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

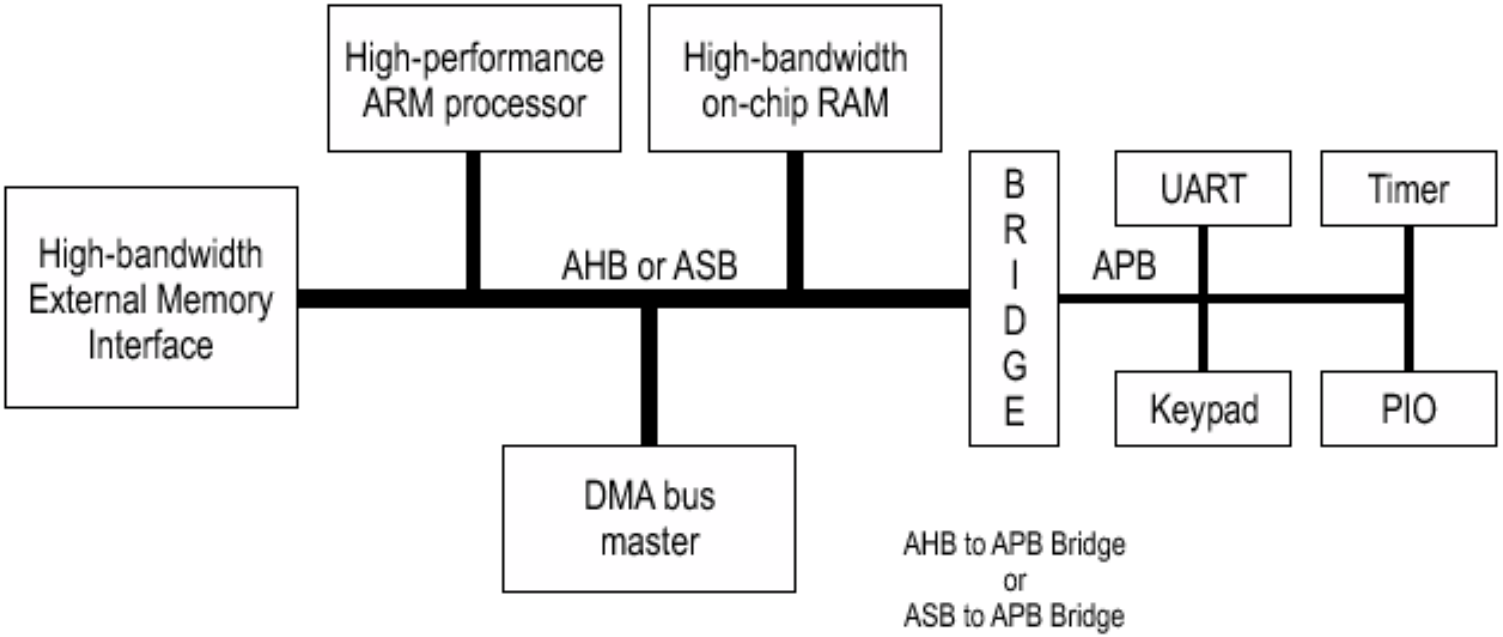
- AMBA system architecture
- Derivatives Bus, AHB, ASB, APB
- Transactions

- 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



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

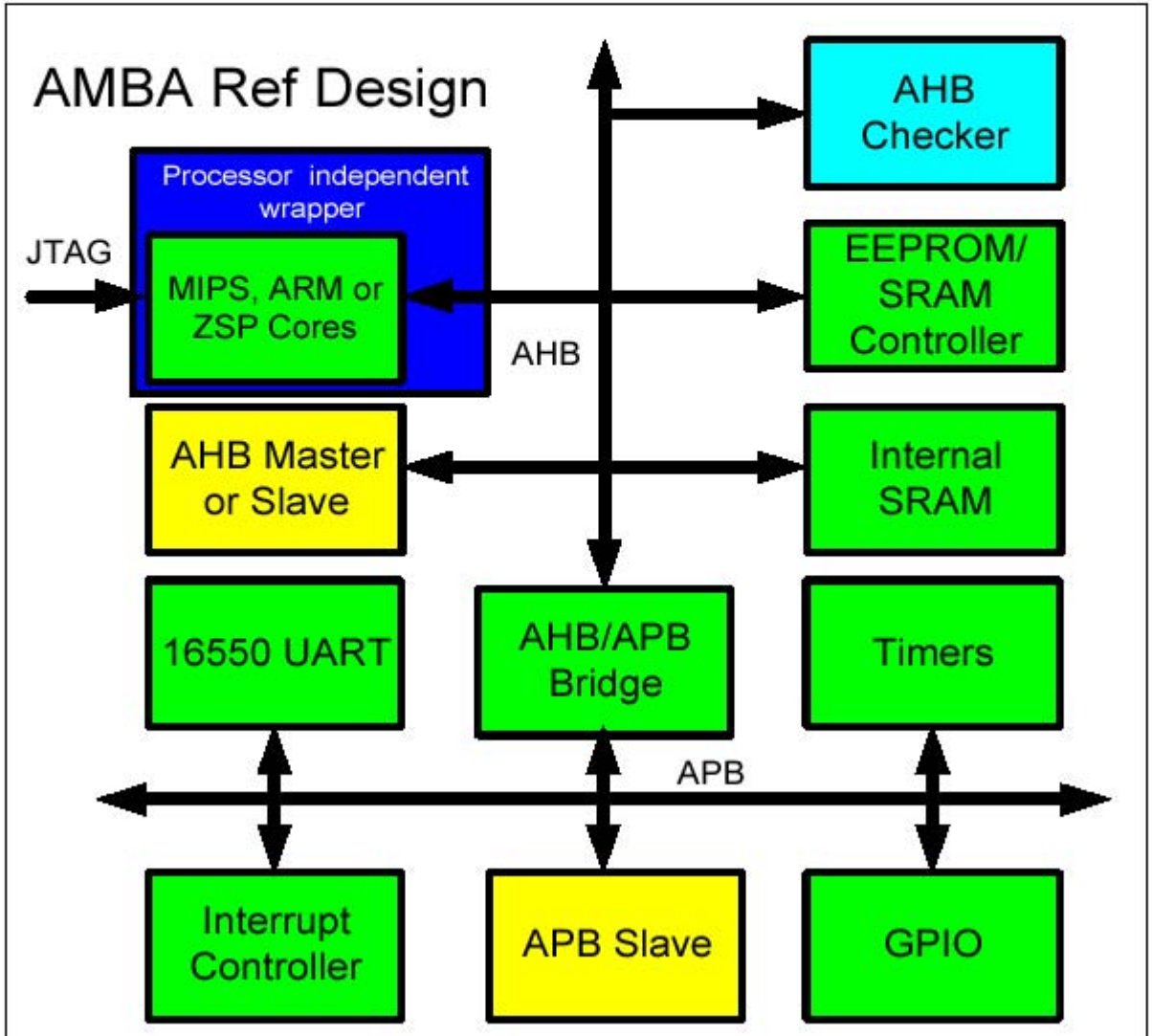
- The AMBA specification has been derived to satisfy four key requirements:
  - to facilitate the *right-first-time* development of embedded microcontroller products with one or more CPUs or signal processors
  - to be *technology-independent* and ensure that highly reusable peripheral and system macrocells 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.

LOGIC

Processor CoreWare® w/ AMBA™



AMBA AHB, new generation

Advanced High-performance Bus

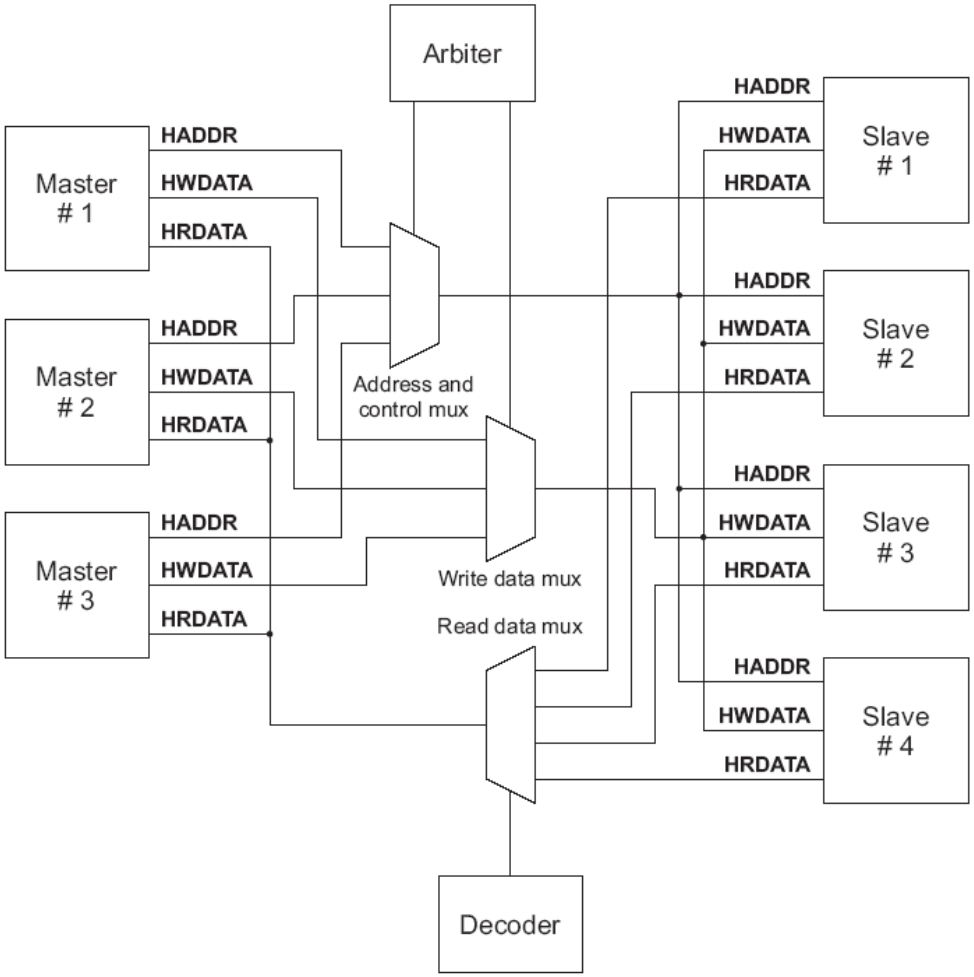
AMBA AHB implements the features required for high-performance, high clock frequency systems including:

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

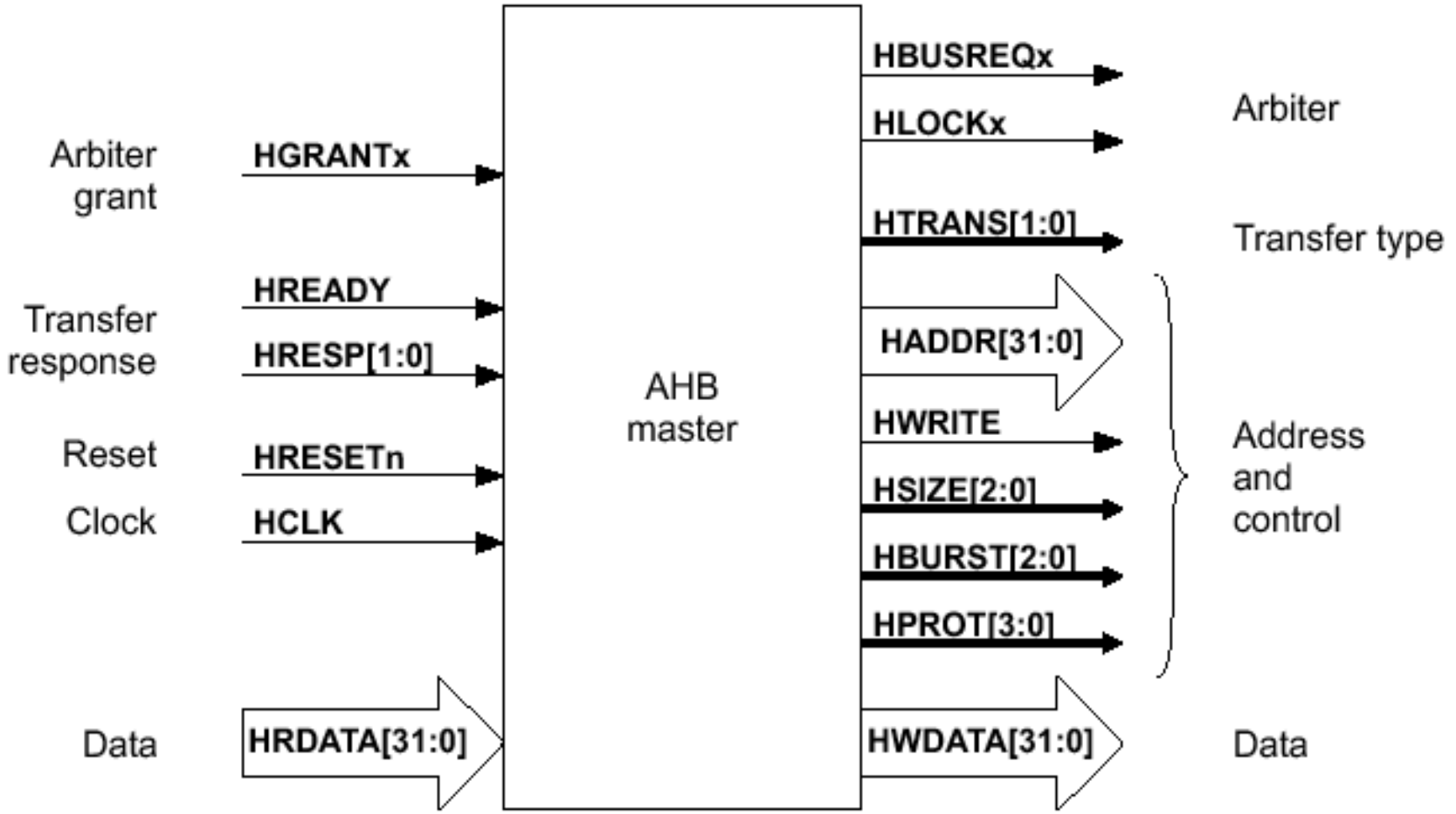
AHB Components

- **AHB master → transfers initiator**
- **AHB slave**
- **AHB arbiter → multi-master**
- **AHB decoder → centralized decoder**

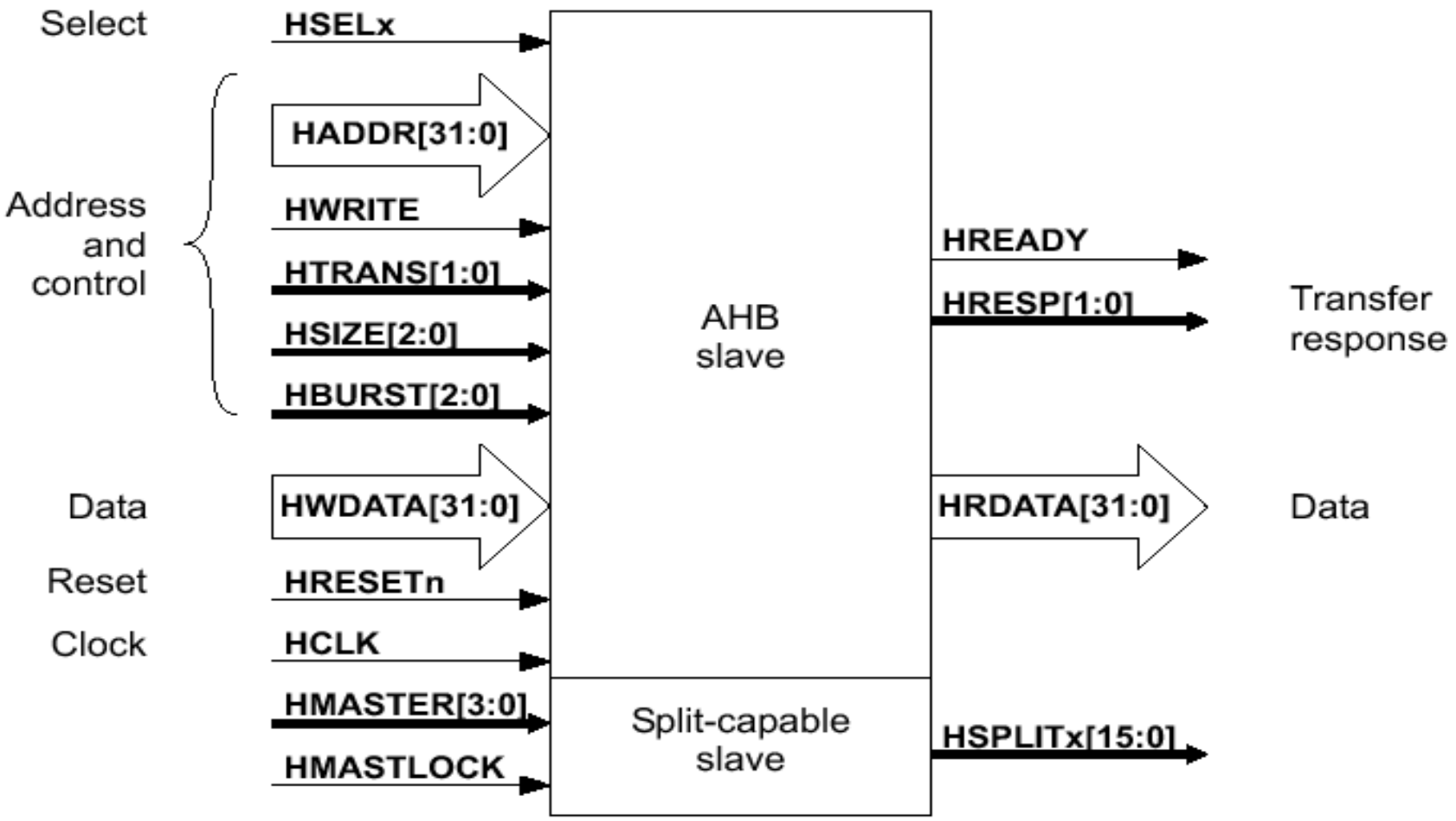
AHB, general view, multi-master



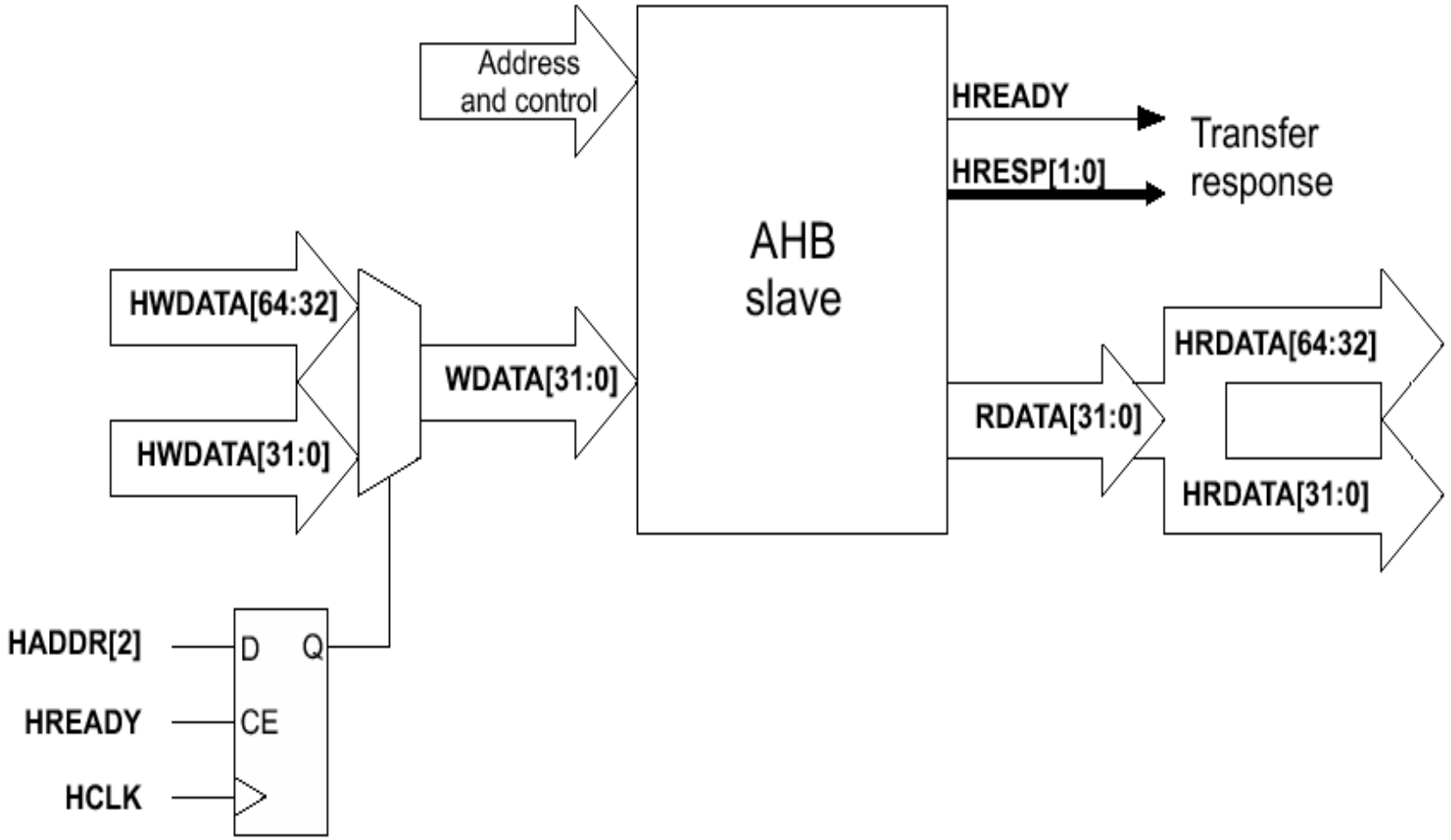
AHB Master


Arbiter grant
HGRANTx
Transfer response
HREADY
HRESP[1:0]
Reset
HRESETn
Clock
HCLK
Data
HRDATA[31:0]
AHB master
HBUSREQx
HLOCKx
Arbiter
HTRANS[1:0]
Transfer type
HADDR[31:0]
HWRITE
HSIZE[2:0]
HBURST[2:0]
HPROT[3:0]
Address and control
HWDATA[31:0]
Data

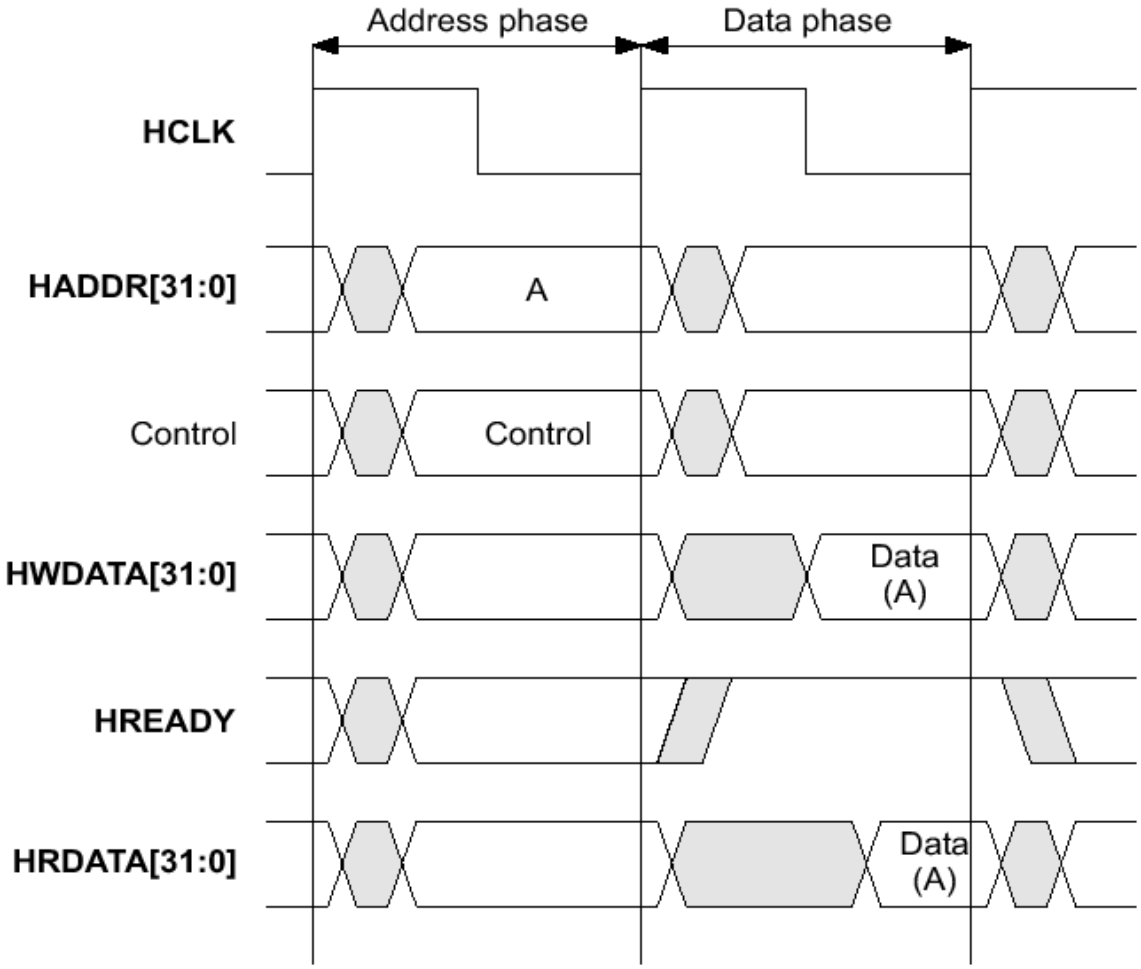
AHB Slave


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]

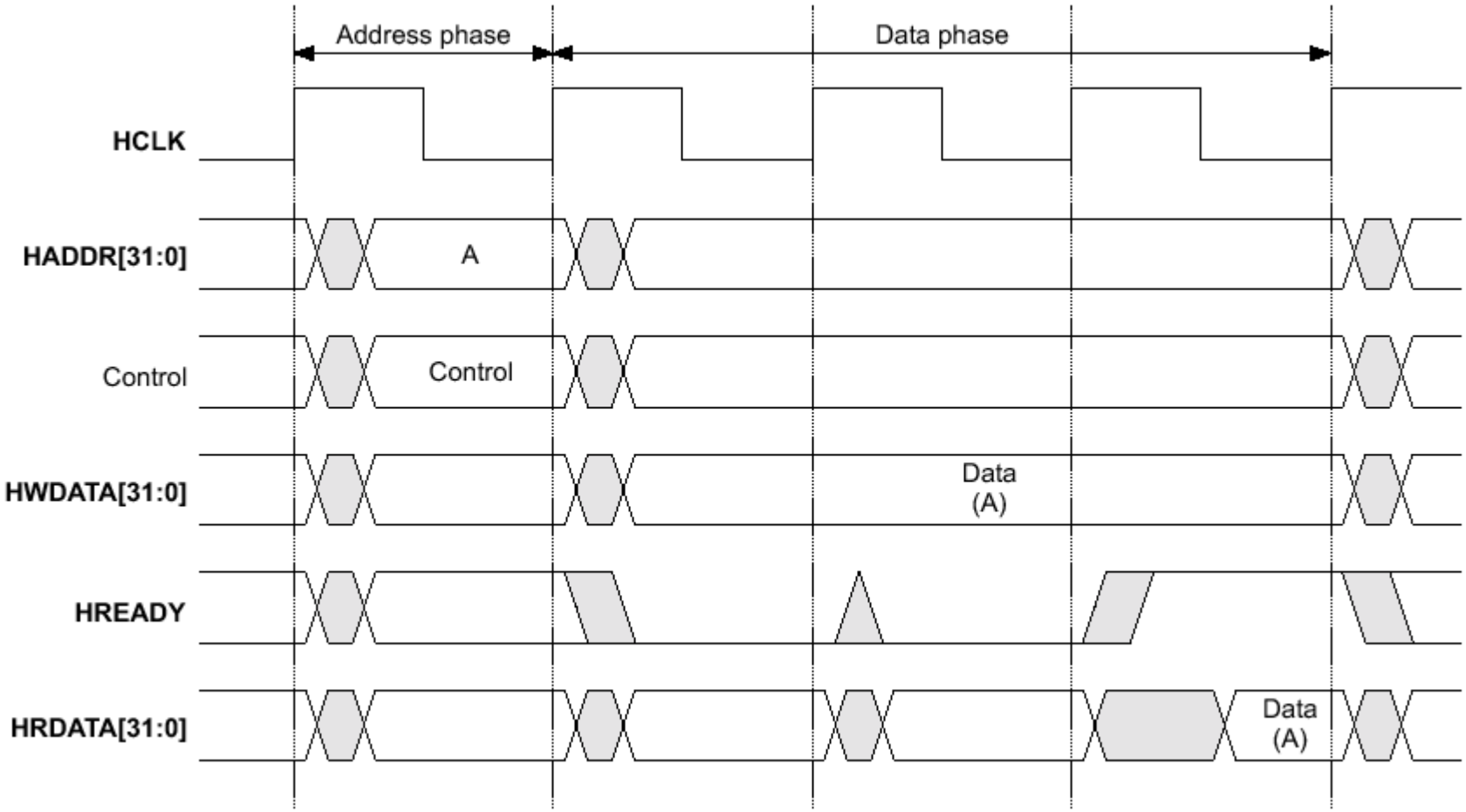
AHB Slave, mux data



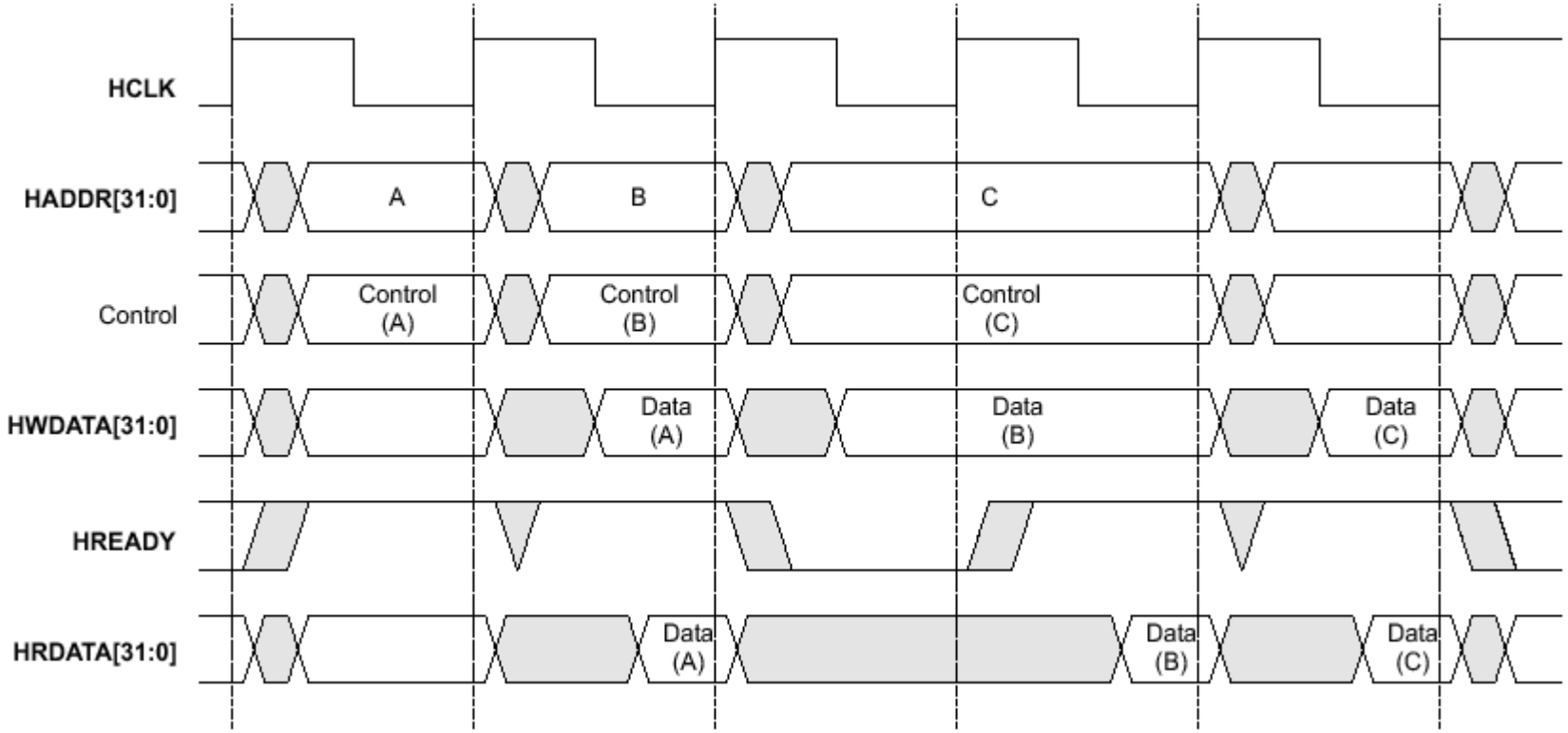
AHB, simple transfert



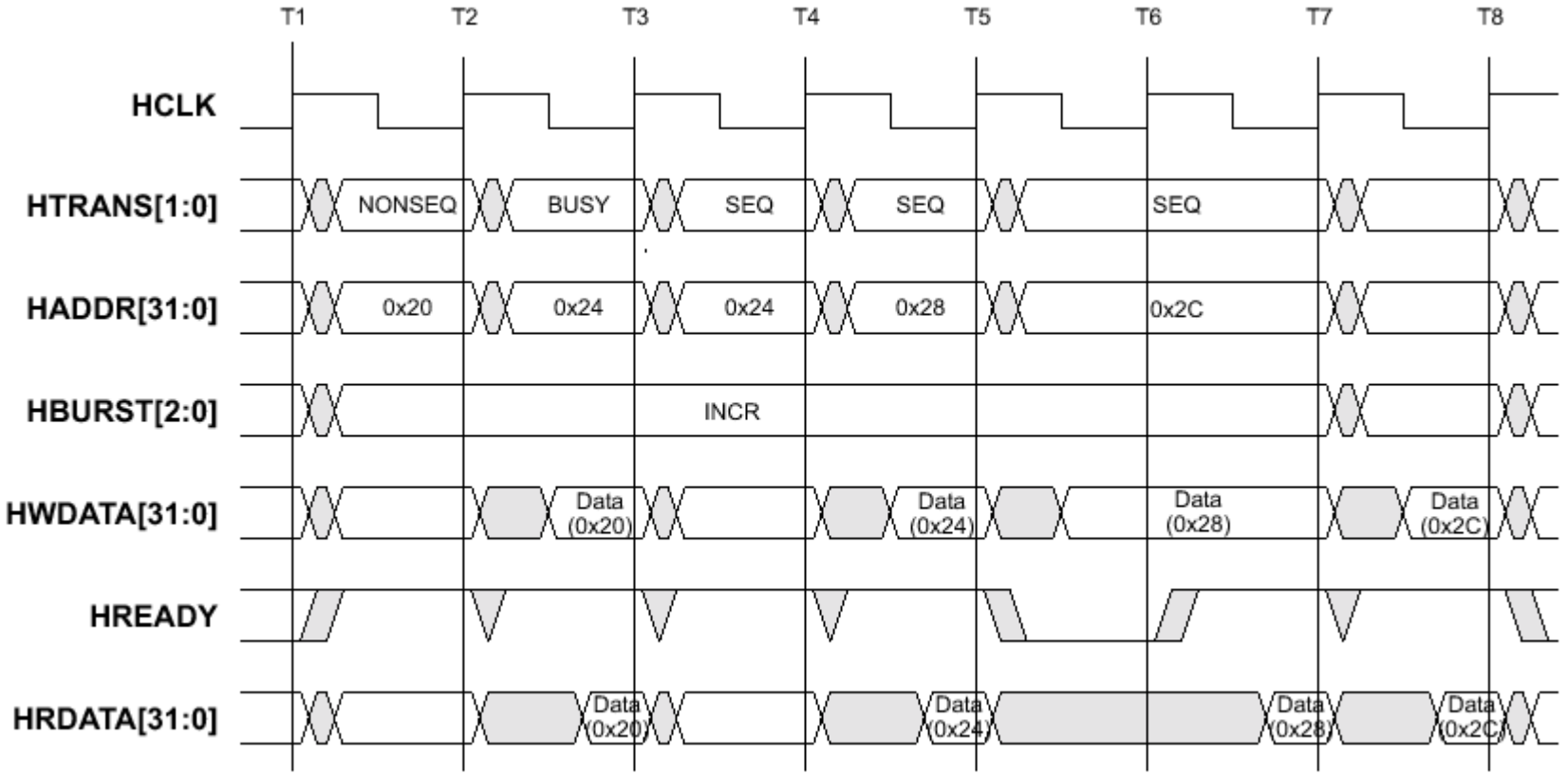
AHB, wait



AHB, multiples transfers



AHB, examples of transfers



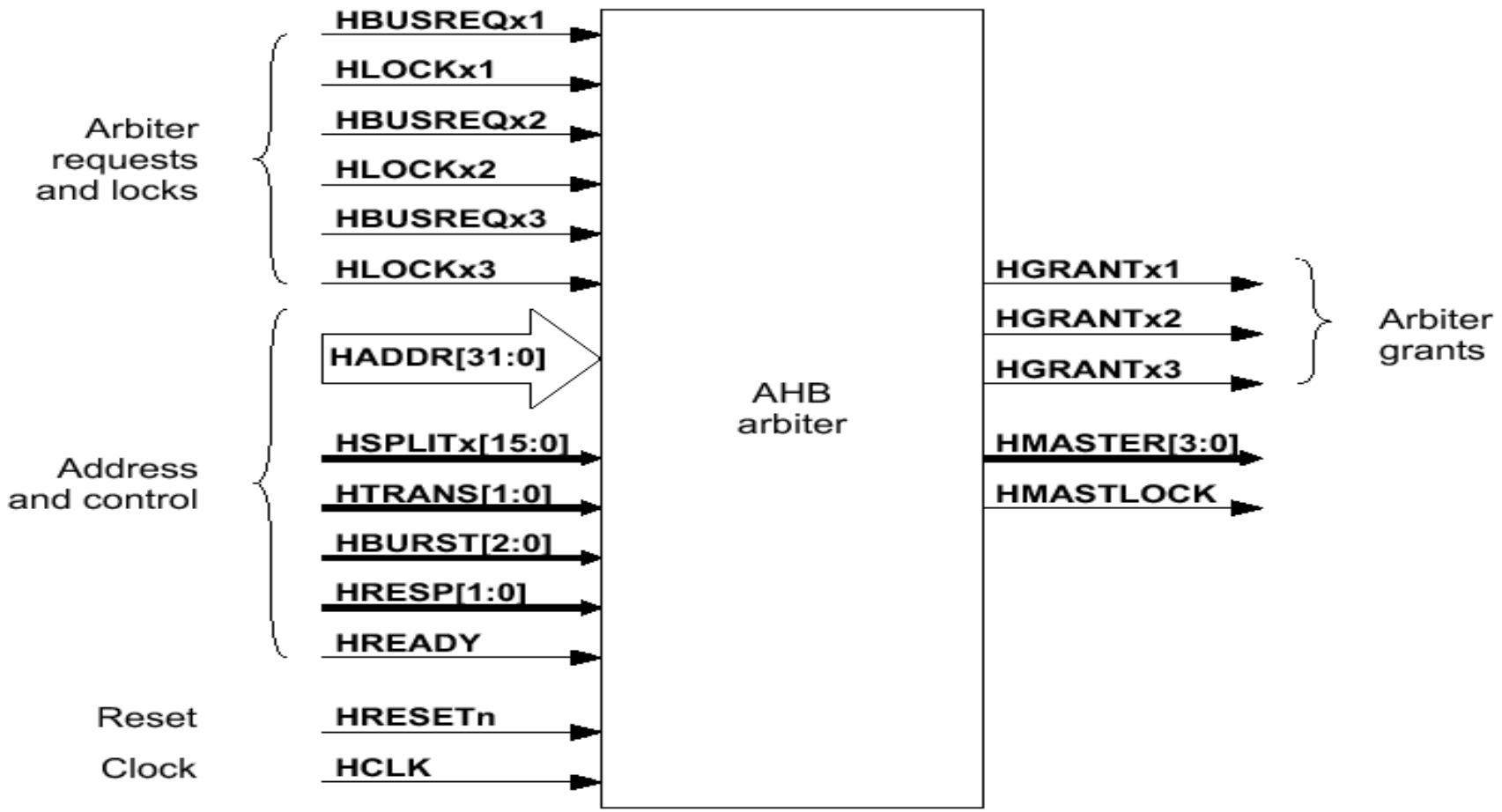
Data bus, little endian

Transfer size	Address offset	DATA [31:24]	DATA [23:16]	DATA [15:8]	DATA [7:0]
Word	0	✓	✓	✓	✓
Halfword	0	-	-	✓	✓
Halfword	2	✓	✓	-	-
Byte	0	-	-	-	✓
Byte	1	-	-	✓	-
Byte	2	-	✓	-	-
Byte	3	✓	-	-	-

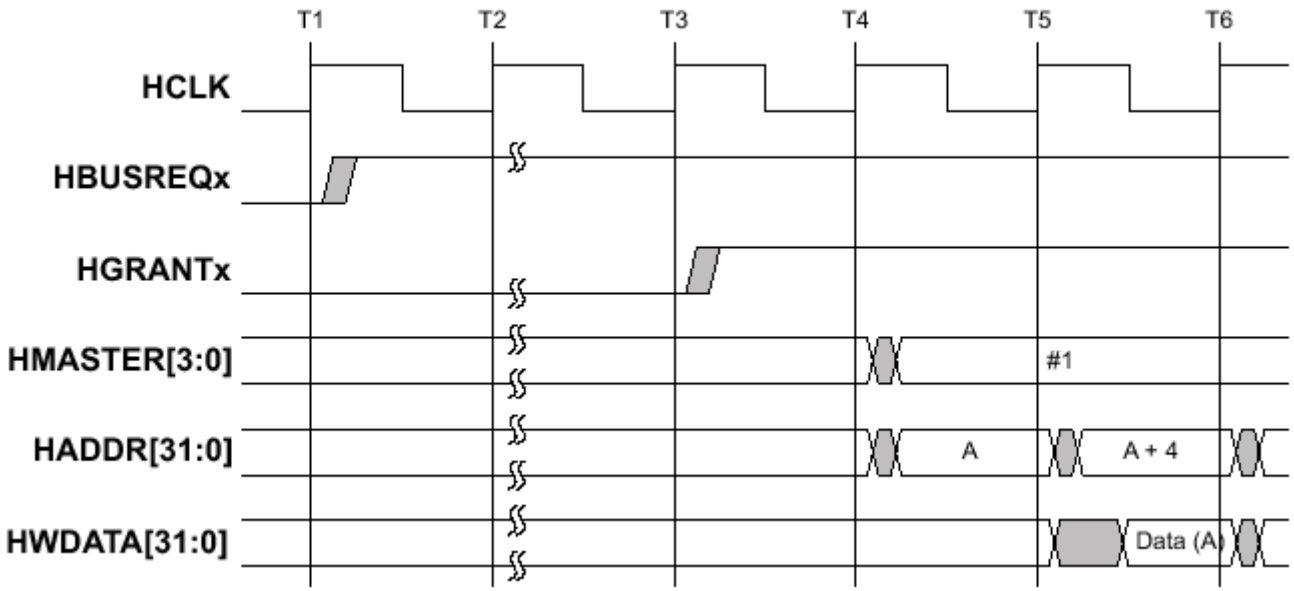
Data bus, big endian

Transfer size	Address offset	DATA [31:24]	DATA [23:16]	DATA [15:8]	DATA [7:0]
Word	0	✓	✓	✓	✓
Halfword	0	✓	✓	-	-
Halfword	2	-	-	✓	✓
Byte	0	✓	-	-	-
Byte	1	-	✓	-	-
Byte	2	-	-	✓	-
Byte	3	-	-	-	✓

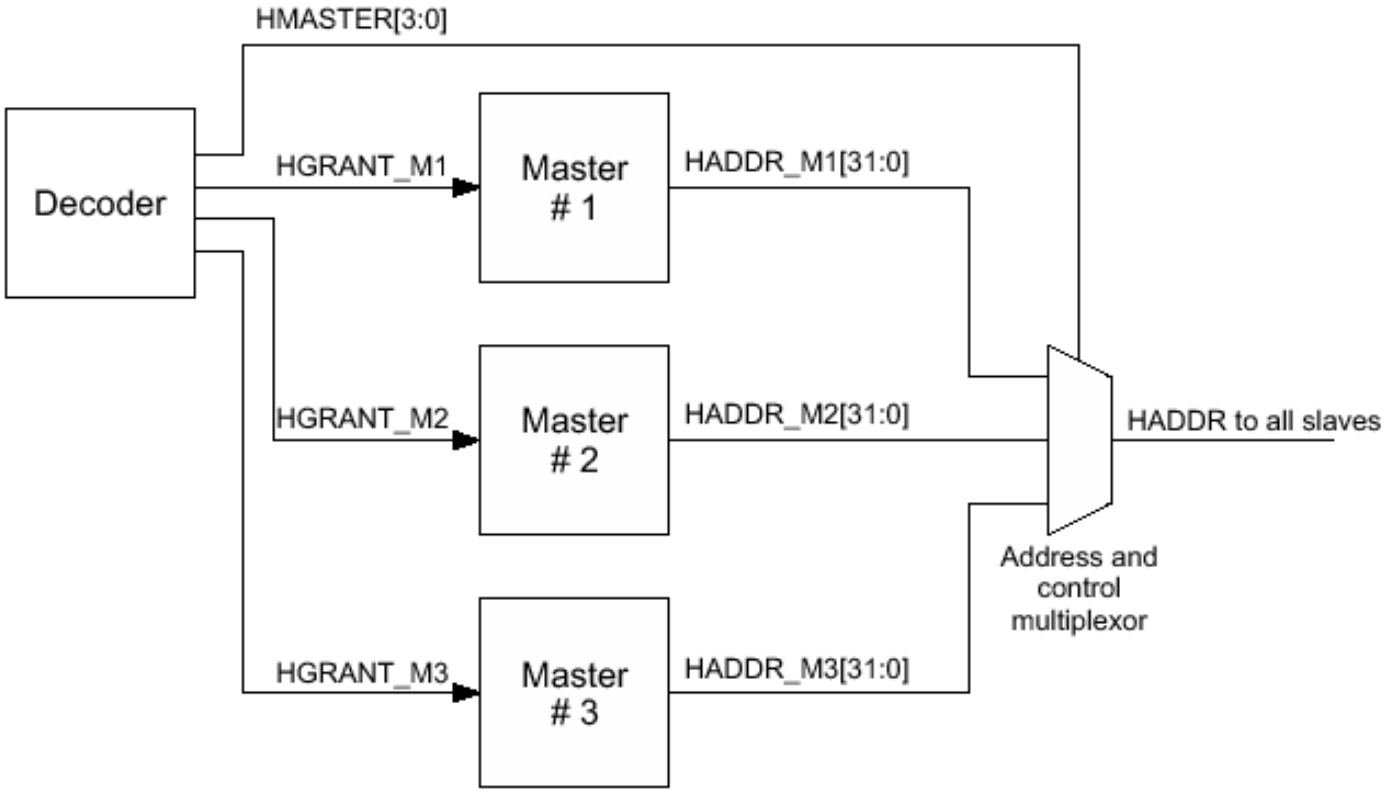
AHB, central arbiter



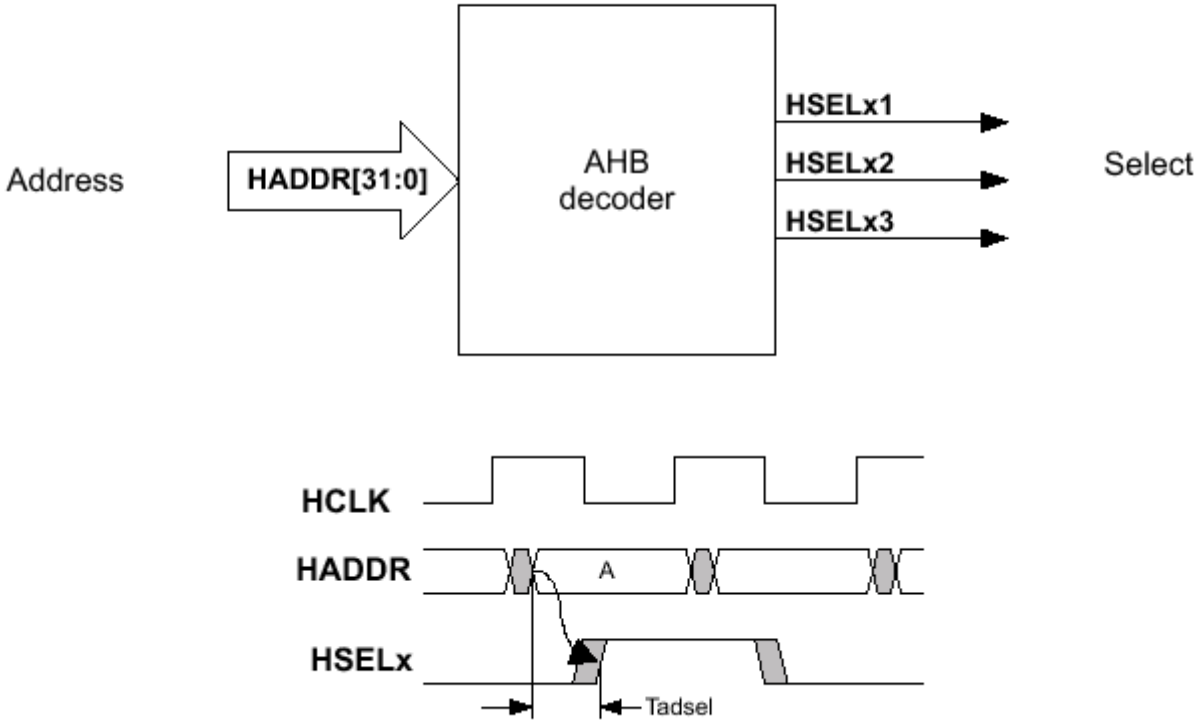
Centralized Arbitration



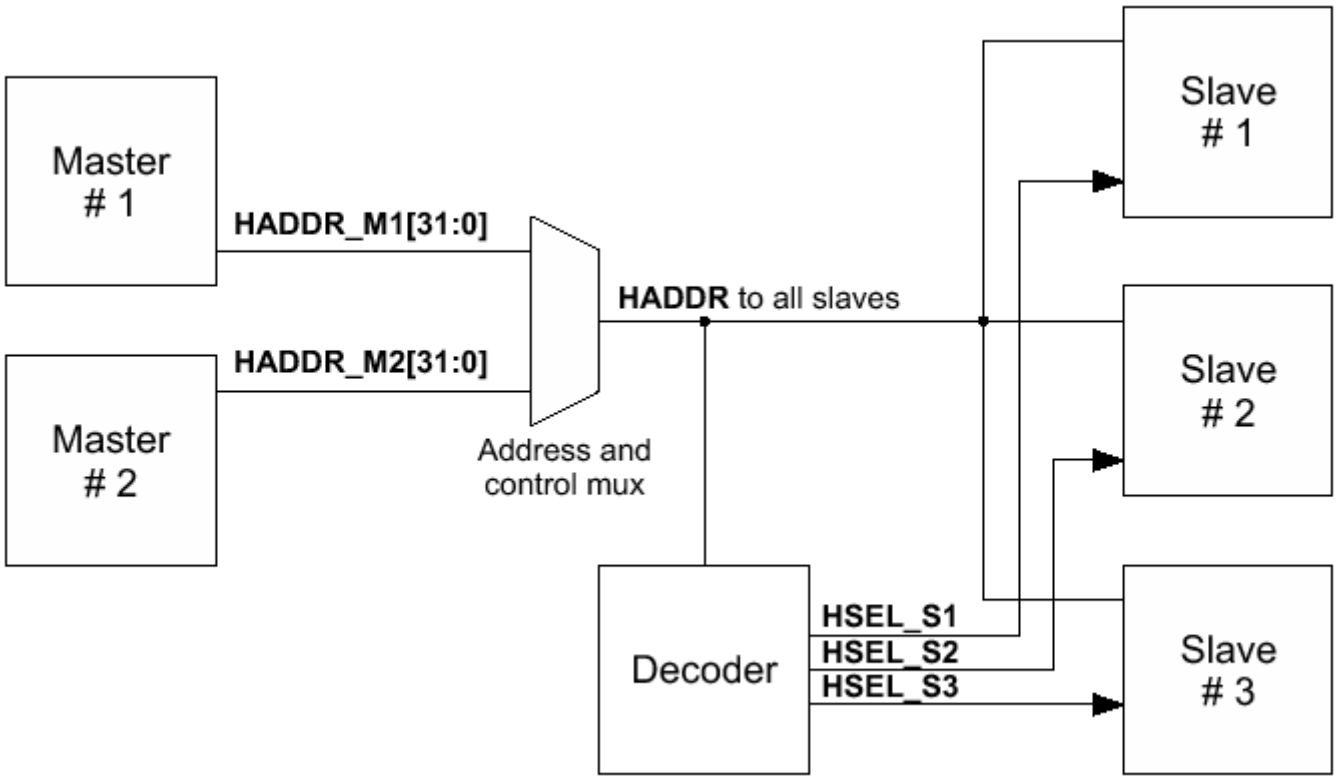
Arbitration



AHB decoder


Address
HADDR[31:0]
AHB
decoder
HSELx1
HSELx2
HSELx3
Select
HCLK
HADDR
A
HSELx
Tadsel

AHB, decoding, example

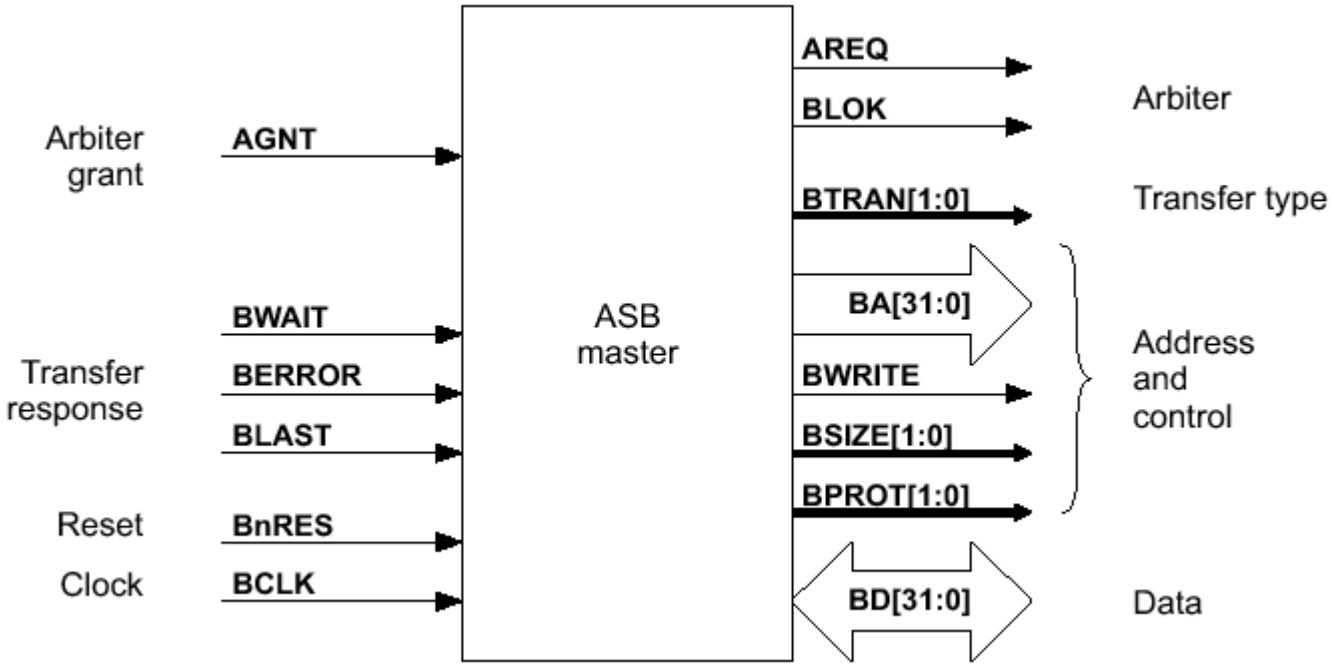


AMBA ASB, old generation

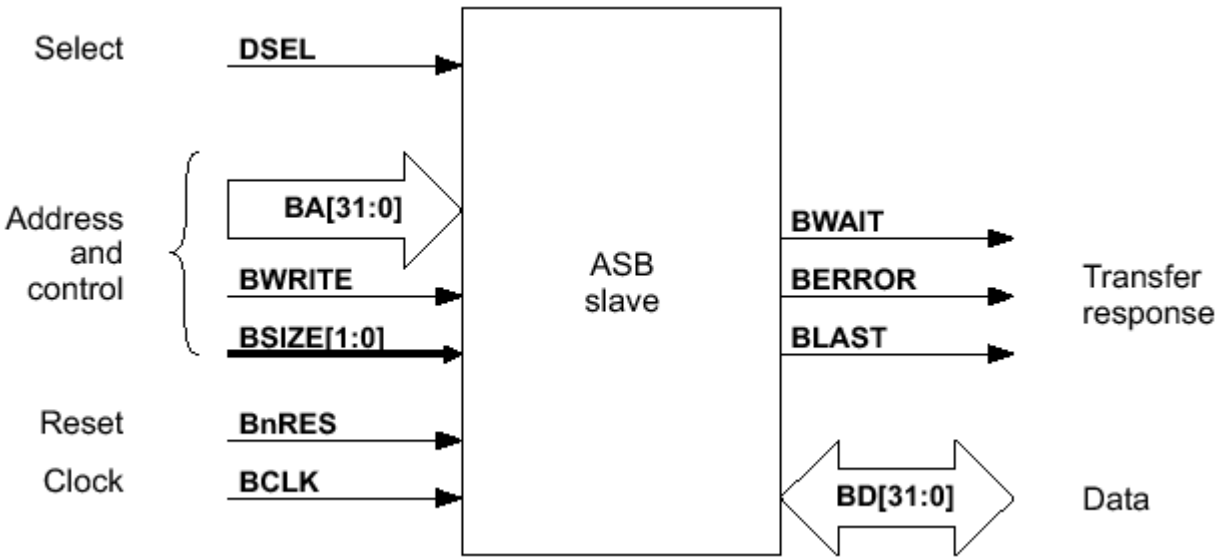
Advanced System Bus

- burst transfers
- pipelined transfer operation
- multiple bus master.

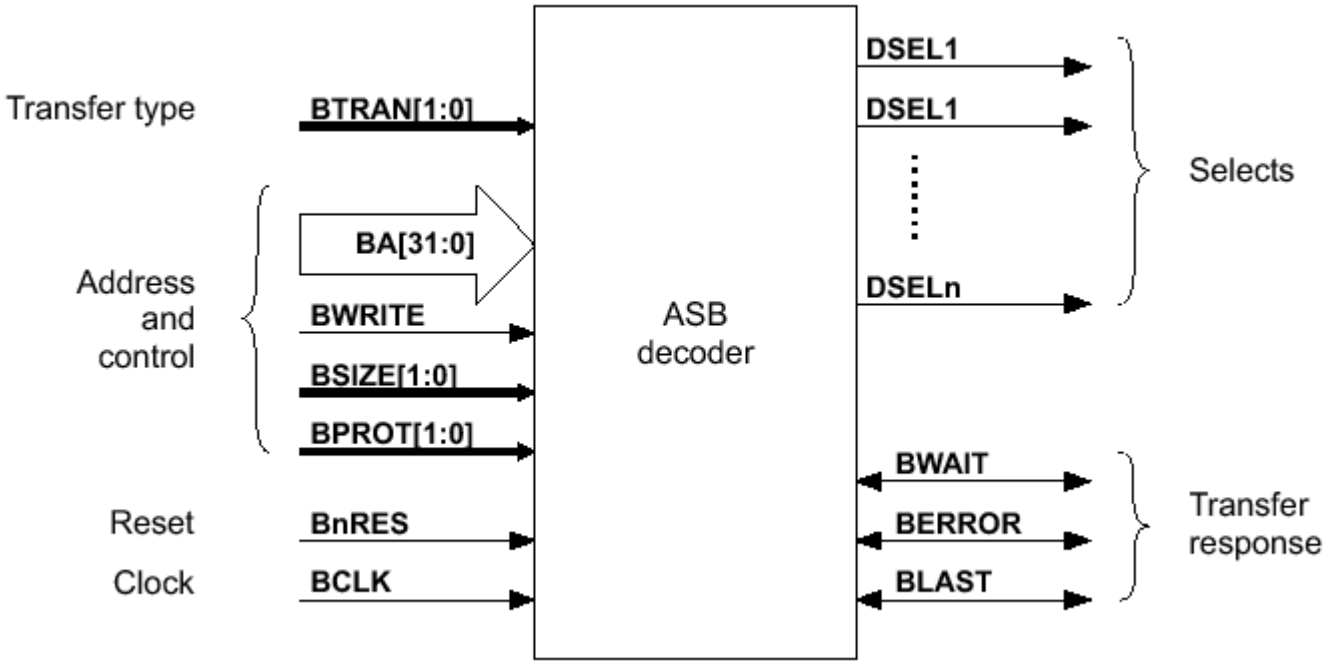
ASB Master


Arbiter grant
AGNT
BWAIT
Transfer response
BERROR
BLAST
Reset
BnRES
Clock
BCLK
ASB master
AREQ
BLOK
BTRAN[1:0]
BA[31:0]
BWRITE
BSIZE[1:0]
BPROT[1:0]
BD[31:0]
Arbiter
Transfer type
Address and control
Data

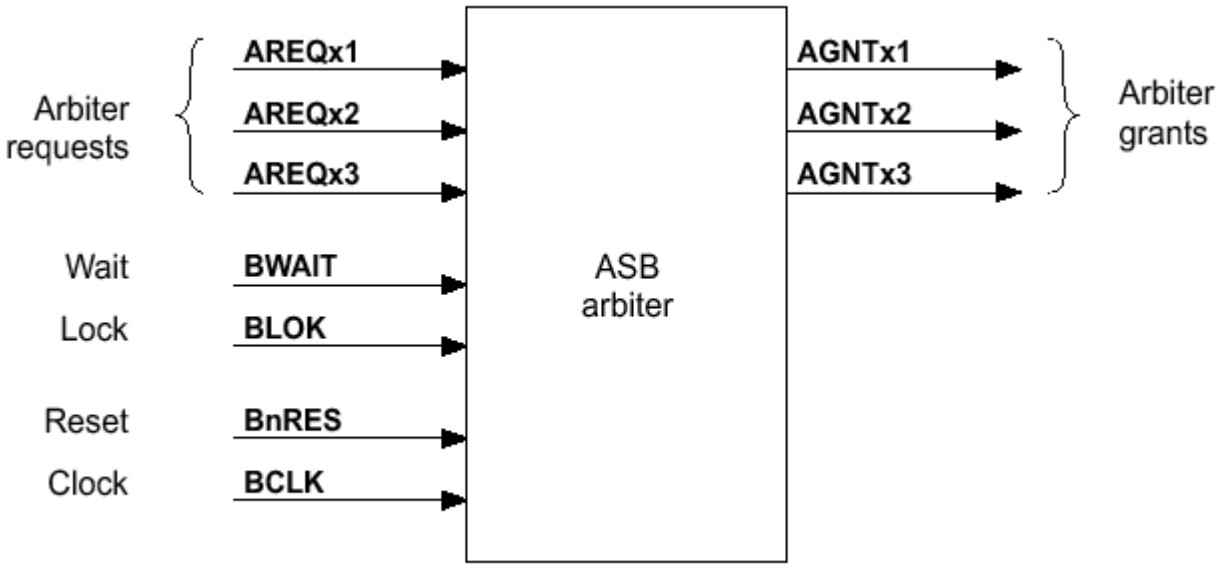
ASB Slave


Select
DSEL
Address and control
BA[31:0]
BWRITE
BSIZE[1:0]
Reset
BnRES
Clock
BCLK
ASB slave
BWAIT
BERROR
BLAST
Transfer response
BD[31:0]
Data

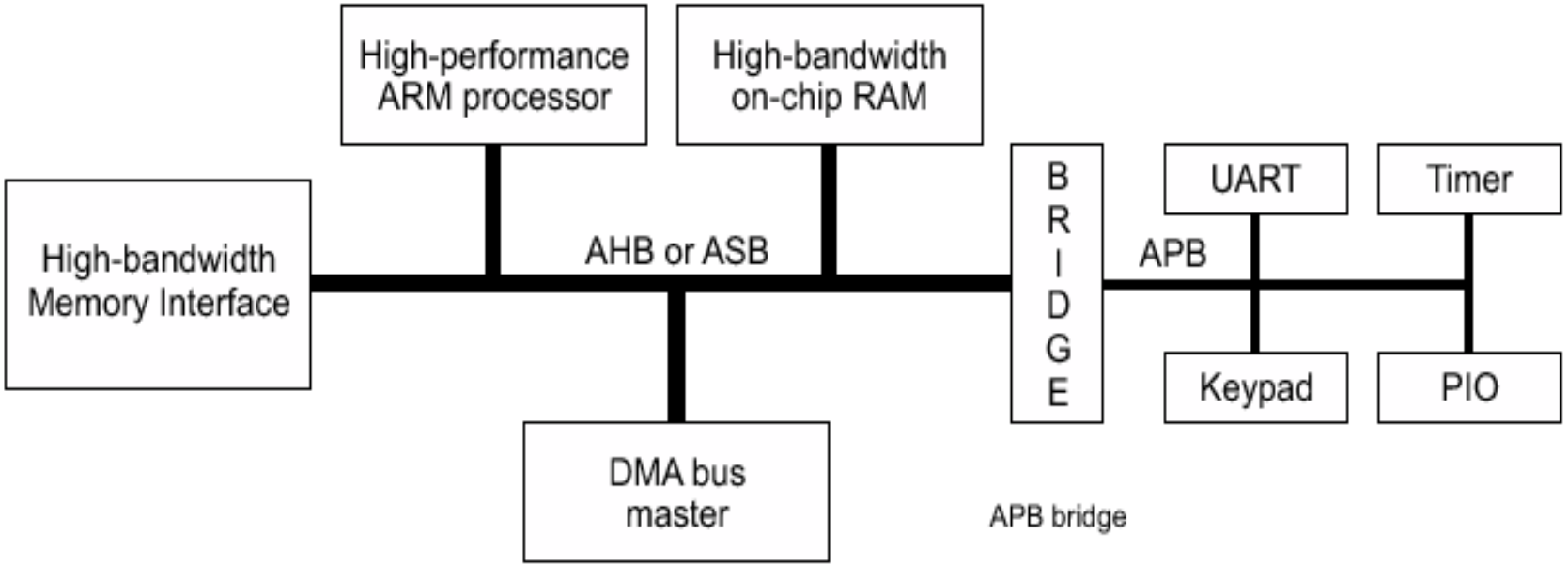
ASB decoder


Transfer type
BTRAN[1:0]
Address and control
BA[31:0]
BWRITE
BSIZE[1:0]
BPROT[1:0]
Reset
BnRES
Clock
BCLK
ASB decoder
DSEL1
DSEL1
DSELn
Selects
BWAIT
BERROR
BLAST
Transfer response

ASB arbiter


Arbiter requests
AREQx1
AREQx2
AREQx3
Wait
BWAIT
Lock
BLOK
Reset
BnRES
Clock
BCLK
ASB arbiter
AGNTx1
AGNTx2
AGNTx3
Arbiter grants

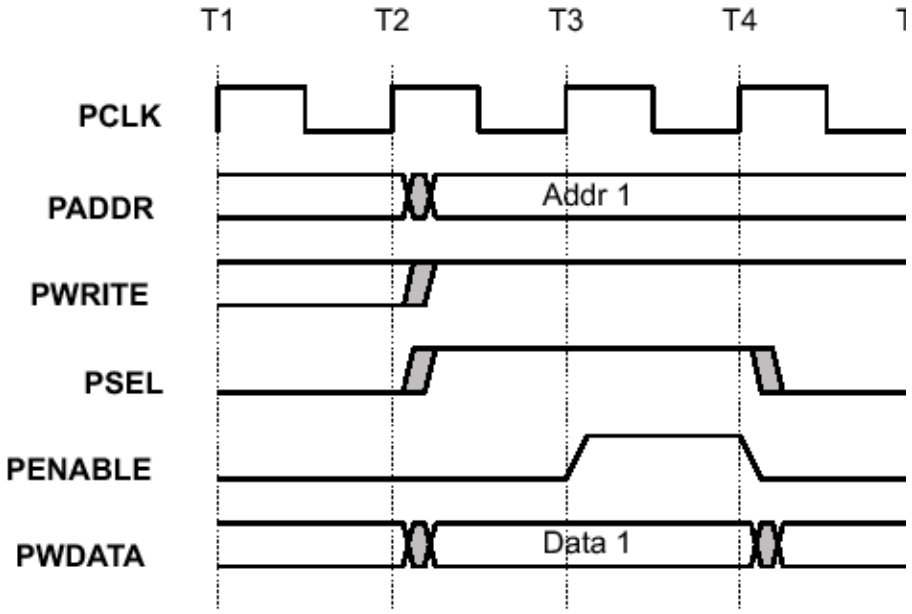
AMBA APB



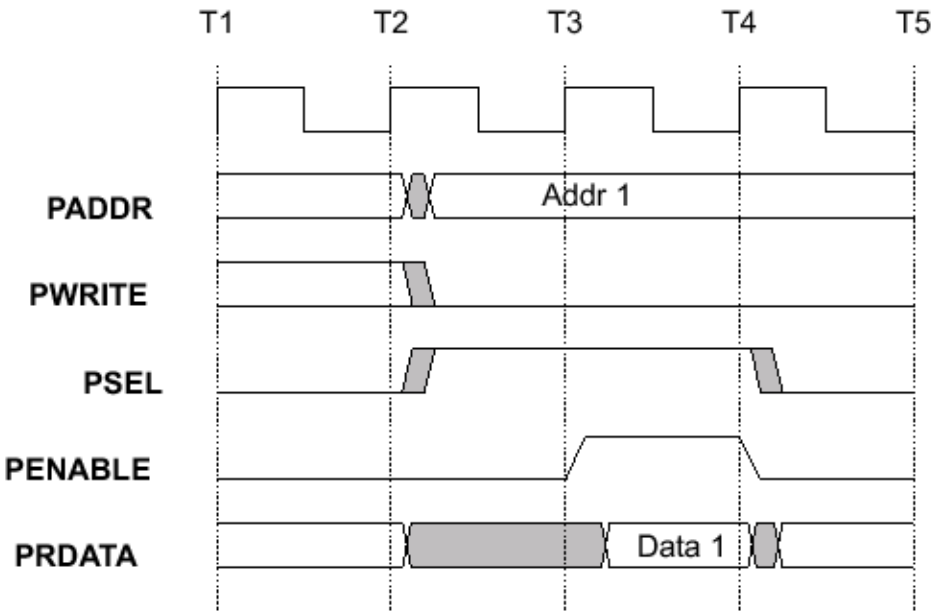
AMBA Advanced Peripheral Bus (APB)

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

APB transferts

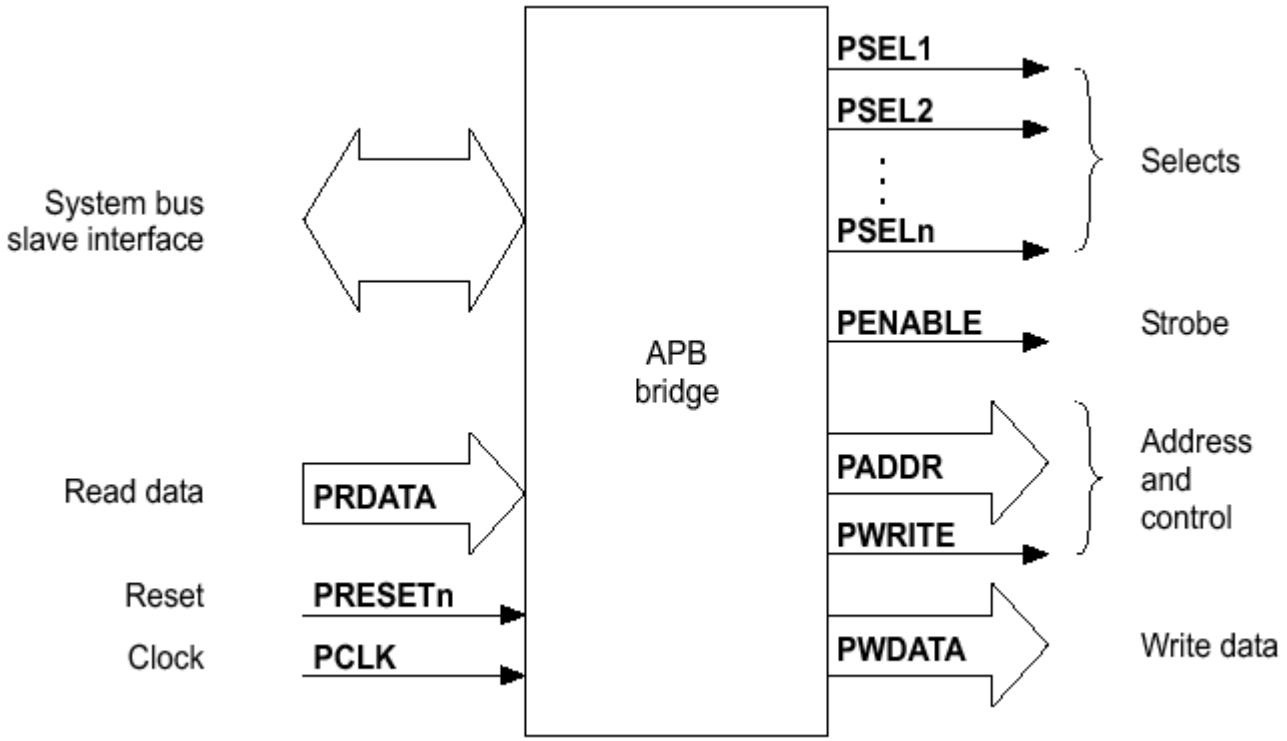


Write transfer

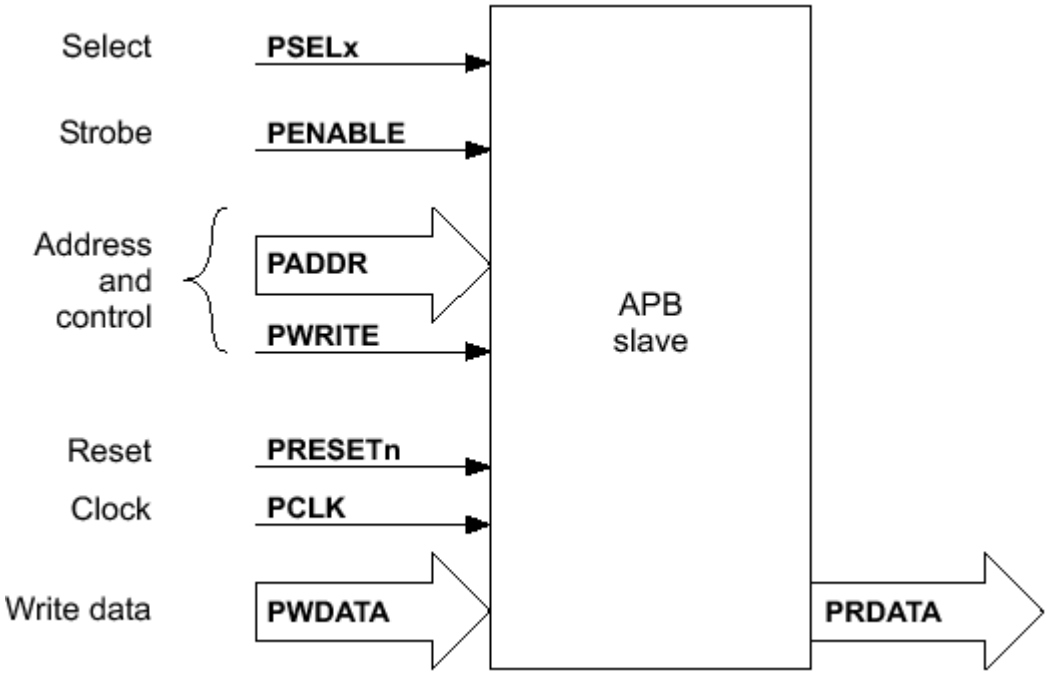


Read transfer

APB bridge

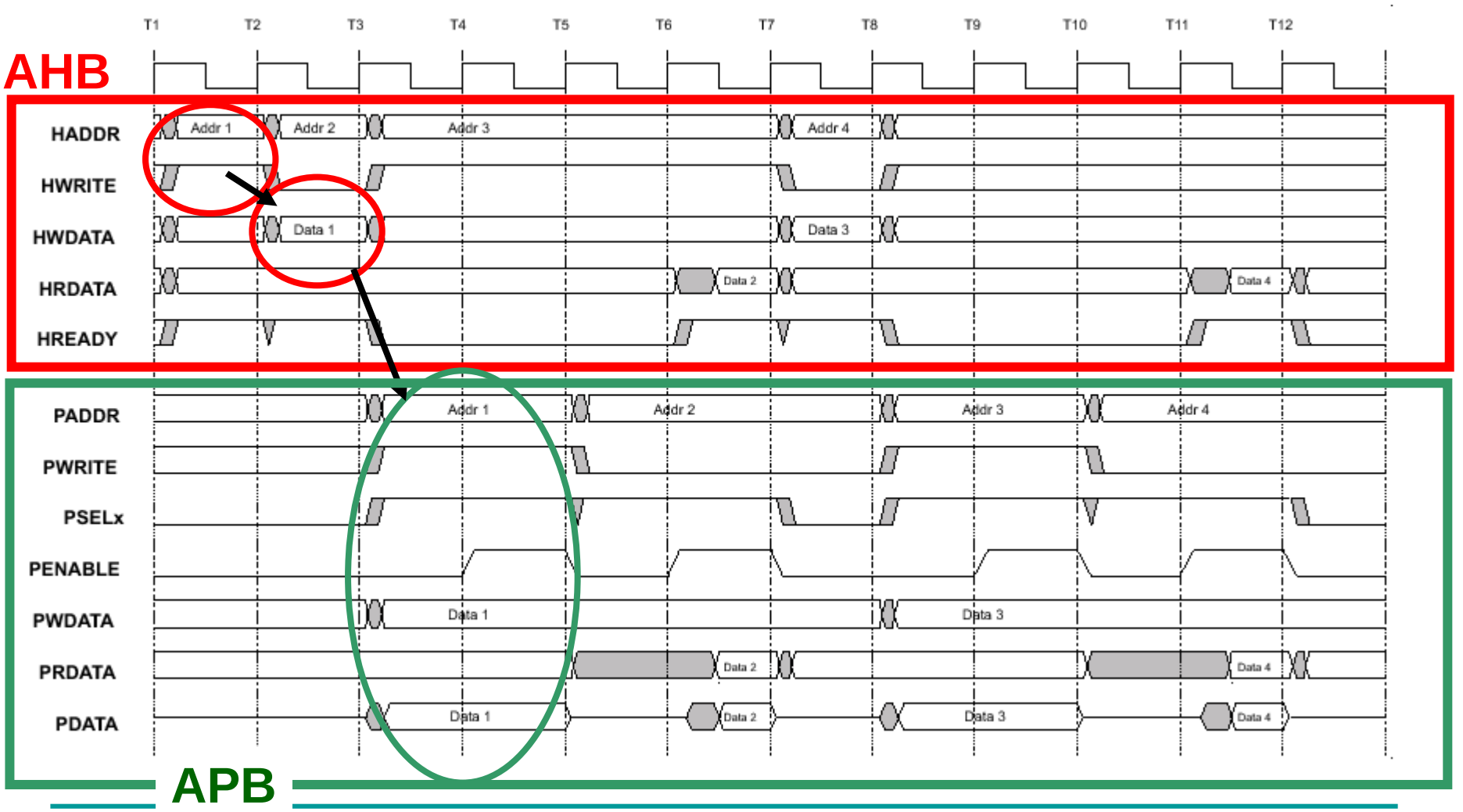

System bus
slave interface
Read data
PRDATA
Reset
PRESETn
Clock
PCLK
APB
bridge
PSEL1
PSEL2
PSELn
Selects
PENABLE
Strobe
PADDR
PWRITE
Address
and
control
PWDATA
Write data

APB Slave


Select
PSELx
Strobe
PENABLE
Address
and
control
PADDR
PWRITE
APB
slave
Reset
PRESETn
Clock
PCLK
Write data
PWDATA
PRDATA

APB, tri-state Bus available depending on the implementation

write cycle



APB, tri-state Bus available depending on the implementation

read cycle