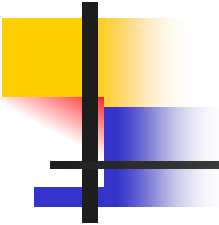


2. AMBA Specification



AMBA background

- ARM사가 선보인 AMBA(Advanced Microcontroller Bus Architecture)는 오픈 표준 버스 규격
- AMBA는 SOC를 구성하는 기능 블록들간의 연결 및 관리 방법으로써 하나 또는 그 이상의 CPU/ DSP를 내장한 임베디드 제품의 적시 출시
- AMBA 버스는 SOC 내부 모듈들을 위한 공통 시스템 버스 (backbone)를 정의함으로써 디자인의 재사용을 한층 강화시키는 장점을 제공

AMBA Overview

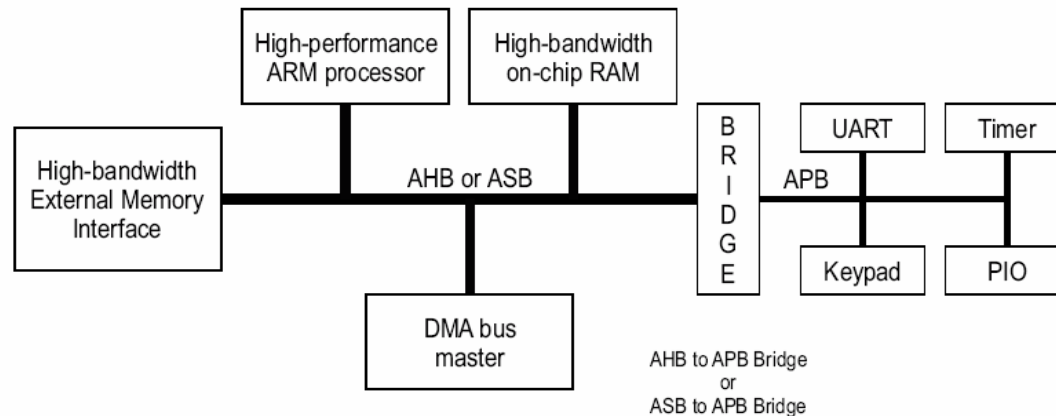
- Advanced Microcontroller Bus Architecture
- SoC Target On-chip bus protocol
- 3 type of bus
 - AHB(the advanced High-performance bus)
 - ASB(the Advanced System Bus)
 - APB(the Advanced Peripheral Bus)
 - AXI(the Advanced eXtensible Interface)
 - spec 3.0

AMBA AHB introduce

- AHB master
 - read/write 요청
 - 한번에 하나의 master만이 bus를 사용 가능
- AHB slave
 - 주어진 address 범위 내에서 read/write 동작 수행
 - 데이터 전송에 대한 success, failure, waiting 상태에 대한 정보를 master에 보고
- AHB arbiter
 - 한번에 하나의 master만이 bus를 사용하도록 함
- AHB decoder
 - decoder는 전송하고자 하는 slave의 어드레스를 decode
 - 하나의 decoder가 모든 AHB상의 slave들을 decode

A typical AMBA-based system

Split transactions enable the slave to defer the operation until it's able to accomplish it, thereby releasing the bus for other accesses. The slave signals a split transaction and saves the master number (HMASTER []). When ready to complete the transaction, the slave signals the arbiter with the master number. When the arbiter grants bus access to the master, it restarts the transaction.



AMBA AHB

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

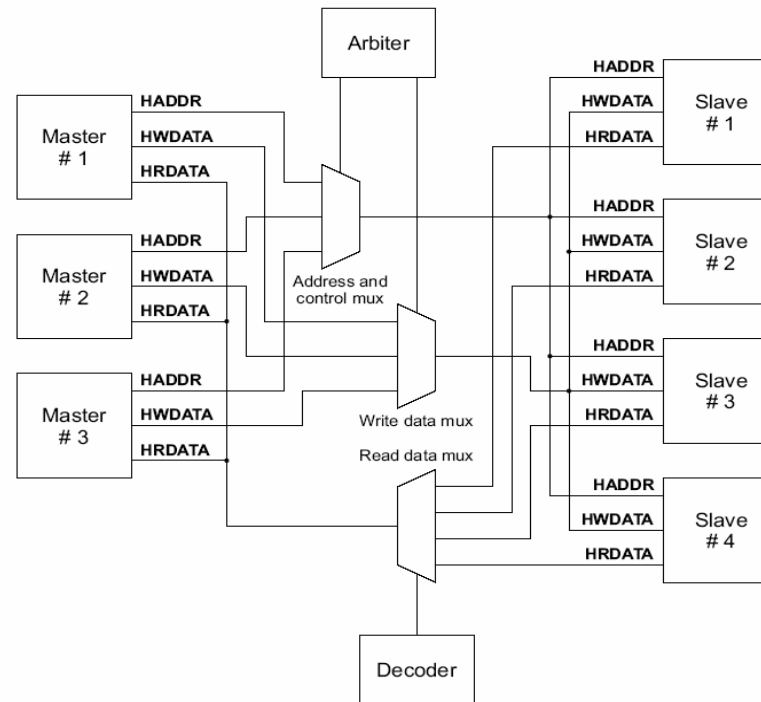
AMBA ASB

- * High performance
- * Pipelined operation
- * Multiple bus masters

AMBA APB

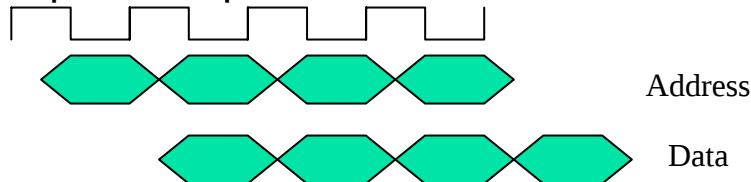
- * Low power
- * Latched address and control
- * Simple interface
- * Suitable for many peripherals

Bus interconnection



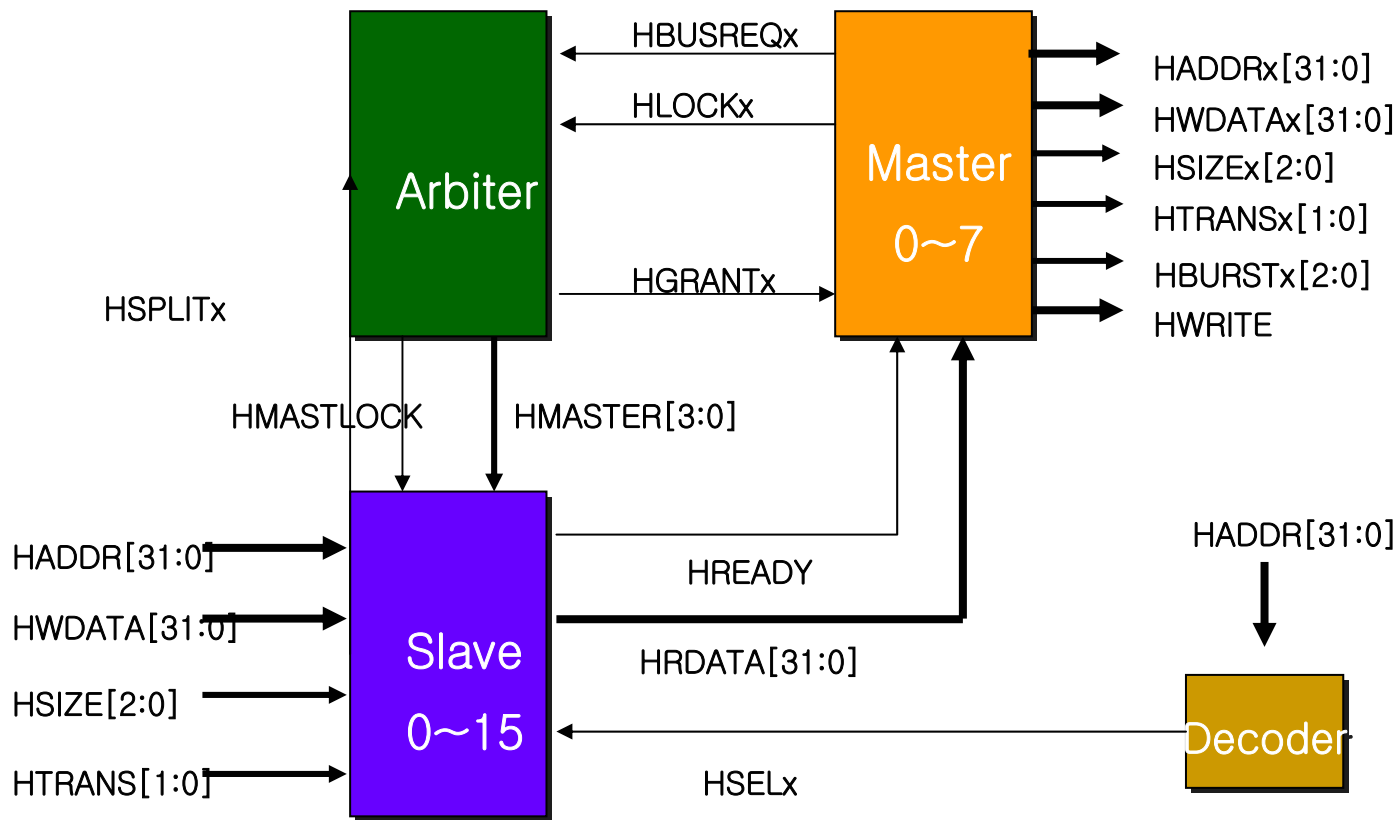
AMBA AHB Overview

- High performance
- Pipelined operation



- Multiple bus masters
 - It is necessary to implement Arbiter.
- Single cycle bus master handover
- Burst transfers
- Split transactions
- Single clock edge operation
 - Rising edge
- Non-tristate implementation
 - Uni-direction vs Bi-direction
- Wide data bus configuration(64/128 bits)

AMBA diagram



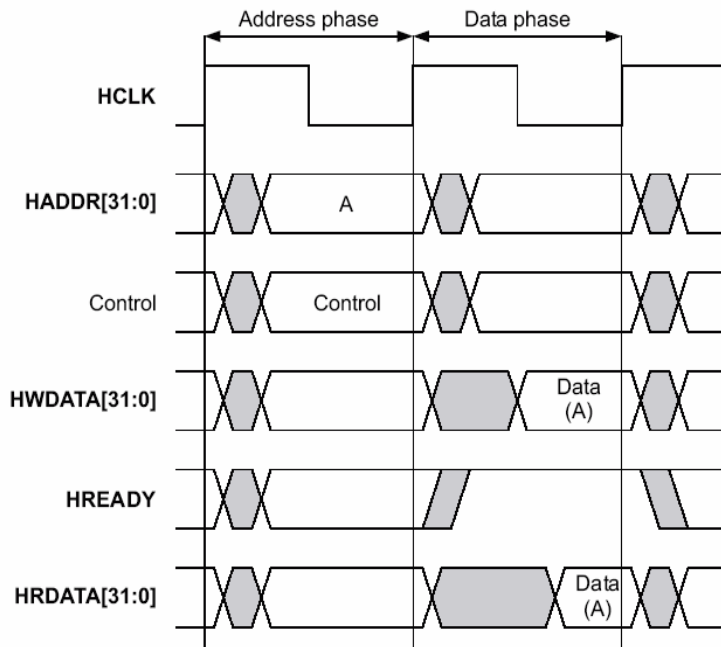
AMBA AHB signal list(1)

Name	Source	Description
HCLK	Clock source	Bus clock (rising edge)
HRESETn	Reset controller	Reset(Active low)
HADDR[31:0]	Master	Address bus
<u>HTRANS[1:0]</u>	Master	Transfer type(nonsequential,sequential,idle,busy)
HWRITE	Master	Transfer direction(high-write/low-read)
HSIZE[2:0]	Master	Transfer size[(8 bit,16 bit,32bit), 최대 1024bit]
HBURST[2:0]	Master	Burst type(4,8,16 beat burst 지원)
<u>HPROT[3:0]</u>	Master	Protection control(opcode fetch or data access) (privileged mode access, user mode access)
HWDATA[31:0]	Master	Write data bus
HSELx	Decoder	Slave select
HRDATA[31:0]	Slave	Read data bus
HREADY	Slave	Transfer done(high-done)
HRESP	Slave	Transfer response(okey,error,retry,split)

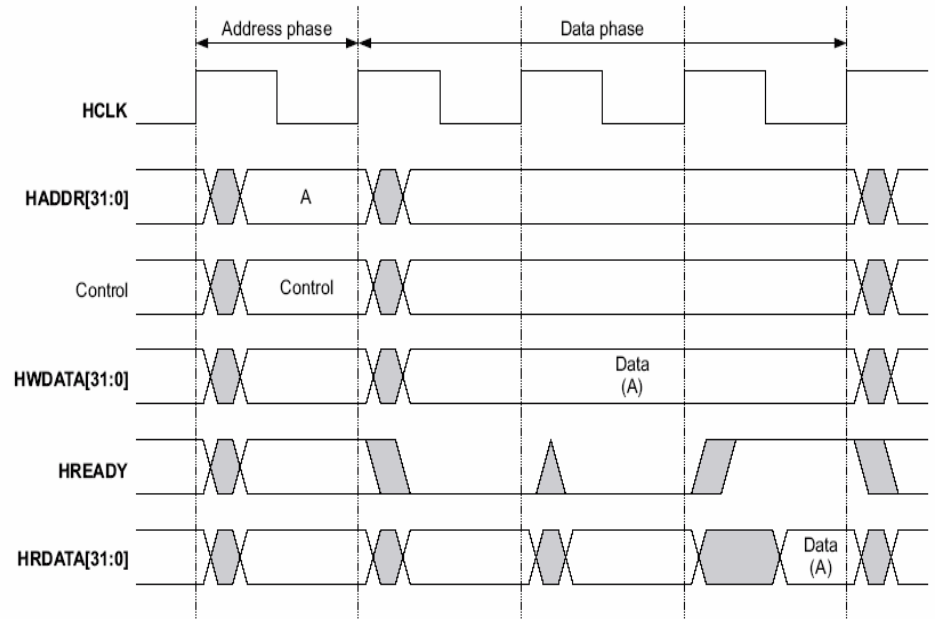
AMBA AHB signal list(2)

Name	Source	Description
<u>HBUSREQx</u>	Master	Bus request(최대 16 bus master)
<u>HLOCKx</u>	Master	Locked transfers(High-독점하다가 low가 될 때 까지 다른 master는 사용 못함)
<u>HGRANTx</u>	Arbiter	Bus grant (버스 사용 – HREADY,HGRANTx가 high일 때)
<u>HMASTER[3:0]</u>	Arbiter	Master number
<u>HMASTLOCK</u>	Arbiter	Locked sequence (HMASTER signal과 같은 timing)
<u>HSPLITx[15:0]</u>	Slave	Split completion request

Basic transfer



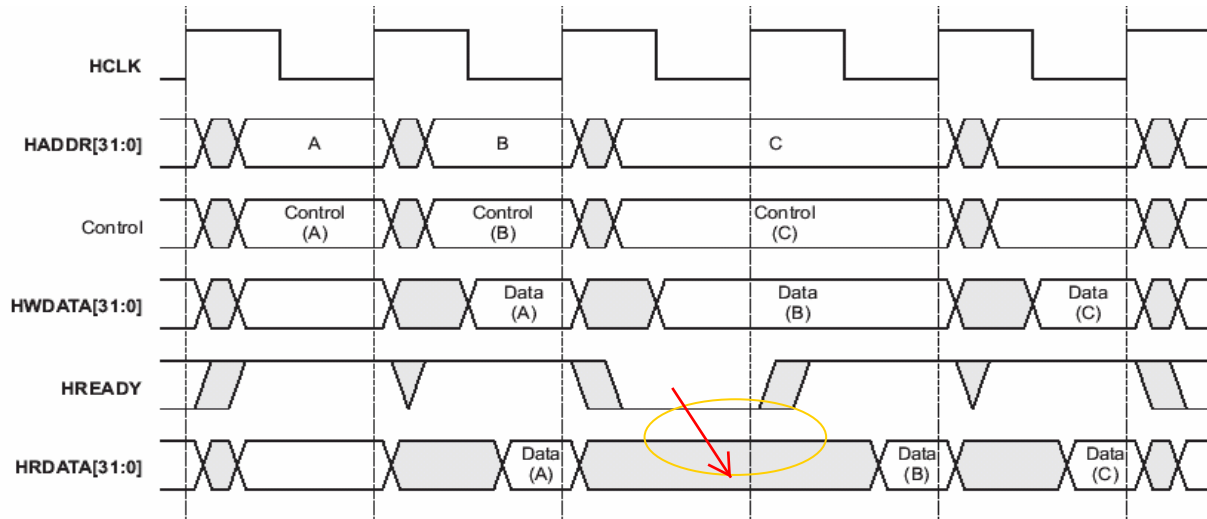
✚ Simple transfer



✚ Transfer with wait states

HREADY 신호가 '0'이면 Data phase는 지속된다

Multiple transfer

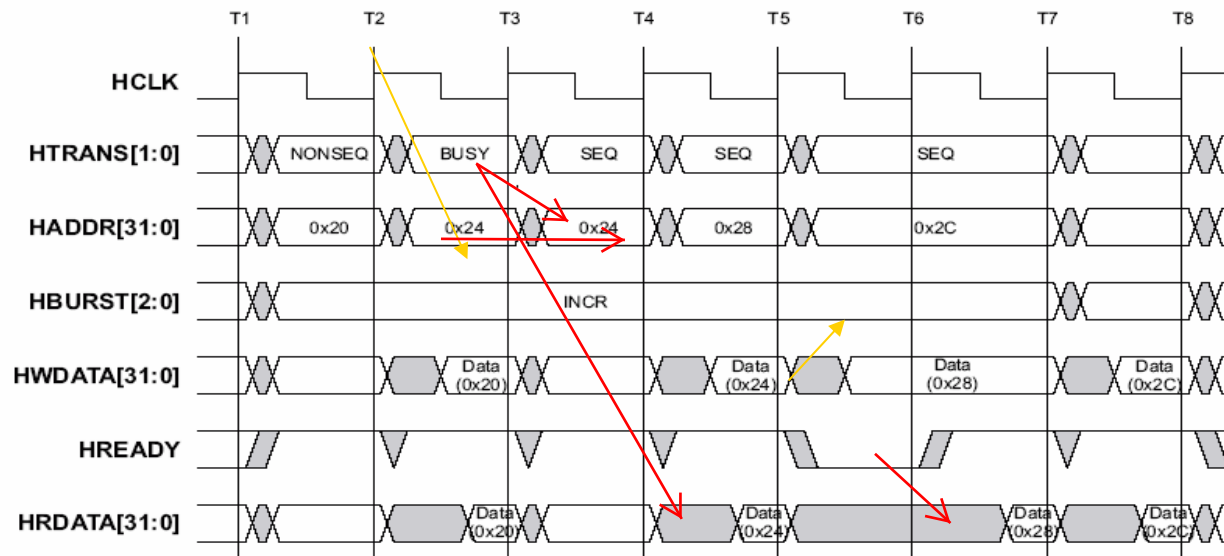


Multiple transfer

Transfer type

HTRANS [1:0]	Type	Description
00	IDLE	데이터 전송을 하고 있지 않은 상태이므로 Master가 Slave로 데이터를 전송하고자 할 때 언제든지 Slave로부터 Okay응답을 받을 수 있다.
01	BUSY	Burst전송 동작 중 IDLE상태를 갖도록 한다. <u>Burst동작 중</u> 데이터를 가져오기 전에 어떤 처리를 위하여 기다리게 될 경우에 중간에 IDLE상태를 만들기 위해 BUSY를 사용한다.
10	NONSEQ	단일 데이터 전송이나 Burst전송의 첫 번째 동작에 사용된다. (전에 전송한 것과 관련없다.)
11	SEQ	Burst로 데이터를 전송할 경우 NONSEQ의 다음 데이터부터 SEQ로 동작한다. (전에 전송한 것과 관련있다.)

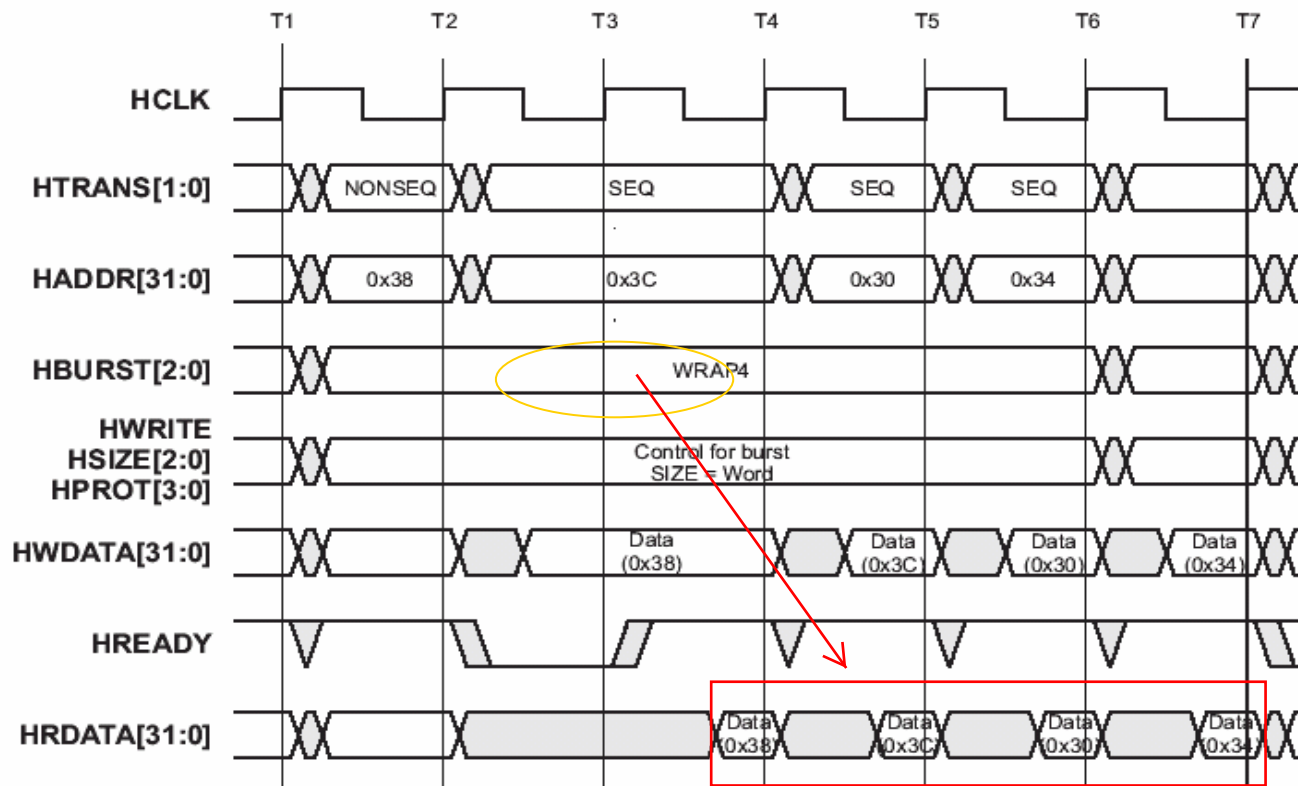
Transfer type examples



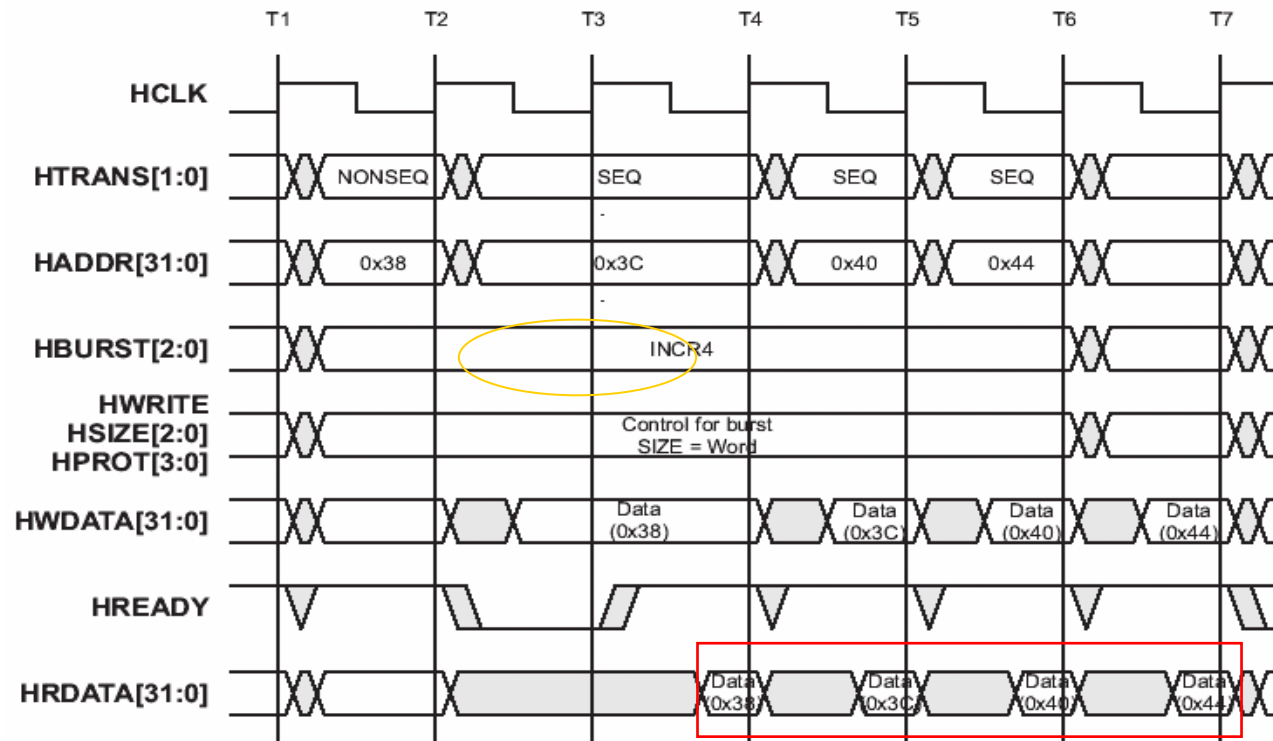
Burst operation

HBURST[2:0]	Type	Description
000	SINGLE	Single transfer
001	INCR	Incrementing burst of unspecified length
010	WRAP4	4-beat wrapping burst
011	INCR4	4-beat increment burst
100	WRAP8	8-beat wrapping burst
101	INCR8	8-beat increment burst
110	WRAP16	16-beat wrapping burst
111	INCR16	16-beat increment burst

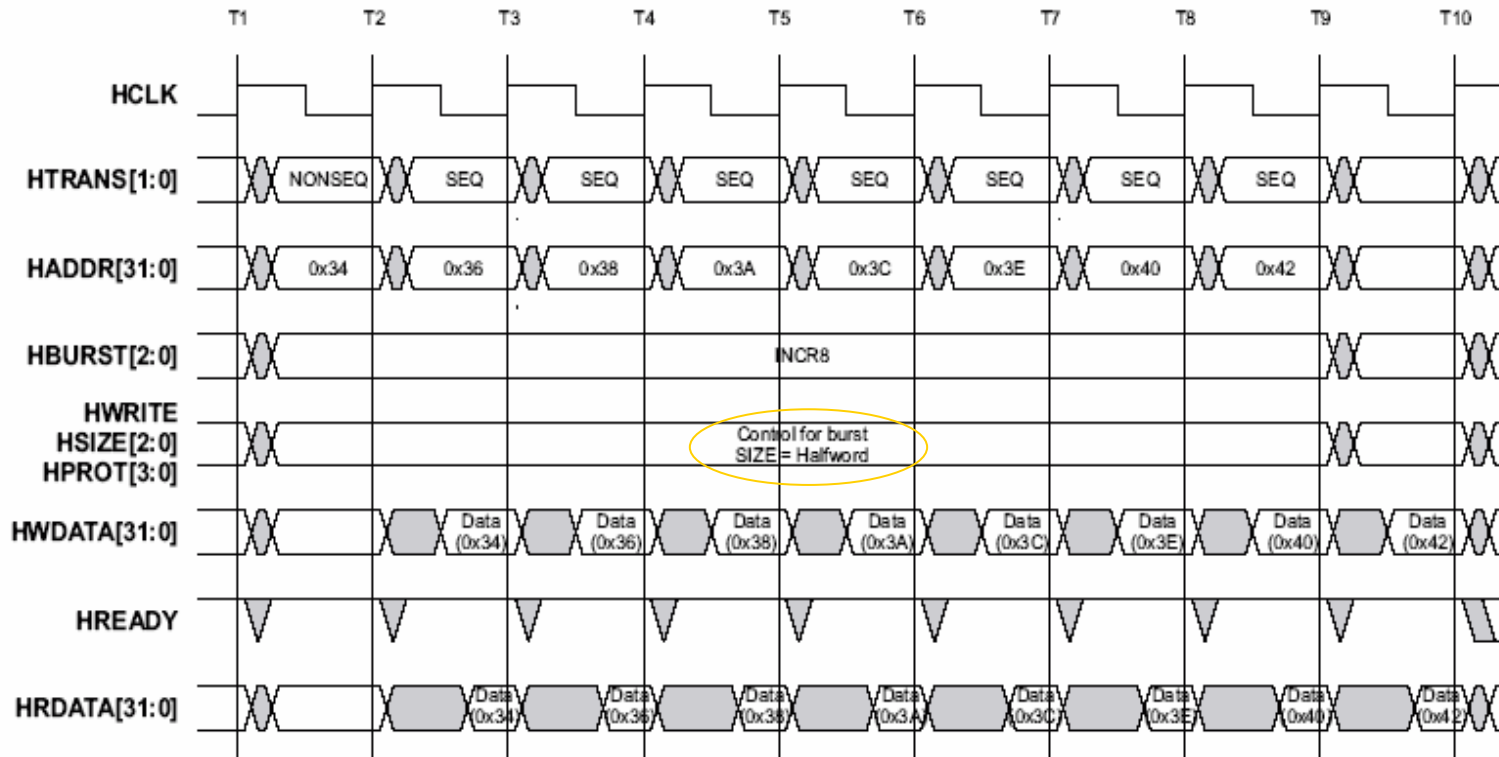
4-beat wrapping burst



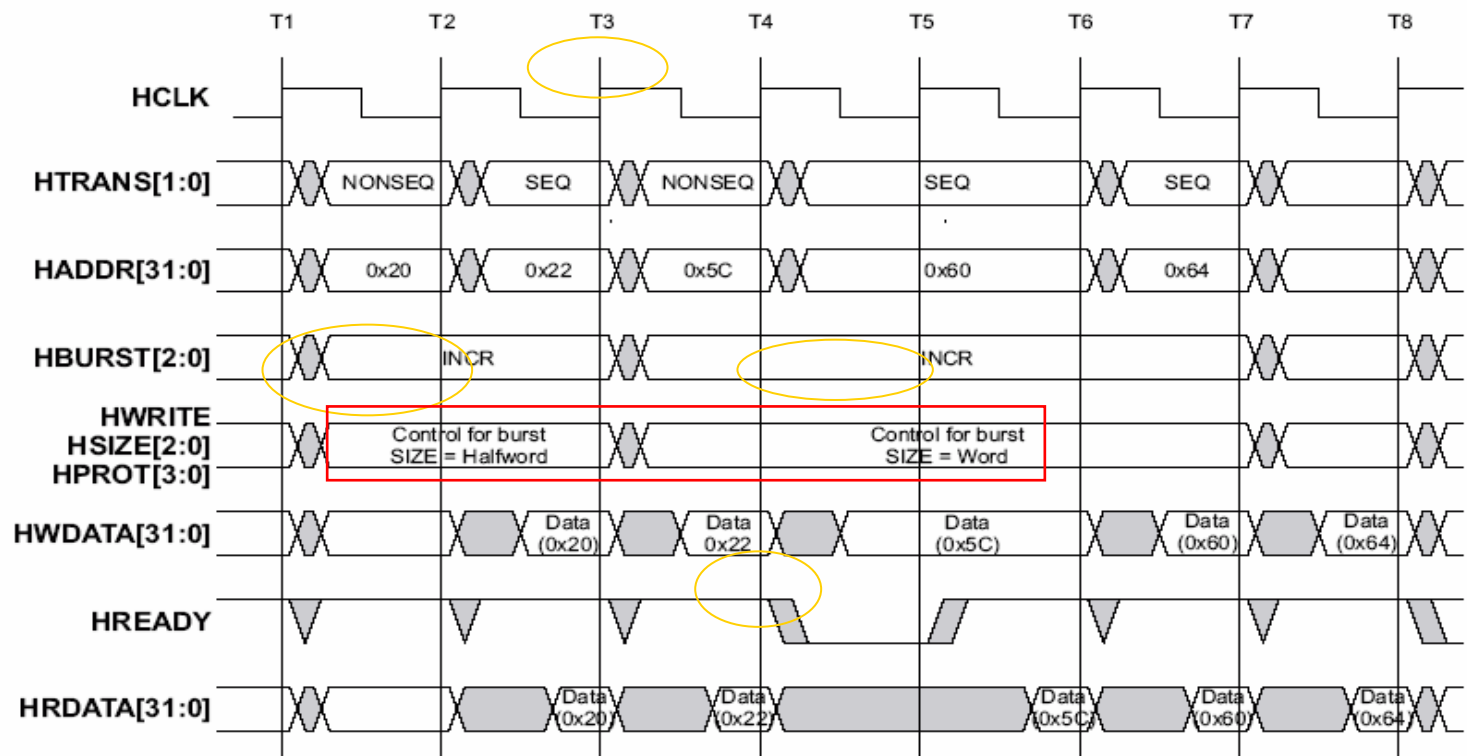
4-beat incrementing burst



8-beat incrementing burst

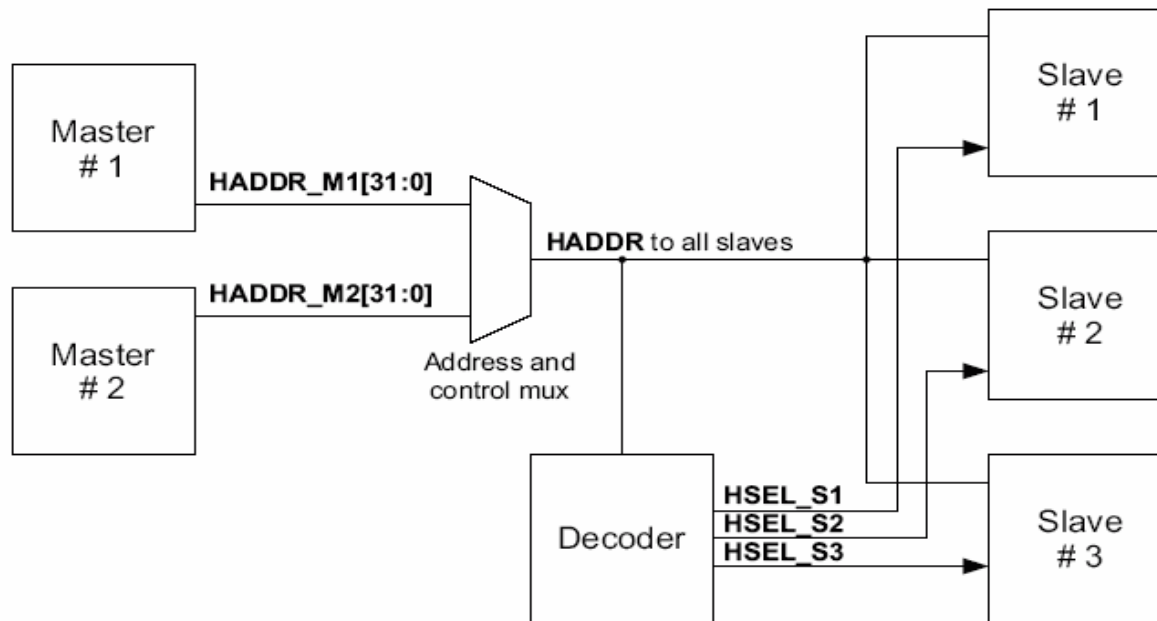


Undefined-length bursts

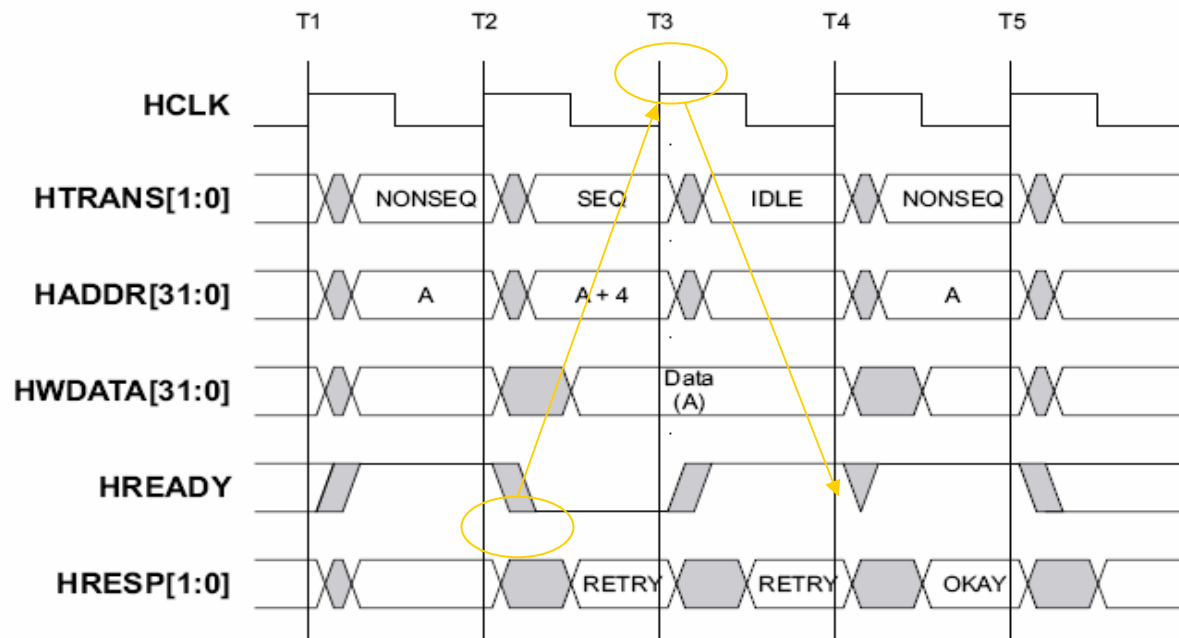


HSIZE[2:0]	Size	Description
000	8bits	Byte
001	16bits	Halfword
010	32bits	Word
011	64bits	-
100	128bits	4-word line
101	256bits	8-word line
110	512bits	-
111	1024bits	-

Address decoding

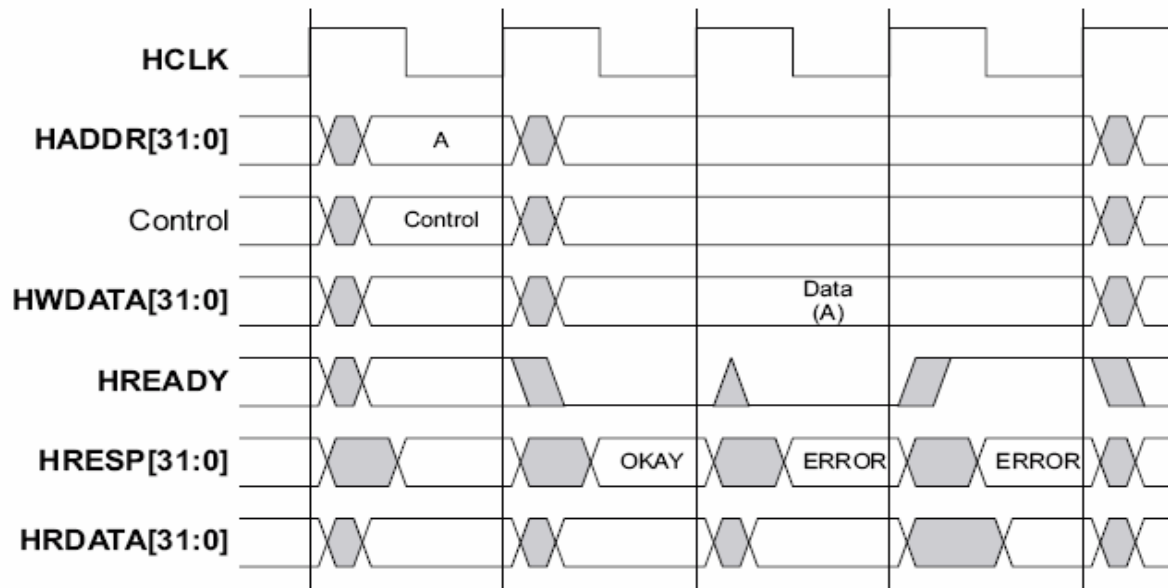


Slave transfer response : retry

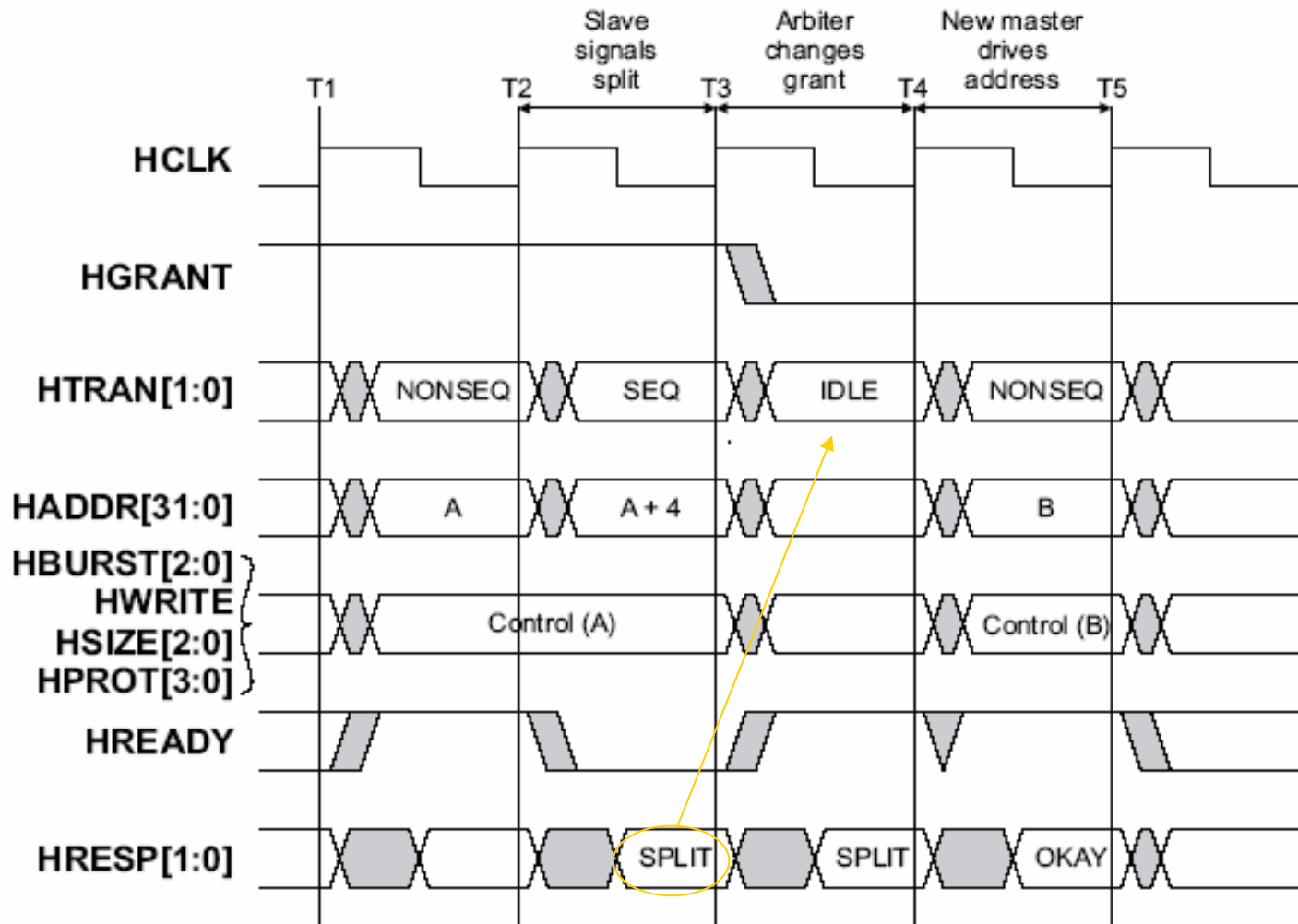


- okey response : 1 cycle
- error, split, retry : 2 cycle 이상

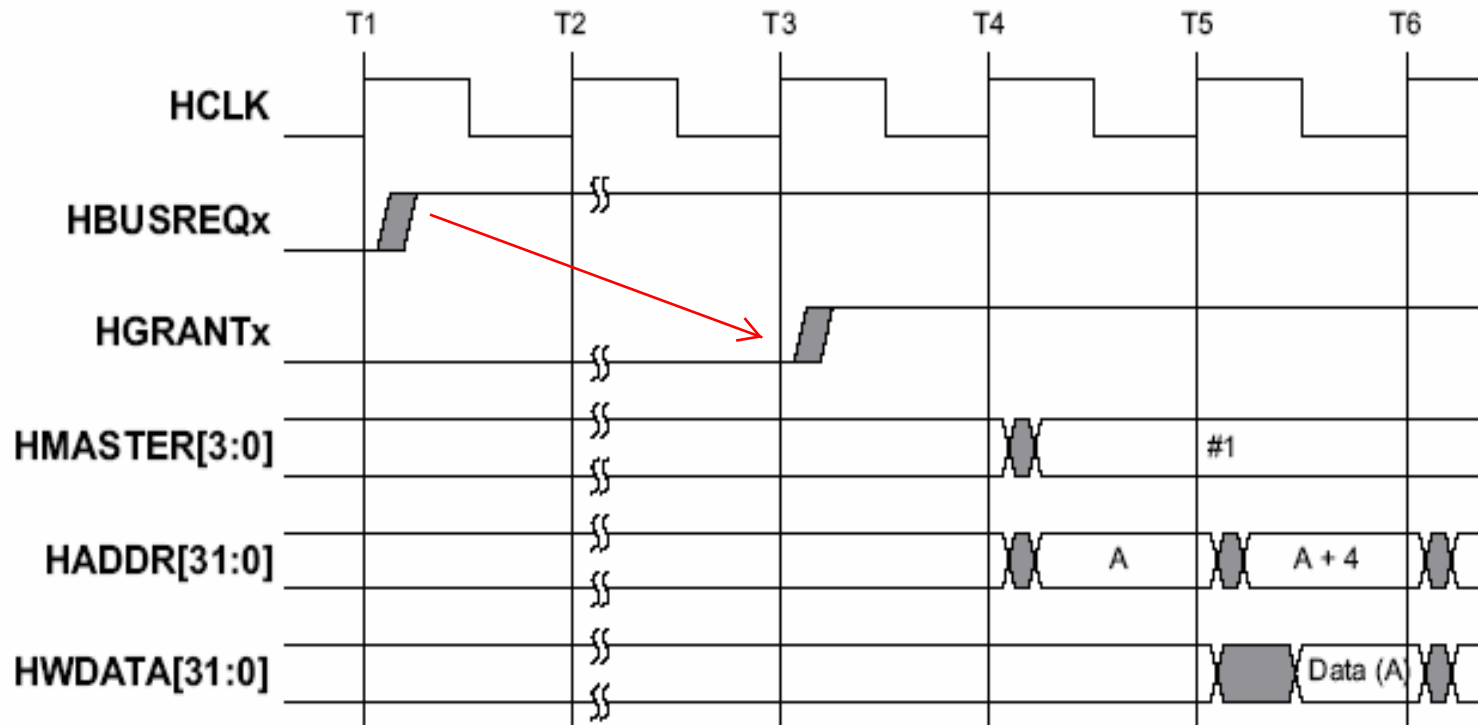
Slave transfer response : error



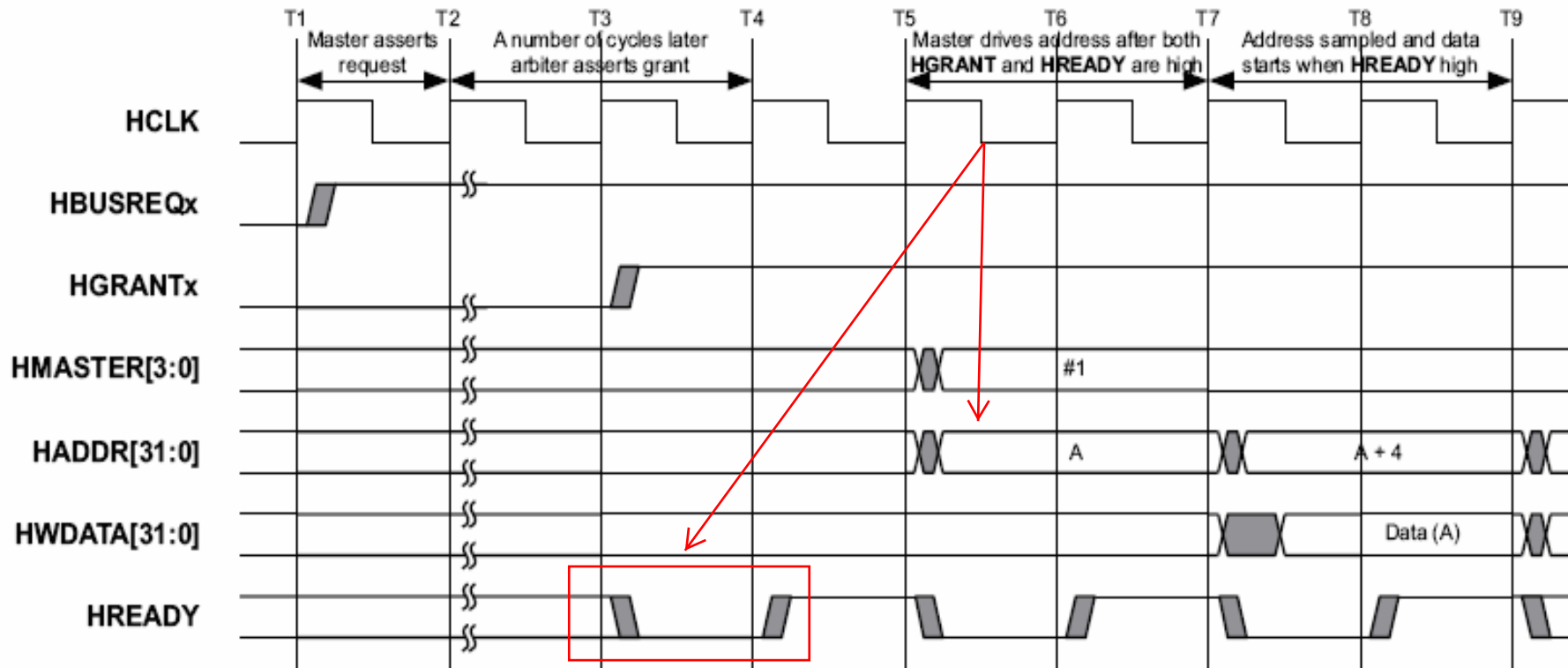
Slave transfer response : split



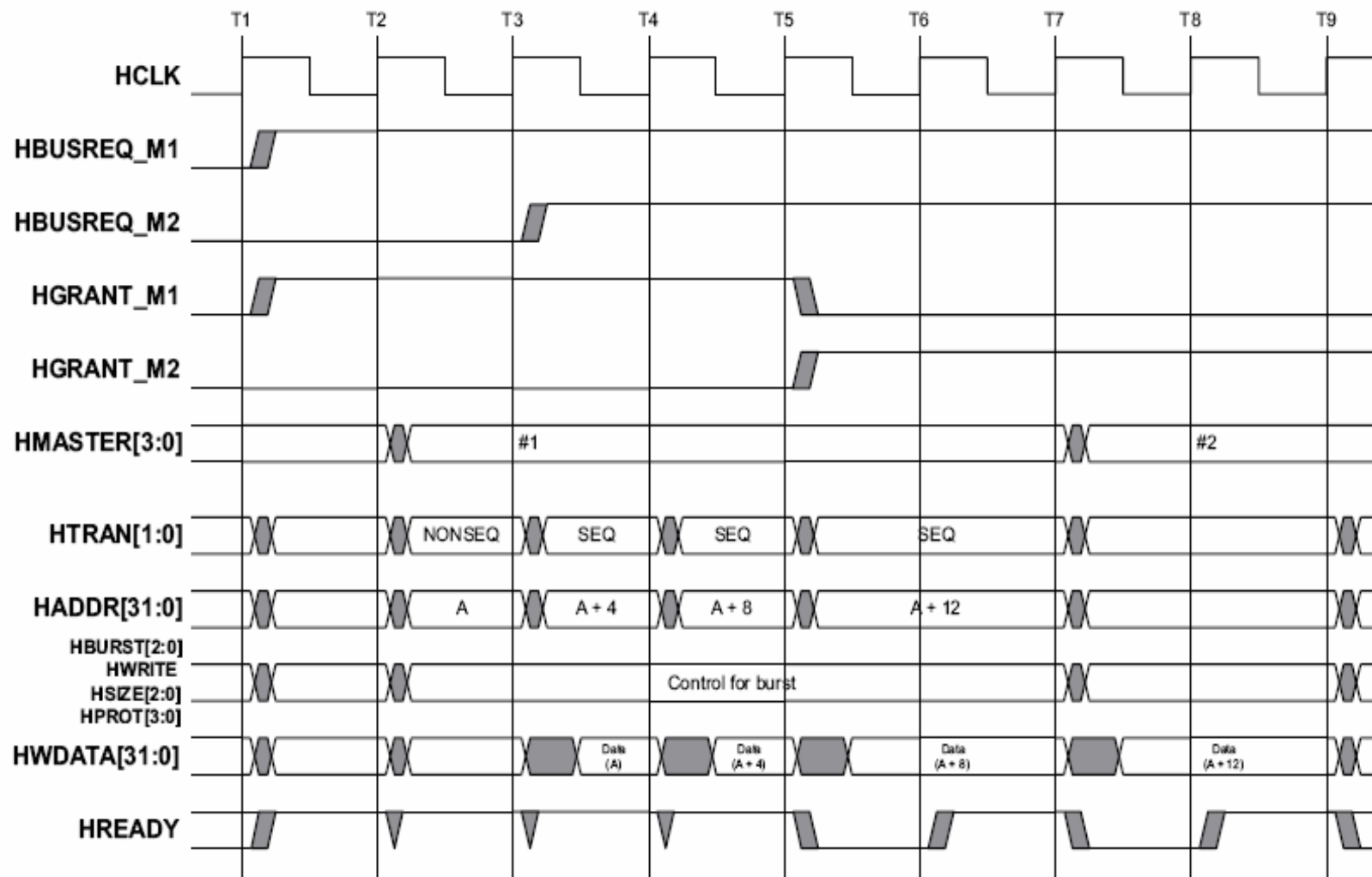
Arbitration : granting bus access



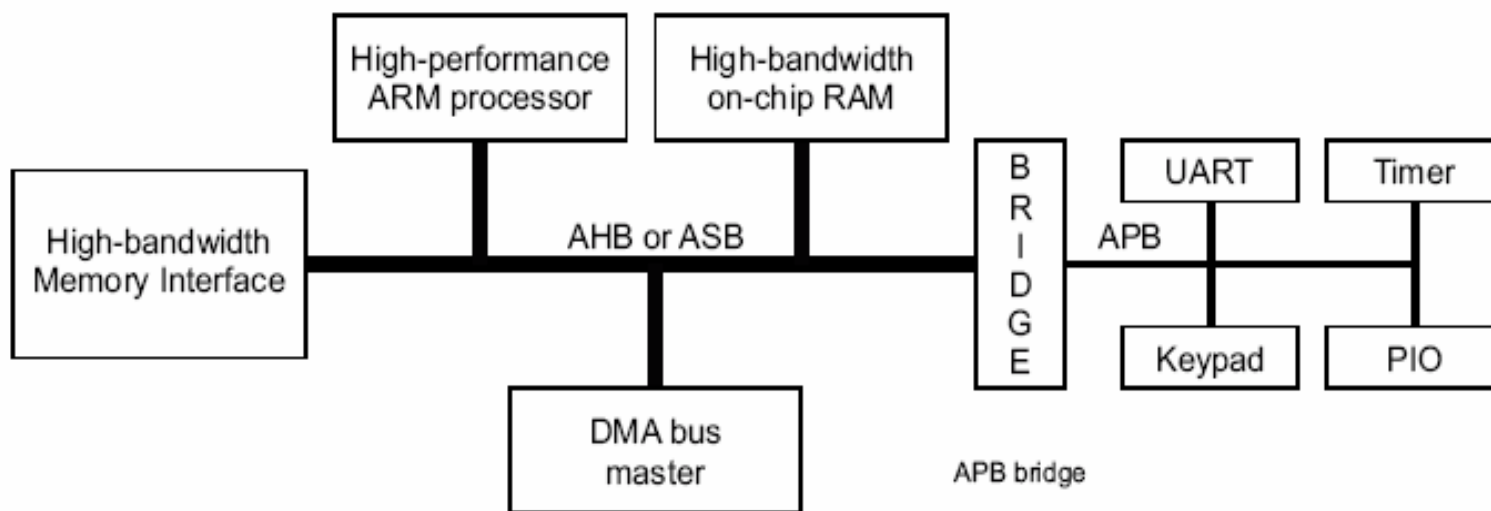
Arbitration : granting bus access with wait states



Arbitration : handover after burst



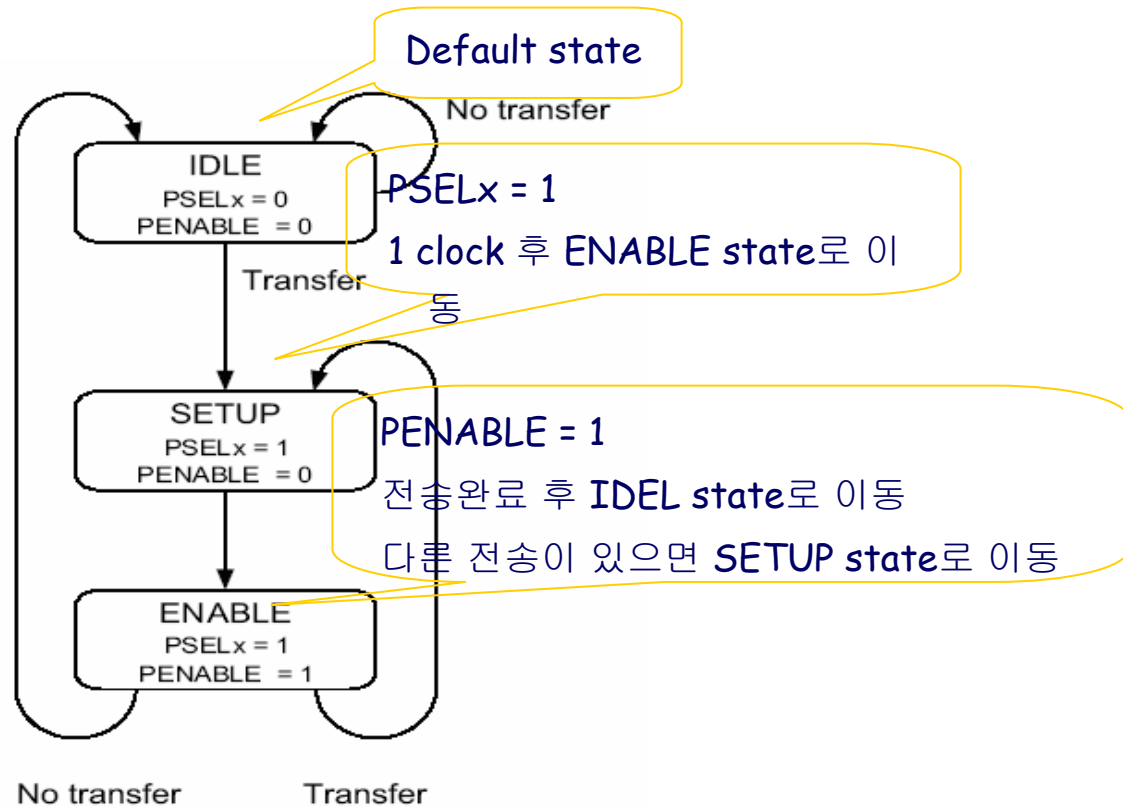
Typical AMBA AHB-based system



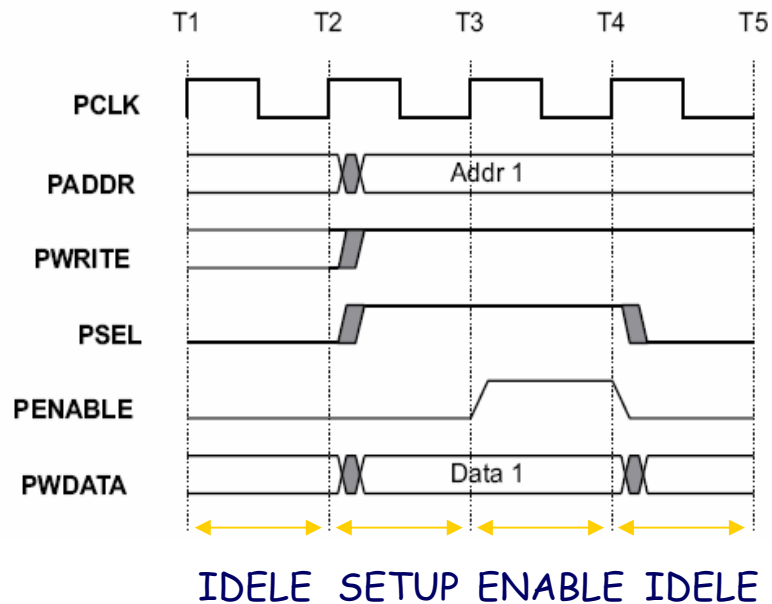
AMBA Advanced Peripheral Bus (APB)

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

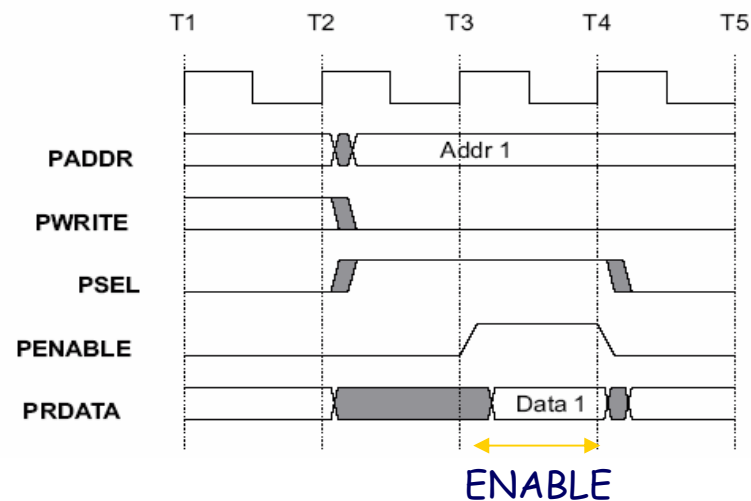
State diagram



APB Write/Read transfer



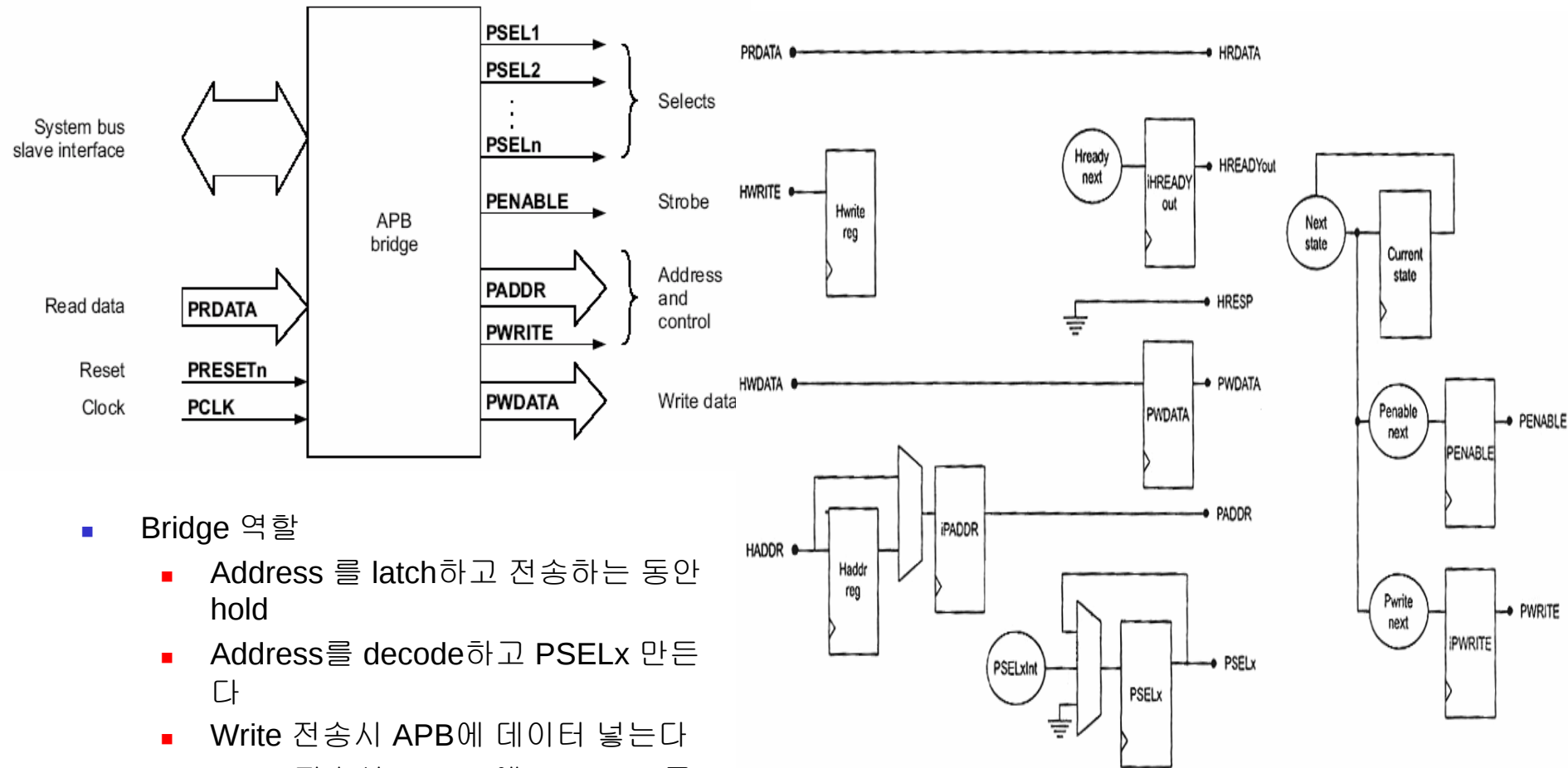
Write transfer



Read transfer

⚡ 전력소모를 줄이는 방법 : address sig와 write sig를 다음 ACCESS가 일어날때까지 바꾸지 않는다

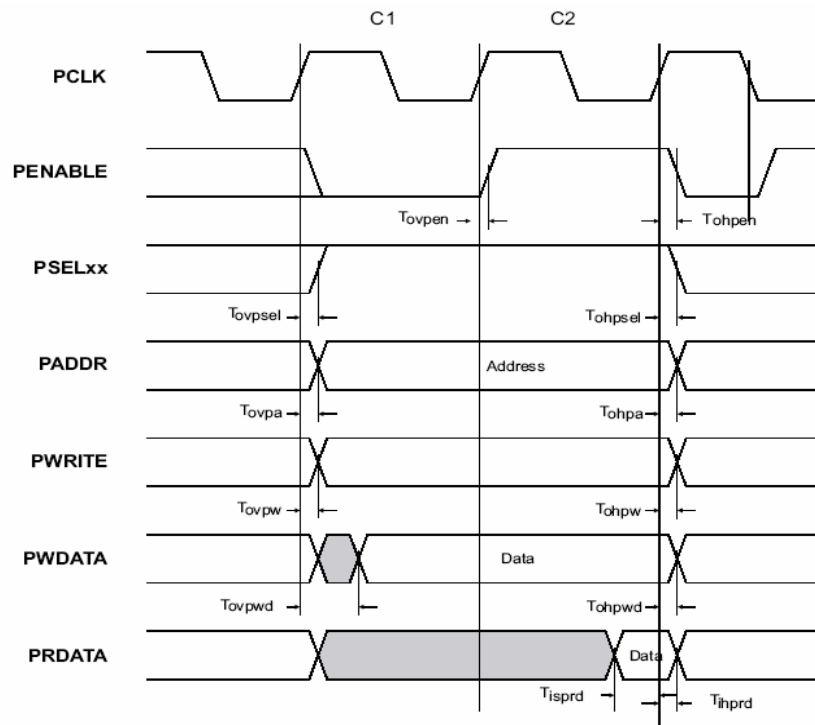
APB bridge



■ Bridge 역할

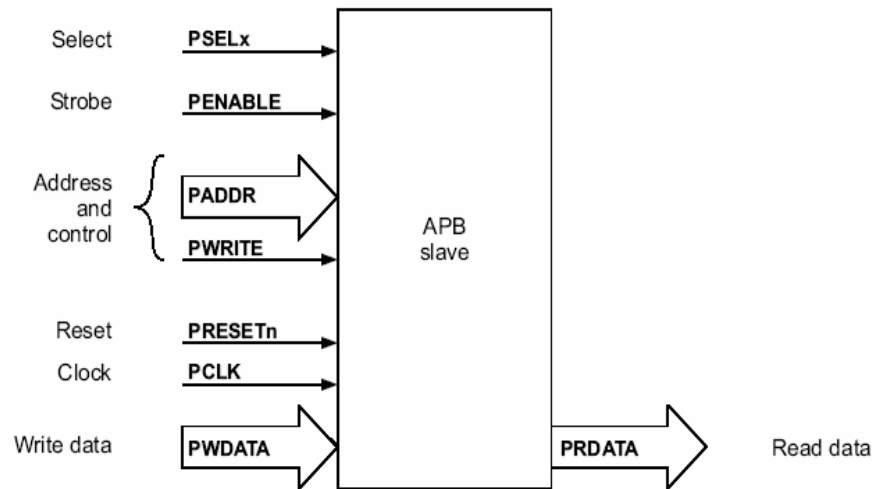
- Address 를 latch하고 전송하는 동안 hold
- Address를 decode하고 PSELx 만든다
- Write 전송시 APB에 데이터 넣는다
- Read 전송시 system에 APB data를 넣는다
- 전송시 PENABLE 를 만든다

APB bridge transfer(timing)

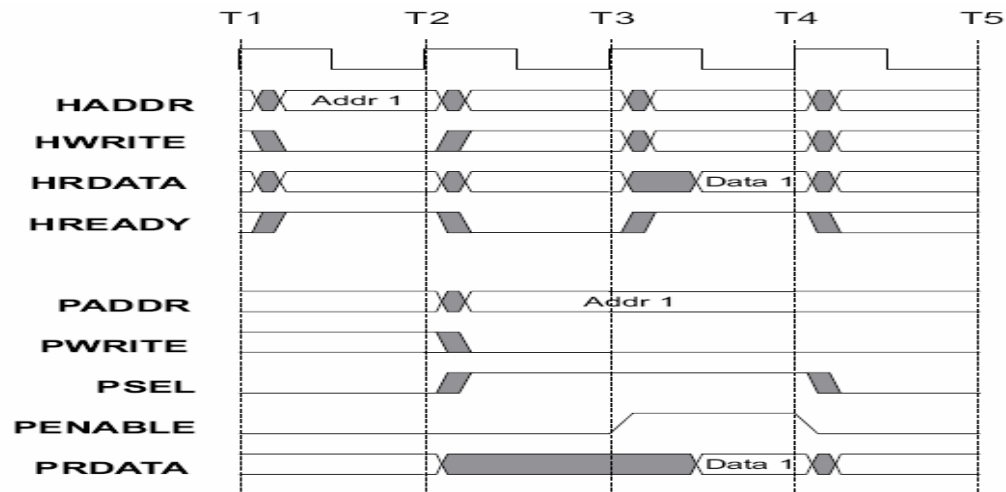


APB Slave

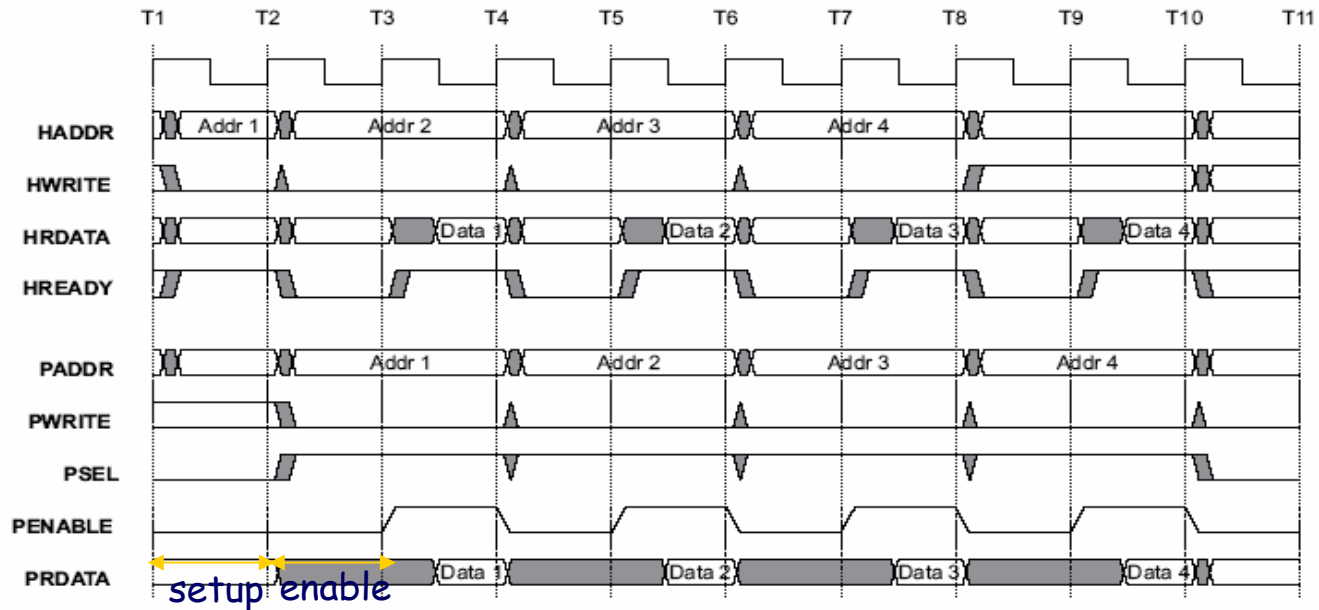
- Write transfer 을 위해 data를 latch
 - PSEL- high 일때 Pclk의 rising edge때
 - PSEL- high 일때 PENABLE의 rising edge때
- Write transfer
 - PSELx , PADDR, PWRITE 에 의해 결정된다
- READ transfer시 data를 data bus에 넣는방법
 - PWRITE- LOW, PSELx-HIGH, PENABLE- HIGH, ADDRESS



Interfacing APB to AHB(read transfer)



Interfacing APB to AHB(Burst of read transfer)



🌈 All read transfer require a single wait state