inc., July 2002.

[7] Vera User Guide, ver. 5.1, Synopsys Inc., June 2002.

[8] SolidAMBA, Averant Inc., Dec. 2003.