

Robei 集成电路实战



AHB简介





AHB简介

1

AHB介绍

2

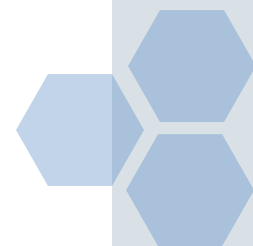
AHB模块结构

3

信号介绍与说明

4

初步设计





AHB简介

随着深亚微米工艺技术日益成熟，集成电路芯片的规模越来越大。数字IC从基于时序驱动的设计方法，发展到基于IP复用的设计方法，并在SOC设计中得到了广泛应用。在基于IP复用的SoC（System on Chip的缩写，称为系统级芯片，也有称片上系统）设计中，片上总线设计是最关键的问题。为此，业界出现了很多片上总线标准。其中，由ARM公司推出的AMBA片上总线受到了广大IP开发商和SoC系统集成者的青睐，已成为一种流行的工业标准片上结构。AMBA规范主要包括了AHB (Advanced High performance Bus)系统总线和APB (Advanced Peripheral Bus)外围总线。





AHB简介

AHB (Advanced High-performance Bus) 高级高性能总线

- 高速、高性能
- **Pipeline**操作
- 支持多个**Master**（最多**16**个）
- 支持**Burst**传输
- 支持**Split**传输
- 单时钟边沿操作





AHB简介

AHB 构成:

- Master （多个支持）
- Slave （多个支持）
- Infrastructure
 - Arbiter
 - M2S Multiplexor
 - S2M Multiplexor
 - Decoder
 - Dummy Slave
 - Dummy Master





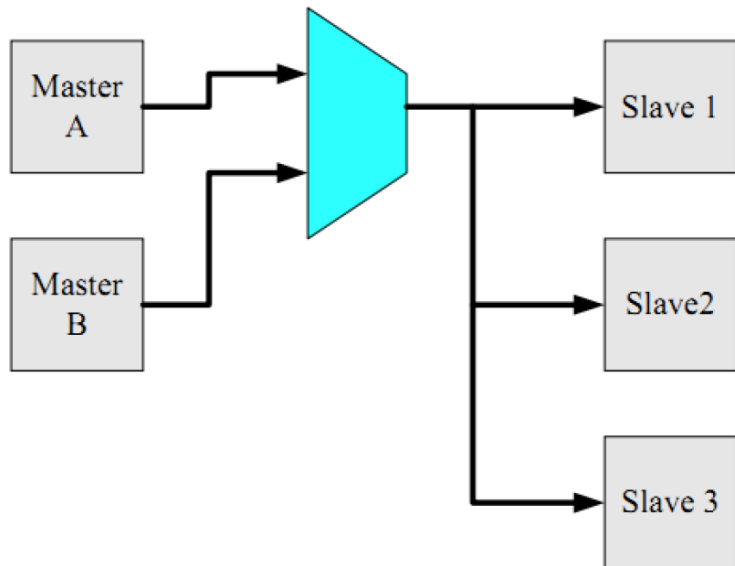
AHB简介

Arbiter: 在多Master设备的系统里面进行仲裁。

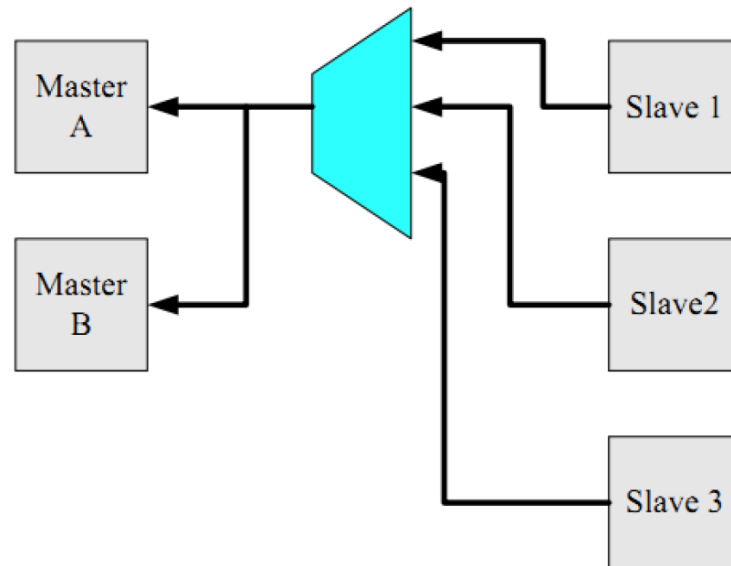
Decoder: 在多Slave设备的系统里面进行选择。

Multiplexor:

Master to Slaver Multiplexor



Slave to Master Multiplexor



允许某个主设备控制总线

Arbiter

发起一个请求给仲裁器

Master # 1

HADDR
HWDATA
HRDATA

Master # 2

HADDR
HWDATA
HRDATA

Master # 3

HADDR
HWDATA
HRDATA

Address and control mux

Write data mux

Read data mux

Decoder

HADDR

HWDATA

HRDATA

Slave # 1

HADDR

HWDATA

HRDATA

仅选中的从设备响应地址/控制信号

HADDR

HWDATA

HRDATA

Slave # 3

HADDR

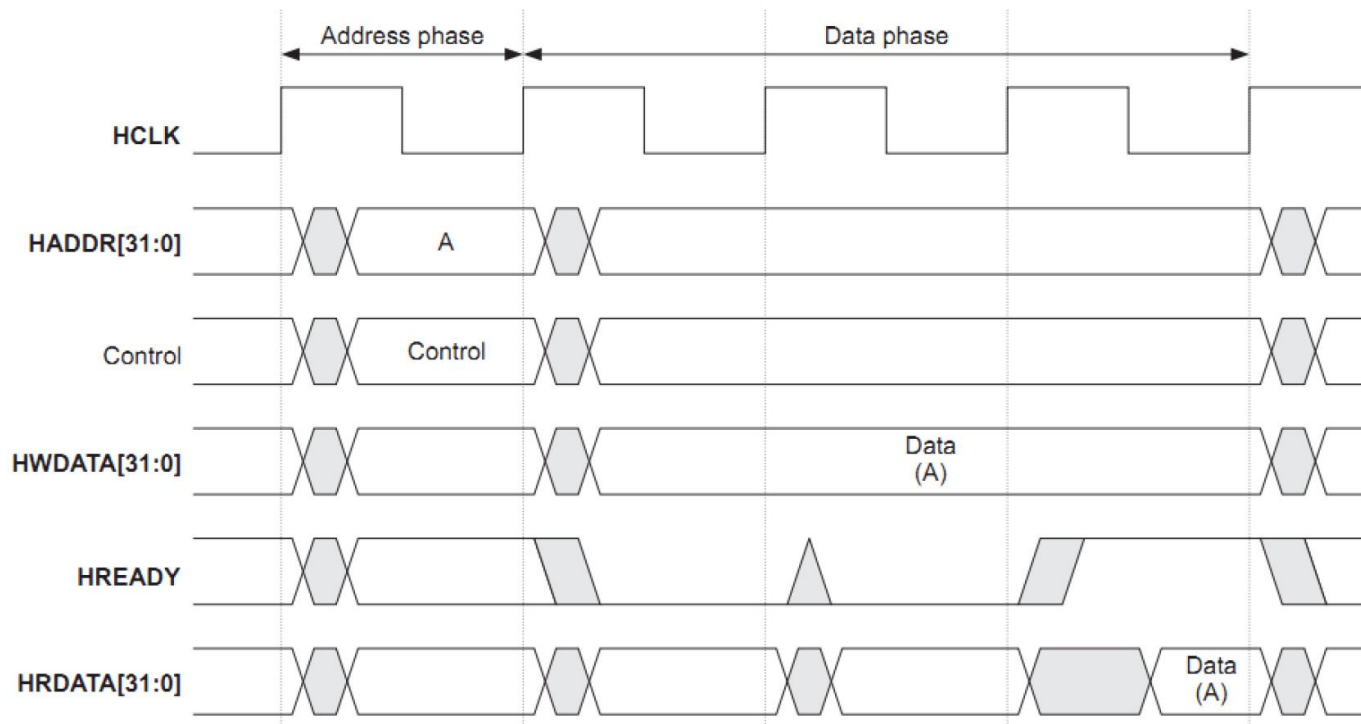
HWDATA

HRDATA

拉高HREADY信号，总线传输完成



AHB简介



A Address

A Data

B Address

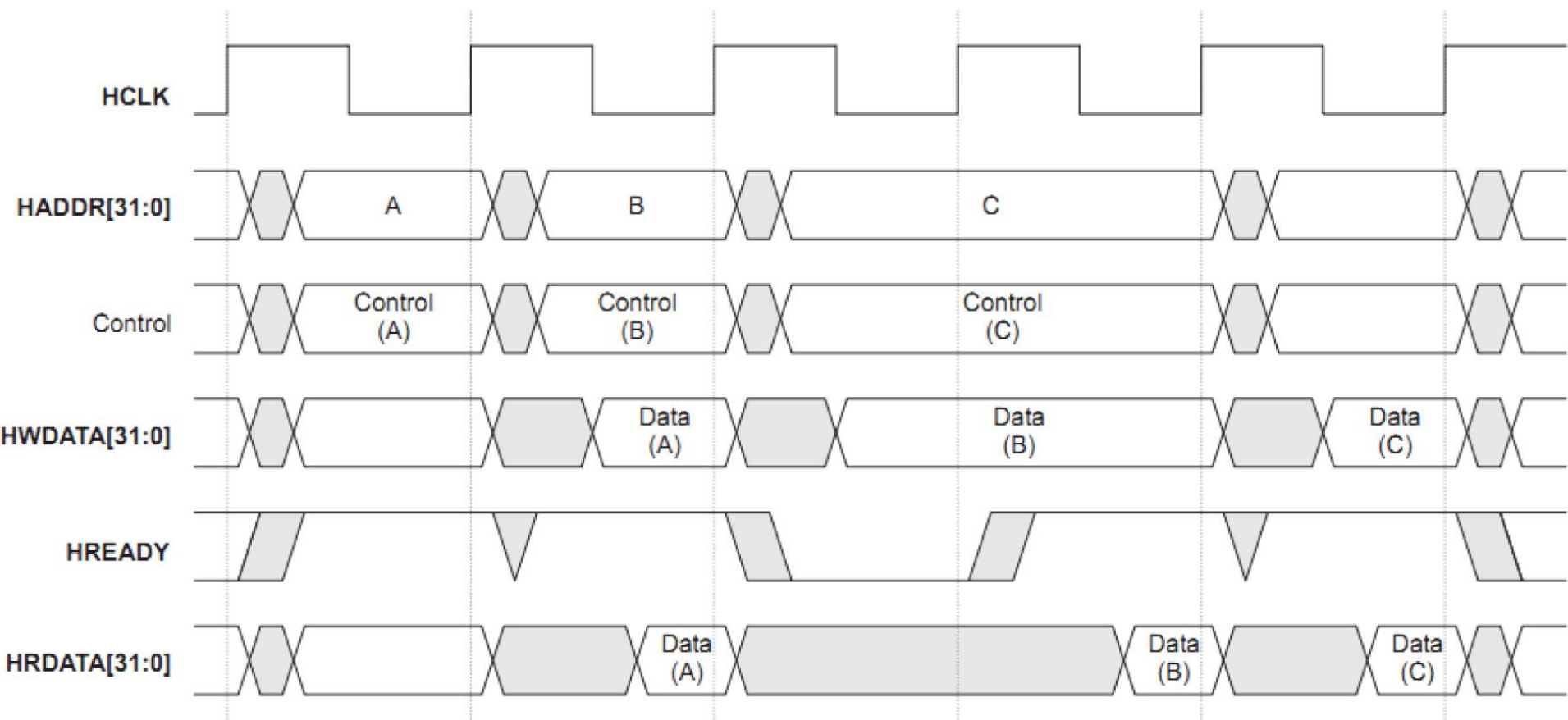
B Data

C Address

C Data



AHB简介



Name	Source	Description
HCLK	Clock source	Bus Clock ◦ All signal timings are related to the rising edge of HCLK ◦
HRESETn	Reset controller	active LOW ◦ reset whole system ◦
HADDR[31:0]	Master	32-bit system bus ◦
HTRANS[1:0]	Master	current transfer type ◦
HWRITE	Master	HIGH : write transfer ◦ LOW : read transfer ◦
HSIZE[2:0]	Master	the size of the transfer ◦ control signal
HBURST[2:0]	Master	Indicates if the transfer forms part of a burst ◦
HPROT[3:0]	Master	Implement some level of protection ◦
HWDATA[31:0]	Master	write data bus ◦
HSELx	Decoder	slave select signal ◦
HRDATA[31:0]	Slave	read data bus ◦
HREADY	Slave	High : transfer done ◦ LOW : extending transfer ◦
HRESP[1:0]	Slave	transfer response ◦
HBUSREQx	Master	bus request ◦ To Arbiter, request signal
HLOCKx	Master	Locked transfer ◦
HGRANTx	Arbiter	Bus grant signal ◦
HMASTER[3:0]	Arbiter	Indicate granted master number ◦
HMASTLOCK	Arbiter	Locked sequence ◦
HSPLITx[15:0]	Slave	Split completion request ◦



AHB简介

AHB中的五个控制信号:

HTRANS[1:0] HWRITE HSIZE[2:0] HBURST[2:0] HPROT[3:0]

HTRANS[1:0]

idle: 空闲传输，用于Master没有数据需要传输时。

busy: 主设备占用总线，但是数据还没有准备好。

nonseq: 非连续传输，当前的地址和控制信号与前一次无关。

seq: 连续传输，当前的地址和控制信号与前一次相关。





AHB简介

AHB中的五个控制信号:

HTRANS[1:0] HWRITE HSIZE[2:0] HBURST[2:0] HPROT[3:0]

HWRITE

- 1: Master将数据通过write data bus (HWDATA[31:0])发往Slave。
- 0: Slave将数据通过read data bus (HRDATA[31:0]) 发往Master。



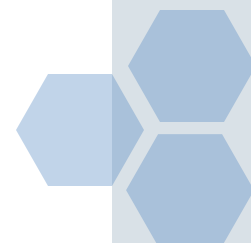
AHB简介

AHB中的五个控制信号:

HTRANS[1:0] HWRITE HSIZE[2:0] HBURST[2:0] HPROT[3:0]

HSIZE[2:0]

HSIZE[2]	HSIZE[1]	HSIZE[0]	Size	Description
0	0	0	8 bits	Byte
0	0	1	16 bits	Halfword
0	1	0	32 bits	Word
0	1	1	64 bits	-
1	0	0	128 bits	4-word line
1	0	1	256 bits	8-word line
1	1	0	512 bits	-
1	1	1	1024 bits	-





AHB简介

AHB中的五个控制信号：

HTRANS[1:0] HWRITE HSIZE[2:0] HBURST[2:0] HPROT[3:0]

HBURST[2:0]

HBURST[2:0]	Type	Description
000	SINGLE	Single transfer
001	INCR	Incrementing burst of unspecified length

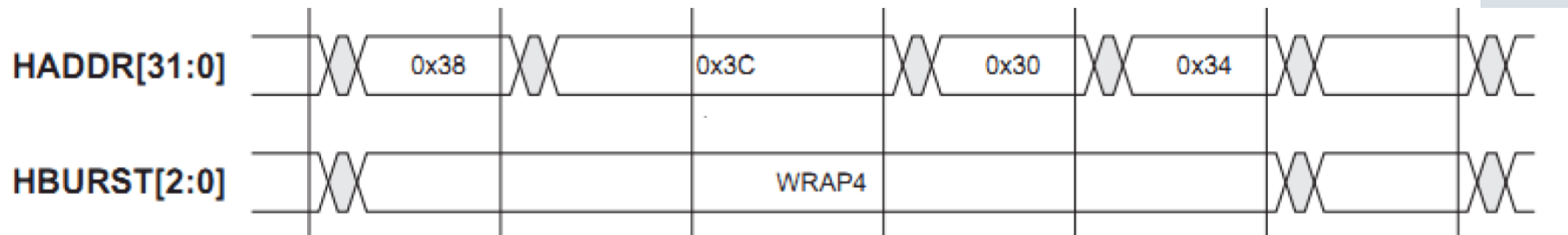
突发传输的结束点并不能预测到，所以还没有有效的方法设计仲裁器以保证在将总线授予另一个主设备时突发传输已经结束。

010	WRAP4	4-beat wrapping burst
011	INCR4	4-beat incrementing burst
100	WRAP8	8-beat wrapping burst
101	INCR8	8-beat incrementing burst
110	WRAP16	16-beat wrapping burst
111	INCR16	16-beat incrementing burst

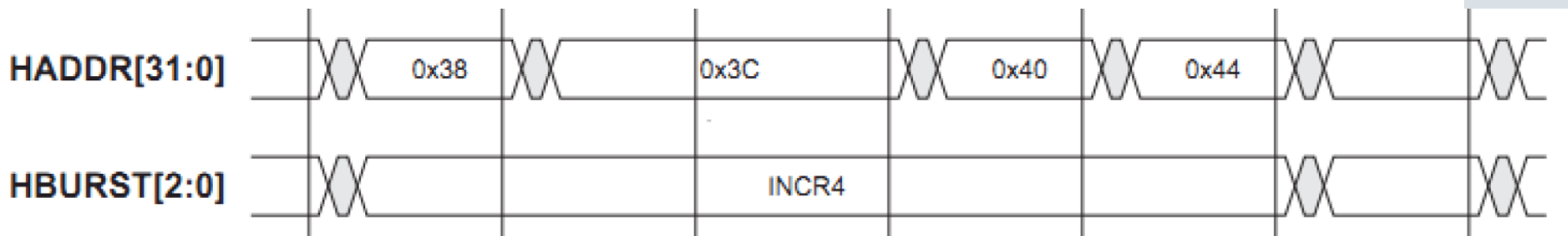


AHB简介

4-beat wrapping:



4-beat increasing:





AHB简介

AHB中的五个控制信号:

HTRANS[1:0] HWRITE HSIZE[2:0] HBURST[2:0] HPROT[3:0]

HPROT[3:0]

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	-	-	Not bufferable
-	1	-	-	Bufferable
0	-	-	-	Not cacheable
1	-	-	-	Cacheable

提供额外的保护信息；若是非必须，可将该信号常置为4'b0001。



AHB简介

AHB中Slave返回Master的信号：
HREADY, HRESP[1:0]

HREADY：在数据传输结束时置高电平；

HRESP[1:0]：

- OKAY：传输成功完成；
- ERROR：传输失败或者出错；
- RETRY：当前传输未完成，请求Master重试；
- SPLIT：当前传输未完成，Arbiter可以释放总线给其他Master使用。





AHB简介

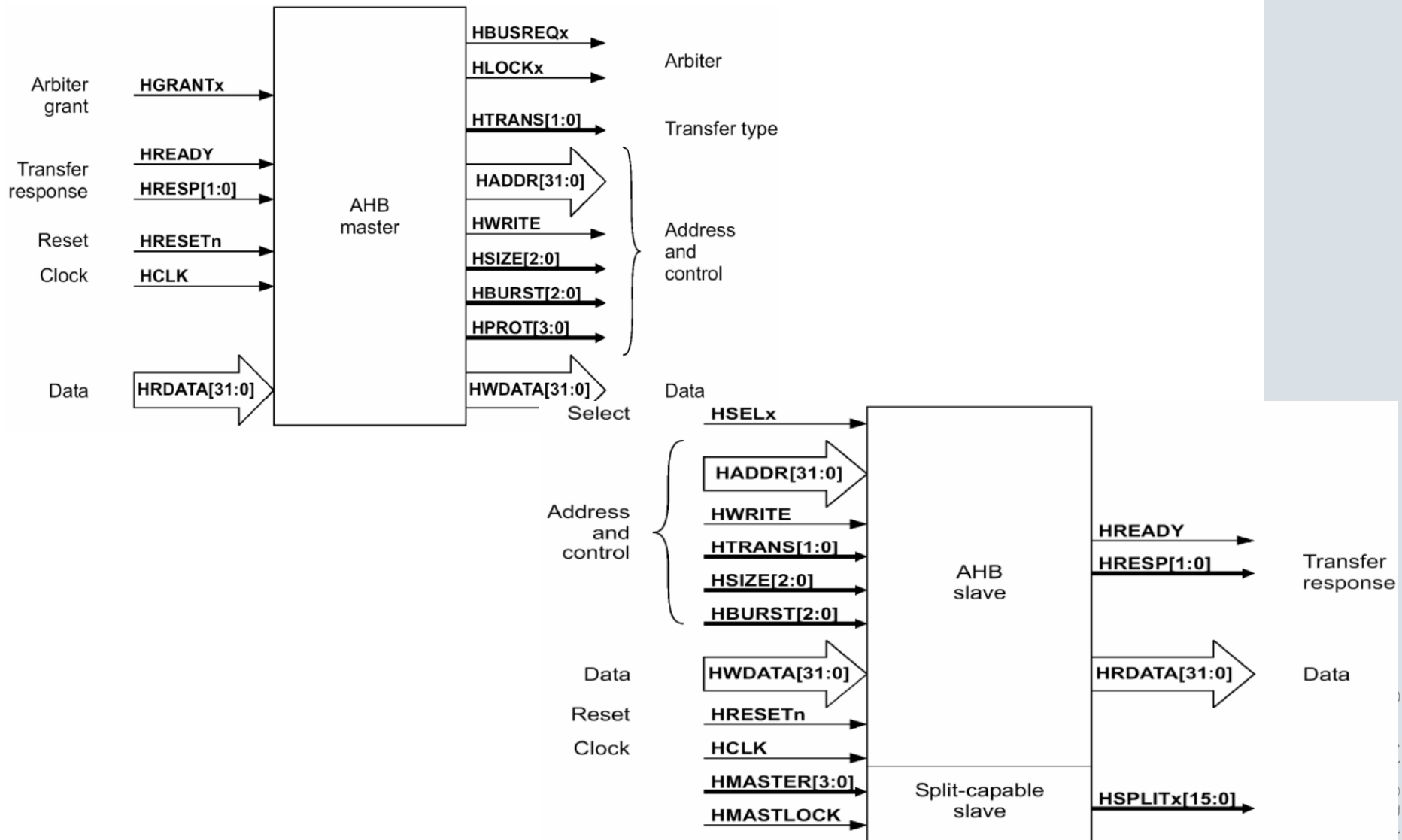
AHB中的仲裁信号:

HBUSREQx, HLOCKx, HGRANTx

- AHB为支持Multi-Master的Bus system, 但同一时间只允许一个master接入总线, 因此需要Arbiter进行Arbitration。
- 当master想要接入总线时, master将HBUSREQ drive high。
- 同一时间可能有多个master请求接入总线, 因此需要Arbiter进行Arbitration。
- Arbiter 在HCLK上升沿采样HBUSREQ, 并将最高优先级master的HGRANT signal drive HIGH, 表示该master可以接入总线。
- 若想要master进行连续传输而不被中断, 则在master申请总线接入时将HLOCK drive HIGH, 告知arbiter transfer不可被中断。
- AHB没有规定的优先级算法, 由设计者根据需求自定。

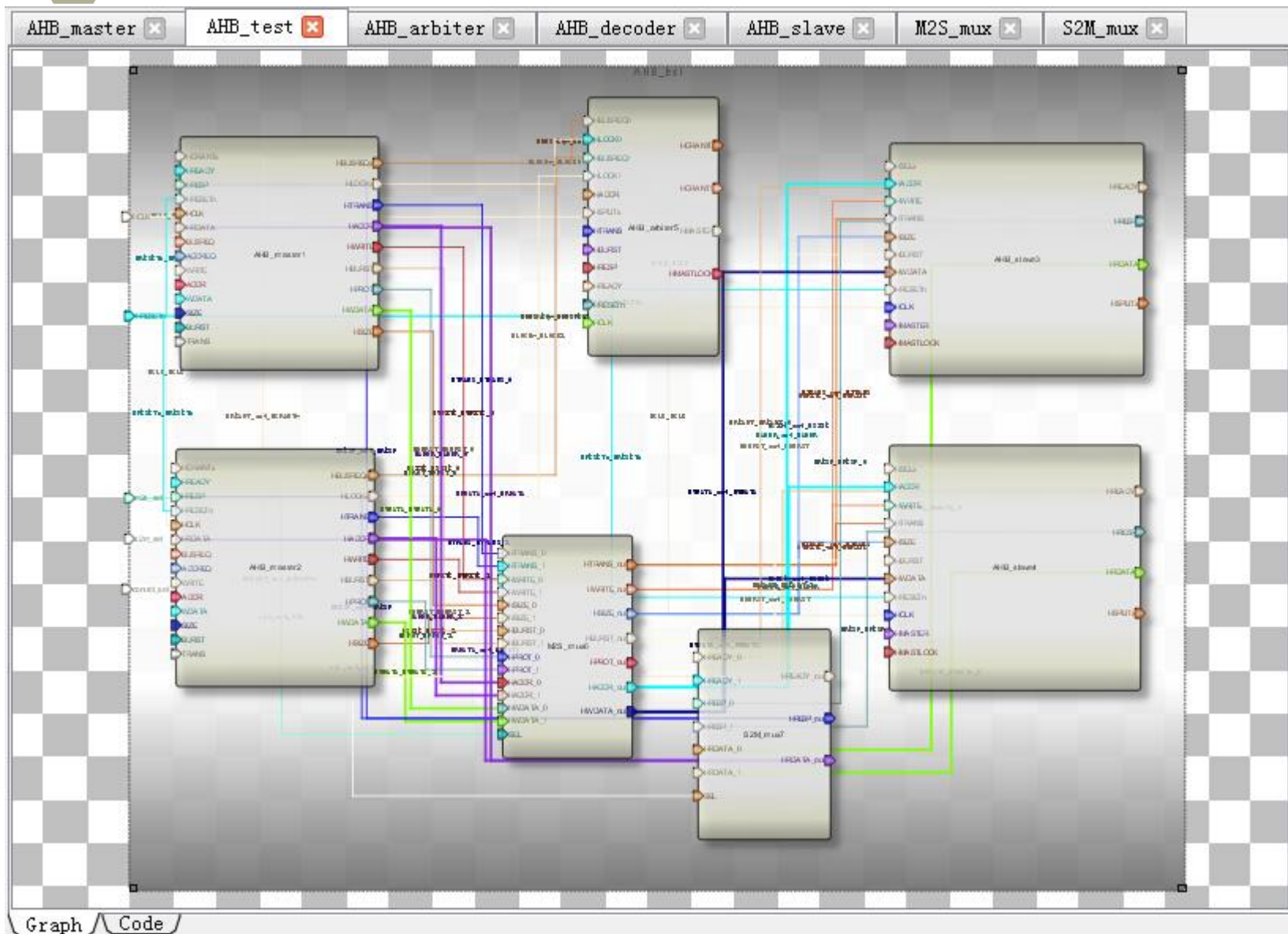


AHB简介





AHB简介





AHB简介

```
AHB_master x AHB_test x AHB_arbiter x AHB_decod
125 // HPROT[3:0]
126 //HPROT[0]: 0:opcode fetch      1:data access
127 //HPROT[1]: 0:user access      1:privileged access
128 //HPROT[2]: 0:not bufferable    1:bufferable
129 //HPROT[3]: 0:not cacheable     1:cacheable
130 //*****
131 parameter okay=2'b00;
132 parameter error=2'b01;
133 parameter retry=2'b10;
134 parameter split=2'b11;
135
136 reg bus_reg;
137 reg addr_reg;
138 reg new_hready;
139 reg old_hready;
140
141 reg [31:0] RDATA;
142 reg [31:0] h_addr;
143
144 always @ (posedge HCLK)
145 begin
146     if (!HRESETn)
147     begin
148         HLOCKx<=0;
149         HBUSREQx<=0;
150         bus_reg<=0;
151     end
152     else if (!bus_reg)
153     begin
154         if (BUSREQ)
155         begin
156             HBUSREQx<=1;
157             bus_reg<=1;
158         end
159         else if (!BUSREQ)
160             HLOCKx<=0;
```

```
AHB_master x AHB_slave x AHB_test x AHB_arbite
94 wire rd_en;
95
96 assign wr_en=(HSELx)&(HTRANS[1])&(HWRITE);
97 assign rd_en=(HSELx)&(HTRANS[1])&(!HWRITE);
98
99 always @ (posedge HCLK)
100 begin
101     if (!HRESETn)
102     begin
103         HREADY<=0;
104         HRESP<=2'b00;
105         HSPLITx<=16'h0000;
106     end
107     else if (counter>=totalbits)
108     begin
109         HREADY<=1;
110         HRESP<=HRESP_okay;
111         HSPLITx<=HSPLITx;
112     end
113 end
114
115 always @ (posedge HCLK)
116 begin
117     if (!HRESETn)
118         counter<=0;
119     else
120     begin
121         case (HBURST)
122             HBURST_single: counter<=totalbits;
123             HBURST_incr: counter<=totalbits;
124             HBURST_wrap4: counter<=counter+4;
125             HBURST_incr4: counter<=counter+4;
126             HBURST_wrap8: counter<=counter+2;
127             HBURST_incr8: counter<=counter+2;
128             HBURST_wrap16: counter<=counter+1;
129             HBURST_incr16: counter<=counter+1;
```

www.robei.com



Thank You!