Bus AMBA

Advanced Microcontroller Bus Architecture (AMBA)

Rene.beuchat@epfl.ch
Rene.beuchat@hesge.ch
Réf: AMBA™ Specification
(Rev 2.0)
www.arm.com
ARM IHI 0011A
What to see

- AMBA system architecture
- Derivatives Bus, AHB, ASB, APB
- Transactions

- Note: Amba, version 3 is define and allows parallel transaction with multi-masters
- www.arm.com
Bus hierarchies

• **Advanced Microcontroller Bus Architecture (AMBA)**
  - Advanced High-performance Bus (AHB)
  - Advanced System Bus (ASB)
  - Advanced Peripheral Bus (APB).
Typical System

- High-performance ARM processor
- High-bandwidth on-chip RAM
- AHB or ASB
- UART
- Timer
- 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
Advanced High-performance Bus (AHB)

- The AMBA AHB is for high-performance, high clock frequency system modules.
- The AHB acts as the high-performance system backbone bus.
- AHB supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macro cell functions.
- AHB is also specified to ensure ease of use in an efficient design flow using synthesis and automated test techniques.
Advanced System Bus (ASB)

- The AMBA ASB is for high-performance system modules.
- AMBA ASB is an alternative system bus suitable for use where the high-performance features of AHB are not required.
- ASB also supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macrocell functions.
Advanced Peripheral Bus (APB)

- The AMBA APB is for low-power peripherals.
- AMBA APB is optimized for minimal power consumption and reduced interface complexity to support peripheral functions. APB can be used in conjunction with either version of the system bus.
Objectives of the AMBA specification

• The AMBA specification has been derived to satisfy four key requirements:
  - to facilitate the *right-first-time* development of embedded microcontroller products with one or more CPUs or signal processors
  - to be *technology-independent* and ensure that highly reusable peripheral and system macrocells can be migrated across a diverse range of IC processes and be appropriate for full-custom, standard cell and gate array technologies
Objectives of the AMBA specification (2)

- to encourage *modular system design* to improve processor independence, providing a development road-map for advanced cached CPU cores and the development of peripheral libraries
- to minimize the silicon infrastructure required to support efficient on-chip and off-chip communication for both operation and manufacturing test.
AMBA AHB, new generation
Advanced High-performance Bus

AMBA AHB implements the features required for high-performance, high clock frequency systems including:

• burst transfers
• split transactions
• single-cycle bus master handover
• single-clock edge operation
• non-tristate implementation
• wider data bus configurations (64/128 bits).
AHB Components

• AHB master  → transfers initiator
• AHB slave
• AHB arbiter  → multi-master
• AHB decoder  → centralized decoder
AHB, general view, multi-master
AHB Master

Arbiter Grant
- HGRANTx

Transfer Response
- HREADY
- HRESP[1:0]

Reset
- HRESETn
- HCLK

Data
- HRDATA[31:0]

AHB Master
- HBUSREQx
- HLOCKx
- HTRANS[1:0]
- HADDR[31:0]
- HWRITE
- HSIZE[2:0]
- HBURST[2:0]
- HPROT[3:0]
- HWDATA[31:0]

Arbiter
- Address and Control

Transfer type

Data
AHB Slave

Select
- HSELx

Address and control
- HADDR[31:0]
- HWRITE
- HTRANS[1:0]
- HSIZE[2:0]
- HBURST[2:0]

Data
- HWDATA[31:0]

Reset
- HRESETn

Clock
- HCLK
- HMASTER[3:0]
- HMASTLOCK

AHB slave

Split-capable slave

HREADY
HRESP[1:0]
Transfer response

HRDATA[31:0]
Data

HSPLITx[15:0]
AHB Slave, mux data
AHB, simple transfert

[Diagram showing AHB transaction phases with HCLK, HADDR[31:0], Control, HWDATA[31:0], HREADY, and HRDATA[31:0].]
AHB, wait
AHB, multiples transfers
AHB, examples of transfers
### Data bus, little endian

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Word</td>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Halfword</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Halfword</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Byte</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>Byte</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>Byte</td>
<td>2</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Byte</td>
<td>3</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
## Data bus, big endian

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Word</td>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Halfword</td>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Halfword</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Byte</td>
<td>0</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Byte</td>
<td>1</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Byte</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>Byte</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
</tbody>
</table>
AHB, central arbiter

Arbiter requests and locks
- HBUSREQx1
- HLOCKx1
- HBUSREQx2
- HLOCKx2
- HBUSREQx3
- HLOCKx3

Address and control
- Haddr[31:0]
- HSPLITx[15:0]
- HTRANS[1:0]
- HBURST[2:0]
- HRESP[1:0]
- HREADY

AHB arbiter

Arbiter grants
- HGRANTx1
- HGRANTx2
- HGRANTx3

Reset
- HRESETn

Clock
- HCLK
Centralized Arbitration

![Diagram showing the timing and signals for centralized arbitration.](image-url)
Arbitration
AHB decoder
AHB, decoding, example

Master # 1
HADDR_M1[31:0]

Master # 2
HADDR_M2[31:0]
Address and control mux

Decoder

HADDR to all slaves

Slave # 1

Slave # 2
HSEL_S1
HSEL_S2
HSEL_S3

Slave # 3
AMBA ASB, old generation
Advanced System Bus

• burst transfers
• pipelined transfer operation
• multiple bus master.
ASB Master

Arbiter grant
- AGNT

Transfer response
- BWAIT
- BERROR
- BLAST

Reset
- BnRES

Clock
- BCLK

ASB master

Arbiter
- AREQ
- BLOK

Transfer type
- BTRAN[1:0]

Address and control
- BA[31:0]
- BWRITE
- BSIZE[1:0]
- BPROM[1:0]

Data
- BD[31:0]
ASB Slave

Select

DSEL

Address and control

BA[31:0]
BWRITE
BSIZE[1:0]

ASB slave

Reset

BnRES

Clock

BCLK

Transfer response

BWAIT
BERROR
BLAST

Data

BD[31:0]
ASB decoder

Transfer type
- BTRAN[1:0]

Address and control
- BA[31:0]
- BWRITE
- BSIZE[1:0]
- BPRT[1:0]

Reset
- BnRES

Clock
- BCLK

ASB decoder

Selects
- DSEL1
- DSEL1
- ...
- DSELn

Transfer response
- BWAIT
- BERROR
- BLAST
ASB arbiter

Arbiter requests:
- AREQx1
- AREQx2
- AREQx3

Wait
Lock
Reset
Clock

Arbiter grants:
- AGNTx1
- AGNTx2
- AGNTx3
AMBA APB

- High-performance ARM processor
- High-bandwidth on-chip RAM
- AHB or ASB
- DMA bus master
- UART
- Timer
- Keypad
- PIO
- APB bridge

**AMBA Advanced Peripheral Bus (APB)**

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

Write transfer

Read transfer
APB bridge

System bus slave interface

Read data

Reset

Clock

PRDATA

PRESETn

PCLK

PSEL1

PSEL2

... PSELn

PENABLE

PADDR

PWRITE

PWRITE

Write data

Selects

Strobe

Address and control
APB Slave
APB, tri-state Bus available depending on the implementation
write cycle
APB, tri-state Bus available depending on the implementation
read cycle