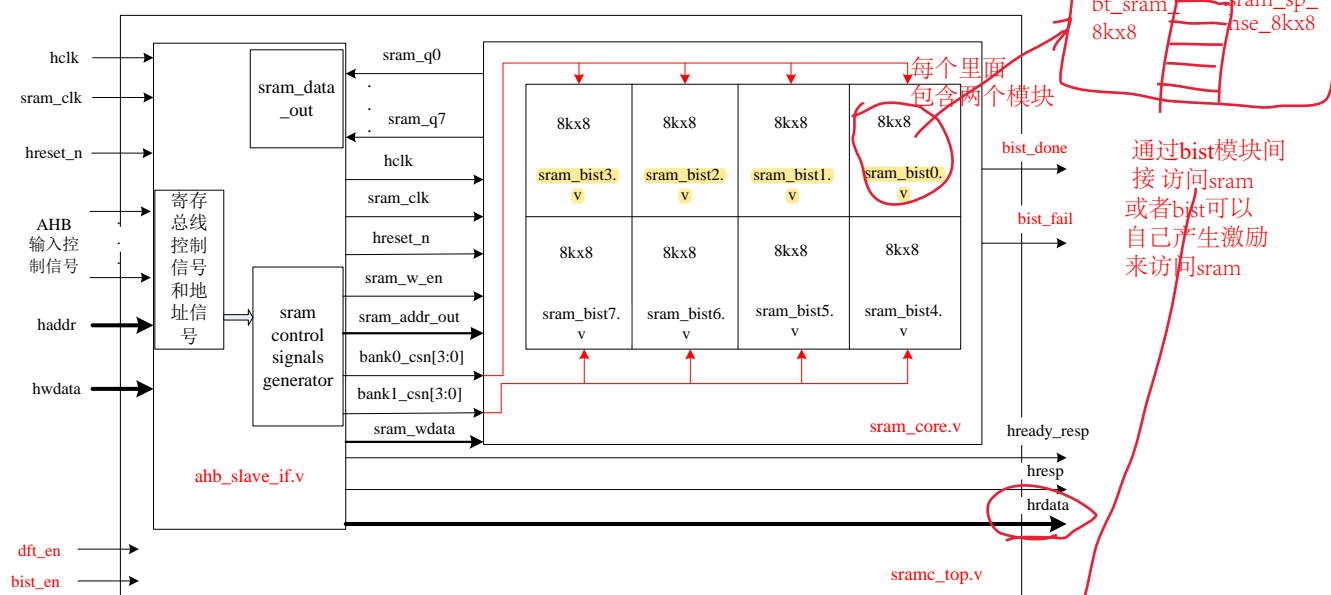


# 基于 AHB 总线的 SRAM 控制器的微架构设计

## SRAM 控制器架构框图

下面给出的是 SRAM 控制器详细的设计框图，如图 1 所示：



拆成8个小块是为了实现低功耗的设计

图 1 SRAM 控制器模块框图

## SRAM 控制器各模块详细描述

### 模块框图

- (1) AHB slave interface 模块：如图 2 所示为 SRAM 控制器的 AHB slave interface，其产生读写控制信号（sram\_r\_en、sram\_w\_en）和 SRAM 控制器片选所需的输入信号（bank0\_csn、bank1\_csn）。包括：AHB 输入控制信号和地址信号寄存模块、SRAM 读写和片选产生模块、SRAM 输出数据接收模块。

march C 算法，  
p1:全写0，  
p2:减地址；读0，比较，写1  
p3:增地址；读1，比较，写0，读0，比较，写1  
p4:减地址；读1，比较，写0，  
p5:增地址；读0，比较，写1，读1，比较，写0  
p6:减地址；读0，比较

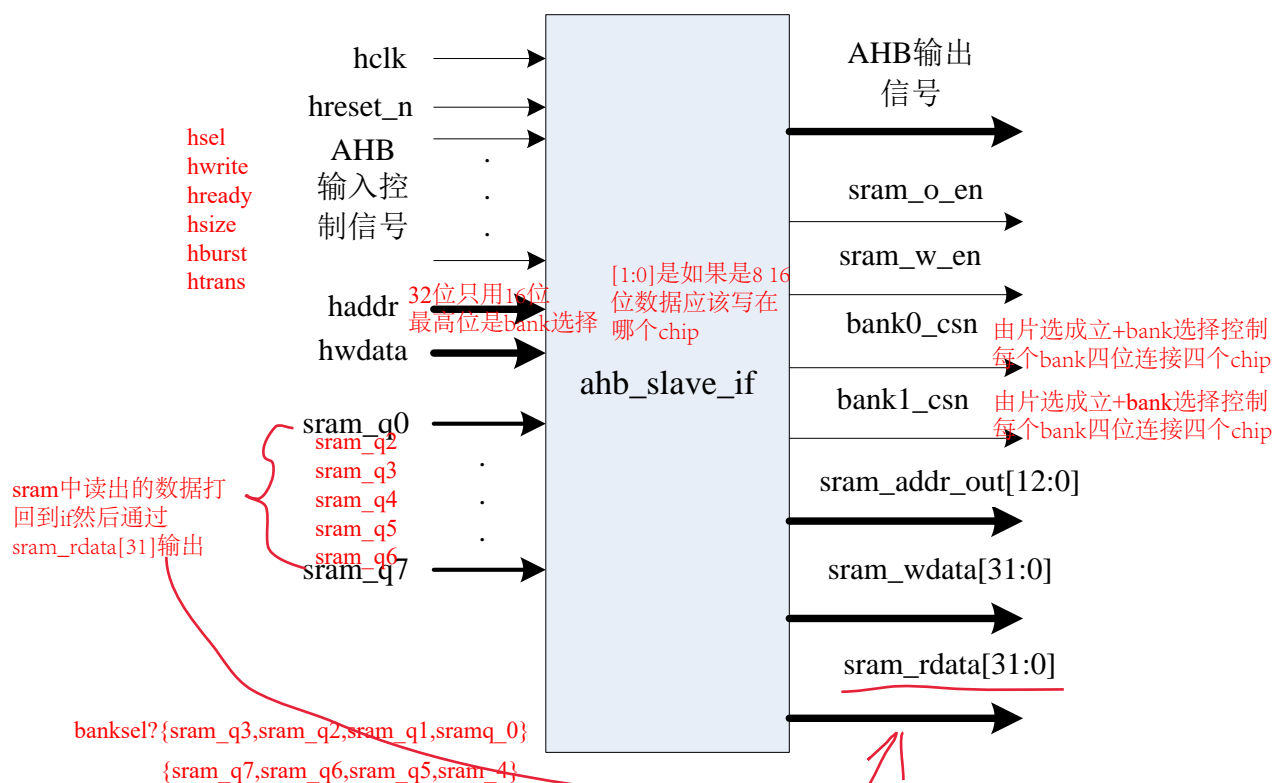


图 2 SRAM 控制器中的 AHB slave 接口

信号功能说明如下：

信号名	方向	位宽	详细说明
ahb_slave_if 模块输出信号			
hclk	input	1	AHB 总线时钟信号
hresetn	input	1	AHB 总线复位信号
hsel	input	1	sram 控制器片选信号
hwrite	input	1	sram 控制器写信号
hready	input	1	sram 控制器准备好信号
hsize	input	[2:0]	传输数据大小
hburst	input	[2:0]	未用到
htrans	input	[1:0]	传输数据类型
haddr	input	[12:0]	AHB 总线的地址输入
hwdata	input	[31:0]	AHB 总线的数据输入
sram_q0 ~ sram_q7	input	[7: 0]	sram 的数据读/写（输出/输入）使能
ahb_slave_if 模块输出信号			

hready_resp	output	1	sram 控制器接收完成的反馈信号
hresp	output	[1:0]	sram 控制器接收状态的反馈信号
hrdata	output	[31:0]	AHB 总线读 sram 的数据
sram_w_en	output	1	sram 的读写使能信号
sram_o_en	output	1	sram 的数据读（输出）有效使能
bank0_csn	output	[3:0]	bank0 块中的 sram 块选择信号
bank1_csn	output	[3:0]	bank1 块中的 sram 选择信号
sram_addr_out	output	[12:0]	发往 sram 的控制器的地址信号
sram_wdata	output	[31:0]	写 sram 的数据信号

- (2) 4 块 SRAM 实例化后的 bank 模块，如图 4 所示：sram\_core.v。该模块受 SRAM 控制器控制，需要重点注意的每块 sram 的片选由图 1 中 sram\_csn[3:0]中的值和 haddr\_reg[15]的值决定的。

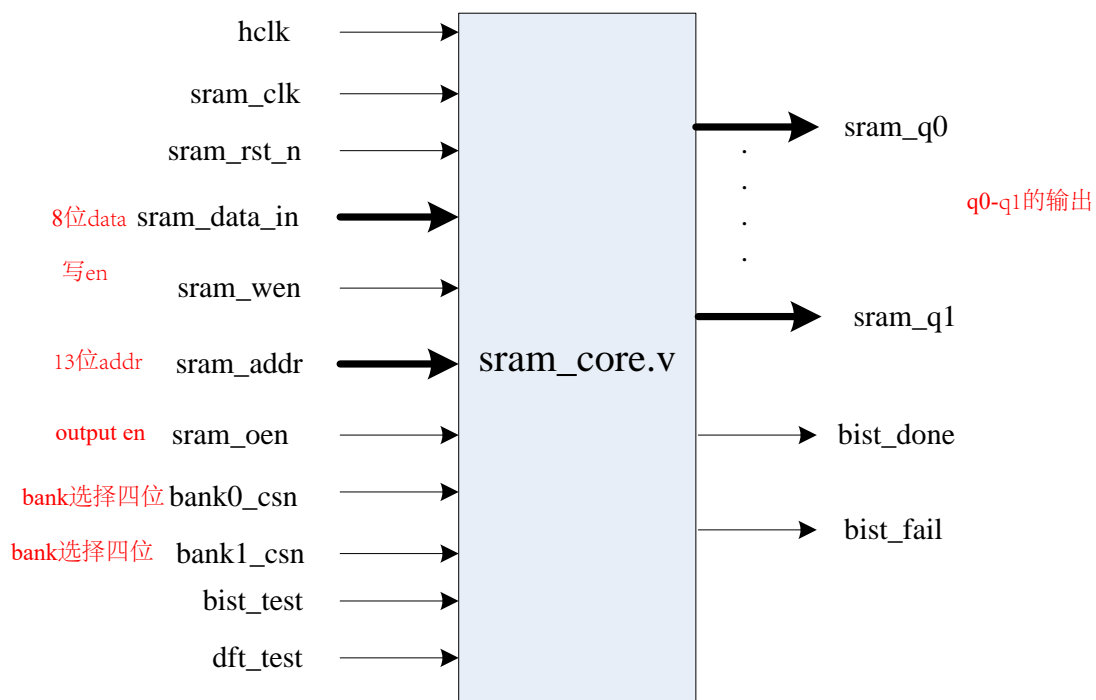


图 4 SRAM Core 框图

信号功能如下：

信号名	方向	位宽	详细说明
sram_core 模块输入信号			

hclk	input	1	系统时钟
sram_clk	input	1	sram 控制器的时钟信号
sram_rst_n	input	1	sram 控制器的复位信号
sram_addr	input	[12:0]	sram 的地址输入
sram_data_in	input	[31:0]	sram 的数据输入
sram_w_en	input	1	sram 的数据读/写（输出/输入）使能
sram_oen	input	1	sram 的数据读（输出）有效使能
bank0_csn	input	[3:0]	sram 的 bank0 块选择信号
bank1_csn	input	[3:0]	sram 的 bank1 块选择信号
bist_test	input	1	sram 的 BIST 测试模式使能信号
dft_test	input	1	sram 的 DFT 测试模式使能信号
sram_core 模块输出信号			
sram_q0 ~ sram_q7	output	[7:0]	sram 的数据输出
bist_done	output	1	BIST 测试模式下的测试结束标志
bist_fail	output	[7:0]	BIST 测试模式下的结果输出

(3) sram\_bist.v 模块，如图 5 所示

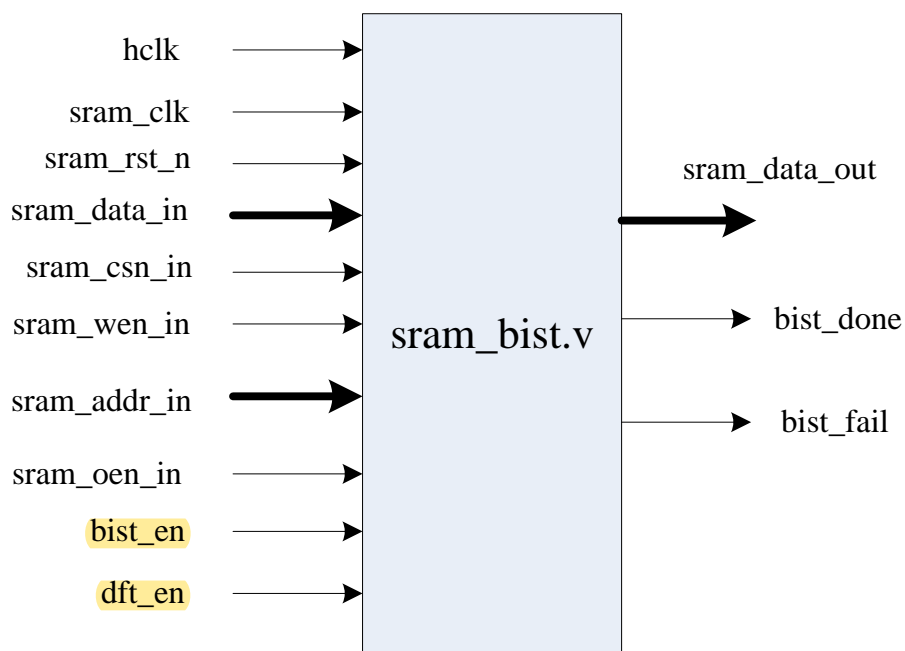


图 5 带有 BIST&DFT 功能的 SRAM 框图

信号功能如下:

信号名	方向	位宽	详细说明
sram_bist 模块输入信号			
hclk	input	1	系统时钟
sram_clk	input	1	sram 控制器的时钟信号
sram_rst_n	input	1	sram 控制器的复位信号
sram_addr_in	input	[12:0]	sram 的地址输入
sram_data_in	input	[7:0]	sram 的数据输入
sram_csn_in	input	1	sram 的片选信号
sram_wen_in	input	1	sram 的数据读/写 (输出/输入) 使能
sram_oen_in	input	1	sram 读 (输出) 数据有效使能
bist_en	input	1	sram 的 BIST 测试模式使能信号
dft_en	input	1	sram 的 DFT 测试模式使能信号
sram_bist 模块输出信号			
sram_data_out	output	[7:0]	sram 的数据输出
bist_done	output	1	BIST 测试模式下结束标志
bist_fail	output	1	BIST 测试模式下的结果输出

(4) BIST 模块, 如图 6 所示

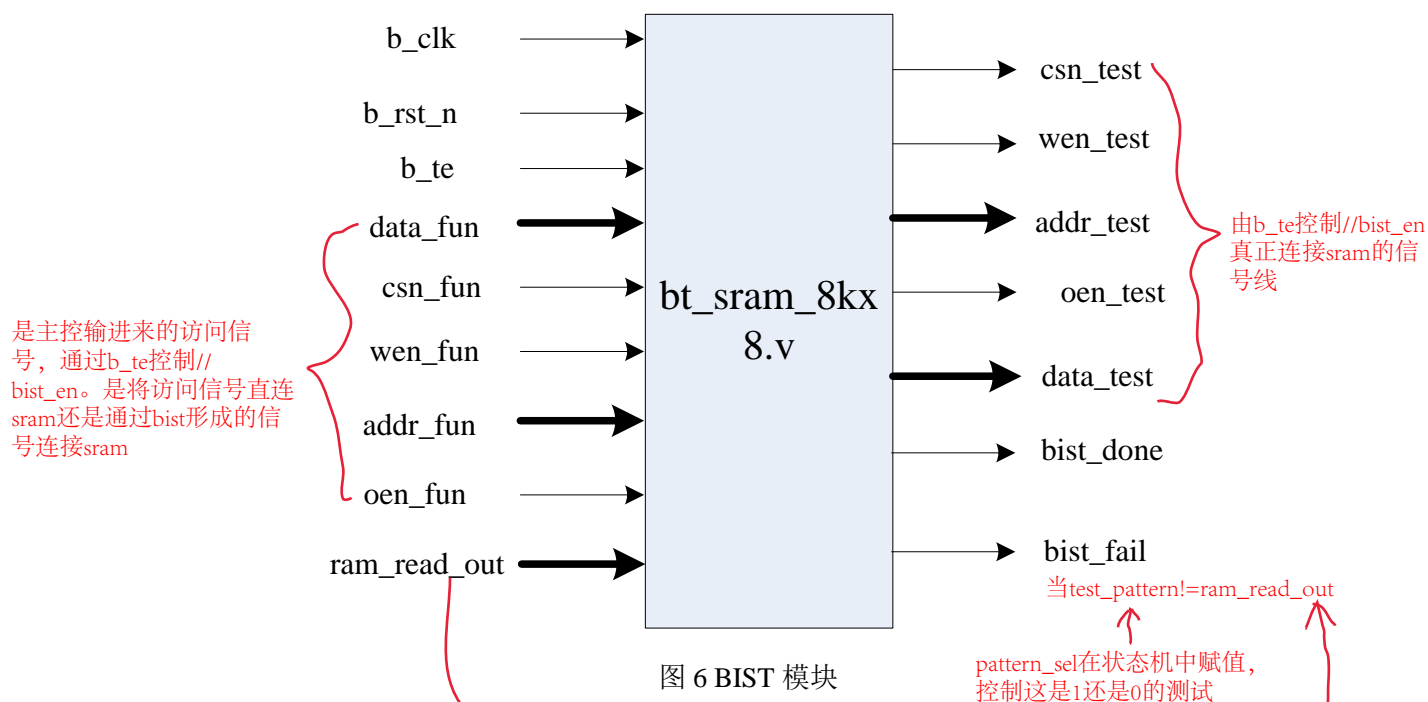


图 6 BIST 模块

信号功能如下：

信号名	方向	位宽	详细说明
BIST 模块输入信号			
b_clk	input	1	BIST 模块工作测试时钟
b_rst_n	input	1	BIST 模块复位
b_te	input	1	BIST 模块工作使能
data_fun	input	[7:0]	总线的输出数据
csn_fun	input	1	总线输出的 sram 的片选使能
wen_fun	input	1	总线输出的 sram 的写使能
addr_fun	input	[12:0]	总线输出的 sram 的地址
oen_fun	input	1	总线输出的读使能
ram_read_out	input	[7:0]	sram 输出的数据的输入
BIST 模块输出信号			
addr_test	output	[12:0]	发往 SRAM 的地址输出
wen_test	output	1	SRAM 读写控制信号
data_test	output	[7:0]	写 SRAM 的输出数据
oen_test	output	1	读 SRAM 数据使能
bist_done	output	1	BIST 模块测试结束标志
bist_fail	output	1	BIST 模块的结果输出

## 顶层信号描述

SRAM 控制器顶层模块，图 6：sramc\_top.v

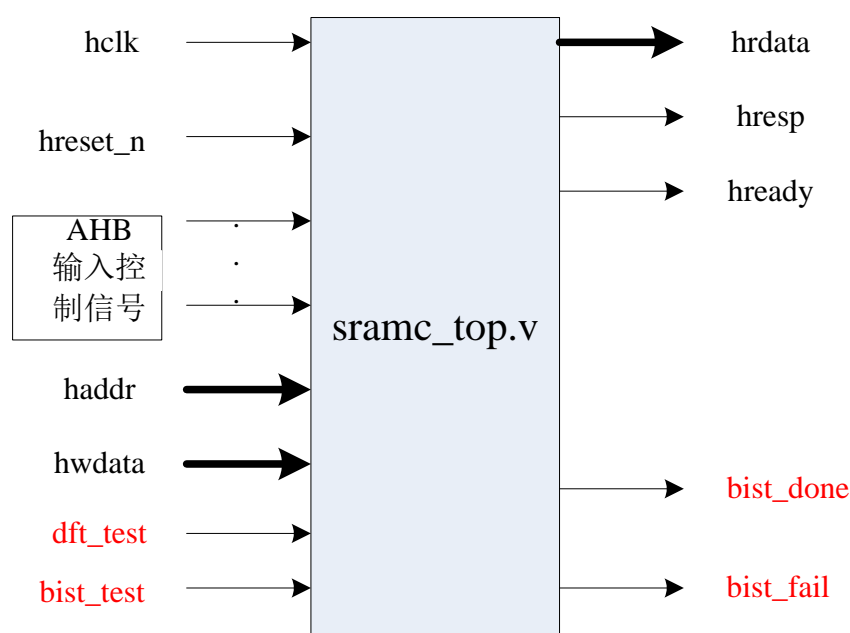


图 6 SRAM 控制器顶层模块

信号功能说明如下：

信号名	方向	位宽	详细说明
sramc_top 模块输出信号			
hclk	input	1	AHB 总线时钟信号
hresetn	input	1	AHB 总线复位信号
hsel	input	1	sram 控制器片选信号
hwrite	input	1	sram 控制器写信号
hready	input	1	sram 控制器准备好信号
hsize	input	[2:0]	传输数据大小
hburst	input	[2:0]	未用到
htrans	input	[1:0]	传输数据类型
haddr	input	[12:0]	AHB 总线的地址输入
hwdata	input	[31:0]	AHB 总线的数据输入
bist_test	input	1	BIST 测试模式使能信号
dft_test	input	1	DFT 测试模式使能信号
sramc_top 模块输出信号			
hready_resp	output	1	sram 控制器接收完成的反馈信号
hresp	output	[1:0]	sram 控制器接收状态的反馈信号
hrdata	output	[31:0]	AHB 总线读 sram 的数据
bist_done	output	1	BIST 测试结束标志
bist_fail	output	[7:0]	BSIT 测试结果, sram 是否功能错误标志

## 模块功能以及设计注意点

本设计分为几大模块: sramc\_top.v、ahb\_slave\_if.v、sram\_core.v, 其中 sramc\_top.v 为顶层模块, 其余为内部模块。

(1) sramc\_top.v 中除了作为 AHB slave 接口的标准输入/出信号外, 需要添加的信号如下:

输入端: dft\_test、bist\_test

输出端: bist\_done、bist\_fail

- (2) ahb\_slave\_if.v 用来生成 SRAM 的读写控制信号和片选所需的信号。如图 1 中，该模块输出将 haddr、hsize 寄存了一拍输出成 haddr\_reg、hsize\_reg。同时还需要对 hwrite 信号进行特别处理（为了满足单周期读写，可以见下面的 SRAM 读写时序），一方面，当写有效时，寄存一拍输出，信号为 sram\_wen，另一方面，当读有效时，模块内部要进行处理下，hwrite 进行逻辑组合输出，信号为 sram\_ren。本模块的读写数据无需进行寄存，其中 haddr[15]用来生成 SRAM 的 bank 选择和 byte/halfword/word 的数据读写操作选择。通过 AHB 接口过来的片选信号和地址总线的逻辑运算产生 SRAM\_CS<sub>n</sub> 的片选值。

表 1 SRAM\_CS<sub>n</sub>[3:0]的编码真值表

HADDR[15] （选通 Bank）	SRAM_CS <sub>n</sub> [3:0]
X	1111
0	1110, ...
1	1101, ...

SRAM\_CS<sub>n</sub>[3:0]的值为 1111 时，为系统复位状态时的值，其他值根据 SRAM 的 Byte/halfword/word 读写操作方式来确定。

SRAM\_CS<sub>n</sub> 片选 SRAM 块示意框图如图 7 所示：

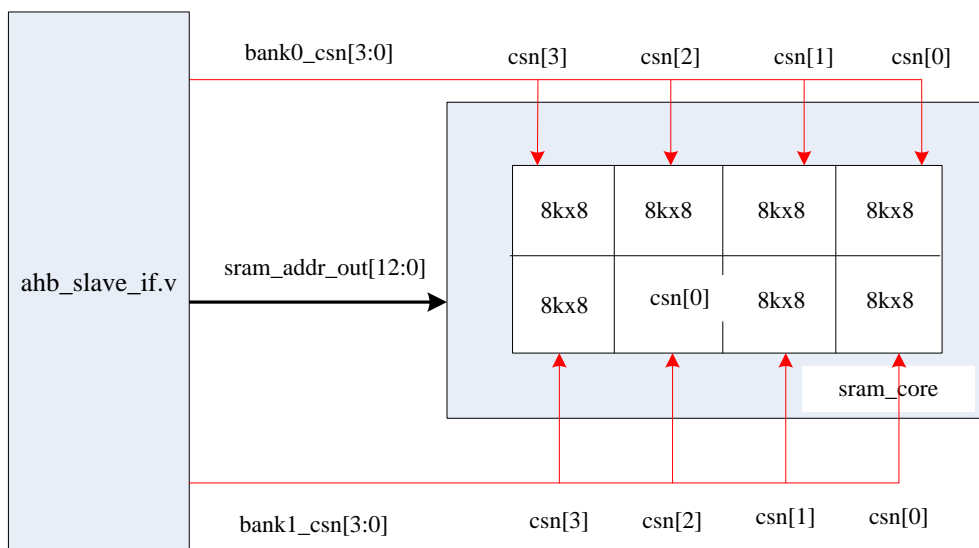


图 7 SRAM 块的片选信号

对 SRAM 的 byte/halfword/word 读写操作，如下表 2 所示：通过 AHB 总线的传输方式 HSIZE 和地址总线的低 2 位的逻辑运算产生 AHB-SRAM 数据位宽选择值——



SRAM\_WEN。

表 2 SRAM\_WEN[3:0]的编码真值表

HSIZE[1:0]	HADDR[1:0]	SRAM_WEN[3:0]	
10 (word)	XX	0000	32位直接是四片全写
01 (half word)	0X	1100	如果是16位，通过haddr控制写在高两位还是低两位
01 (half word)	1X	0011	
00 (byte)	00	1110	如果是8位那就通过haddr来控制写在哪一个chip
00 (byte)	01	1101	
00 (byte)	10	1011	
00 (byte)	11	0111	

在选定某个 bank 后，当确定数据是以 byte/halfword/word 中一种读写方式操作时，通过 sram\_csn 片选使能选定比如高 16 位进行读写，或者低 8 位、或者一个 word（整行）的读写操作的 SRAM 块。

- (3) sram\_core.v 模块包括 8 个点有 BIST&DFT 功能的 SRAM 块。通过顶层模块的输入信号 bist\_test 模式时，进入到 BIST 测试，使能选通 MUX 后会为 SRAM 提供相应的测试数据，地址以及使能信号等。此处特别注意的是将 BIST 模块封装到 SRAM core 中，目的是为了减少外部模块的连线，能更方便地对 SRAM 类型的 FIFO 进行自测，如图 8 所示。

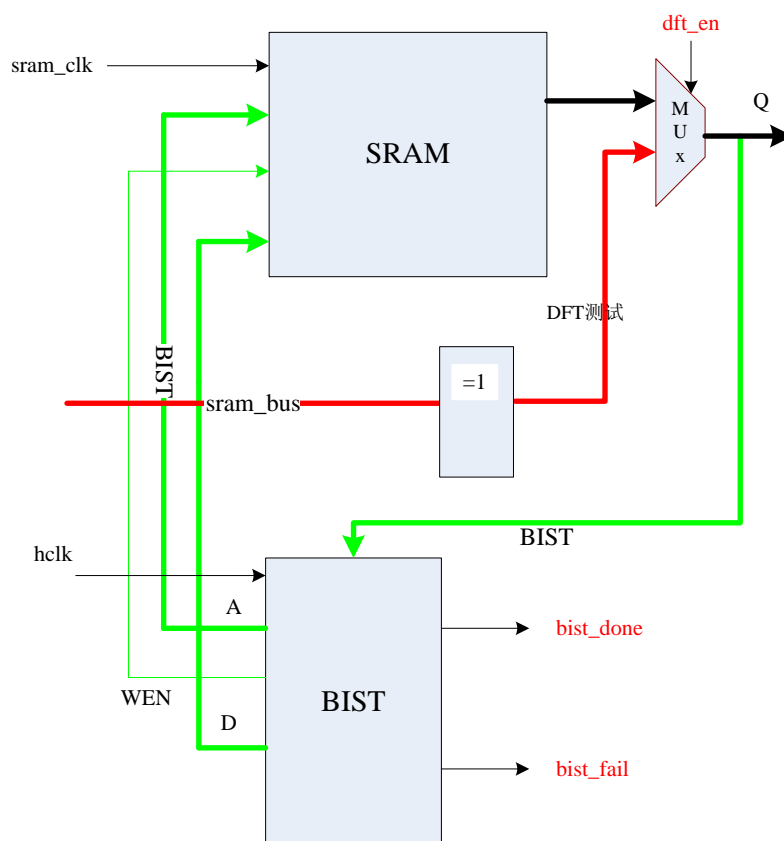


图 8 sram 控制器中的 DFT/BIST 模块的封装

同时在本模块还需完成另外一个功能——DFT 测试模式的选择。由于 SRAM 块类似于模拟模块，在进行 DFT 测试时，扫描链会在此断掉，所以需要进行一个处理——bypass 掉 SRAM 块，通过将 SRAM 的输入和输出信号进行如图 1 中所示那样，对 SRAM 输入的信号进行异或并输出到共同的输出口 Q 端，从而形成一个完整的扫描链路径。

- (4) bist 模块主要包括以下部分：模式选择器、测试模式发送器、BIST 状态机、结果分析模块。如图 9 所示。

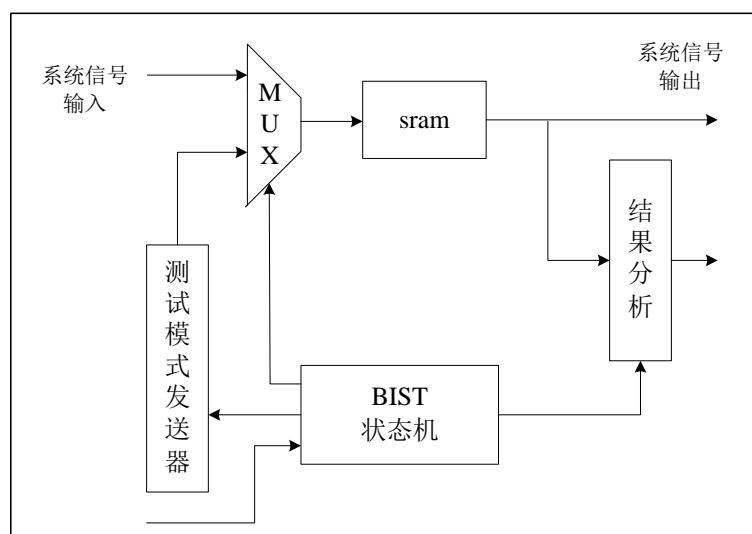


图 9 bist 模块结构

各个模块的作用如下：

**模式选择器：**在 sram bist 测试模式信号下选择 SRAM 的输入信号，正常工作模式 SRAM 接收系统信号；测试模式 SRAM 接收 sram bist 产生的测试信号。

**测试模式发送器：**其包括两个部分，一个是地址产生器，用来遍历整个 sram 的地址序列，或递增，或递减；另外一个数据产生器，产生测试数据，写入 sram 的同时，会发给结果分析模块作为期望读出的正确值，与实际读出的值进行比较。

**BIST 状态机：**根据测试算法对其他部分发出信号，主要控制测试模块的地址和数据的产生和读写方式。

**结果分析模块：**采用比较器，如果读出的结果和先前写入的数据一致则说明存储器无故障，否则存储器存在故障。

本 BIST 电路采用确定性 BIST，通过有限状态机产生确定的预先设定好的值，而这些值是根据 March C 算法来确定的。BIST 测试原理如下：

先按地址递增的顺序进行以下操作：

写 '0'（初始化）；

读 ‘0’，写 ‘1’；

读 ‘1’，写 ‘0’；

读 ‘0’。

再按地址递减的顺序进行以下操作：

读 ‘0’，写 ‘1’；

读 ‘1’，写 ‘0’；

读 ‘0’。

## 附

参考文献