

## 一、NetBus 是什么？

NetBus 是一种 SOC 片内总线，类似 AXI、AHB 总线。

## 二、NetBus 和其他总线的区别是什么？

NetBus 没有主从区分，支持 32 个设备端点，支持优先级，支持实时非实时进行划分，支持按区域进行划分，能够搭建不同等级的网络，能够最大程度发挥系统的实时性和吞吐率。

## 三、NetBus 的特点是什么？

NetBus 可以类似网络那样“组网”，而且不同的网络可以通过网关节点进行不同网络的连接，可以把网络访问的冲突隔离开。

## 四、NetBus 的数据

每个 NetBus 的设备端点信号分为以下几种：

input	WCLK,
input [DATA_WIDTH*9+13:0]	WDATA,
input	WVALID,
output	WREADY,
input	RCLK,
output [DATA_WIDTH*9+13:0]	RDATA,
output	RVALID,
input	RREADY,

WCLK : 发送时钟。

WDATA : 发送数据，宽度可以调节，DATA\_WIDTH 为数据的字节宽度，如果等于 4，数据是 32 位。

WVALID : 发送数据有效，类似 AXI 的有效信号。

WREADY : 发送准备有效、类似 AXI 的 READY 信号。

RCLK : 接收时钟。

RDATA : 接收数据。宽度可以调节，DATA\_WIDTH 为数据的字节宽度，如果等于 4，数据是 32 位。

RVALID : 接受数据有效，类似 AXI 的有效信号。

RREADY : 接收准备有效，类似 AXI 的 READY 信号。

其中 WDATA 和 RDATA 数据又分为以下结构：

input [DATA_WIDTH*8-1:0]	DATA_X,
input [DATA_WIDTH-1:0]	STRB,
input [1:0]	CMD,
input [4:0]	DID,
input [4:0]	SID,
input	FIRST,
input	LAST,

DATA\_X : 数据位。

STRB : 写操作有效指示位，在 FIRST 位为高位时为帧后续的长度。

CMD : 命令, 10 为读命令, 11 为读回应, 00 为写命令, 01 为写回应。  
DID : 为目的地址。  
SID : 为源地址。  
FIRST : 指出是帧的第一拍。  
LAST : 指出是帧的最后一拍。

## 五、NetBus 的搭建

NetBus 主要是由 multiplexer 组成, 在 src 目录下已经实现

NetBusMUX2、NetBusMUX3、NetBusMUX4、NetBusMUX5、NetBusMUX6、NetBusMUX7、NetBusMUX8。  
。拿 NetBusMUX2 做介绍, 其他的类似。

```
NetBusMUX2#(  
    parameter DATA_WIDTH = 4,  
    parameter PORT_ROUTE = 32'h00000000,  
    parameter FIFO_DEPTH = 4,  
    parameter SYNC_STAGES = 2,  
    parameter REAL_TIME_PORT0 = 0,  
    parameter REAL_TIME_PORT1 = 0  
)(  
    input                                RESETn,  
  
    input                                WCLK,  
    input [DATA_WIDTH*9+13:0] WDATA,  
    input                                WVALID,  
    output                                WREADY,  
  
    input                                RCLK,  
    output [DATA_WIDTH*9+13:0] RDATA,  
    output                                RVALID,  
    input                                RREADY,  
  
    //+++++ 优先级最高  
    output                                WCLK0,  
    output [DATA_WIDTH*9+13:0] WDATA0,  
    output                                WVALID0,  
    input                                WREADY0,  
  
    input                                RCLK0,  
    input [DATA_WIDTH*9+13:0] RDATA0,  
    input                                RVALID0,  
    output                                RREADY0,  
  
    //+++++ 优先级次高  
    output                                WCLK1,  
    output [DATA_WIDTH*9+13:0] WDATA1,  
    output                                WVALID1,  
    input                                WREADY1,  
  
    input                                RCLK1,  
    input [DATA_WIDTH*9+13:0] RDATA1,  
    input                                RVALID1,  
    output                                RREADY1  
);
```

DATA\_WIDTH : 数据的字节数宽度, 如果为 4 是 32 位宽。

PORT\_ROUTE : 本端口需要接收的地址号, 一共是 32 比特, 假如 ID 为 4, 那么 bit4 置 1。

FIFO\_DEPTH : 接收的 fifo 深度, 4 的深度为 2 的四次方。

SYNC\_STAGES: 跨时钟异步的拍数, 800M 以下一般为 2, 以上为 3。

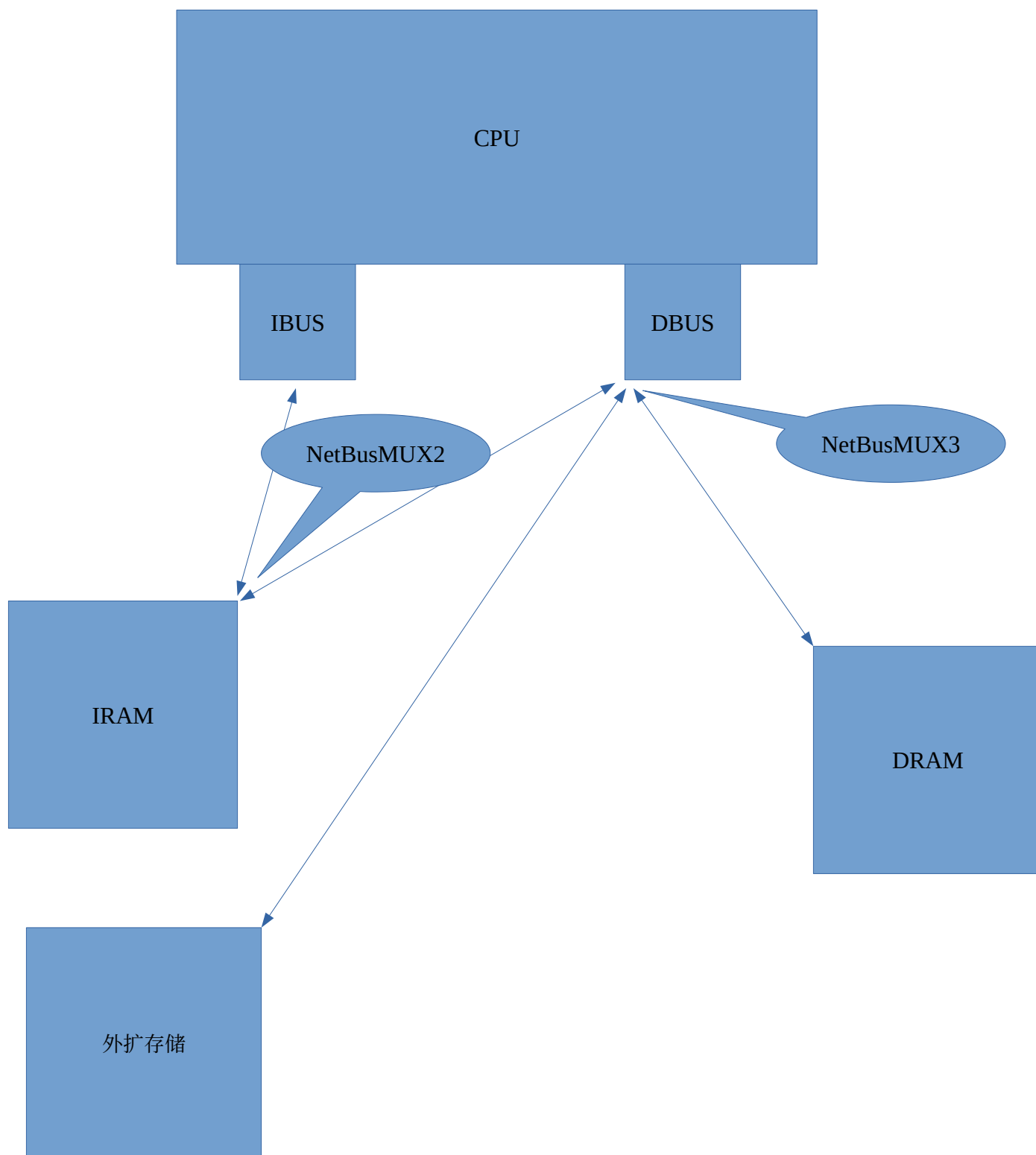
REAL\_TIME\_PORT0: 置 1 表示 port 0 为实时性端口。

REAL\_TIME\_PORT1: 置 1 表示 port 1 为实时性端口。

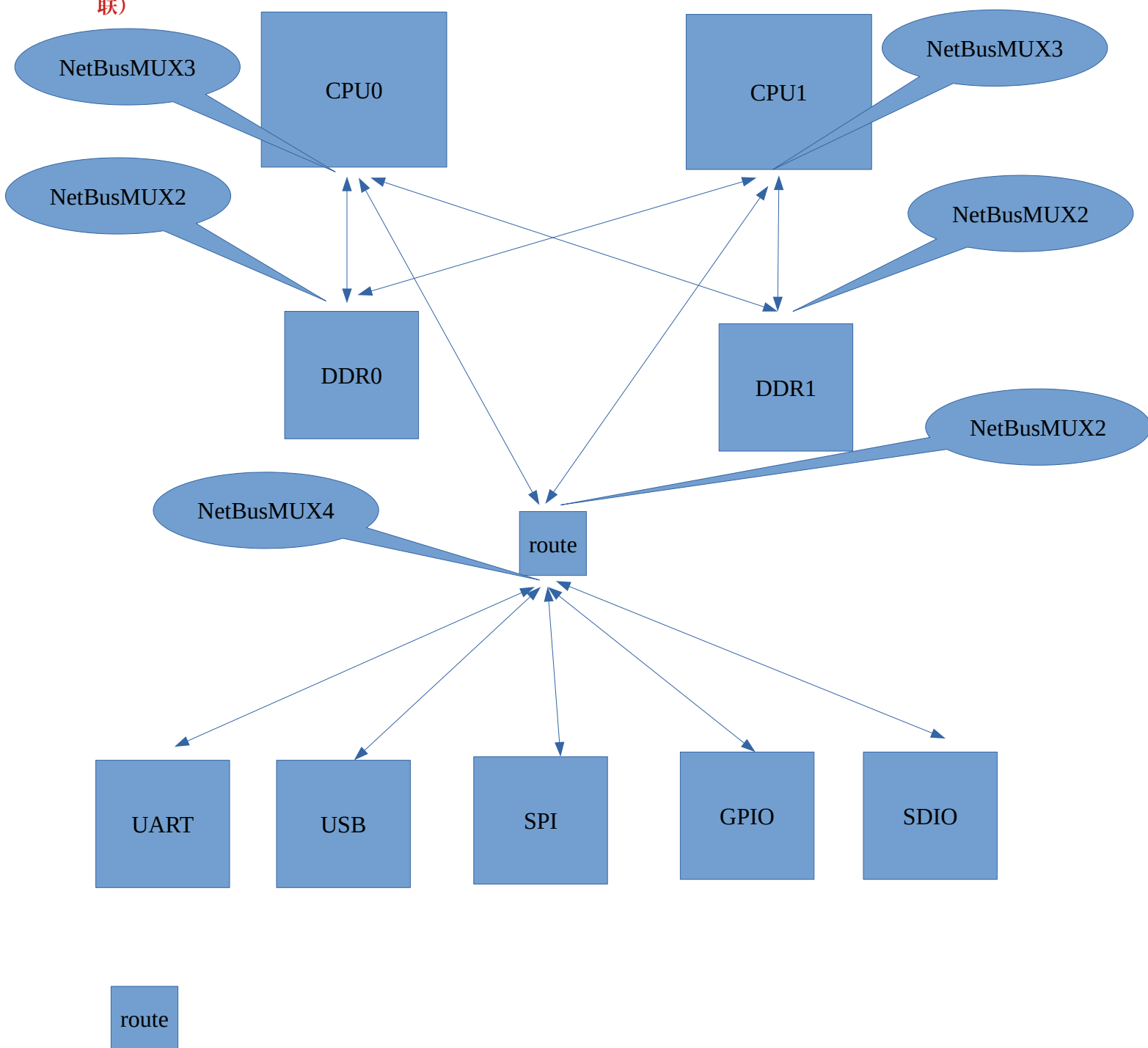
RESETn : 异步复位。

## 六、NetBus 的搭建示例 1

一个哈佛结构的计算机系统，**DBUS** 可以访问 **IRAM** 是为了有时需要指令的搬移。



七、NetBus的搭建示例 2（一个非对称的双 CPU 系统，通过一个 route 把高速和低速区分开，同时能保证互联）



**route** 就是把 NetBusMUX2 和 NetBusMUX4 的接收和发送交叉互联。

八、优先级的划分

比如 NetBusMUX2 中 端口划分为 0 的 比 1 的优先级高，在数据同时到达的情况下优先处理 0 的端口

## 九、实时特性

**REAL\_TIME\_PORT0** 表示 0 端口是否要求实时性，实时性的端口只要有数据就会抢占权，而非实时的需要接收到 **LAST** 位有效后才会有抢占的机会，速度低的设备不要打开实时性，速度高的设备打开实时性。

## 十、路由设置

就是把可能会传输的数据 ID 号设置在 MUX 的 **PORT\_ROUTE** 里，比如上图 **route** 模块里，CPU0、CPU1、UART、USB、SPI、GPIO、SDIO 的 ID 号设置在 NetBusMUX2、NetBusMUX3、的 **PORT\_ROUTE** 属性里，数据帧就会被正常传递。