

亿海神针系列 FPGA 器件用户指南

EQ6HL45 器件

UG208 (v2.4) 2023.07.31



版权声明

Copyright © 2023 该文档所有版权归属中科亿海微电子科技（苏州）有限公司所有。未经本公司书面许可，任何单位和个人不得以任何方式或理由对文档、信息、材料的任何部分进行使用、复制、修改、抄录、传播或与其它产品捆绑使用、销售。该文档信息准确并可信，但中科亿海微电子科技（苏州）有限公司不对该文档内容的准确性、完整性以及信息和其它内容使用中发生的技术问题以及可能给使用者造成的损失承担责任。中科亿海微电子科技（苏州）有限公司保留对该文档及其内容的修改、更新、解释权。

中科亿海微电子科技（苏州）有限公司

苏州市吴中区甪直镇长虹北路169号吴淞江商务区A幢2层和B幢裙楼1层, 215000

北京市丰台区南四环西路128号诺德中心1号楼西配楼7层, 100070

成都市高新区高朋大道3号东方希望大厦A座508, 610000

电话：010-51136695

邮箱：support@ehiway.cn

网址：www.ehiway.cn

版本说明

日期	版本	修订
2021.07.02	2021.1.1	新编
2021.09.28	2021.1.2	修改格式、页眉
2021.10.18	2021.1.3	修改内容
2021.11.01	2021.1.4	修改图 1
2021.12.24	2021.1.5	增加免责声明，修正 DSP 模块数量；修改主动串行配置模式示意图；修正可编程寄存器个数；修改表 30、图 69、图 70、图 79、图 88、图 90；修改 PLL 参数；
2022.01.25	2022.1.1	修改表 2；更改文字内容、标点符号、错别字
2022.02.15	2022.1.2	修改表 24 JTAG 数据，删除 10.2 特殊管脚
2022.03.15	2022.1.3	修改器件芯核电路阵列规模
2022.04.29	2022.1.4	删除 10.1.1 特殊管脚表述； 修改质量等级中 IO 数与表 2 一致 修改逻辑阵列数
2022.06.27	2.0	修改图 70，修改页眉，变更版本号
2022.08.01	2.1	修正文档中的图片、表格编号
2022.12.30	2.2	修改表 35 和表 41 单位
2023.05.17	2.3	删除表 43 和修改相应文本
2023.07.31	2.4	修改参数

目录

1 EQ6HL45 特性	1
2 逻辑模块 (LC)	4
2.1 基本逻辑单元 (BLE)	4
2.1.1 基本结构	4
2.1.2 工作模式	5
2.1.3 控制逻辑模块 (CL)	6
2.2 可编程互连资源	7
3 时钟网络	8
3.1 时钟网络	8
3.1.1 概述	8
3.1.2 全局时钟网络	9
3.1.3 局域时钟网络	9
4 输入输出模块	10
4.1 IOBLK	10
4.2 I/O 接口标准	13
4.3 可编程电流驱动	14
4.4 可编程摆率控制	14
4.5 可编程弱上拉	14
4.6 可编程开漏输出	15
4.7 总线保持功能	15
4.8 LVDS 功能	15
5 嵌入式存储器 (BRAM)	18
5.1 功能特性	18
5.2 工作模式	19
5.2.1 Packed 模式	19
5.2.2 单端口模式	20
5.2.3 简单双端口模式	21
5.2.4 真双端口模式	22
5.2.5 移位寄存器模式	24
5.2.6 ROM 模式	24
5.2.7 FIFO 模式	25
5.3 时钟模式	25
5.3.1 独立时钟模式	25
5.3.2 输入/输出时钟模式	26
5.3.3 读写时钟模式	27
5.3.4 单时钟模式	28
5.4 读写冲突处理	28
5.4.1 同一端口读写冲突处理	28
5.4.2 混合端口读写冲突处理	29
5.4.3 读写冲突处理机制的适用范围	29
5.5 上电情形和存储器初始化	29

5.6 字节使能功能	29
6 嵌入式乘法器 (DSP)	31
6.1 DSP 功能概述	31
6.2 DSP 结构	31
6.2.1 乘法单元	32
6.2.2 加法、输出单元	33
6.2.3 端口描述	35
6.3 DSP 工作模式	36
6.3.1 简单乘法模式	36
6.3.2 乘累加模式	38
6.3.3 双乘加模式	38
6.3.4 四乘加模式	40
6.3.5 FIR 的应用	40
6.3.6 有符号数的乘法	41
6.3.7 Scanout 的处理	42
6.3.7.1 简单乘法模式	42
6.3.7.2 乘累加模式	42
6.3.7.3 双乘加模式	42
6.3.7.4 四乘加模式	42
6.3.7.5 36 位乘法模式	42
7 JTAG 电路	44
7.1 概述	44
7.2 电路结构	44
7.2.1 接口信号说明	45
7.2.2 子模块电路功能说明	45
7.2.3 操作时序	46
7.3 功能模块	46
7.3.1 TAP 控制器	46
7.3.2 指令寄存器	47
7.3.3 边界扫描寄存器	48
7.3.4 旁路寄存器	48
7.3.5 器件标志寄存器	48
7.4 边界扫描指令	48
7.4.1 外部测试指令(EXTEST)	48
7.4.2 采样指令 (Sample) 和预加载指令(Preload)	49
7.4.3 旁路指令(Bypass)	49
7.4.4 高阻指令(HIGHZ)	49
7.4.5 钳位指令(CLAMP)	49
7.4.6 器件标志指令(IDCODE)	49
7.4.7 配置控制器异步复位指令(JPROG_B)	49
7.4.8 配置指令(CFG_IN)	50
7.4.9 解冻指令(JSTART)	50
7.4.10 回读指令(CFG_OUT)	50
7.4.11 BitProbe 指令(BitProbe)	50

8 配置	51
8.1 概述	51
8.2 配置管脚	51
8.3 配置模式	52
8.3.1 主动串行（Master Serial）模式	53
8.3.2 被动串行（Slave Serial）模式	54
8.3.3 主动并行模式	55
8.3.4 被动并行模式	56
8.3.5 JTAG 模式	57
8.3.6 SPI Master 模式	57
8.3.7 SPI Slave 模式	61
8.3.8 Daisy-chain 模式	63
8.4 回读模式	64
8.4.1 被动并行回读模式	64
8.4.2 JTAG 回读模式	65
8.5 Multiboot	67
8.5.1 概述	67
8.5.2 Multi-boot 码流配置	67
8.5.3 Multi-boot ICAP 配置	68
8.5.3.1 接口信号说明	68
8.5.3.2 接口协议说明	69
8.5.3.3 接口时序说明	69
8.5.4 状态寄存器	69
8.6 配置流程	70
8.7 配置芯片	72
9 电特性参数	73
9.1 静态电特性参数	73
9.2 动态电特性参数	74
9.2.1 基本逻辑单元	74
9.2.2 互连资源开关参数	75
9.2.3 时钟网络和 PLL	75
9.2.4 I/O 通道	76
9.2.5 嵌入式存储器 BRAM	76
9.2.6 DSP	77
10 板级电接口设计参考	78
10.1 电源部分设计	78
10.1.1 电源电压需求	78
10.1.2 PLL 电源部分设计	78
附录 A	80

1 EQ6HL45 特性

EQ6HL45 是一款基于静态存储器（以下简称 SRAM）结构的 FPGA 产品(可编程逻辑门阵列)，具有大容量、高性能、高性价比的特点。该芯片阵列规模为 92×56，其中 76 列逻辑阵列（Logic Cluster，以下简称 LC），系统门容量达到 425 万门。具有高速时钟管理功能、丰富的 I/O 资源、支持多种配置方式等特点。

其主要特性：

- 基于标准 40nm CMOS 工艺设计加工
- 有效门容量达到 425 万门
 - 具有 42500 个 4 输入 LUT；
 - 具有 42500 个可编程寄存器；
 - 具备 392 个 4.5Kbit 的嵌入式存储器单元（包含奇偶校验位），最大可实现 1764K bit 的双端口 RAM；
 - 具备 56 个高速 18bit×18bit 乘法器。
- 高速时钟管理功能实现功耗、延时、抖动和偏斜的综合优化
 - 具有 8 个可编程 PLL 实现频率综合；
 - 最高时钟管理频率可达 500MHz；
 - 最多可提供 16 路全局时钟信号。
- 最大可提供 232 个可编程用户 I/O
 - 支持多种 I/O 标准：3.3V/2.5V/1.8V/1.5V-LVTTL/LVCMOS 标准，支持 SSTL2/SSTL3 标准，最高输入/输出频率可达 250MHz；
 - 最多提供 116 对 LVDS 标准的差分接口，最高传输速率可达 640Mbps；
 - 支持片上数字控制终端电阻（Digital Control Impedance, DCI）；
 - 具有可编程的 I/O 驱动能力、摆率控制、总线保持、弱上拉等功能；
 - 具有兼容 IEEE 1149.1 标准的边界扫描电路。
- 提供多种配置方式
 - 支持主动串行、被动串行、JTAG、主动并行、被动并行、SPI 等配置模式；
 - 建议配置速率在 1 MHz~15 MHz。
- 质量等级
 - 采用塑料 CSG324 封装形式，可提供 232 个用户 IO, 116 对 LVDS 差分对，工作环境满足最宽-55℃~125℃的温度范围；
 - 采用塑料 CSG225 封装形式，可提供 160 个用户 IO, 78 对 LVDS 差分对，工作环境满足最宽-55℃~125℃的温度范围；
 - 采用塑料 FTG256 封装形式，可提供 186 个用户 IO, 91 对 LVDS 差分对，工作环境满足最宽-55℃~125℃温度范围；
 - ESD 耐压大于 2000V。

EQ6HL45 器件基本参数见表 1, 器件封装及最大可用 I/O 数见表 2。

表 1 EQ6HL45 器件基本参数

器件	逻辑资源			最大 I/O 资源		嵌入式存储器		时钟资源		DSP
EQ6HL45	LUT	BLE	LC	USER I/O	LVDS 差动对	存储单元	BRAM 总容量	PLL	全局时钟	18b×18b 乘法器
	42500	42500	4250	232	116	392	1764K	8	16	56

表 2 器件封装及最大可用 I/O 数

封装	CSG324	CSG225	FTG256
封装尺寸 (mm)	15*15	13*13	17*17
间距 (mm)	0.8	0.8	1.0
用户 IO 数 (/LVDS 差分对数)	232/116	160/78	186/91

EQ6HL45 的可编程结构主要由以下几个功能模块部分构成: 逻辑阵列模块 (LC)、内部互联资源、可编程 IO 单元 (以下简称 IOE)、PLL、DSP 和嵌入式存储器等, 如图 1 所示。

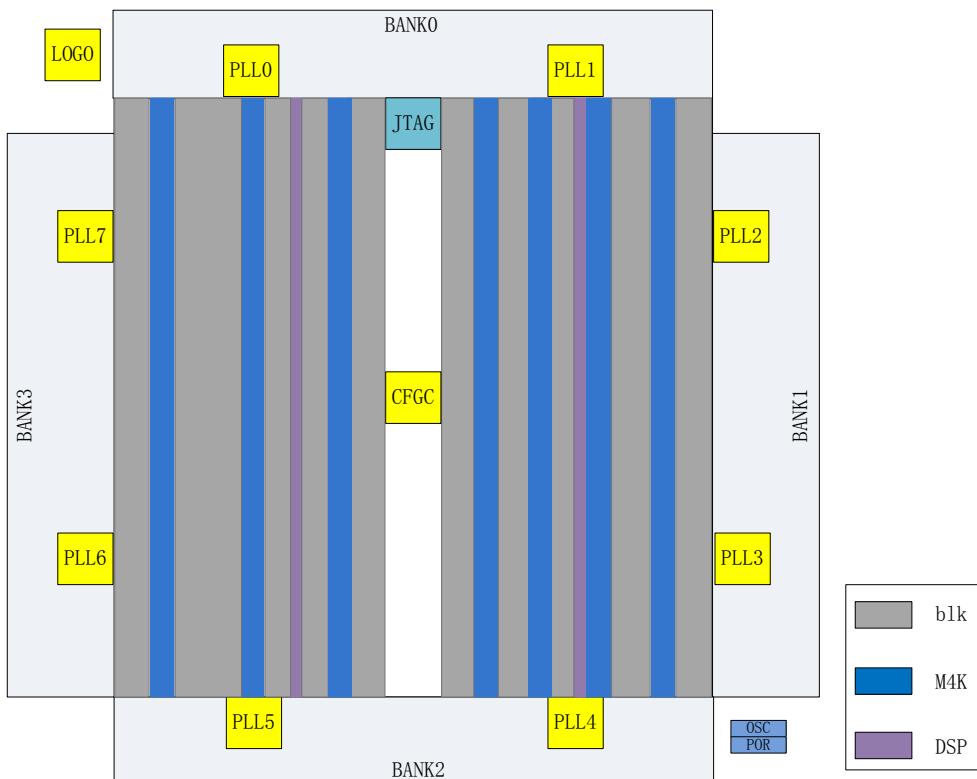


图 1 EQ6HL45 整体结构示意图

器件芯核电路阵列规模为 92×56 , LC 是一个相对独立的实体, 它可以独自完成一定的逻辑功能, 也可以通过位于 LC 之间的连线通道进行相互之间的通信, 以实现更复杂的逻辑功能。

内部连线资源模块包括连线通道 (以下简称 Track) 和连线通道交叉开关(以下简称 Switch)等。连线通道又分为水平连线通道和垂直连线通道, 它们都是由一组规则的连线单元

组成，负责实现逻辑阵列块之间的连接。

逻辑阵列四周为可编程 I/O 单元，负责实现芯片内部与外围电路的通信，EQ6HL45 最大可以提供 232 个用户可编程 I/O。

EQ6HL45 提供了独立的可编程 PLL 模块，用户可以方便的对 PLL 进行配置，完成频率综合、相移、可调占空比等功能。

EQ6HL45 提供了 14 个数字信号处理 (DSP) 单元。每一个 DSP 单元可以同时完成 8 个 $9b \times 9b$ 的乘法、或 4 个 $18b \times 18b$ 的乘法、或 1 个 $36b \times 36b$ 的乘法，且乘法结果可以进行加减、累加和求和等运算。每一个 DSP 单元包含 18 位输入移位寄存器用于常见的数字信号处理应用，如 FIR、IIR 等。

器件具有 392 个嵌入式存储器单元，共 1764Kb RAM，可以实现以下功能：实现 RAM、ROM、Shift Register 和 FIFO 等不同形式的存储功能以满足用户对不同形式存储器的需求；实现不同的工作模式，包括端口模式(单端口模式、双端口模式)和时钟模式(单时钟模式、读写时钟模式、输入输出时钟模式和独立时钟模式)；实现对输入输出信号的控制，包括：输入输出信号的寄存和清零实现；字节使能的实现；字宽的配置；写期间读模式的配置等；此外，用户可以通过编程配置写入初始化信息来实现对嵌入式可编程存储器 (BRAM) 的内容进行初始化。

2 逻辑模块 (LC)

LC (Logic Cluster) 是 FPGA 中的逻辑阵列模块，每个 LC 中包含 10 个 BLE (Basic Logic Element)、1 个控制逻辑电路。BLE 是最基本的可编程逻辑单元，控制逻辑电路负责产生同一个 LC 中每个 BLE 的控制信号。LC 的基本结构如图 2 所示。

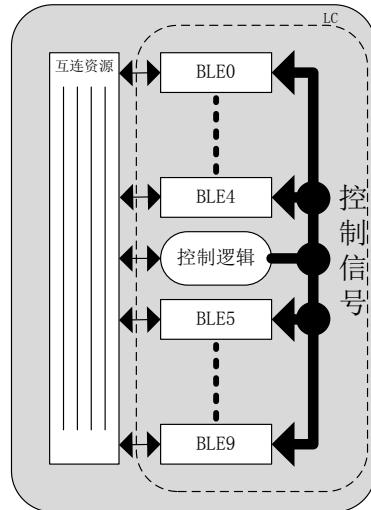


图 2 LC 的结构示意图

2.1 基本逻辑单元 (BLE)

2.1.1 基本结构

基本逻辑单元 (BLE) 是 FPGA 中的最小可编程功能单元，其基本结构如图 3 所示。

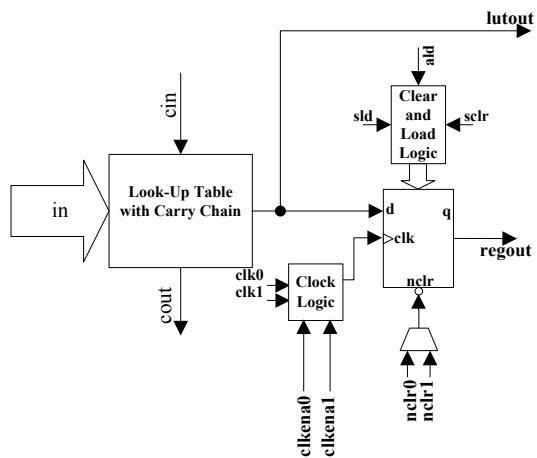


图 3 BLE 的结构示意图

每个 BLE 包含一个 4 输入查找表 (Look-Up Table, LUT)、一个由时钟控制的可编程触

发器、一个进位链、一个查找表级联链和一个寄存器级联链等，可实现任意 4 输入（luta、lutb、lutc、lutd）组合或时序逻辑电路功能。

BLE 中的触发器为 D 触发器，其端口如图 3 所示：数据输入端（d）、时钟端（clk）、异步置数端（ald）、异步清零端（nclr）。

查找表逻辑函数真值表存储在 16 个 SRAM 单元中，可以实现任意四输入的组合逻辑功能。对于时序逻辑功能，查找表的输出通过触发器寄存输出，而对于组合逻辑功能，查找表的输出旁路触发器后直接输出到 BLE 的输出端。

2.1.2 工作模式

BLE 有两种工作模式：普通模式和算术模式。

(1) 普通模式

BLE 普通模式用来实现通用的逻辑功能，其连接关系如图 4 所示。在普通模式下，查找表的 4 个输入端 luta、lutb、lutc、lutd 来自于 LC 内部的局部互连资源。此外，查找表输入 lutc 可以来自同一 BLE 的寄存器输出，lutd 可以来自上一个 BLE 的查找表级联输出。查找表可以实现任意的 4 输入功能函数（种类共有 2^{16} 种）。

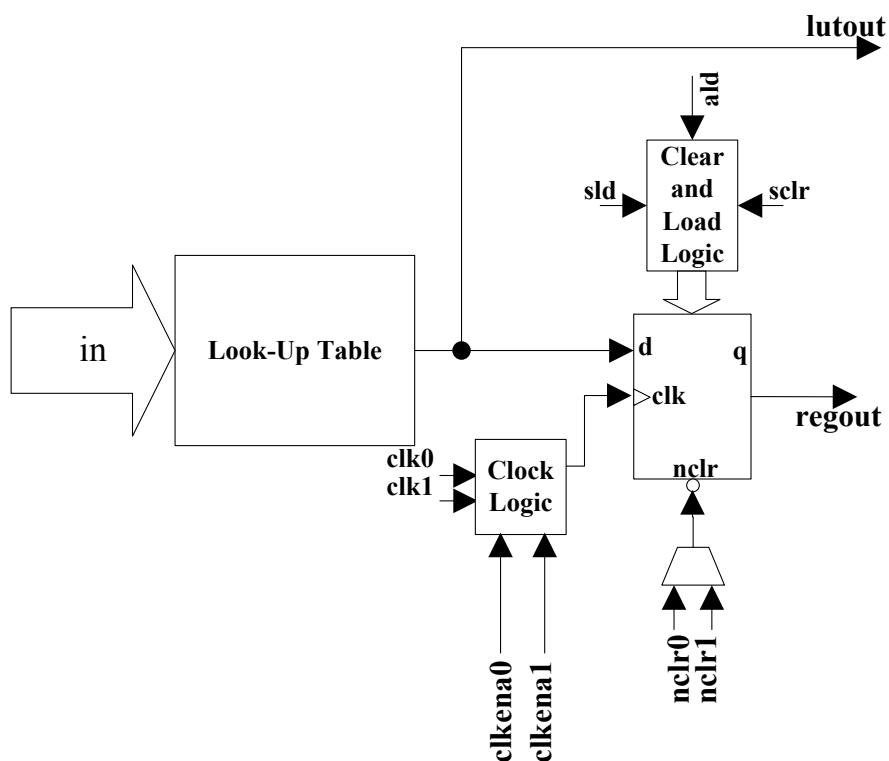


图 4 BLE 普通工作模式下连接关系示意图

(2) 算术模式

BLE 算术模式用于实现不同的算术功能（如加法、减法），其连接关系如图 5 所示。在算术模式下，算术逻辑的操作数（luta 和 lutb）来自于 LC 内部的局部互连资源，进位输入来自于上一个 BLE 的进位输出或控制逻辑 CL。每个 BLE 可以实现一位的加法或减法操作：当加减控制信号 arith 为高时，BLE 实现的是 luta、lutb 和进位输入的加法操作；当 arith 为低时，BLE 实现的是 lutb 和 luta 的减法操作。算术模式不局限于实现加法和减法，对于累

加、比较等逻辑均可通过进位链来实现。

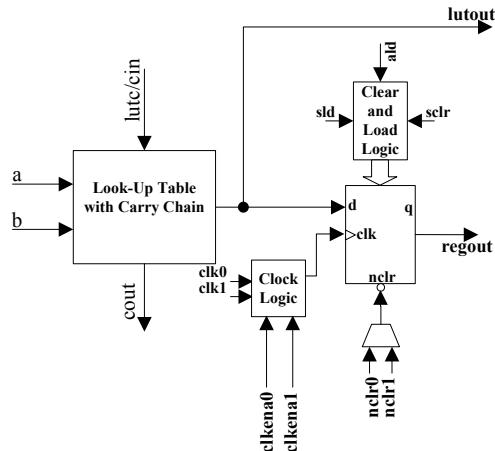


图 5 BLE 算术工作模式下连接关系示意图

2.1.3 控制逻辑模块 (CL)

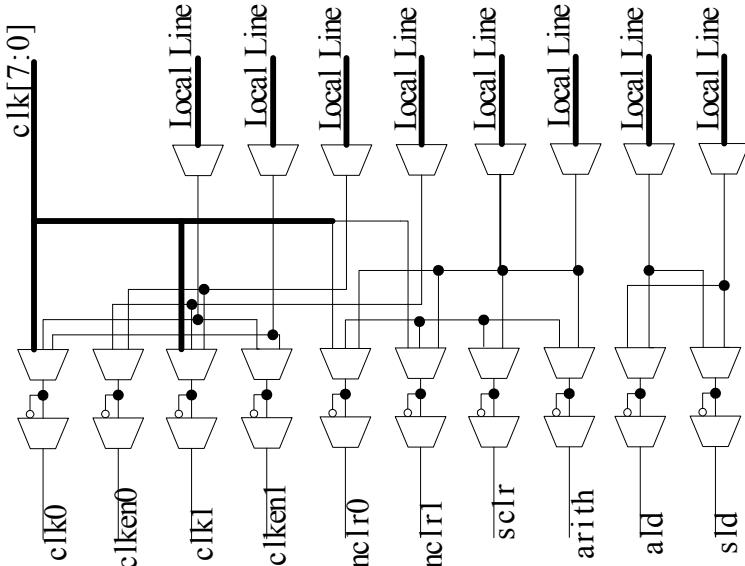


图 6 LC 控制逻辑功能示意图

控制逻辑模块为 LC 中的 BLE 提供的信号包括：2 路时钟信号（clk0 和 clk1）、2 路时钟使能信号（clken0 和 clken1）、2 路异步清零信号（nclr0 和 nclr1）、1 路同步清零信号（sclr）、1 路异步置数信号（alid）、1 路同步置数信号（sld）、加减控制信号（arith）等。

控制信号来源于全局时钟和互连资源，全局时钟可以作为时钟信号、异步清零信号的来源，互连资源可以作为时钟信号、时钟使能信号、异步清零信号、同步清零信号、异步置数信号、同步置数信号、加减控制信号的来源。

2.2 可编程互连资源

可编程互连资源包括跨越一定数目 LC 的互连通道、实现互连通道之间连接以及 LC 与互连通道之间连接的交叉开关。

互连通道包括水平互连通道和垂直互连通道。水平互连通道示意图如图 7 所示：

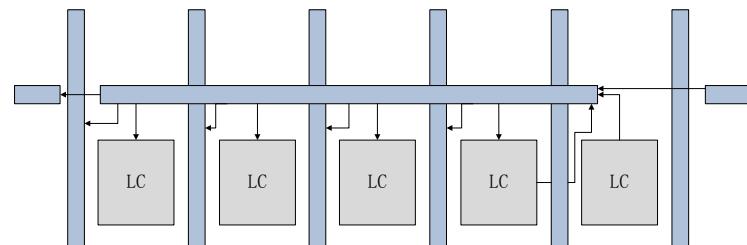


图 7 水平通道示意图

垂直互连通道示意图如图 8 所示：

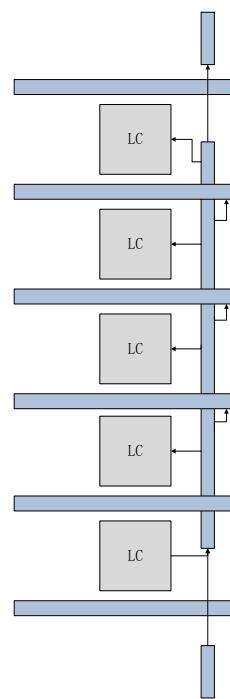


图 8 垂直通道示意图

3 时钟网络

3.1 时钟网络

3.1.1 概述

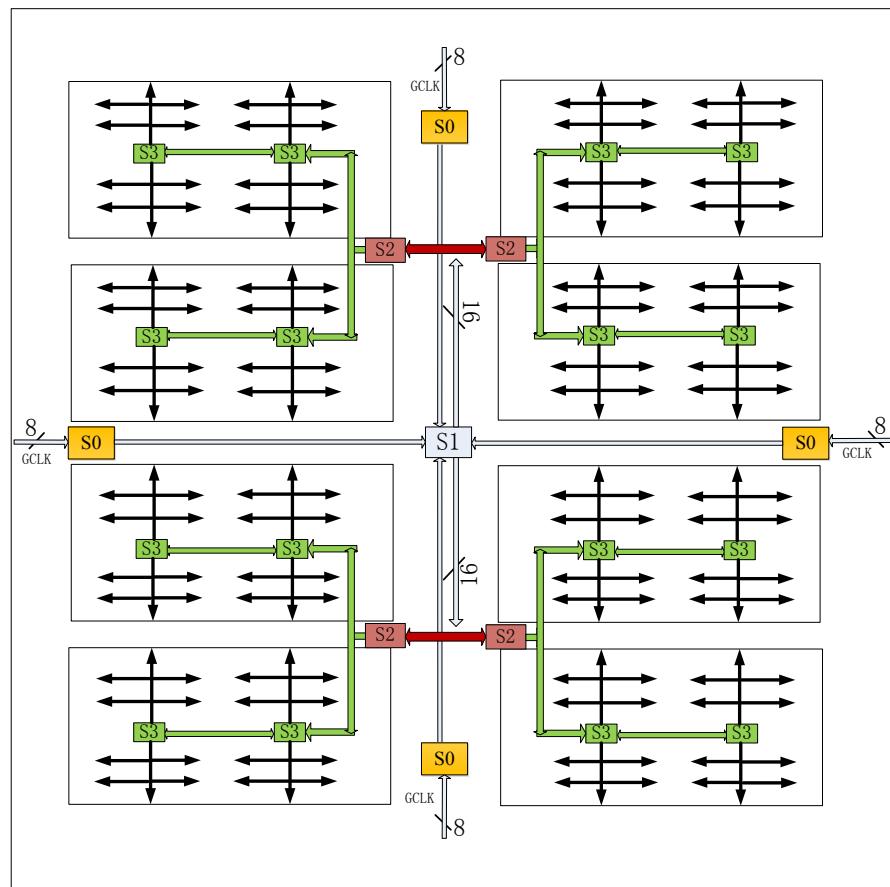


图 9 全局时钟网络整体分布示意图

EQ6HL45 系列 FPGA 产品提供了稳定快速的时钟信号输入资源，包括全局时钟和局域时钟。全局时钟信号来自 EQ6HL45 的上下左右四边的中间输入，共 32 路时钟信号，通过 S0 进入时钟网络，每个象限有 16 路时钟信号，通过 S0 至 S3 共 4 级模块分布到内核各个时钟区域。局域时钟从 EQ6HL45 芯片芯核的四周提供输入，通过 S3 进入时钟网络。时钟网络提供的功能包括：

1. 时钟对解冻信号的同步；
2. 时钟分频功能，支持 2, 4, 6, 8 分频；
3. 门控时钟功能，控制信号来自通道布线资源；
4. 提供 8 路从通道布线资源上时钟网络的接口，信号来自上下左右 iotrack；
5. 时钟区域：S3 提供最大为 32*32 个 BLK 区域的时钟；每个 4*4 的 BLK 由一个门控 buffer 控制，控制该 4*4BLK 模块的时钟开启或关闭，在该区域不使用时钟的时候关闭时钟，以降低动态功耗；

6. 提供时钟末端 pin 脚引出。

3.1.2 全局时钟网络

全局时钟 GCLK 按照象限分布，共 4 个象限。每个象限分成 2 个时钟区域，提供 8 个 GCLK 时钟信号。

全局时钟信号经由 S0~S3 共 4 级的 GCLKMUX 进入芯片的各个逻辑资源模块，其中时钟信号在器件中心的 S2 模块由 4 个 16 选 8 的全连通 MUX 分别为每个象限产生 8 路时钟信号，四个象限的时钟相互独立，选通机制相同。

全局时钟信号来源包括每边的 2 个 PLL 的输出时钟、时钟专用管脚以及通道布线资源。

3.1.3 局域时钟网络

局域时钟为 FPGA 核提供更小延时的时钟信号，从 EQ6HL45 系列 FPGA 芯片的四周进入到 S3，通过 S3 分布到核的部分区域中，最大可提供一个 32×24 个 CLB 的区域时钟。

局域时钟信号来源包括每边的 2 个 PLL 的输出时钟和时钟专用 IO 管脚。

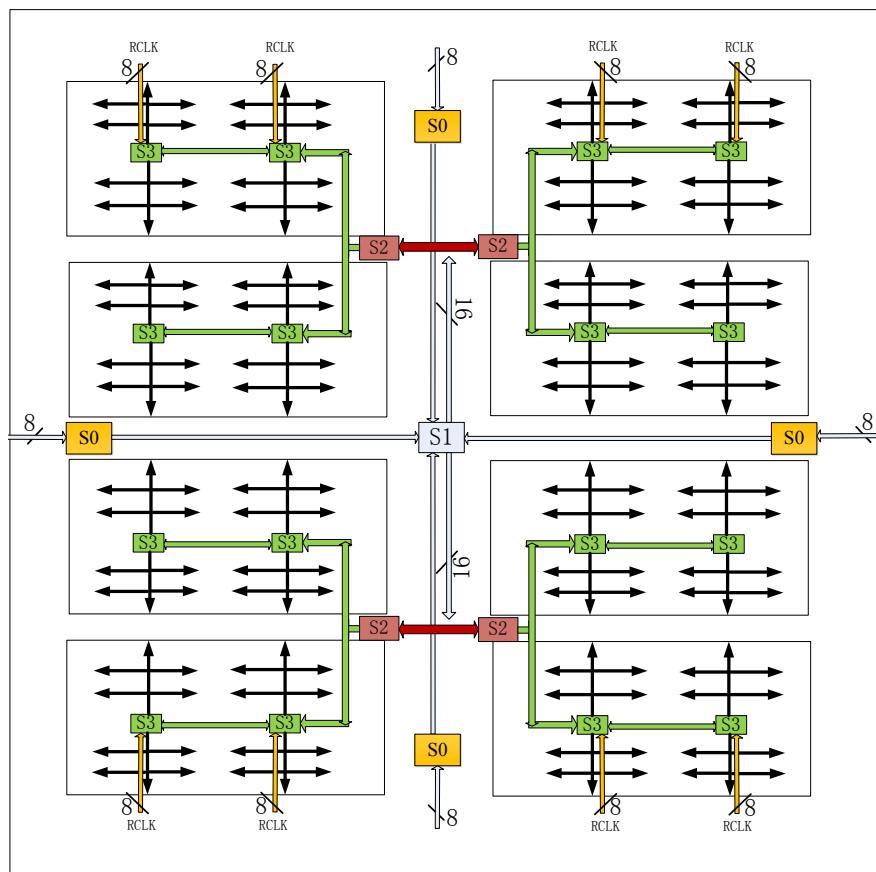


图 10 局域时钟网络分布示意图

4 输入输出模块

4.1 IOBLK

IOBLK 由 SwitchBox、ConnectionBox、控制逻辑和 IOE 组成：SwitchBox 实现通道之间的信号跳转，并实现 I/O 输入信号连接至通道；ConnectionBox 实现通道与 IOE 的控制逻辑之间的信号连接；控制逻辑为 IOE 产生各种控制信号；IOE 实现输入信号、输出信号以及使能信号的组合/时序模式、正向/反向模式。

IOE 分布在芯片四周的 IOBLK(Input Output BLOCK)中，每个列 IOBLK 包含 6 个 IOE，每个行 IOBLK 包含 3 个 IOE。其中列 IOBLK 可以连接垂直通道，行 IOBLK 可以连接水平通道、垂直通道和直连通道。EQ6HL45 的输入信号可以通过 IOE 输入到逻辑阵列，逻辑阵列的输出信号可以通过 Control Logic 和 IOE 输出至 FPGA 的输出管脚，其信号传输路径如图 11 所示。

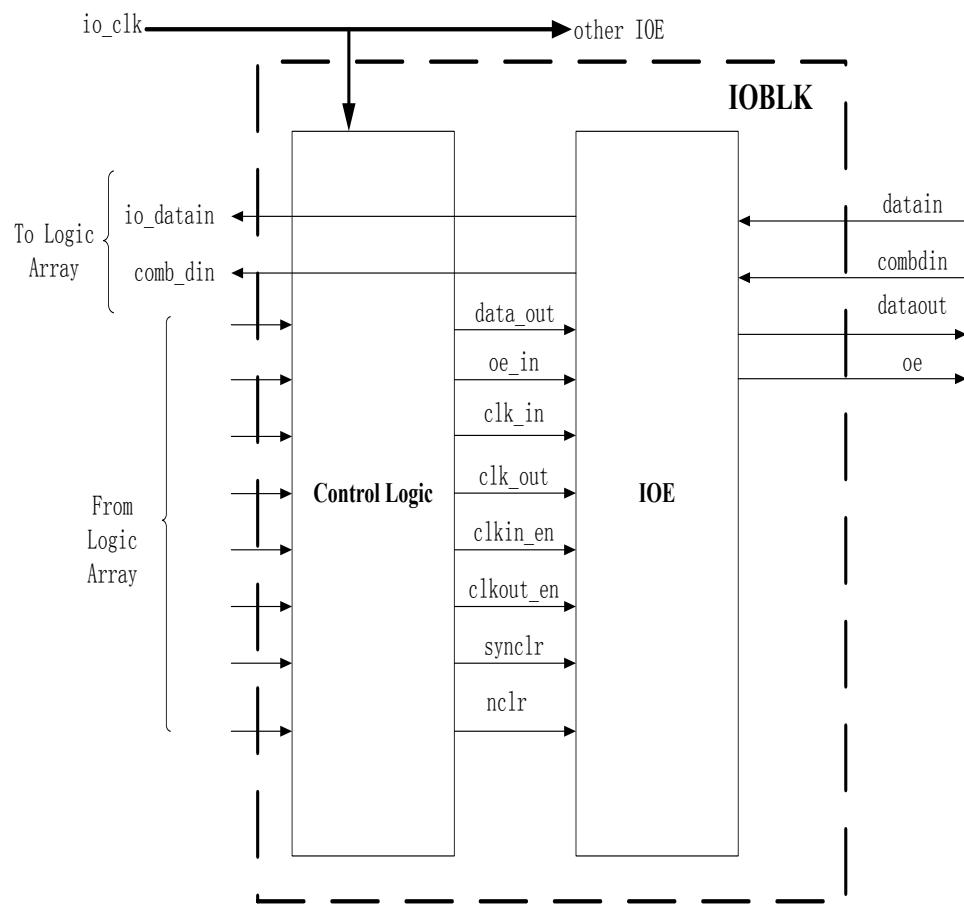


图 11 IOBLK 的信号传输路径示意图

输入输出模块 IOE (Input Output Element)为 EQ6HL45 的输入输出管脚与内部逻辑提供了可编程的接口，EQ6HL45 的 IOE 结构如图 12 所示。每个 IOE 包括一个输入寄存器，一个输出寄存器和一个输出使能寄存器，分别实现对输入数据、输出数据、输出使能信号的寄

存功能。

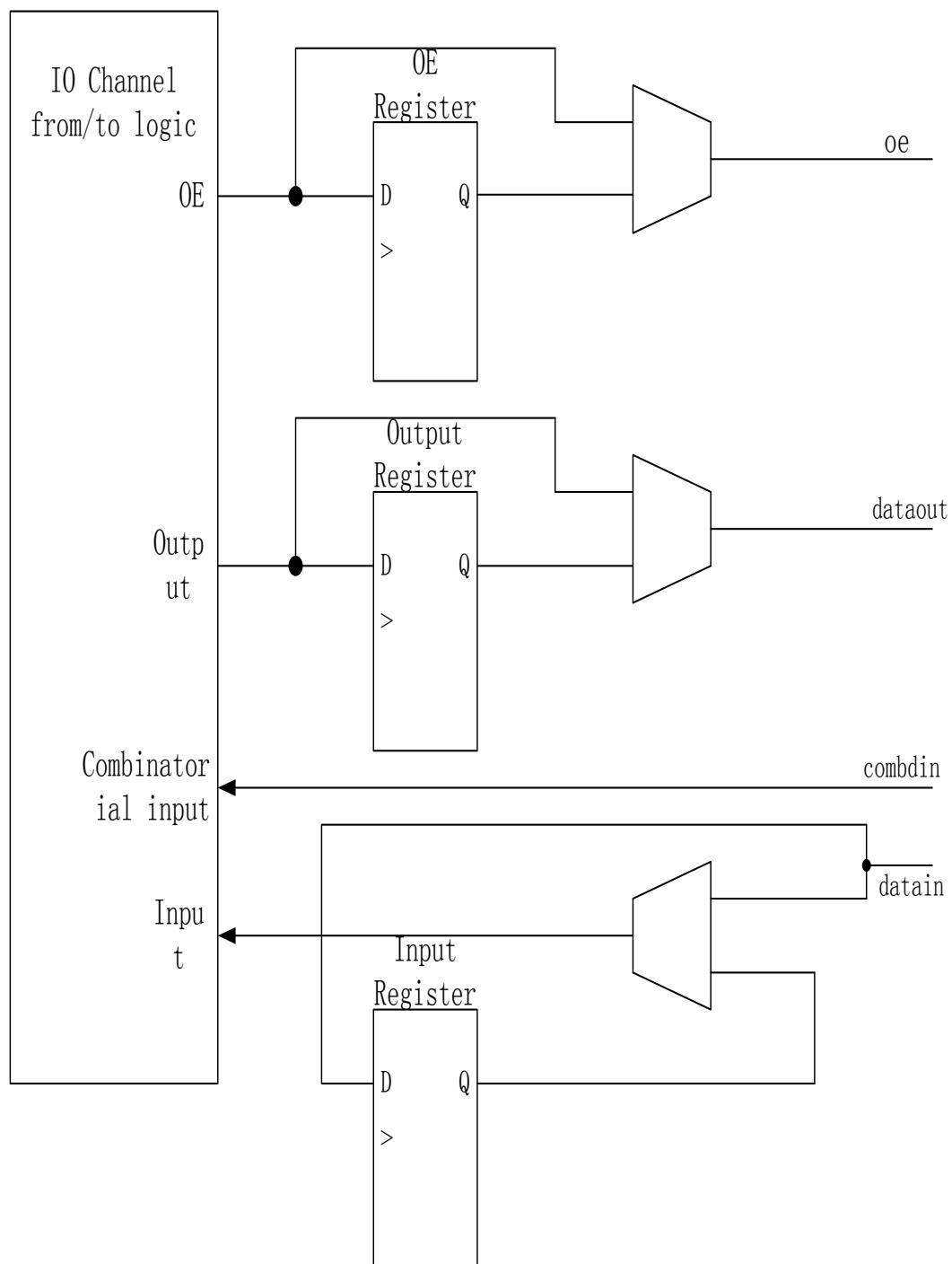


图 12 IOE 结构示意图

IOE 可以实现快速组合和普通的两种输入模式，其中普通输入包含正向模式、反向模式、组合模式和时序模式；对于输出信号，也可以实现正向模式、反向模式、组合模式和时序模式。在普通双向工作模式下，可以通过使用输入寄存器来满足输入信号的快速建立时间的要求。输入寄存器有它自己的独立于输出寄存器和输出使能寄存器的输入和时钟。输出寄存器可以用来满足快速 clock-output 的要求。输出使能寄存器和输出寄存器共用同样的时钟和时钟使能信号。所有寄存器还可以实现同步/异步的 PRSET/CLEAR 或无效 (none) 功能、上电

(POWERUP) 的低电平有效(0: enable)或高电平有效(1: enable)。

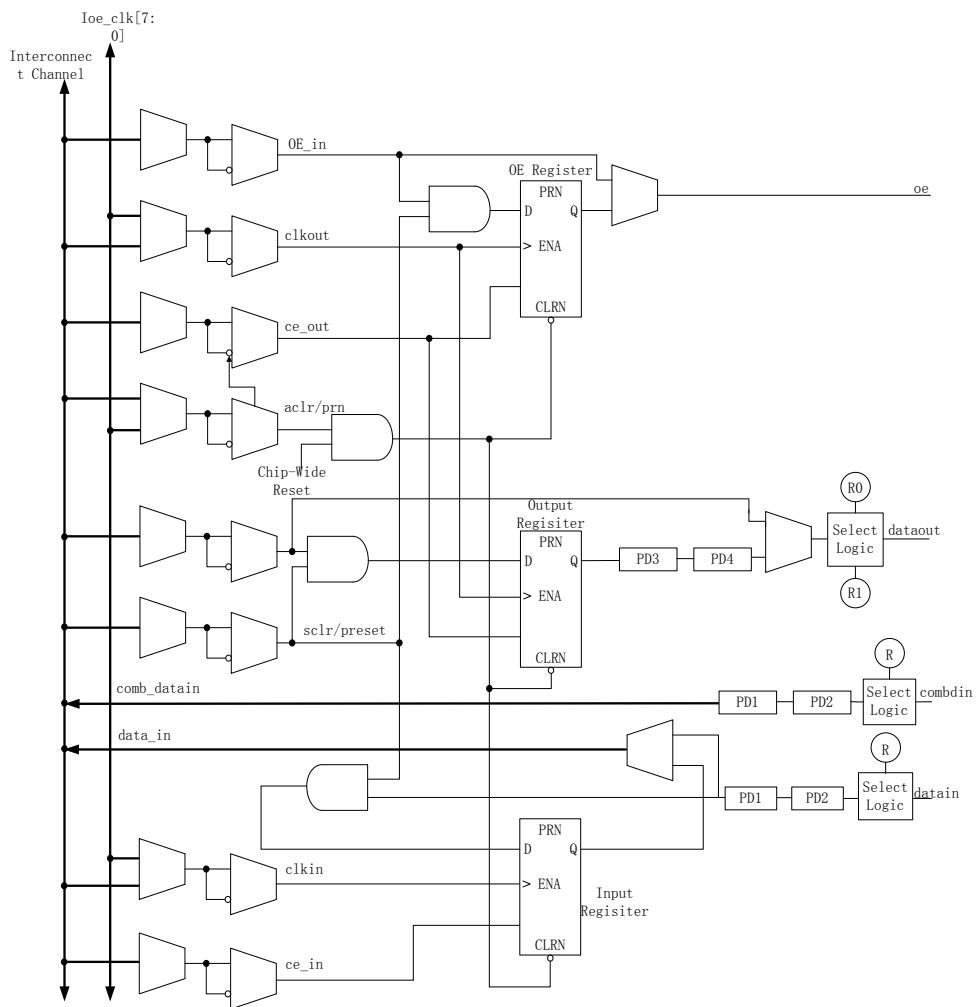


图 13 IOE 结构示意图

图 13 是双向模式下 IOE 的配置示意图，输入端具有可编程的延时单元 (PD1、PD2)，其作用包括：1) 用以时钟输入信号和数据输入信号的零保持时间；2) 为实现总线数据对齐而进行的延时调整；输出端具有可编程的延时单元 (PD3、PD4) 用以调节时钟沿到输出数据的延时。IOE 的输出部分还可以将时钟信号通过数据输出管脚送到芯片外面。

在 IOE 中有两条组合输入路径 (combdin、datain) 可以到达逻辑阵列，这两条路径的延时可分别进行调节，这样就可以调整输入管脚到芯核内不同位置寄存器的延时。控制逻辑为 IOE 产生输入时钟信号 (clk_in)、输出时钟信号 (clk_out)、输出使能信号 (oe)、输入时钟使能信号 (ce_in)、输出时钟使能信号 (ce_out)、异步清零信号 (aclr) 和同步清零信号 (sclr)，如图 14 所示。

IOE 的输入信号 combdin 和 datain 在配置阶段的电平状态不作要求，用户无需特殊处理，配置阶段传播到芯片内部的实际输入信号为固定电平。IOE 的输出信号在配置阶段对外表现为低电平，配置完成后由用户电路决定。EQ6HL45 的 IOE 个数较多，对于用户没有用到的 IOE，其输入状态通过配置固定，输出状态可以通过配置设置为高或低。

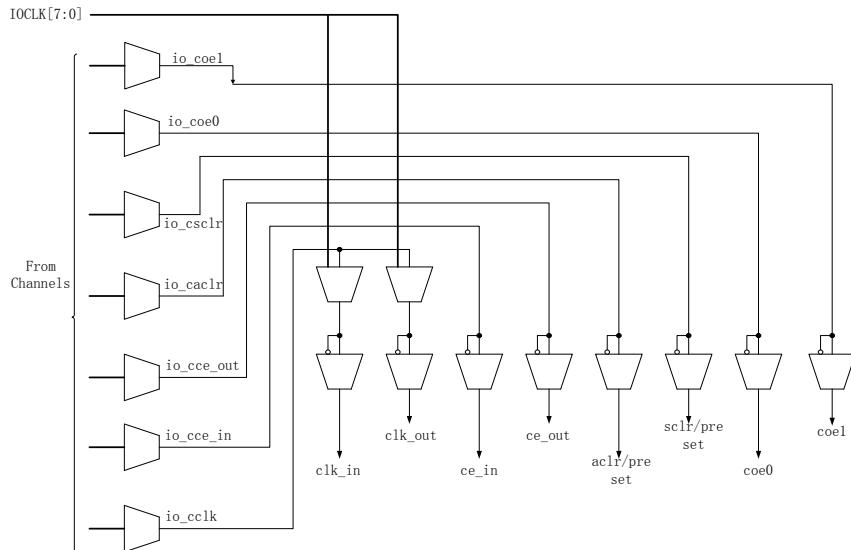


图 14 控制逻辑示意图

4.2 I/O 接口标准

表 3 EQ6HL45 支持的 I/O 标准

I/O 标准	类型	I/O 电源电压
3.3-V LVTTL/LVC MOS	单端	3.3V
2.5-VLVC MOS	单端	2.5V
1.8-V LVC MOS	单端	1.8V
1.5-V LVC MOS	单端	1.5V
SSTL_3	单端	3.3V
SSTL_2	单端	2.5V
SSTL_18	单端	1.8V
LVDS	差分	2.5V/3.3V

EQ6HL45 器件共有 4 个 I/O BANK，每个 I/O BANK 支持独立的 I/O 电源供电，允许用户在 I/O BANK 间使用不同的电平标准，从而提供了更大的灵活性。

4.3 可编程电流驱动

在 LVTTL 和 LVCMOS 标准下，EQ6HL45 器件的每个 USER I/O 都具有可编程电流驱动功能，用户可以通过软件对每个 USER I/O 的驱动电流大小进行配置。

表 4 EQ6HL45 可编程驱动电流

I/O 标准	I_{OH} 和 I_{OL} 驱动电流 (mA)
LVTTL/LVCMOS (3.3V)	2、4、6、8、12、16、24
LVCMOS (2.5V)	2、4、6、8、12、16
LVCMOS (1.8V)	2、4、6、8、12
LVCMOS (1.5V)	2、4、6、8、

4.4 可编程摆率控制

EQ6HL45 中 USER I/O 的输出缓冲器都具有可编程的输出摆率控制功能。用户可以通过软件对每个 USER I/O 的摆率进行独立的配置：当用户将 I/O 配置为高摆率输出时，I/O 将输出高速变化的信号，具有较小的上升及下降时间，满足高速信号传输的需求；当用户将 I/O 配置为低摆率输出时，I/O 输出的信号具有较大的上升及下降时间，能够降低同时开关噪声，但会明显增大传输延时。

4.5 可编程弱上拉

每个 USER I/O 在用户模式下都提供一个可选的弱上拉电阻：在被配置为输入且管脚悬空时，弱上拉电阻可以有效的将输出拉升至 VDDIO 的电平；在被配置为输入但有输入信号时，弱上拉电阻不影响其输入功能。可编程弱上拉电路原理图如图 15 所示，在芯片上电清零之后至配置过程完成之前这段时间内，弱上拉电路处于工作状态。

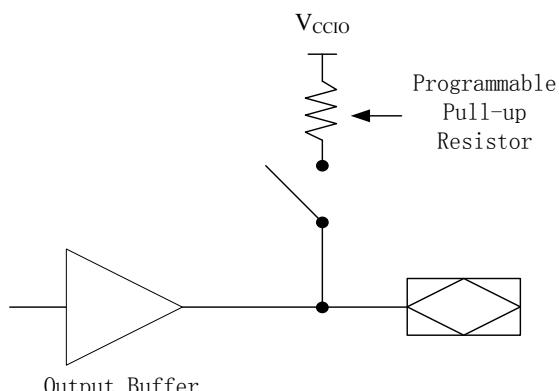


图 15 可编程弱上拉电路原理图

4.6 可编程开漏输出

EQ6HL45 的每个 USER I/O 都可以支持可选的漏极开路 OD (Open-Drain) 输出，为用户提供系统级的线与功能。用户可以通过软件对 USER I/O 的开路输出功能进行配置。

可编程开漏输出功能位于 Pre-Driver 逻辑控制模块，由 OD_EN 信号控制。如图 16 所示，输入信号 (IN)、三态控制信号 (OE) 和可编程开漏输出控制信号 (OD_EN) 经过逻辑运算后，转化为开漏控制信号 PMOS_C 和 NMOS_C，分别控制 PMOS 通路和 NMOS 通路。

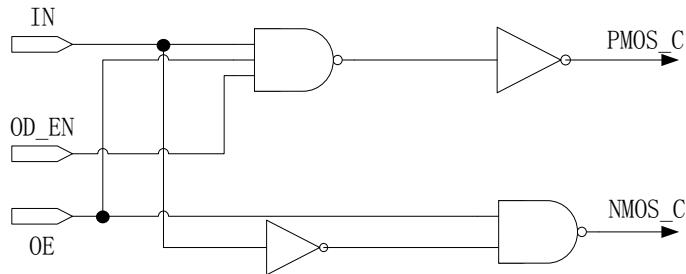


图 16 Pre-Driver 逻辑控制模块示意图

4.7 总线保持功能

每个 USER I/O 都具有独立可选的总线保持电路，用以保持 I/O 引脚最后驱动的状态。在配置时需注意，当 I/O 引脚被配置为差动输入/输出模式时，必须关闭总线保持电路，总线保持电路仅在 USER I/O 输入模式下有效，且和可编程弱上拉电路不能同时工作。

4.8 LVDS 功能

EQ6HL45 可以为用户提供支持 LVDS 电平标准的接口，可以方便的实现 LVDS 接收和发送的功能。

表 5 LVDS 输入/输出电平标准参数说明

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
VCCIO	I/O 供电电压	/	2.375	2.5	2.625	V
VOD	差动输出电平	RL=100Ω	250	350	550	mV
ΔVOD	差动输出电平范围	RL=100Ω	/	/	50	mV
VOS	共模输出电平	RL=100Ω	1.125	1.25	1.375	V
ΔVOS	共模输出电平范围	RL=100Ω	/	/	50	mV
VID	差动输入电平	0.1V≤VCM≤2.4V	100	/	650	mV
VIN	输入电平范围	/	0	/	2.5	V
VCM	共模输入电平	0.1V≤VID≤0.65V	100	/	2400	mV
RL	输入端差动负载匹配电阻	/	90	100	110	Ω

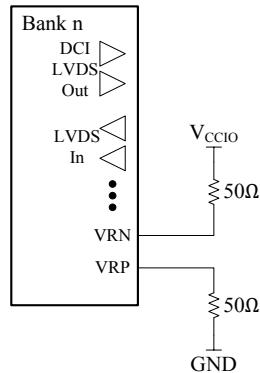


图 17 LVDS 输出使用 DCI 模式的 Bank 示意图

EQ6HL45 提供基于 DCI (Digital Control Impedance) 数字控制终端电阻的 LVDS 差分输出, 通过将所用 Bank 的 VRP 与 VRN 管脚分别接 50Ω 的下拉和上拉电阻(如图 17 所示), 并将 I/O 配置为 LVDS 输出 DCI 模式即可实现。

当 EQ6HL45 采用 LVDS 输出 DCI 模式发送数据时, 无需外接电阻网络, 可以直接接 50Ω 同轴电缆, 如图 18 所示。LVDS 输入所需的差分对之间的电阻也可以通过片内配置实现, 如图 19 所示。

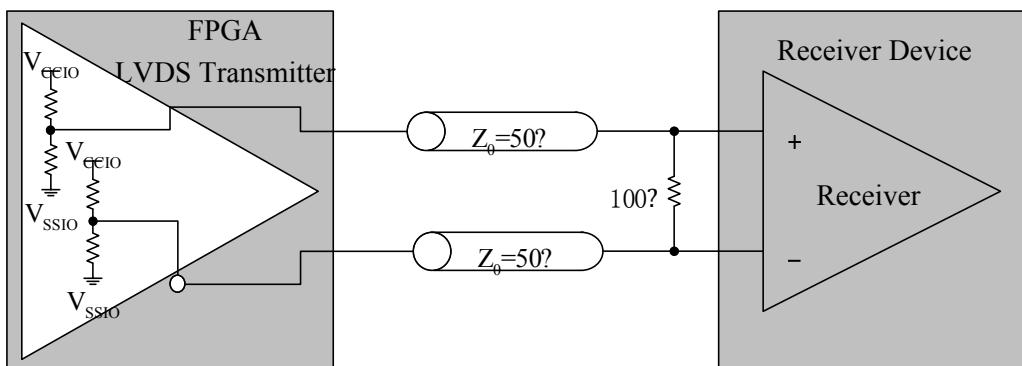


图 18 LVDS 输出使用 DCI 模式示意图

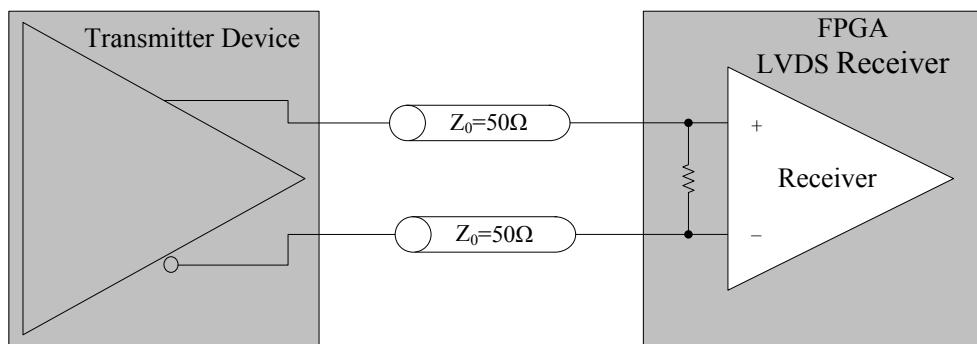


图 19 LVDS 输入可配置电阻示意图

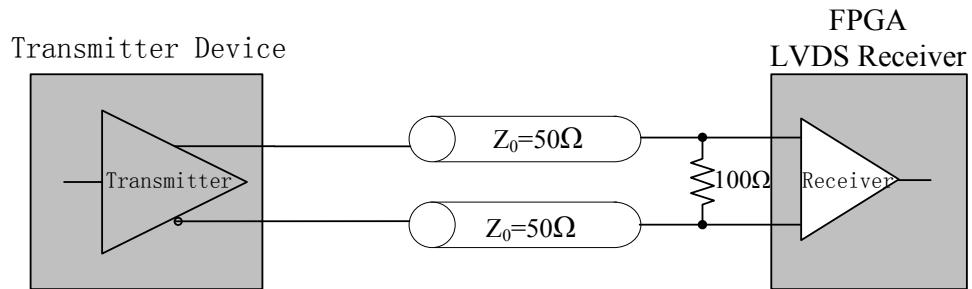


图 20 LVDS 输入示意图

此外，EQ6HL45 也可以支持采用外接电阻网络实现 LVDS 功能。如图 20 所示，当 EQ6HL45 采用 LVDS 标准接收数据时，外接 LVDS 发送器需使用 50Ω 同轴电缆，同时 EQ6HL45 LVDS 接收器的输入端需跨接 100Ω 的匹配电阻。

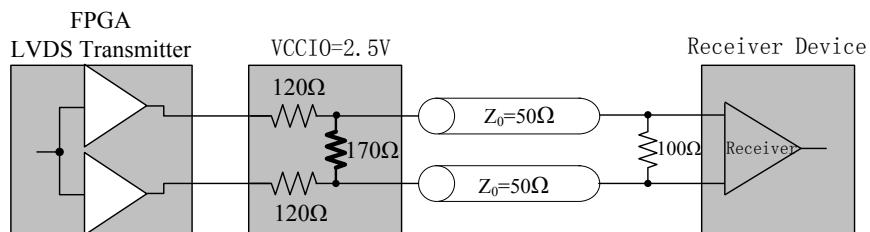


图 21 LVDS (VCCIO=2.5V) 输出示意图

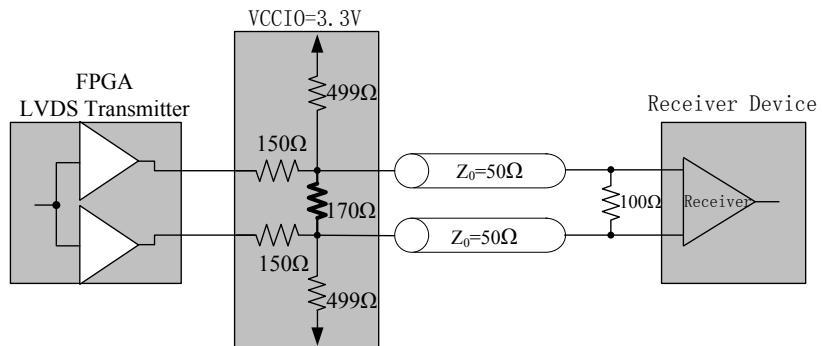


图 22 LVDS (VCCIO=3.3V) 输出示意图

当 EQ6HL45 采用 LVDS 标准发送数据时，推荐使用 50Ω 同轴电缆，同时通过外接的电阻网络进行分压和阻抗匹配。2.5V 和 3.3V 两种不同 VCCIO 电压下所要添加的电阻网络的匹配电阻值如图 21 和图 22 所示，并在 LVDS 接收器的输入端跨接 100Ω 的终端电阻。

5 嵌入式存储器（BRAM）

EQ6HL45 共包括 392 个可独立编程的 4.5Kbit 的嵌入式存储器 BRAM（Block RAM）。

支持的功能模式如下：

- (1) 每个 BRAM 的存储阵列的大小为 128×36 ;
- (2) 支持的端口模式： packed、单端口、简单双端口和真双端口；
- (3) 支持的读写冲突处理模式： read first 和 write first；
- (4) 支持的字宽模式： 1、2、4、8、9、16、18、32、36；
- (5) 支持的时钟模式： 单时钟、读写时钟、输入输出时钟和独立时钟；
- (6) 支持对输入数据的字节使能功能；
- (7) 支持对输入信号的清零功能；
- (8) 支持寄存器的上电清零功能；
- (9) 支持输出数据的同步和异步输出选择功能；
- (10) 支持初始化回读功能。

5.1 功能特性

表 6 BRAM 功能特性

特性	说明
性能	250MHz
存储容量（包含奇偶校验位）	4,608
配置模式 (深度 × 宽度)	4K×1 2K×2 1K×4 512×8 512×9 256×16 256×18 128×32 128×36
校验位	支持
字节使能	支持
单端口存储器	支持
简单双端口存储器	支持
真双端口存储器	支持
嵌入式移位寄存器	支持
ROM	支持
FIFO 缓冲器	支持
简单双端口混合宽度	支持
真双端口混合宽度	支持
存储器初始化文件 (.mif)	支持
混合时钟模式	支持

特性	说明
上电状态	寄存输出端被清零
寄存器清零	当清零信号加到寄存器时，寄存器被清零
同端口“写期间读”	输出设置为新数据
混合端口“写期间读”	输出设置为旧数据

5.2 工作模式

5.2.1 Packed 模式

两个单端口模式下且容量不超过 2K 比特的存储器可配置成一个 BRAM 的模式称为 Packed 模式。在大量使用小容量存储器的应用中，Packed 模式可以显著提高存储资源的利用率。Packed 模式的存储空间映射关系如图 23 所示，A 端口使用 BRAM 的上半个存储空间，B 端口使用 BRAM 的下半个存储空间。

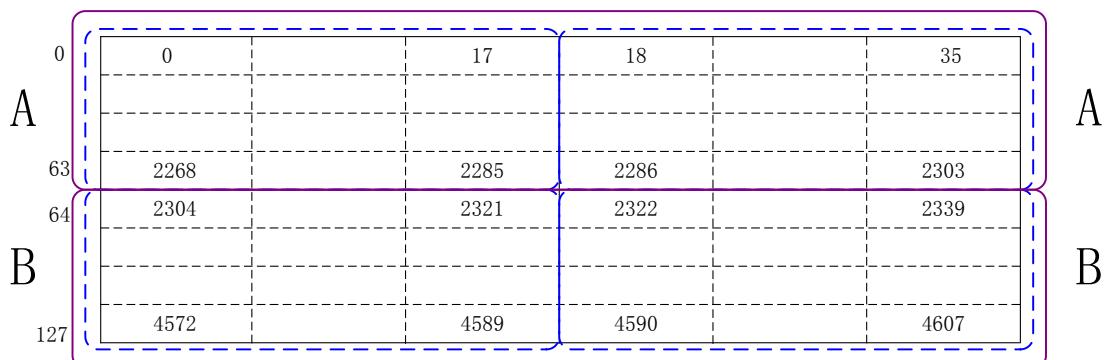


图 23 Packed 模式的存储空间映射关系图

Packed 模式下，一个 BRAM 可以用作两个独立的单端口存储器，每个独立的存储器最大支持 18 位的数据宽度，其端口如图 24 所示。

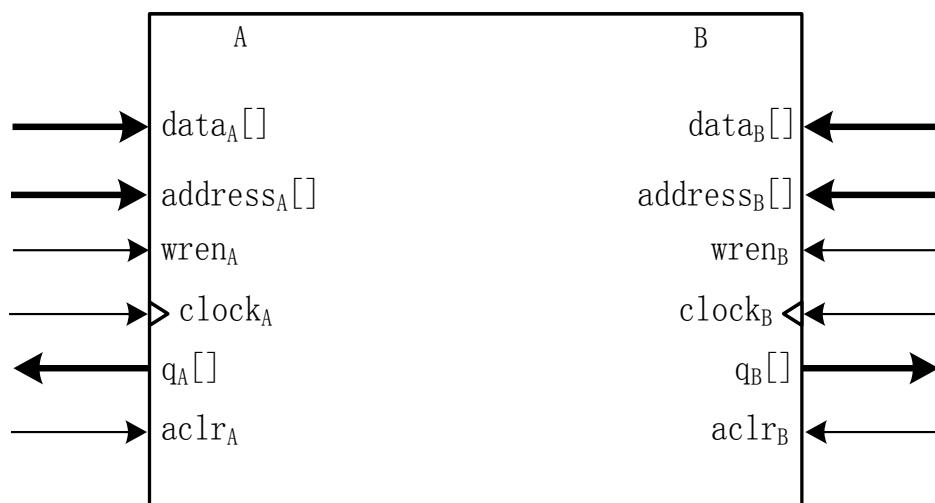


图 24 Packed 模式的端口示意图

Packed 模式下 BRAM 端口宽度配置如表 7 所示。

表 7 Packed 模式下的端口宽度配置

端口宽度	支持情况
$4K \times 1$	√
$2K \times 2$	√
$1K \times 4$	√
512×8	√
512×9	√
256×16	√
256×18	√

Packed 模式下的读写时序关系如图 25 所示。

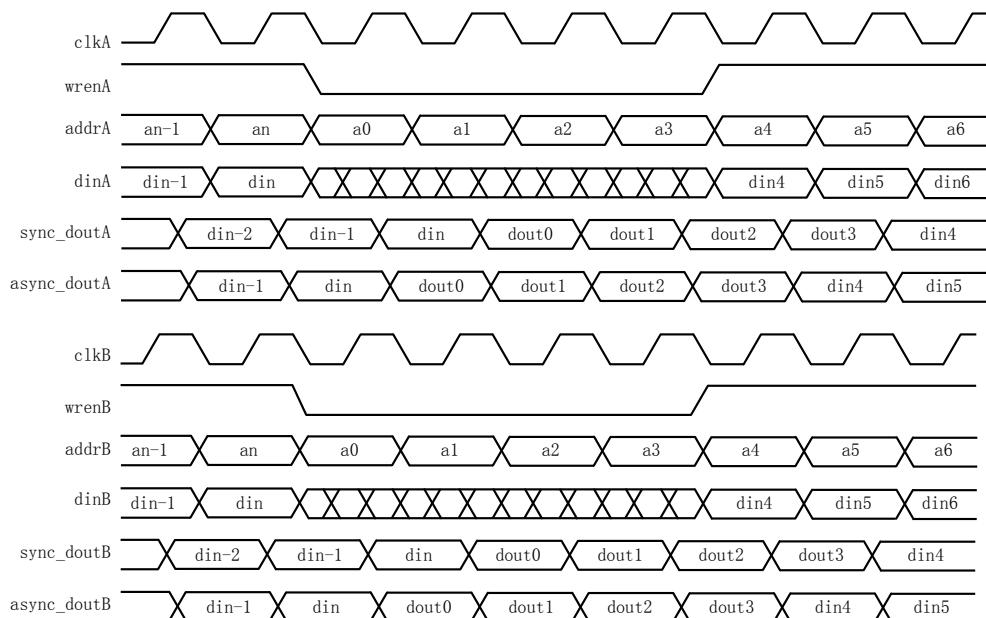


图 25 Packed 模式下的读写时序关系图

5.2.2 单端口模式

单端口模式允许在同一时间进行读或写的操作，在单端口工作模式下，其端口如图 26 所示。

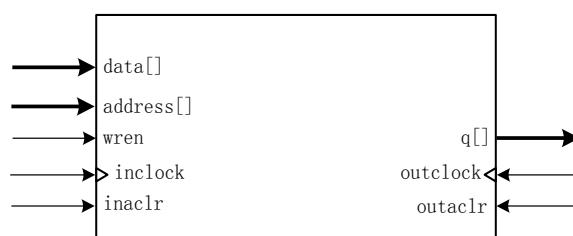


图 26 单端口模式的端口示意图

单端口模式下 BRAM 端口宽度配置如表 8 所示。

表 8 单端口模式下的端口宽度配置

端口宽度	支持情况
4K×1	√
2K×2	√
1K×4	√
512×8	√
512×9	√
256×16	√
256×18	√
128×32	√
128×36	√

单端口模式下的读写时序关系如图 27 所示。

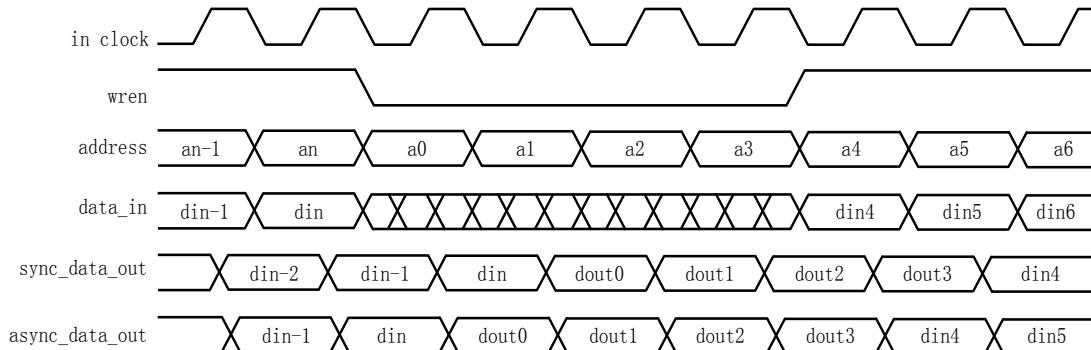


图 27 单端口模式下的读写时序关系图

5.2.3 简单双端口模式

简单双端口模式允许在同一时间对一个端口执行写操作，另一个端口执行读操作，其端口如图 28 所示。

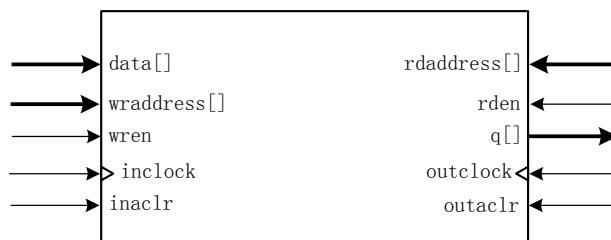


图 28 简单双端口模式的端口示意图

简单双端口模式下 BRAM 端口宽度配置如表 9 所示。

表 9 简单双端口模式下的端口宽度配置

读端口	写端口								
	4K×1	2K×2	1K×4	512×8	256×16	128×32	512×9	256×18	128×36
4K×1	√	√	√	√	√	√	/	/	/
2K×2	√	√	√	√	√	√	/	/	/
1K×4	√	√	√	√	√	√	/	/	/
512×8	√	√	√	√	√	√	/	/	/
256×16	√	√	√	√	√	√	/	/	/
128×32	√	√	√	√	√	√	/	/	/
512×9	/	/	/	/	/	/	√	√	√
256×18	/	/	/	/	/	/	√	√	√
128×36	/	/	/	/	/	/	√	√	√

简单双端口模式下的读写时序关系如图 29 所示。

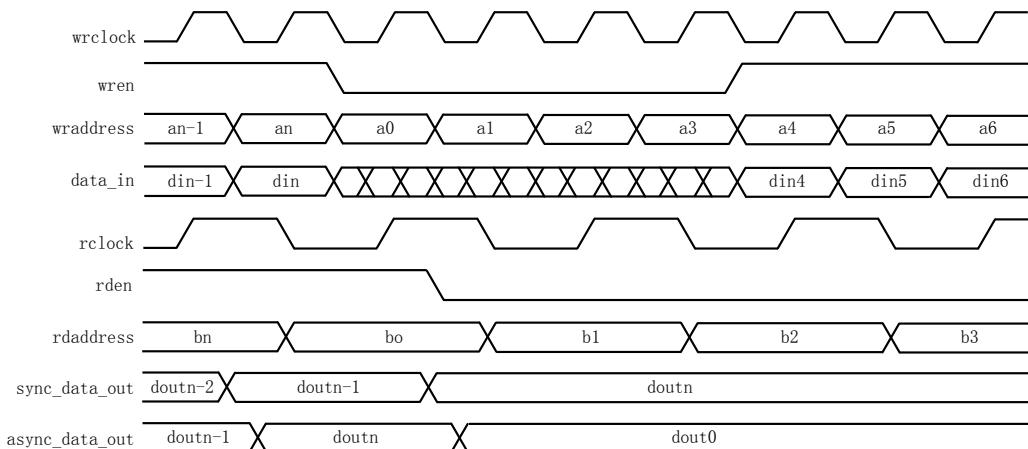


图 29 简单双端口模式下的读写时序关系图

5.2.4 真双端口模式

在真双端口模式下，AB 端口都可以独立对存储空间进行读写操作，与 Packed 模式不同之处在于其存储空间是共用的，其端口如图 30 所示。

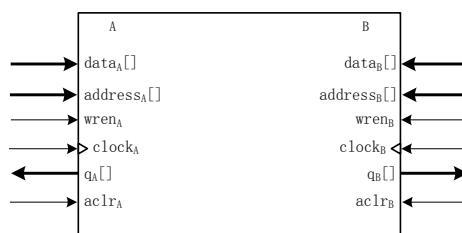


图 30 真双端口模式的端口示意图

真双端口模式下 BRAM 端口宽度配置如表 10 所示。

表 10 真双端口模式下的端口宽度配置

读端口	写端口						
	4K×1	2K×2	1K×4	512×8	256×16	512×9	256×18
4K×1	√	√	√	√	√	/	/
2K×2	√	√	√	√	√	/	/
1K×4	√	√	√	√	√	/	/
512×8	√	√	√	√	√	/	/
256×16	√	√	√	√	√	/	/
512×9	/	/	/	/	/	√	√
256×18	/	/	/	/	/	√	√

真双端口模式下的读写时序关系如图 31 所示。

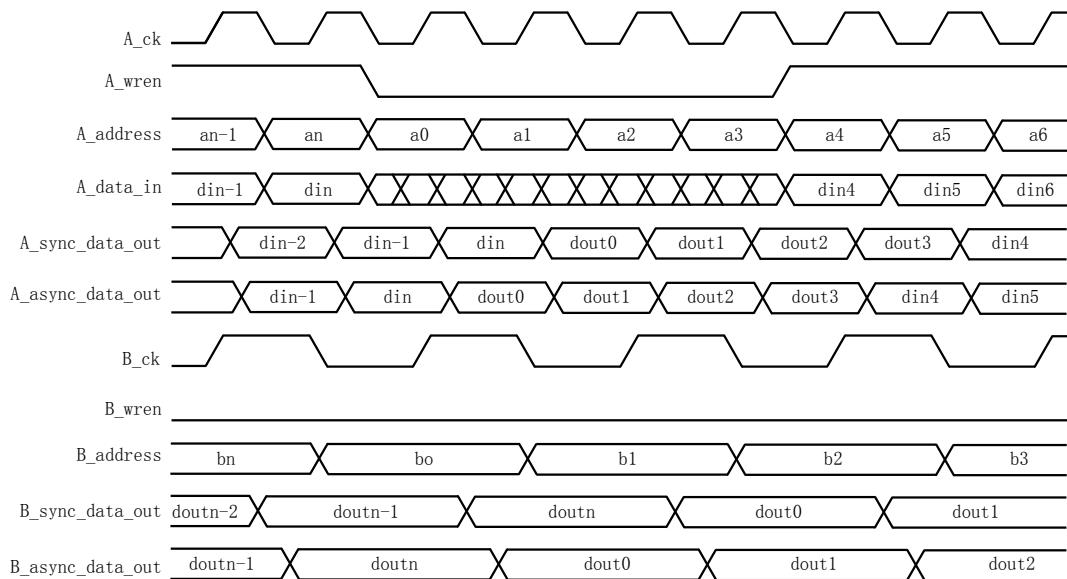


图 31 真双端口模式下的读写时序关系图

5.2.5 移位寄存器模式

BRAM 支持移位寄存器模式，格式为 $(w \times m \times n)$ ，输入数据宽度为 w ，抽头长度为 m ，抽头数为 n ，也可以通过级联生成更大规模的移位寄存器，其配置如图 32 所示。

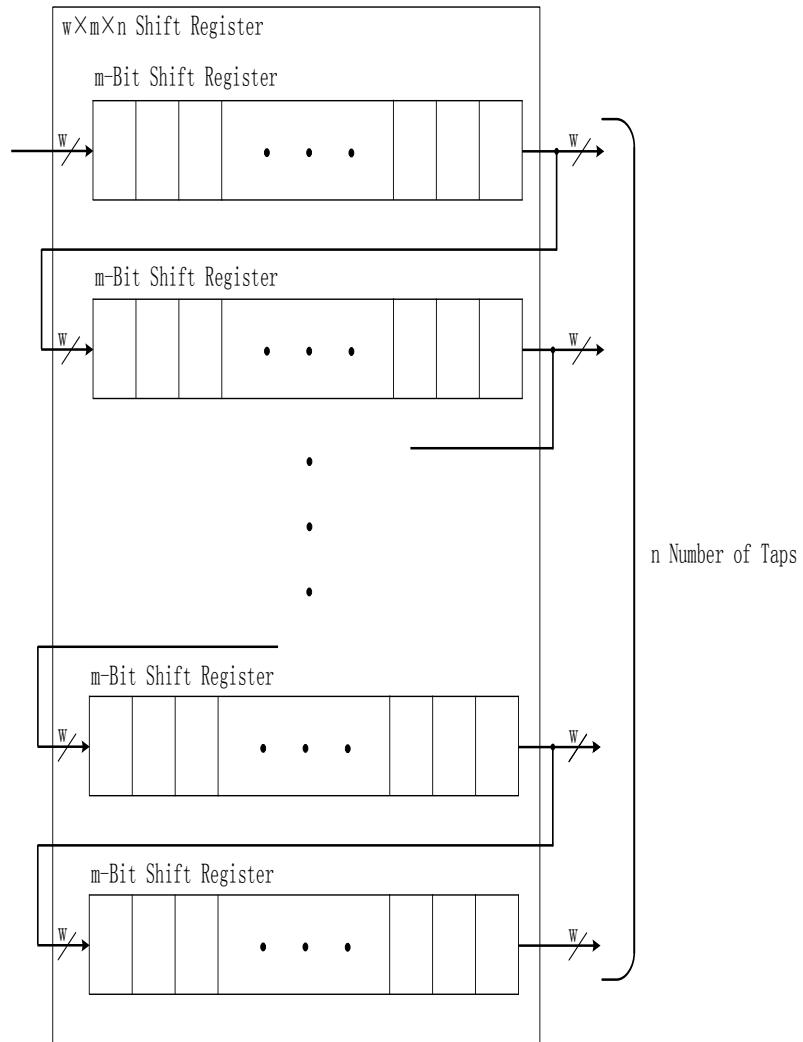


图 32 移位寄存器配置示意图

5.2.6 ROM 模式

BRAM 支持 ROM 模式，使用.mif 文件对 ROM 进行初始化配置。输入地址寄存，输出数据可以选择寄存或直接输出，输出寄存器可以异步清零。ROM 的读操作等同于 RAM 的单端口模式的读操作。

5.2.7 FIFO 模式

可利用软核通过 BRAM 实现 FIFO 模式，支持单时钟和双时钟的 FIFO 缓冲器，不支持混合字宽模式，原理如图 33 所示。

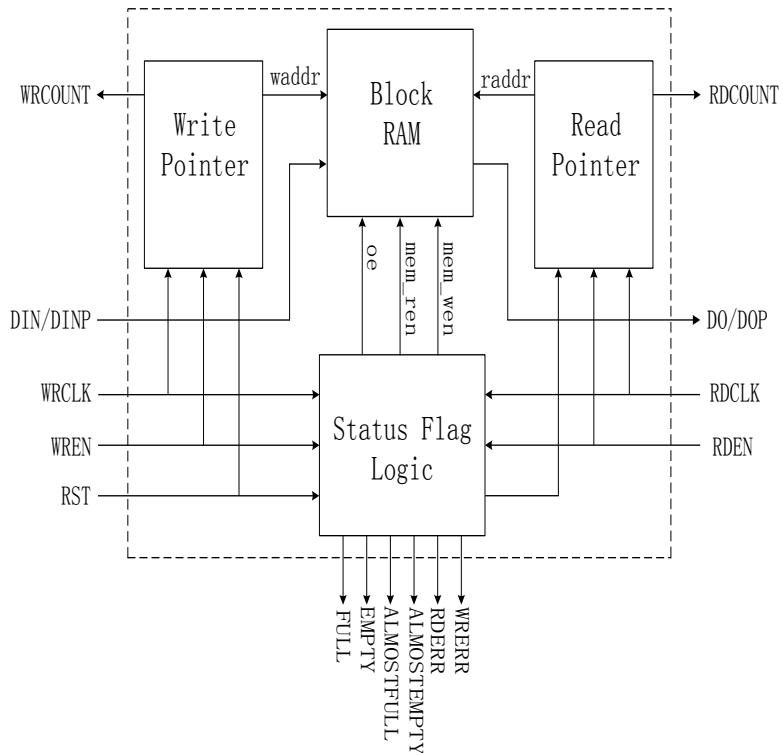


图 33 FIFO 模式实现原理示意图

5.3 时钟模式

BRAM 单元提供四种时钟模式：独立时钟模式、输入/输出时钟模式、读/写时钟模式和单时钟模式。BRAM 在不同工作模式下的所支持的时钟模式如表 11 所示。

表 11 BRAM 在不同工作模式下支持的时钟模式

	Packed 模式	单端口模式	简单双端口模式	真双端口模式
独立时钟模式	/	/	/	√
输入/输出时钟模式	/	√	√	√
读写时钟模式	/	/	√	/
单时钟模式	√	√	√	√

5.3.1 独立时钟模式

BRAM 支持真双端口工作模式下的独立时钟模式，如图 34 所示。在这种模式下，每个端口(A 和 B)支持独立的时钟，时钟 A 控制所有 A 端口上的寄存器，时钟 B 控制所有 B 端

口上的寄存器。每个时钟支持独立的时钟使能。该模式下的寄存器支持异步清零功能。

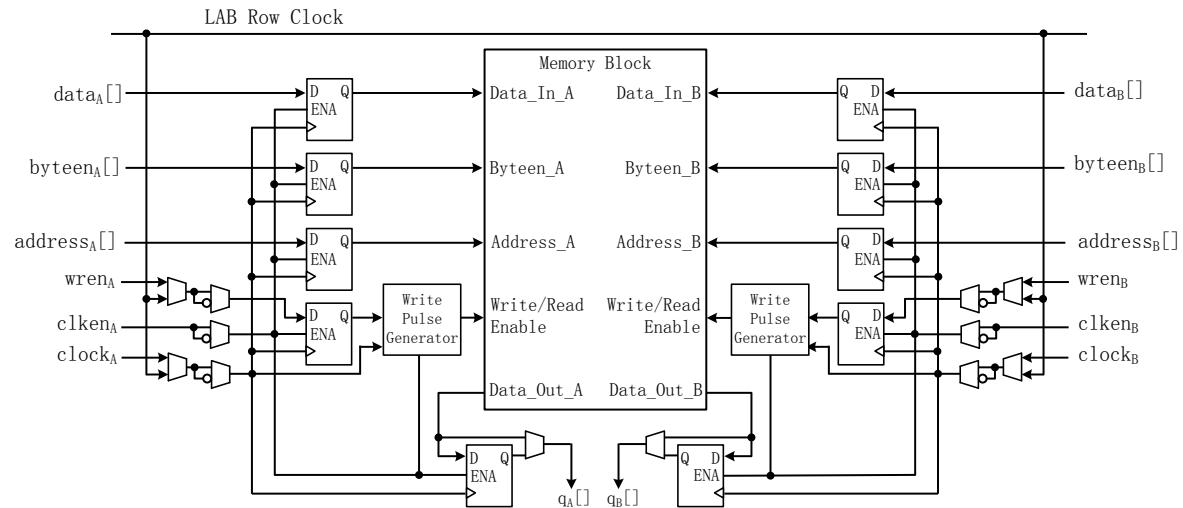


图 34 独立时钟模式

5.3.2 输入/输出时钟模式

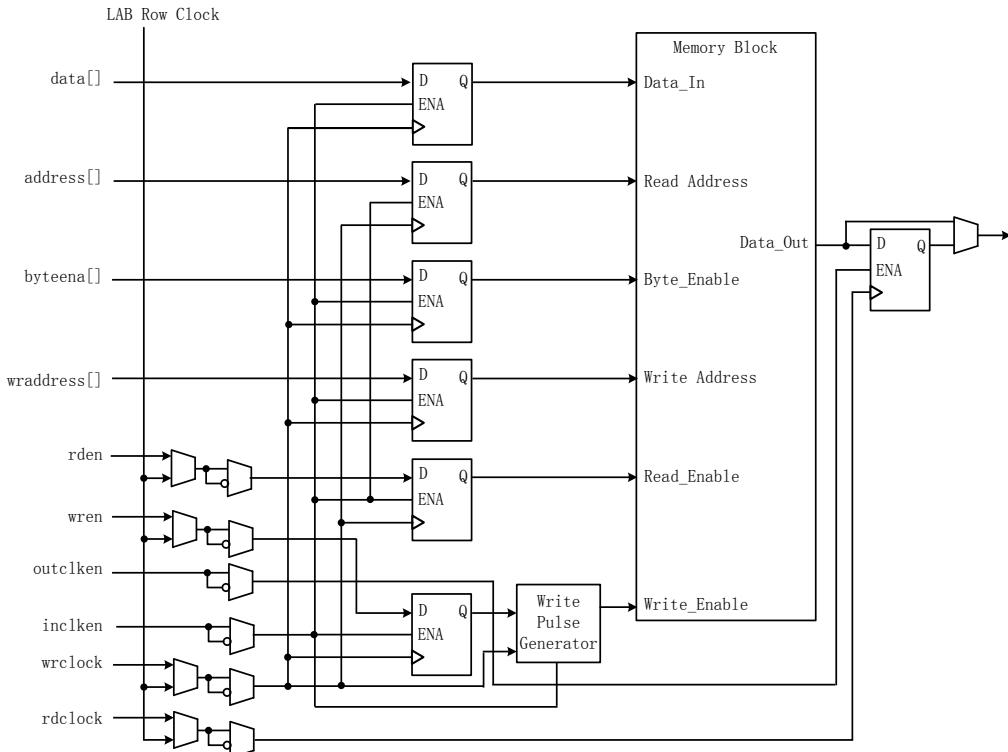


图 35 简单双端口输入输出时钟模式

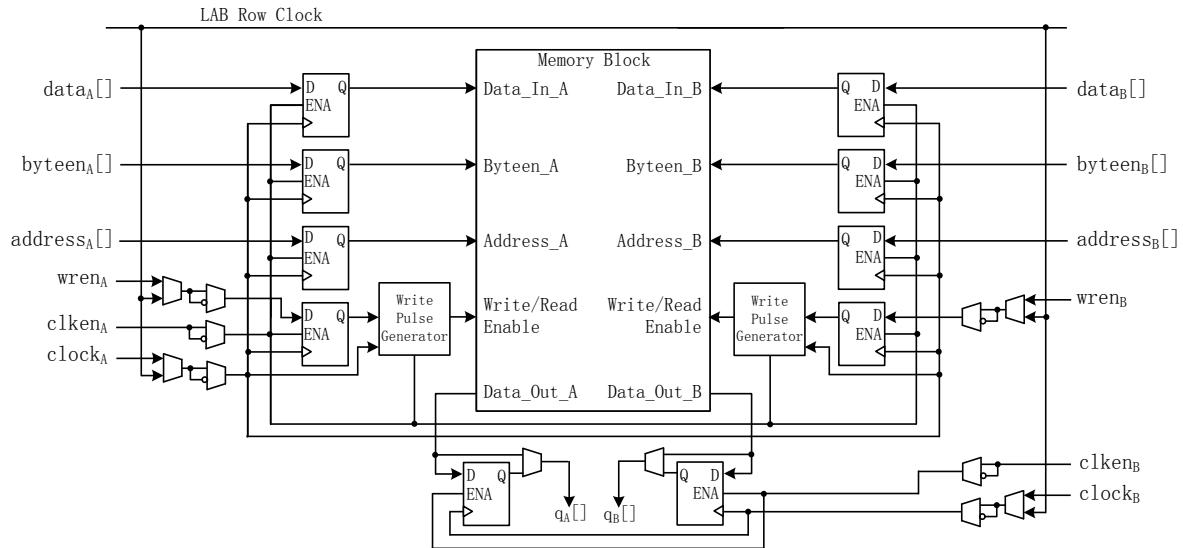


图 36 真双端口输入输出时钟模式

BRAM 支持单端口、简单双端口和真双端口工作模式下的输入/输出时钟模式，如图 35 和图 36 所示。在这种模式下，一个时钟控制所有端口的输入信号（数据输入、写使能和地址输入）的寄存器，另一个时钟控制所有端口的数据输出信号的寄存器。每个时钟支持独立的时钟使能。该模式下的寄存器支持异步清零功能。

5.3.3 读写时钟模式

BRAM 支持简单双端口模式下的读写时钟模式，如图 37 所示。在这种模式下，写时钟控制写操作相关信号（数据输入，写地址和写使能信号）的寄存器，读时钟控制读操作相关信号（数据输出、读地址和读使能信号）的寄存器。每个时钟支持独立的时钟使能。该模式下的寄存器支持异步清零功能。

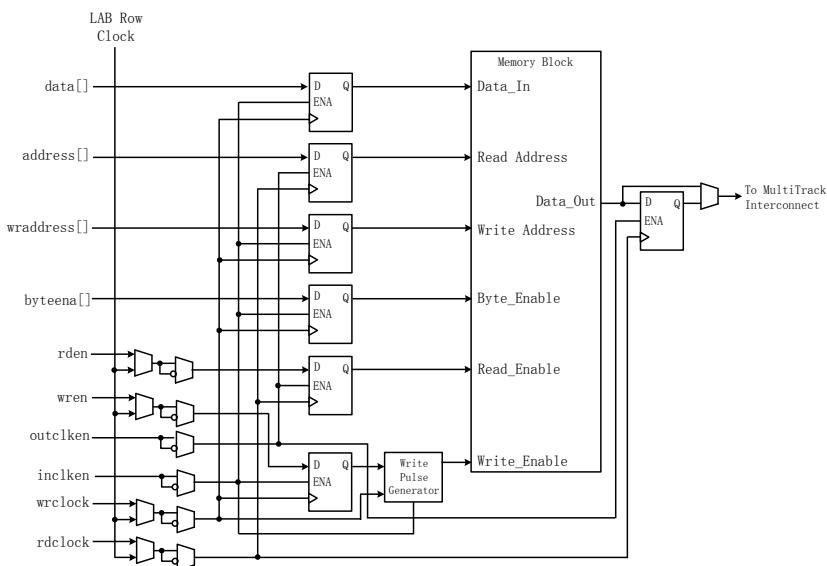


图 37 简单双端口下的读写时钟模式

5.3.4 单时钟模式

BRAM 在 Packed 模式、单端口模式、简单双端口和真双端口工作模式下支持单时钟模式，在单端口模式下，时钟对寄存器的控制如图 38 所示。在这种模式下，不需要进行同时的读和写操作。该模式下的寄存器支持异步清零功能。

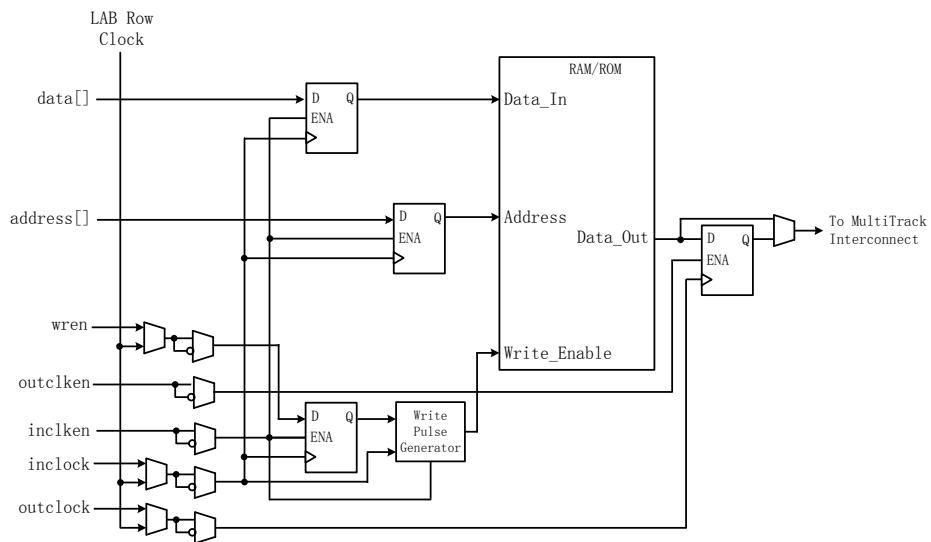


图 38 单端口单时钟模式

5.4 读写冲突处理

5.4.1 同一端口读写冲突处理

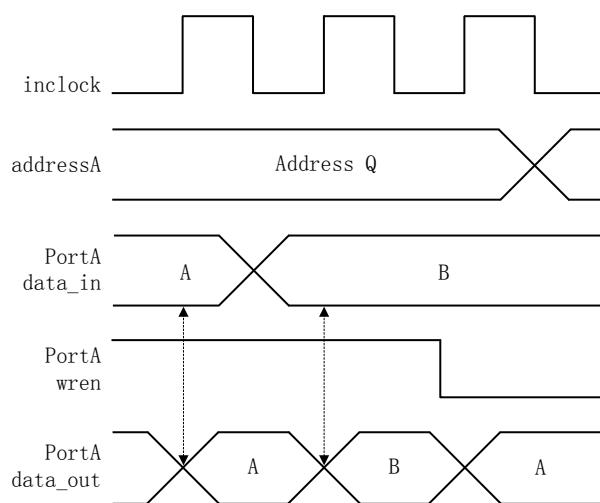


图 39 同一端口读写冲突处理模式时序图

同一端口读写冲突处理模式仅应用在单端口或真双端口模式，在此模式下提供新数据输出。

出（数据随出）功能。新数据在写周期的时钟上升沿在读端口有效，其时序如图 39 所示。

5.4.2 混合端口读写冲突处理

混合端口读写冲突处理模式仅应用在简单双端口和真双端口模式，两个端口在同一时钟周期对相同地址进行读写操作时，读端口输出 BRAM 该地址存储的原有数据，写端口写入新数据，其时序关系如图 40 所示。

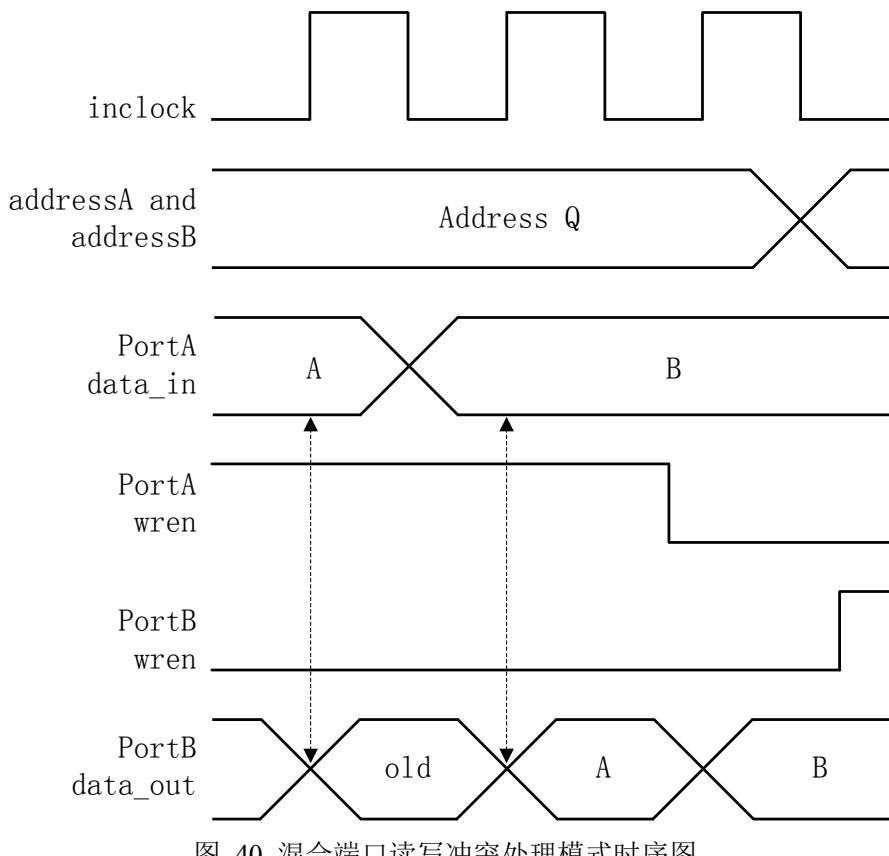


图 40 混合端口读写冲突处理模式时序图

5.4.3 读写冲突处理机制的适用范围

当两个不同的时钟应用于简单双端口和真双端口模式时，不支持混合端口读写冲突处理机制。

5.5 上电情形和存储器初始化

在上电时，复位信号对输出寄存器进行异步清零，在配置期间数据的输出端口为低电平。

5.6 字节使能功能

在 8/9 比特、16/18 比特、32/36 比特位宽模式下支持字节使能功能，可以对用户输入数

据的一个或多个字节进行写入屏蔽，未屏蔽的字节会被写入存储空间，其对应关系如表 12 所示。

表 12 字节使能与相应字节的对应关系

packed 模式、单端口（使用 A 端口）、简单双端口（A 写 B 读）、真双端口						
字节使能	字宽					
	× 8	× 9	× 16	× 18	× 32	× 36
[0] = 1	dinAi[7:0]	dinAi[8:0]	dinAi[7:0]	dinAi[8:0]	dinAi[7:0]	dinAi[8:0]
[1] = 1	/	/	dinAi[15:8]	dinAi[17:9]	dinAi[15:8]	dinAi[17:9]
[2] = 1	/	/	/	/	dinBi[7:0]	dinBi[8:0]
[3] = 1	/	/	/	/	dinBi[15:8]	dinBi[17:9]
单端口（使用 B 端口）、简单双端口（B 写 A 读）						
字节使能	字宽					
	× 8	× 9	× 16	× 18	× 32	× 36
[0] = 1	dinBi[7:0]	dinBi[8:0]	dinBi[7:0]	dinBi[8:0]	dinBi[7:0]	dinBi[8:0]
[1] = 1	/	/	dinBi[15:8]	dinBi[17:9]	dinBi[15:8]	dinBi[17:9]
[2] = 1	/	/	/	/	dinAi[7:0]	dinAi[8:0]
[3] = 1	/	/	/	/	dinAi[15:8]	dinAi[17:9]

6 嵌入式乘法器 (DSP)

6.1 DSP 功能概述

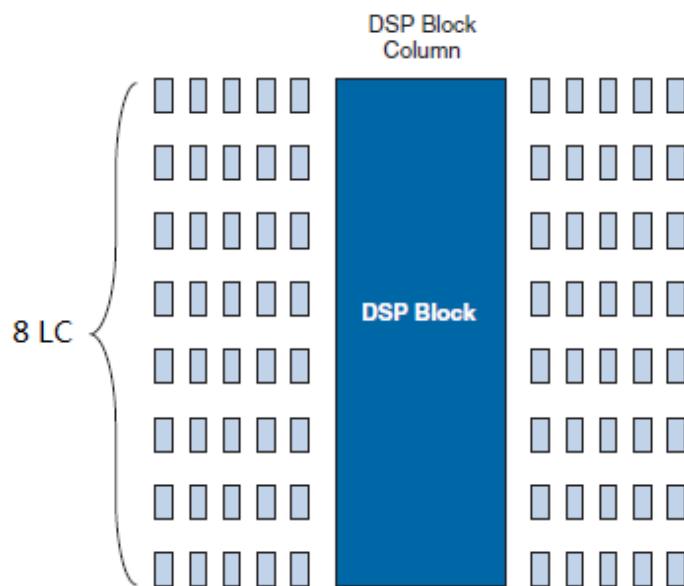


图 41 DSP 模块分布示意图

DSP 模块主要用于高速数字信号处理，可以实现 FIR、IIR、FFT、DCT 等算法，这些算法都包含乘加、乘累加等基本运算。DSP 模块支持下列五种算术运算：乘法、加法、减法、累加、求和，通过这几种运算可以满足大多数的数字信号处理功能需求。

一个 DSP 模块可以同时完成：

- 8 个 9b×9b 的乘法
- 4 个 18b×18b 的乘法
- 1 个 36b×36b 的乘法

并且乘法结果可以进行加减、累加和求和等运算。

一个 DSP 模块占驻 8 个 LC 的高度，共有 2 列 DSP，每列包括 7 个 DSP 模块，共 14 个 DSP 模块。

6.2 DSP 结构

一个 DSP 模块由下列几个基本单元组成：

- 乘法单元
- 加法、减法、累加单元
- 求和单元
- 输出单元
- 输出寄存器

6.2.1 乘法单元

乘法单元包括输入寄存器、乘法器和流水寄存器（pipeline）。

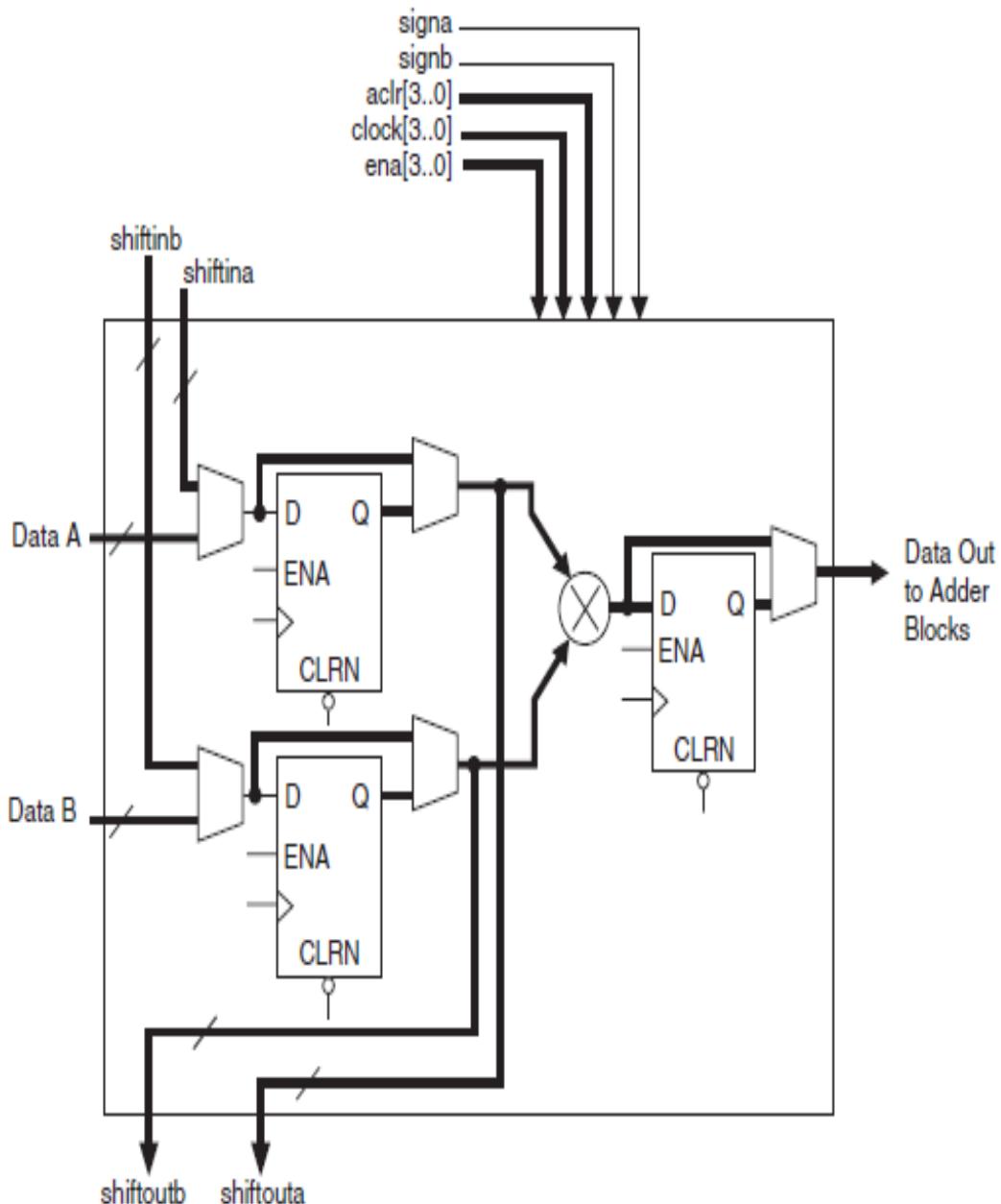


图 42 乘法单元结构示意图

输入的操作数可以通过输入寄存器锁存一拍也可以直接送给乘法器。DSP 模块有下列几组控制信号：clock[3:0]、ena[3:0]、aclr[3:0]。输入寄存器送给乘法器同时也可驱动两组专用移位输出链 shiftouta 和 shiftoutb。从一个乘法器的移位输出可以直接送给同一个 DSP 模块内（或下一个 DSP 模块）相邻的乘法器，从而形成一个寄存器链。移位寄存器很适合实现 FIR 功能，当实现 9b×9b 或 18b×18b 的乘法功能时，无需在 DSP 模块外部产生寄存器

链，可以大大节省 LE 和布线资源。乘法单元级联结构如下图 43 所示。

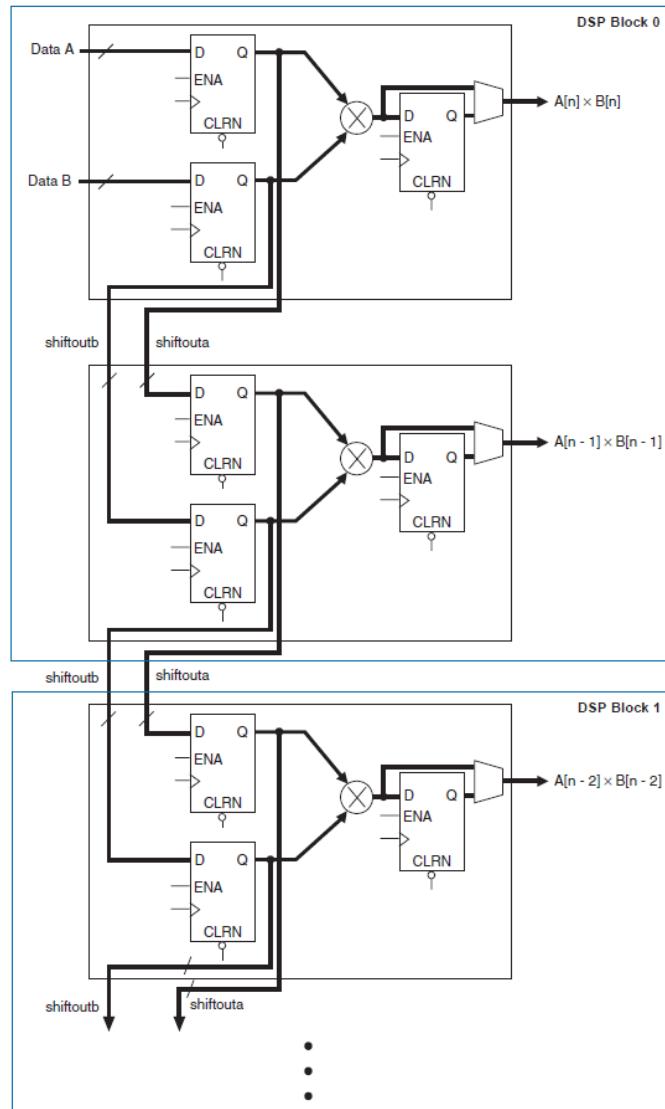


图 43 乘法单元级联结构图

乘法器可以实现 $9b \times 9b$ 、 $18b \times 18b$ 、 $36b \times 36b$ 位的乘法运算。根据位宽不同，一个 DSP 模块可以并行完成多个乘法运算。

乘法的操作数可以是无符号数和有符号数，signa 和 signb 分别用来表示两个操作数是否是有符号数，逻辑 1 表示有符号数，0 表示无符号数。

乘法器的输出结果可以送到流水线寄存器或直接旁路输出，流水线寄存器用以提高 DSP 的性能，很适合乘积后还需加减的运算。对于 $36b \times 36b$ 的乘法，流水线寄存器可以提高性能，而对于 $18b \times 18b$ 及以下的乘法器，流水线只能增加延迟时钟周期而不能提高性能，这是因为 $36b \times 36b$ 的最后一步和加操作是在加法模块里实现的。

6.2.2 加法、输出单元

(1) 输出单元由下列部分组成

1. 加法、减法、累加器

2. 求和部件

3. 输出选择部件

4. 输出寄存器

(2) 输出单元支持的配置模式

1. 单纯输出接口

2. 累加器

3. 一级加法

4. 两级加法 (第一级加法器可以作动态加减控制)

5. 36 位乘法器的最后一级

(3) 输出单元的结构图

输出单元的输出可以用输出寄存器锁存，输出单元不能脱离乘法单元独立使用。图 44 为输出单元的结构图。

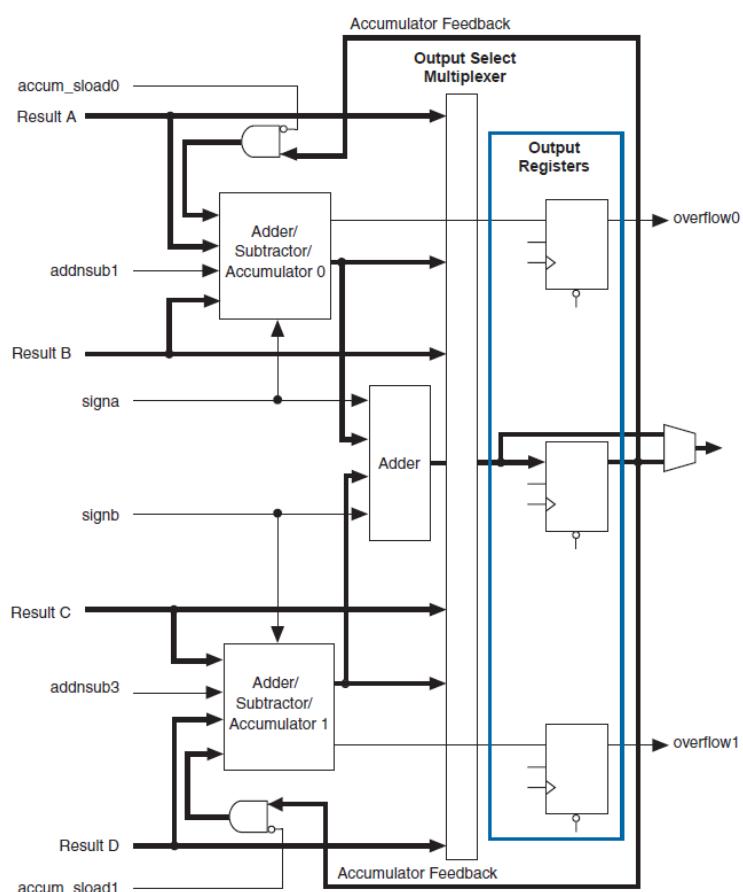


图 44 加法和输出单元结构图

加减累加器是输出单元的第一级。可以被配置成累加器或加法减法器。

1. 累加器

当配置成累加器时，输出单元的输出结果要反馈回累加器。可以通过 `accum_sload[1:0]` 信号来异步清除累加器的结果，通过这个清除过程可以不丢失时钟周期来实现累加的重新开始。

`overflow` 信号当累加器上溢或下溢时会在时钟沿变成高电平，在下一时钟沿有可能变成低电平，建议用户使用此信号时进行锁存。

2. 加/减法

addnsub 信号选择做加法还是做减法: 值为 1 时进行加法运算, 值为 0 时进行减法运算。
addnsub 信号来自外部逻辑, 可以动态控制每一时钟周期的加减运算。加减法部件所用的符号数控制信号 **signa**、**signb** 和乘法器的一致, 并且可以通过寄存器延迟 1~2 个时钟周期。

3. 求和部件

加减累加器的输出送到可选用的求和部件, 用以将加减法的结果进行求和, 此部件在 FIR 算法中很重要。

4. 输出选择部件

不同输出单元输出的数据通过输出选择部件进行选择后输出, 根据 DSP 的操作模式, 乘法单元、加减累加器、求和部件的输出可以直接旁路到输出端口。

5. 输出寄存器

输出寄存器可以对输出进行锁存, 该部件受 **clock[3:0]**、**aclr[3:0]**、**ena[3:0]** 等信号控制, 输出寄存器在任何模式下都可以使用, 还作为累加器的一个组成部分 (累加器必须是寄存输出)。

6.2.3 端口描述

表 13 DSP 端口

端口名称	位宽	功能描述
用户使用端口		
dataa, datab	[71:0]	乘法的乘数 A 和 B, 分别有 72 位。 1-9 位乘法: 可以分别接 8 个乘法的输入, 每个乘法以 9 位输入 10-18 位乘法: 可以分别接 4 个乘法的输入, 每个乘法以 18 位输入 19-36 位乘法: 由软件将输入分配到不同的 4 个 18 位乘法器的输入端
dataout	[143:0]	共 144 位 详见“不同模式不同位宽下的输入输出”列表
clock	[3:0]	四个时钟输入, 可通过时钟选择配置位应用于任何内部寄存器上
ena	[3:0]	时钟使能信号, 此信号与 clock 成对应用, ena 对 clock 进行门控后接到寄存器时钟端
aclr	[3:0]	寄存器的异步清零信号, 可通过选择配置位应用于任何内部寄存器上, 此信号和 clock 无关, 可以独立连接
signa, signb	1bit	符号数标识符 1: 有符号数; 0: 无符号数 作用于所有单元, 缺省接 1
addnsub	[1:0]	加减选择 1: 加法; 0: 减法 用于控制两个第一级加减单元的加减法, addnsub[1] 控制上面 (9 位模式下有两个) 加法单元, addnsub[0] 控制下面加法单元 (9 位模式下有两个), 缺省为 1。
accum_sload	[1:0]	对累加器的反馈数进行清零 1: 清零; 0: 普通累加, 缺省值为 0 accum_sload[0] 控制上面的加减累加单元 accum_sload[1] 控制下面的加减累加单元

6.3 DSP 工作模式

DSP 模块支持四种工作模式。

表 14 DSP 工作模式

操作模式	9 b×9b	18b×18b	36b×36b
简单乘法 (Simple Multiply)	可以实现 8 个乘法和 8 组输出	可以实现 4 个乘法和 4 组输出	可以实现 1 个乘法
乘累加 (MAC, Accumulator)	可以实现 2 个 34 位结果的 MAC	可以实现两个 52 位结果的 MAC	—
双乘加 (Two Multiply- Adder)	可以实现 4 个双乘加	可以实现 2 个双乘加	—
四乘加 (Four Multiply- Adder)	可以实现 2 个四乘加	可以实现 1 个四乘加	—

6.3.1 简单乘法模式

简单乘法模式 (Simple Multiplier Mode) 模式下，DSP 模块实现单个乘法运算。

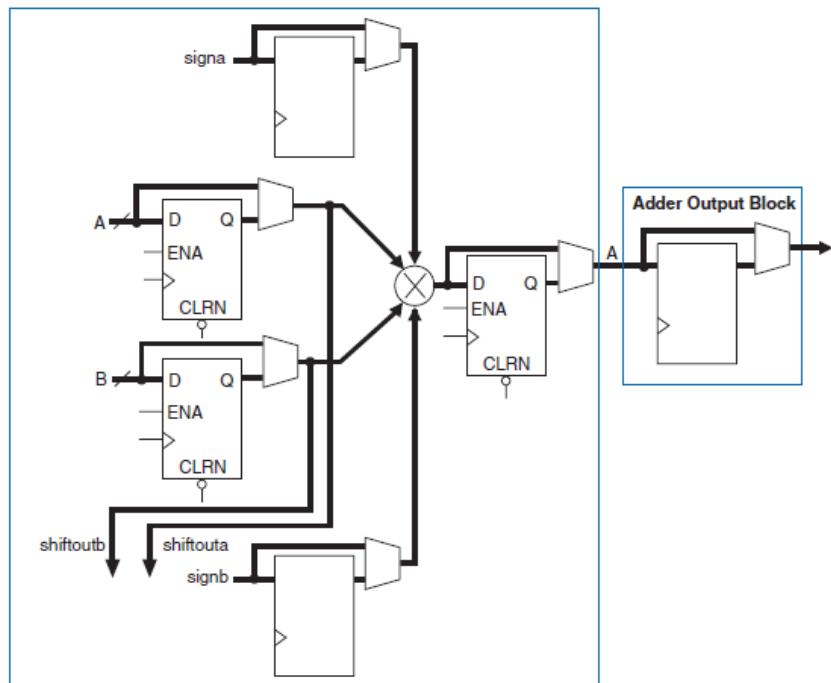


图 45 简单乘法模式结构图

1. 9-Bit / 18-Bit 乘法器

一个 DSP 模块可以支持实现 8 个 9 位以下的乘法器或 4 个 10~18 位的乘法器。

乘法器操作数可以是有符号数和无符号数或者它们的组合，signa、signb 信号可以动态设置或寄存，输入输出也可以独立寄存。流水线寄存器可以提高 DSP 的性能。

2. 36-Bit 乘法器

36 位乘法器是简单乘法模式的一个子集。整个 DSP 只可以用来实现一个 36 位的乘法，其中 4 个 18 位乘法器分别接收输入的一部分。输出单元将 4 个部分和进行相加，可以使用乘法器和求和部件之间的流水寄存器来提高性能。36 位乘法器支持符号数操作，其结构如图 46 所示。

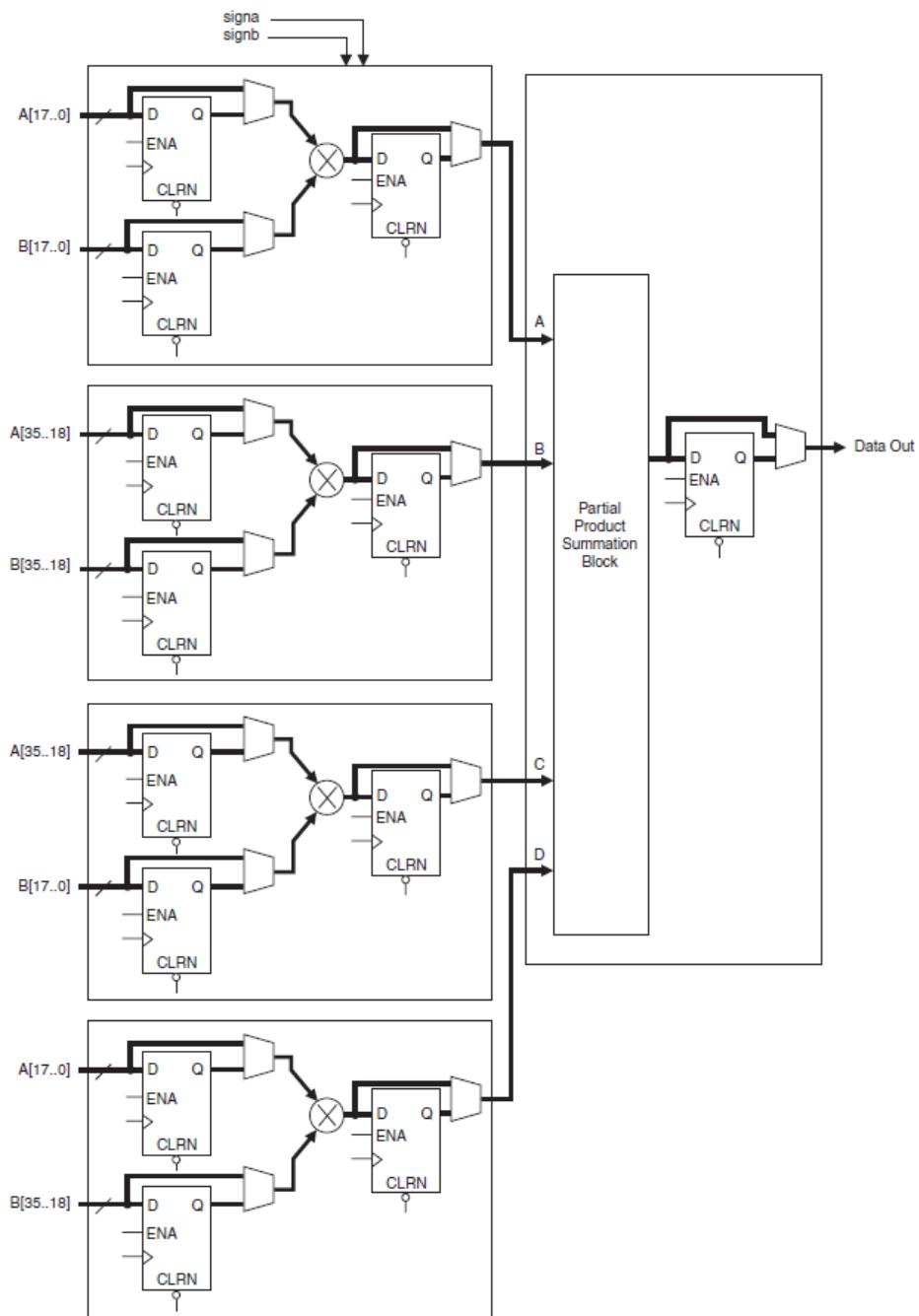


图 46 简单 36 位乘法模式结构图

6.3.2 乘累加模式

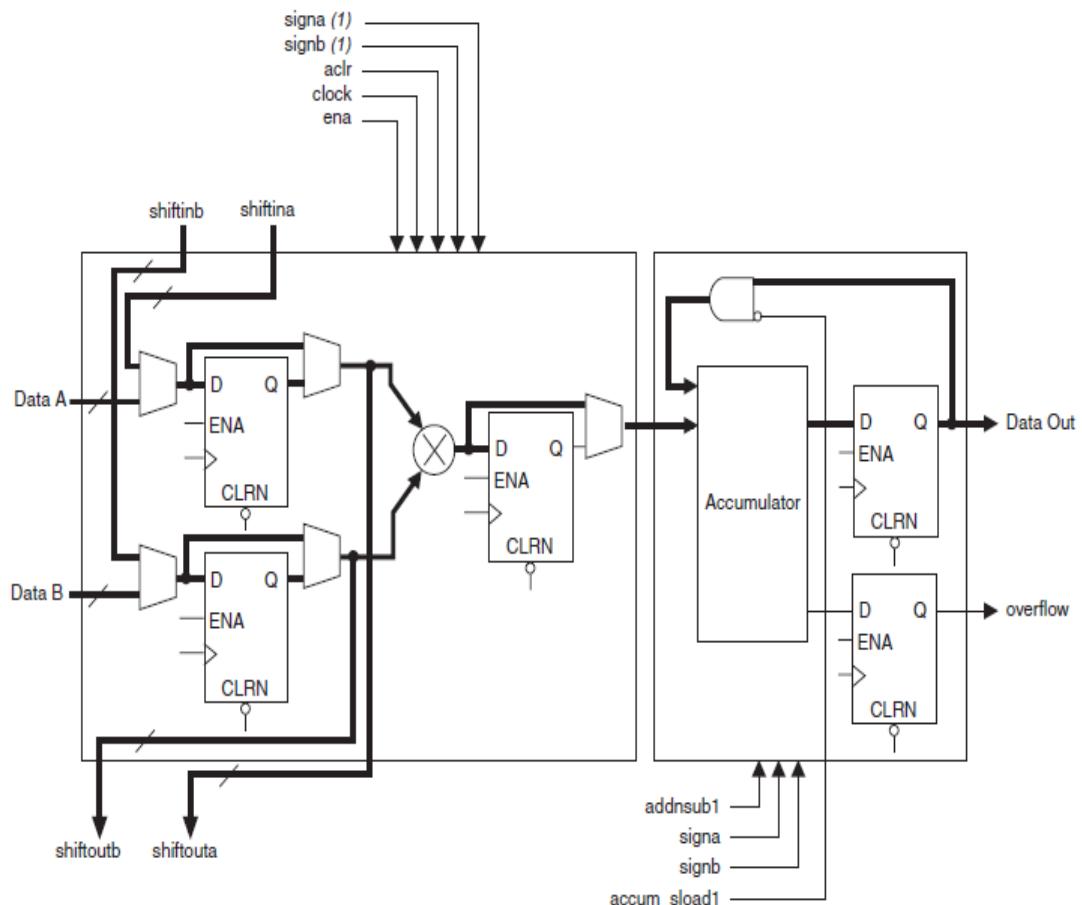


图 47 乘累加结构图

在 MAC (Multiply Accumulator Mode) 模式下，乘法器的输出送到输出单元，输出单元被配置成累加或减法。可以在一个 DSP 模块里实现 2 个 18 位的 MAC。如果实现低于 18 位的 MAC，软件会将数据位放置在 18 位乘法器的高位，18 位乘法器剩余低位置为 0。

MAC 的输出位数最高可以到 52 位，最多可支持 2^{16} 个 36 位的乘法结果累加。在此模式下，DSP 模块可以使用输出寄存器和 accum_sload、overflow 信号。 accum_sload 信号同步加载乘法结果到累加器的输出，此信号可以寄存 1~2 个时钟周期或不寄存，因此 DSP 模块就可以不用损失任何时钟周期数来开始一个新的累加运算。overflow 信号表示累加结果的上溢或下溢，在下一时钟沿马上变成低电平，建议用户使用此信号时进行锁存。此外，可以通过 addnsub 信号动态完成累加或累减。

如果要实现一个累加功能，可以通过乘以 1 再累加来在一个 DSP 中实现。

6.3.3 双乘加模式

双乘加模式 (Two Multiplier-Adder Mode): 使用输出单元对乘法单元的结果进行一级加减运算，一般在 FFT 等算法里可以用到这种模式。另外，这种模式下 1 个 DSP 模块可以输出 2 个 18 位乘法结果的加/减的结果，或输出 4 个 9 位乘法器的加/减的结果。比如一个复数乘法可以表示成：

$$(a + jb) \times (c + jd) = (a \times c - b \times d) + j \times (a \times d + b \times c)$$

此模式下，一个 DSP 模块可以用一个加减单元计算实部($a \times c - b \times d$)，用另一个加减单元计算虚部($a \times d + b \times c$)。

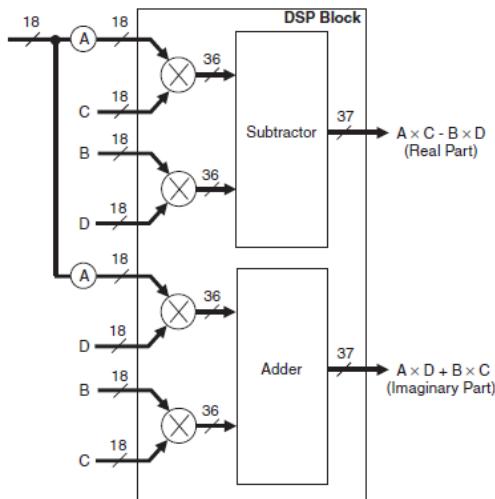


图 48 双乘加模式结构图

对于 9 位以下的乘法，一个 DSP 可以实现两个复数的乘法。其结构如图 49 所示。

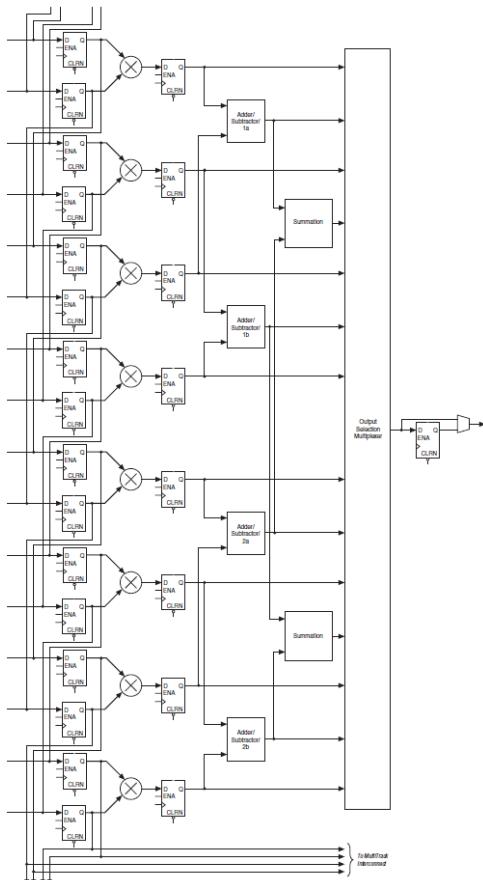


图 49 复数乘累加模式结构图

6.3.4 四乘加模式

在四乘加（Four Multiplier-Adder Mode）模式下，可以实现一维和二维的滤波器。DSP 模块将加减单元的结果在求和部件里求和，一个 DSP 模块可以实现一个 18 位或两个 9 位的求和，其中乘法结果的位宽必须一样。

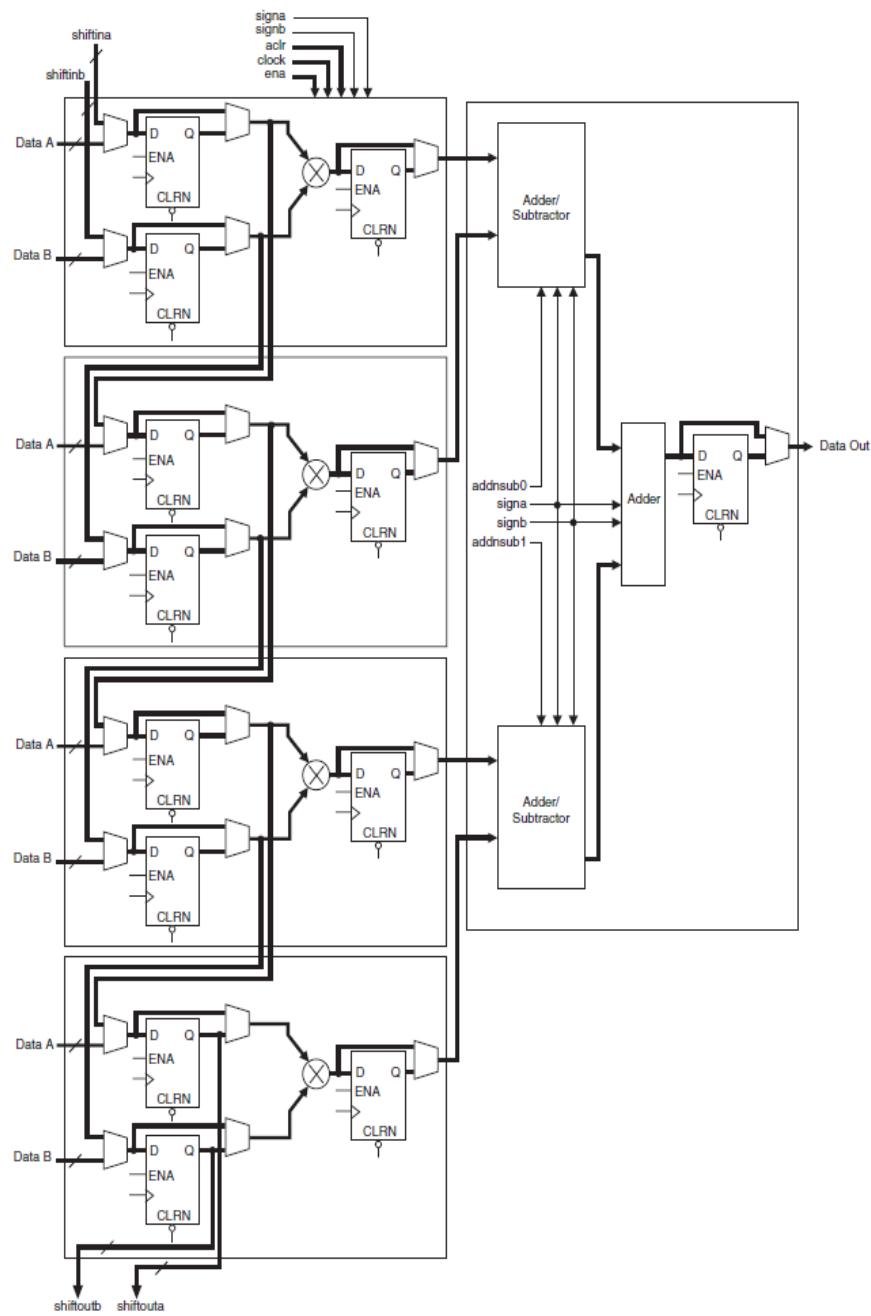


图 50 四乘累加模式结构图

6.3.5 FIR 的应用

四乘加模式适用于实现 FIR，DSP 模块将滤波器数据对应的乘法器输入置为移位输入模

式，系数所对应的另一组输入可以根据需要置为串行或并行加载。一个 DSP 模块可以完全实现一个 4 tap 的 18 位的 FIR，对于大于 4 tap 的 FIR，可以将多个 DSP 模块串起来使用，但需要在外面用 LE 逻辑单元实现加法。

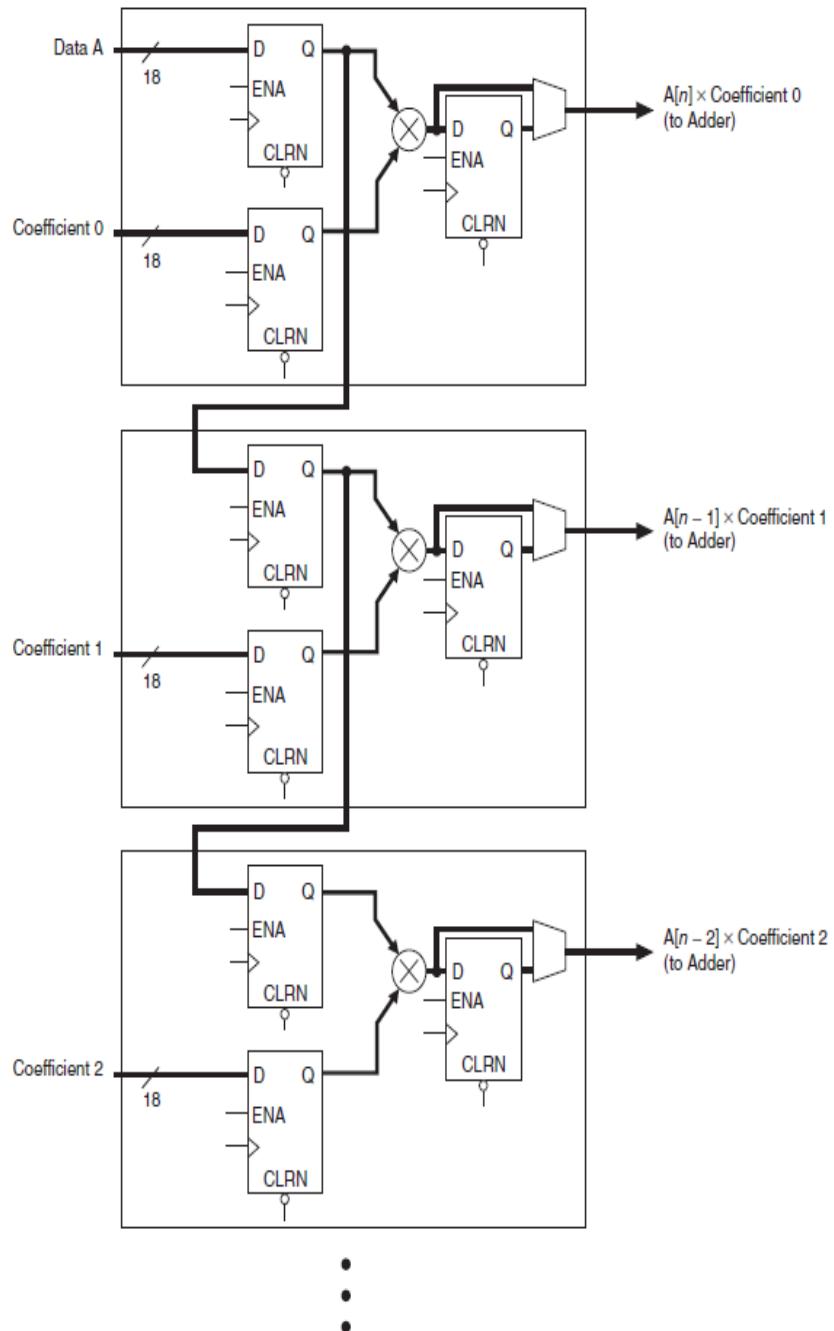


图 51 FIR 应用结构图

6.3.6 有符号数的乘法

signa 和 signb 是 DSP 模块的符号位，signx 在乘法模块只能寄存一个时钟周期，而在随后的 mac_out 模块可寄存 1~2 个时钟周期，第一时钟周期用于输入同步，第二时钟周期用于输出需要的 pipeline。

乘法模块运算结果的符号位标识和 signx 的关系如表 15 所示：

表 15 符号乘法运算结果

Data A	Data B	Result
Unsigned	Unsigned	Unsigned
Unsigned	Signed	Signed
Signed	Unsigned	Signed
Signed	Signed	Signed

从上表可以看出，只有两个输入都是无符号数时，结果才是无符号数；否则结果都是有符号数。当用户没有指定是否有无符号时，signx 端口缺省接为 VCC 高电平，也就是缺省为有符号数。

当两个操作数里有一个为有符号数时，则有符号数为补码（用户直接补码输入），乘法结果也为补码，补码结果直接输出或送给输出模块进行加减累加，结果仍为补码。

6.3.7 Scanout 的处理

DSP 模块最后一个乘法器的 Scanout 信号既可以通过专门的管脚连到下一个 DSP 模块，也可以通过输出连接到通道上，如果需要上通道，则 scanout 必须和输出端口复用，在某些模式下就要占用一些乘法器的输出端口，造成一些乘法器无法使用，下面将分模式说明。

6.3.7.1 简单乘法模式

在 18 位简单乘法模式下，输出为 144 位。如果 scanout 需要走通道，则最后一个乘法器无法使用，空出来的 36 个输出端口可以给第三个乘法器的 36 个 scanout 使用。在 9 位简单乘法模式下，一共有 8 个乘法器（编号 0~7）。第 4 个乘法器如果 scanout 要走通道，则第 6 个乘法器无法使用；第 5 个乘法器如果 scanout 要走通道，则第 7 个乘法器无法使用。

6.3.7.2 乘累加模式

此模式下，两个 MAC 的输出占用 106 个端口，如果两个 scanout 要走通道，则需要增加 72 个端口，无法实现；但如果上面一个 MAC（即第二个乘法模块）的 scanout 走通道，则可以支持该功能。36 位 Scanout 输出可以分别通过输出端口 a 中的[71:54]和输出端口 b 中的[143:126]上通道。

6.3.7.3 双乘加模式

在 9 位模式下空闲 68 个输出端口，在 18 位模式下空闲 70 个输出端口，完全可以满足最后一个乘法器的 scanout 上通道。36 位 Scanout 输出可以通过输出端口[71:54]和[143:126]上通道（对于 9b×9b 模式，只能用到[71:55]和[143:127]，剩下的两位使用[107]和[35]，最终的 36 位（B 和 A）从高到低依次为 {[143:127], [107], [71:55], [35]}）。

6.3.7.4 四乘加模式

在此模式下，完全可以满足最后一个乘法器的 scanout 上通道，36 位 Scanout 可以通过输出端口[71:54]和[143:126]上通道。

6.3.7.5 36 位乘法模式

此模式下，scanout 不可用。

每列的第一个 DSP 不能使用 scanin，最后一个 DSP 不能使用 scanout。

Scanin 或 scanout 只能用于相同的时钟域，否则 Hold 时序无法保证。Scanout 同时输出

到专用端口和复用输出端口上，软件可以根据布局布线的需要自行决定如何与下一个 DSP 相连。DSP 的输入可以通过配置位选择数据来源于 scanin 还是 data。

表 16 不同 DSP 模式下不同位宽的输入输出排列表

width mode	9b×9b (i=0~7)		18b×18b (i=0~3)		36b×36b	
	输入	输出	输入	输出	输入	输出
简单乘法	8 个乘法: [9i+8:9i]	[18i+17:18i]	4 个乘法: [18i+17:18i]	[36i+35:36i]	按 36 位模式的输入连接方式	[71:0]
乘累加 MAC			两个 MAC, [35:18] [71:54] (第一和第三个乘法模块不使用)	上面 MAC: [51:0] overflow0 接[52]; 下面 MAC: [123:72] overflow1 接 [124]		
双乘加	四组: ([8:0][26:18]) ([17:9][35:27]) ([44:36][62:54]) ([53:45][71:63])	对应的输出 [18:0] [54:36] [90:72] [126:108]	两组: 0~1 一组, 2~3 一组: [18i+17:18i]	上面一组: [36:0] 下面一组: [108:72]		
四乘加	两组: ([8:0][26:18] [44:36][62:54]) ([17:9][35:27] [53:45][71:63])	[19:0] [91:72]	4 个输入: [18i+17:18i]	[37:0]		

说明：

- (1) 输入或输出不同位宽均采用向高位对齐
- (2) 9 位模式下的 MAC 也采用 18 位模式来实现。输入输出向高位对齐即可。

7 JTAG 电路

7.1 概述

JTAG 电路是为实现与 IEEE 1149.1-2001 协议兼容而加入的特定电路结构，其功能是完成 IEEE 1149.1-2001 协议所规定的各种强制要求的功能以及一些可选的功能。支持 JTAG 协议，具备边界扫描测试功能。

边界扫描是一种可测试结构技术，它是在芯片的 I/O 端口增加寄存器，在测试模式下，可连接成边界扫描通道。边界扫描测试可以通过专用的 JTAG 接口进行 I/O 管脚值的捕获、预加载、钳位等功能实现板级互联的测试。

7.2 电路结构

JTAG 电路主要包含 7 个模块：TAP 控制器、指令寄存器及其判断逻辑、边界扫描寄存器、旁路寄存器、器件 ID 寄存器、配置寄存器、BitProbe 寄存器。系统电路结构如图 52 所示。

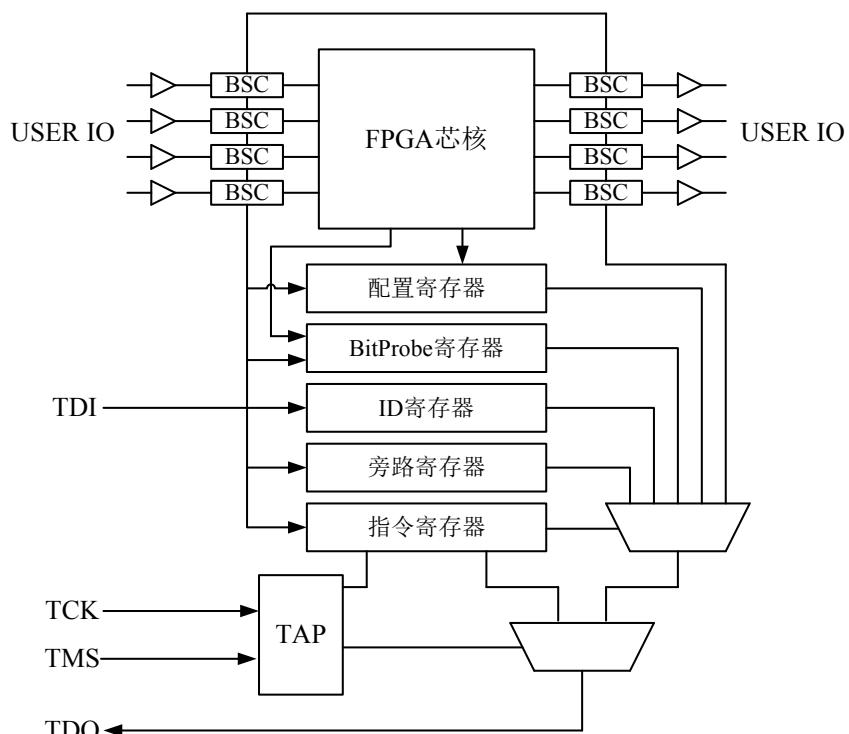


图 52 系统结构图

7.2.1 接口信号说明

表 17 JTAG 输入输出接口

Port_Name	Width	Direction	Description
TCK	1	输入	JTAG 测试时钟
TMS	1	输入	JTAG 测试模式选择
TDI	1	输入	JTAG 数据输入
TDO	1	输出	JTAG 三态数据输出

- (1) TCK: JTAG 的测试时钟输入，为 TAP 控制器和 JTAG 的寄存器提供时钟。
- (2) TMS: JTAG 的测试模式选择输入，用于控制 TAP 控制器的状态转换，在 TCK 的上升沿有效。在无驱动时，需通过上拉保持为“1”。
- (3) TDI: JTAG 的串行数据输入，用于指令寄存器和数据寄存器数据的输入。TAP 控制器的状态和当前的指令决定了 TDI 输入的寄存器位置。TDI 的值在 TCK 的上升沿有效，在无驱动时，需通过上拉保持为“1”。
- (4) TDO: JTAG 的串行输出，用于输出所有指令寄存器和数据寄存器的数据。TAP 控制器的状态和当前的指令决定了 TDO 输出的内容。当没有串行输出时，保持为高阻状态，在测试时钟 TCK 的下降沿有效。

7.2.2 子模块电路功能说明

表 18 子模块电路功能描述

名称	功能
TAP Controller	TAP 控制器
Instruction Register	指令寄存器
BoundaryScanRegister	边界扫描寄存器
DeviceID Register	器件 ID 寄存器
BypassRegister	旁路寄存器
ConfigurationRegister	配置寄存器
BitProbeRegister	BitProbe 寄存器
OutputLogic	输出控制逻辑

7.2.3 操作时序

图 53 显示了 JTAG 电路从复位状态进入加载指令模式的过程。

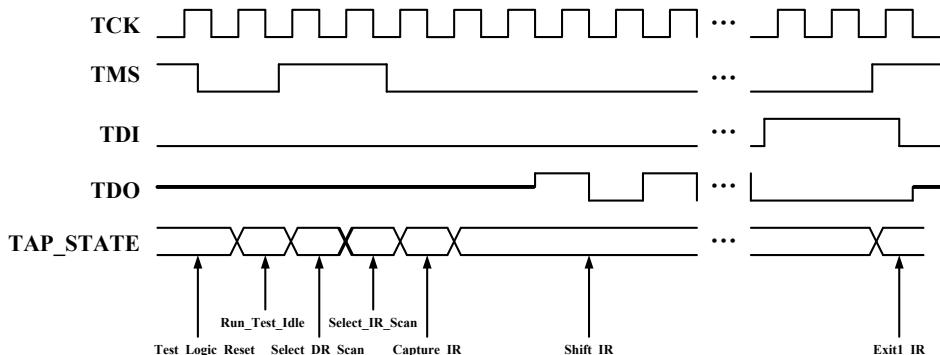


图 53 JTAG 加载指令的时序

JTAG 电路对数据寄存器的操作同具体的指令相关，图 54 显示了 SAMPLE 指令下数据移位操作的过程。

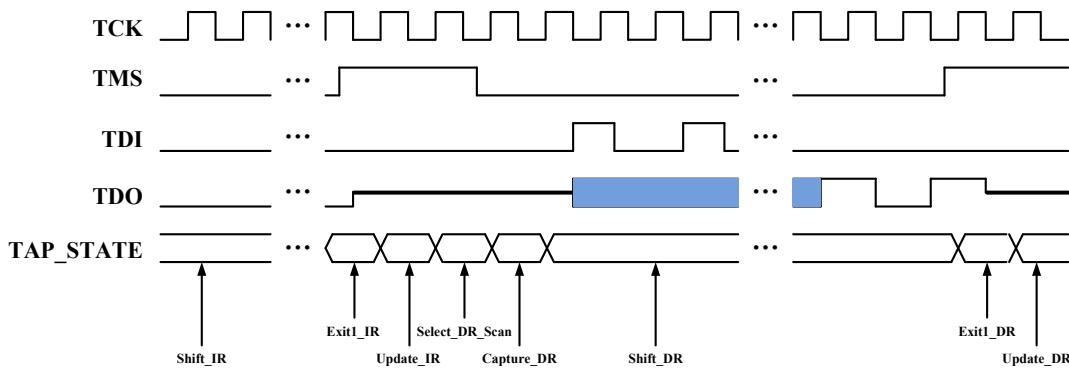


图 54 JTAG 电路数据移位操作的时序

7.3 功能模块

7.3.1 TAP 控制器

TAP 控制器是一个 16 位状态的同步有限状态机，它产生数据寄存器及指令寄存器的控制信号，其状态定义如下表 19。

表 19 TAP 控制器状态定义

State	编码	功能描述
Test_Logic_Reset	1111	复位状态，系统模式，指令寄存器复位为 IDCODE 指令
Run_Test_Idle	1100	由指令寄存器中的指令决定当前的状态和操作
Select_DR_Scan	0111	辅助控制器状态
Capture_DR	0110	在 TCK 上升沿，数据被加载到选中数据寄存器的移位部分

State	编码	功能描述
Shift_DR	0010	数据从 TDO 串行移出，同时从 TDI 移入新的数据
Exit1_DR	0001	辅助控制器状态
Pause_DR	0011	暂停 TDI 和 TDO 之间的移位操作
Exit2_DR	0000	辅助控制器状态
Update_DR	0101	TCK 下降沿，移位寄存器中的数据被写入其并行输出端上
Select_IR_Scan	0100	辅助控制器状态
Capture_IR	1110	在 TCK 上升沿，指令寄存器的移位部分捕获数据。
Shift_IR	1010	串行移出捕获的指令数据，同时移入新的指令数据
Exit1_IR	1001	辅助控制器状态
Pause_IR	1011	暂停指令寄存器的移位操作
Exit2_IR	1000	辅助控制器状态
Update_IR	1101	在 TCK 下降沿，将移入的指令锁存，成为当前指令

TAP 控制器的状态转换发生在 TCK 的上升沿，其状态转换如图 55 所示。

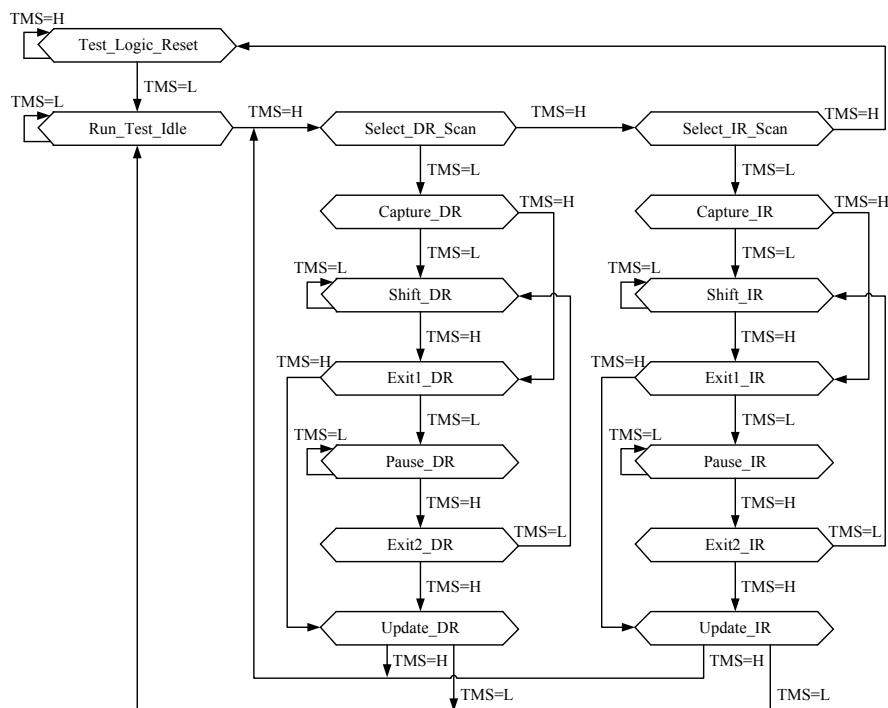


图 55 TAP 控制器状态转换示意图

7.3.2 指令寄存器

指令寄存器由 TAP 控制器控制，用于产生对数据寄存器的选择及控制信号。指令寄存器由移位寄存器单元及相应的锁存器单元组成。移位寄存器在 CAPTURE-IR 时并行加载，在 SHIFT-IR 时进行由 TDI 到 TDO 的移位，并在 UPDATE-IR 时，将移位寄存器的内容传递到锁存器单元中。

7.3.3 边界扫描寄存器

边界扫描寄存器是标准规定必须包含的数据寄存器，也是 JTAG 结构中最重要的数据寄存器，构成了扫描路径。边界扫描寄存器由边界扫描单元串连组成，每个边界扫描单元都对应了芯片的一个管脚，介于芯片管脚和芯片内核逻辑之间，在逻辑上相当于一个内置的测试点。

7.3.4 旁路寄存器

旁路寄存器是一个单数据位寄存器。当选中旁路寄存器时，在 Capture-DR 状态下加载为 0，在 Shift-DR 状态下将 TDI 的数据串行移出。对旁路寄存器的操作不影响系统的正常工作。

7.3.5 器件标志寄存器

器件标志寄存器提供关于芯片生产商的名称、芯片号和版本号的二进制信息，当读出器件的标志信息时，不影响芯片的正常工作。在 Captur-DR 状态下，器件标志寄存器捕获器件标志信息，在 Shift-DR 状态下将其移出。

7.4 边界扫描指令

表 20 列出了所支持的边界扫描指令。

表 20 边界扫描指令

边界扫描指令	指令代码[5:0]	功能描述
EXTEST	00_0110	板级芯片互连测试指令
IDCODE	00_0001	板上芯片识别，移出器件标志信息
SAMPLE/PRELOAD	00_0011	采样/预加载板上芯片 I/O 管脚的值
CLAMP	00_1110	钳位输入/输出，并使能旁路寄存器
HIGHZ	00_1100	三态输出管脚，并使能旁路寄存器
BYPASS	11_1111	使能 BYPASS
JPROG_B	00_1101	配置控制器异步复位指令
CFG_IN	00_0101	JTAG 配置指令
JSTART	00_1111	JTAG 解冻指令
CFG_OUT	00_0100	JTAG 回读指令
BIT_PROBE	00_0010	BIT_PROBE 指令

7.4.1 外部测试指令(EXTEST)

外部测试指令用于板级芯片互连测试，将边界扫描寄存器置于 TDI 和 TDO 之间。在

CAPTURE-DR 状态下，所有的芯片输入被捕获进入相应的边界寄存单元的移位寄存器，在 SHIFT-DR 状态下，边界寄存器进行移位，用户可以读出捕获的输入信息。

7.4.2 采样指令（Sample）和预加载指令(Preload)

采样指令和预加载指令是 IEEE 1149.1-2001 标准规定必须包含的指令，其将边界寄存器置于 TDI 和 TDO 之间，但不影响系统逻辑的正常工作。

采样动作发生在 CAPTURE-DR 状态，所有与边界寄存器单元相连的芯片输入输出信号，都被加载到边界寄存器单元的移位寄存器部分，从而获得了当前芯片输入输出管脚的状态。在 SHIFT-DR 状态下，将采样信息通过移位寄存器移出。

在所采样的数据被移出的同时，新的数据被移入。移位寄存器接收到这些数据后，在 UPDATE-DR 状态将数据加载到锁存器，从而实现预加载功能。

7.4.3 旁路指令(Bypass)

旁路指令将旁路寄存器置于 TDI 和 TDO 之间，其目的是为了产生一位的移位通路，使信号旁路穿过此芯片，而不影响芯片正常的工作。旁路指令是 1149.1 标准规定所必须包含的指令，其指令代码为 11_1111。

7.4.4 高阻指令(HIGHZ)

高阻指令是一个可选指令，它将旁路寄存器作为 TDI 和 TDO 之间的目标寄存器。当高阻指令有效时，所有系统逻辑的输出都被置于高阻状态。

7.4.5 钳位指令(CLAMP)

钳位指令是一个可选指令，它使所有的输出和双向引脚都受控于边界寄存器，从而使芯片的输出引脚被设置为预设值。这条指令可用于芯片组的测试，对于在测试中没有涉及到的逻辑电路引脚，需加一个静态的保护值。

7.4.6 器件标志指令(IDCODE)

器件标志寄存器包含了 32 位的标识芯片生产商名称、芯片号和版本号的数据。在 Capture-DR 状态下，这些数据值被捕获进移位单元，通过移位操作由 TDO 输出。

7.4.7 配置控制器异步复位指令(JPROG_B)

配置控制器异步复位指令是与配置相关的指令，它将旁路寄存器作为 TDI 和 TDO 之间的目标寄存器。当该指令成为当前指令时，将控制信号 JPROG_B_En 拉高，对配置控制器进行异步复位。

7.4.8 配置指令(CFG_IN)

配置指令对 FPGA 进行 JTAG 配置，它将配置接收寄存器作为 TDI 与 TDO 之间的目标寄存器。当该指令成为当前指令时，配置控制器对配置存储器进行最后一遍清零，开始接收从 JTAG 来的时钟和数据信号。

7.4.9 解冻指令(JSTART)

解冻指令对 JTAG 配置后的芯片进行解冻，它将解冻接收寄存器作为 TDI 与 TDO 之间的目标寄存器。当该指令成为当前指令时，配置控制器开始对 FPGA 芯片进行解冻。解冻完毕后输出 IO 使能信号，释放 IO 进入用户模式，成功解冻后 TDO 输出由低变高。

7.4.10 回读指令(CFG_OUT)

回读指令通过 JTAG 端口对 FPGA 中的码流进行回读，回读内容由预先加载的回读命令确定。回读指令将回读接收寄存器作为 TDI 与 TDO 之间的目标寄存器。 JTAG 回读需先加载 CFG_IN 指令，将不同回读命令下载到配置接收寄存器。

7.4.11 BitProbe 指令(BitProbe)

内嵌逻辑分析仪 BitProbe 可通过 JTAG 端口监控器件内部信号的状态。该指令将 BitProbe 接收寄存器作为目标寄存器放在 TDI 和 TDO 之间。当 BitProbe 指令成为当前指令时，TCK、TMS、TDI 旁路到芯片内部作为 BitProbe 的输入，同时 BitProbe 接收寄存器接收来自 BitProbe 的信号并通过 TDO 输出。

8 配置

8.1 概述

EQ6HL45 支持 8 种配置模式，包括：主动串行（Master Serial）、被动串行（Slave Serial）、主动并行（Master SelectMAP）、被动并行（Slave SelectMAP）、JTAG、主动 SPI（SPI Master）、被动 SPI（SPI Slave）和 Daisy-chain 的下载方式，支持被动并行和 JTAG 模式下的配置存储器（CRAM）回读与 BRAM 初始化值回读功能。

8.2 配置管脚

EQ6HL45 配置管脚如下表 21 所示。

表 21 EQ6HL45 配置管脚定义

名称	方向	宽	类型	说明
DONE	Bi-dir	1	双向开漏	(1) 作为输出，高电平标志配置的数据装载阶段已经完成 (需要再经过解冻和释放 IO 即可进入用户模式) (2) 作为输入，低电平强行推迟进入用户模式后用户电路开始工作的时间
CCLK	Bi-dir	1	双向 OE	配置时钟
D[0]	Bi-dir	1	双向 可控制方向、可置为高阻	(1) 串行配置数据的输入 (2) 并行配置数据的第 0 位输入 (3) 被动并行模式下作为回读数据的第 0 位输出
D[1]	Bi-dir	1	与 D[0]相同	(1) 并行配置数据的第 1 位输入； (2) 被动并行模式下作为回读数据的第 1 位输出
D[7:2]	Bi-dir	6	复用双向 复用状态下可控制方向、可置为高阻	(1) 主动和被动并行配置模式下作为数据输入； (2) 被动并行模式下作为回读数据输出
BUSY/DOUT	Output	1	专用单向 可置为高阻	(1) 串行配置时作为 daisy-chain 输出； (2) 并行配置时作为输入数据的超速指示； (3) 被动并行回读时作为数据有效的指示
RDWR_B	Input	1	复用单向	控制 D 端口的方向； 0：D 端口工作于输入状态，数据从片外写给 FPGA； 1：D 端口工作于输出状态，FPGA 将回读数据送至片外
SPI_MASTERR_CSB	Output	1	专用单向	Spi master 的使能端输出
SPI_SLAVE_CSB	Input	1	专用单向	Spi slave 的使能输入
SPI_DO	Output	1	专用单向	Spi master output 和 spi slave output
M2	Input	1	专用单向	配置模式选择
M1	Input	1	专用单向	配置模式选择

名称	方向	宽	类型	说明
M0	Input	1	专用单向	配置模式选择
PROG_B	Input	1	单向	配置逻辑同步复位信号，低电平有效
INIT_B	Bi-dir	1	双向开漏	(1) 上电后、数据装载开始前，片内将其置低时，表明FPGA 正在将配置存储器清零 (2) 数据装载开始前，从片外将其强行置低，可推迟配置 (3) 在配置数据装载期间被片内拉低，表明配置出错

8.3 配置模式

表 22 EQ6HL45 配置模式管脚设定

管脚设置 配置模式	M2	M1	M0	CCLK 方向	数据宽度
SPI Master	1	0	1	输出	1
主动串行	0	0	1	输出	1
主动并行	1	0	0	输出	8
被动并行	1	1	0	输入	8
被动串行	1	1	1	输入	1
SPI Chain Slave	0	0	0	输入	1
SPI Slave	0	1	1	输入	1
JTAG	x	x	x		1

说明：

- (1) M2、M1 和 M0 应当始终被设置为恒定的直流电平，在配置期间严禁翻转，否则会导致配置错误。
- (2) 只有被动并行模式和 JTAG 模式能够对配码进行回读。
- (3) Daisy-chain 配置在串行模式（Master serial 和 slave serial）以及 SPI（X1）模式下有效。
- (4) 前四种模式均可与 XCF32P 型 PROM 配合使用，SPI 模式可以与 Flash 搭配使用。
- (5) 在串行模式下，配置时钟 CCLK 的每个周期将 1 位配码写入器件；在并行模式下，配置时钟 CCLK 的每个周期将 8 位配码写入器件。
- (6) 主动模式下 CCLK 由器件内部产生；被动模式下由片外时钟驱动 CCLK 管脚。
- (7) 在 JTAG 模式下，DONE 和 INIT_B 依然有效。
- (8) 在主动串行、主动并行、被动串行、被动并行、SPI Master、SPI Slave、Daisy-chain 的模式下均支持码流加密。
- (9) 在 SPI Master、SPI Slave、主动串行、主动并行模式下均支持码流压缩。

8.3.1 主动串行（Master Serial）模式

在主动串行模式下，可以使用 PROM 对 FPGA 进行串行配置，原理如图 56 所示。

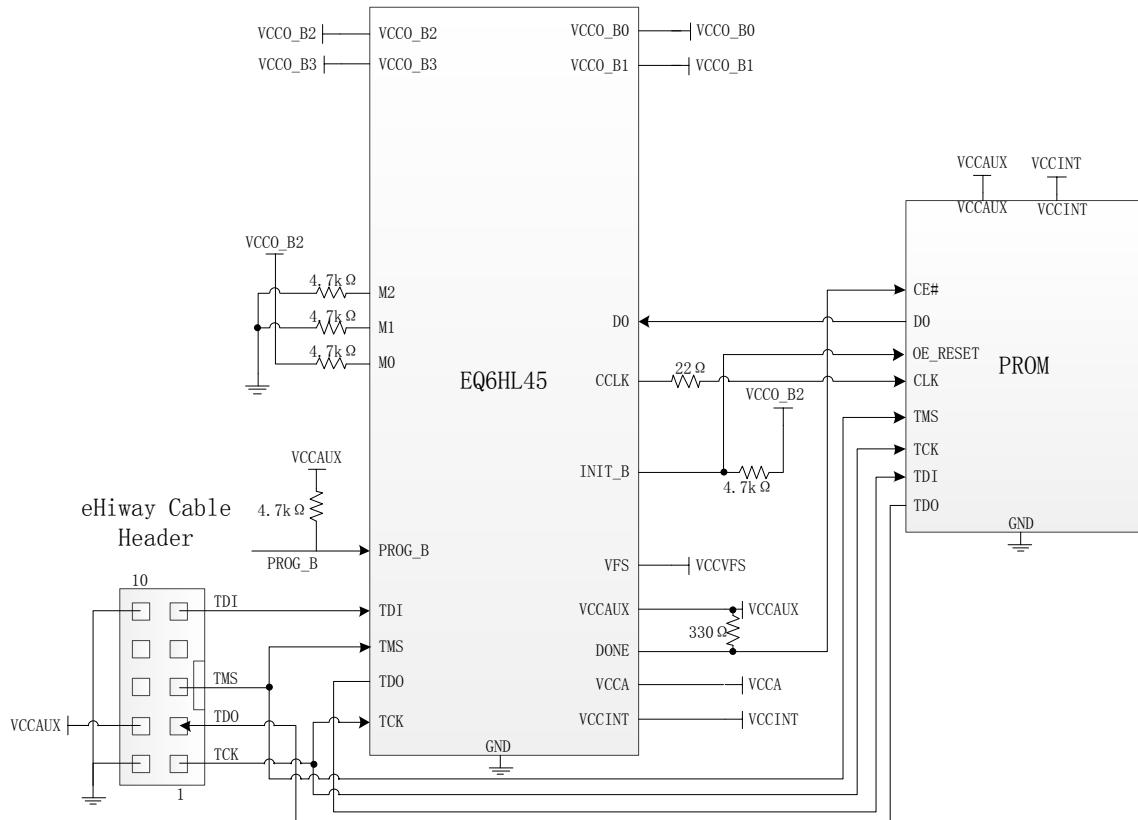


图 56 主动串行配置模式示意图

在选用配置芯片时，应确保 PROM 支持器件所生成的 CCLK 频率。

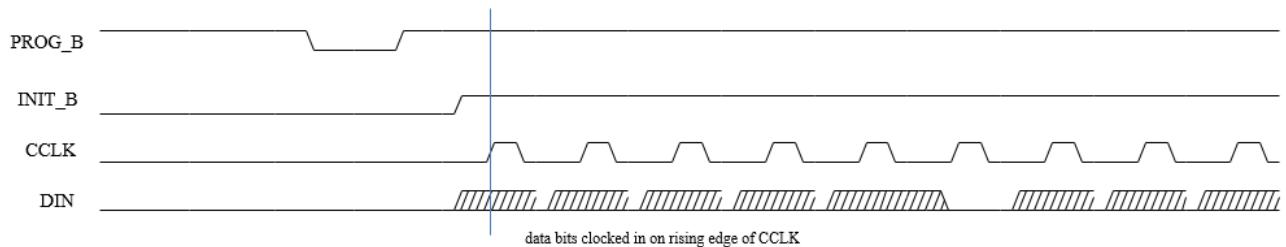


图 57 主动串行配置时序图

主动串行模式下的时序如图 57 所示，需等待初始化完成后，FPGA 才开始送出 CCLK 时钟，器件在 CCLK 上升沿对 DIN 进行采样。在烧写 PROM 时需开启 Serial Mode 选项。

8.3.2 被动串行 (Slave Serial) 模式

在被动串行模式下，FPGA 的 CCLK 管脚由片外时钟源来驱动，每个 CCLK 周期装载 1 个配置位，被动串行模式下的原理图如图 58 所示，时序如图 59 所示。

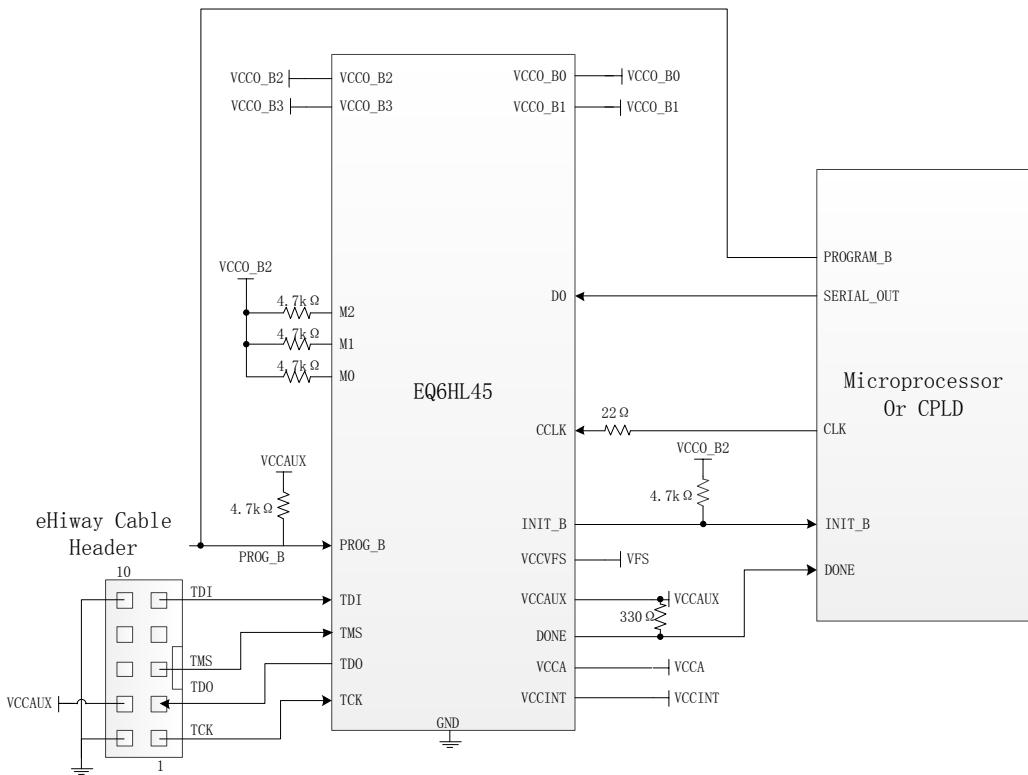


图 58 被动串行配置模式示意图

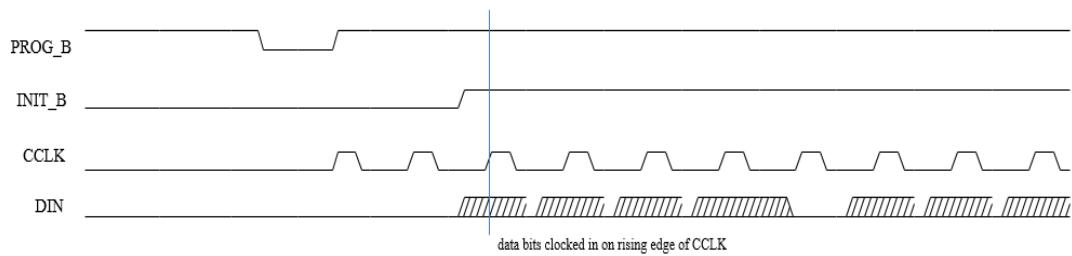


图 59 被动串行配置时序图

8.3.3 主动并行模式

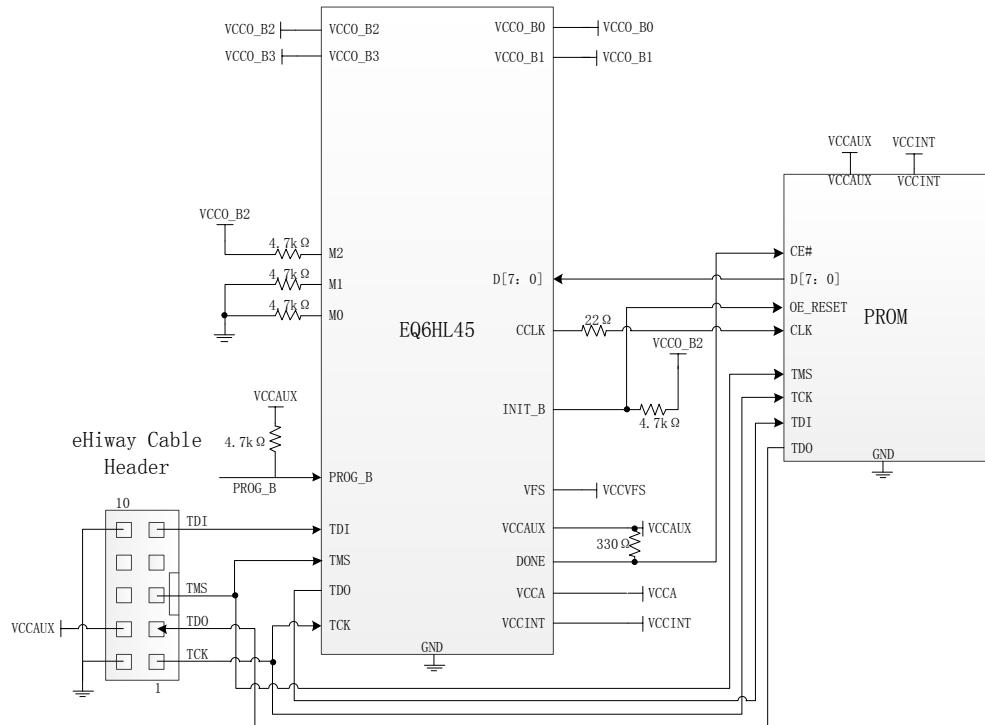


图 60 主动并行配置模式示意图

在主动并行模式下，可以使用 PROM 对 FPGA 进行并行配置，原理如图 60 所示。

说明：

- (1) DONE 必须外接上拉电阻。
- (2) INIT_B 必须外接上拉电阻。
- (3) CCLK 在主动并行模式下是 FPGA 的输出信号，用于对数据总线 DATA 上的数据进行同步。
- (4) 在烧写 PROM 时，需开启 Parallel Mode 选项。
- (5) PROG_B 信号可以通过按键开关下拉到地，用于复位 FPGA。

主动并行模式下的时序如图 61 所示。

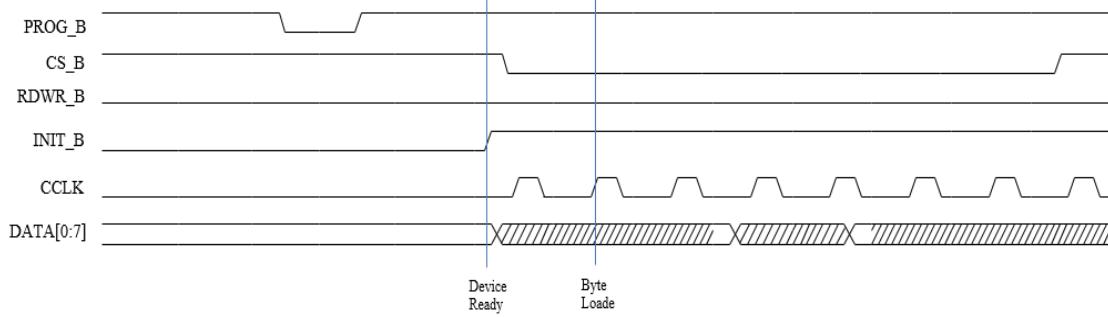


图 61 主动并行配置时序图

8.3.4 被动并行模式

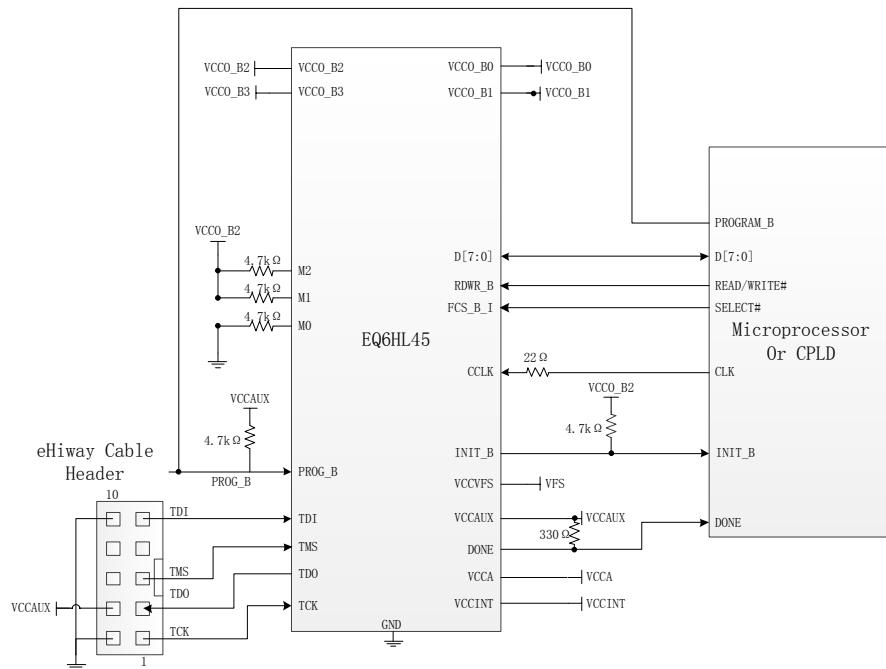


图 62 被动并行配置模式示意图

在被动并行模式下，FPGA 的 CCLK 管脚由片外时钟源来驱动，每个 CCLK 周期装载 1 个配置位，被动并行模式下的原理如图 62 所示。

说明：

- (1) DONE 必须外接上拉电阻。
- (2) INIT_B 必须外接上拉电阻。
- (3) RDWR_B 的定义与 Virtex-II 相同。被动并行模式下，D 是双向 IO，配置 FPGA 时是输入、回读 FPGA 配码时是输出，其方向受 RDWR_B 控制。
- (4) CS_B 是无效输入，接 0 或者 1 均可。
- (5) CCLK 来自片外，用于对数据总线 DATA 上的数据进行同步。
- (6) 在使用 iMPACT 烧写 XCF32P PROM 时，必须开启并行模式。
- (7) 上图中的 PROG_B 可以通过按键开关下拉到地，用于将 FPGA 复位。
- (8) 配置完成后如果对配码进行回读，必须小心谨慎地处理 RDWR_B。

被动并行模式下的时序如图 63 所示。

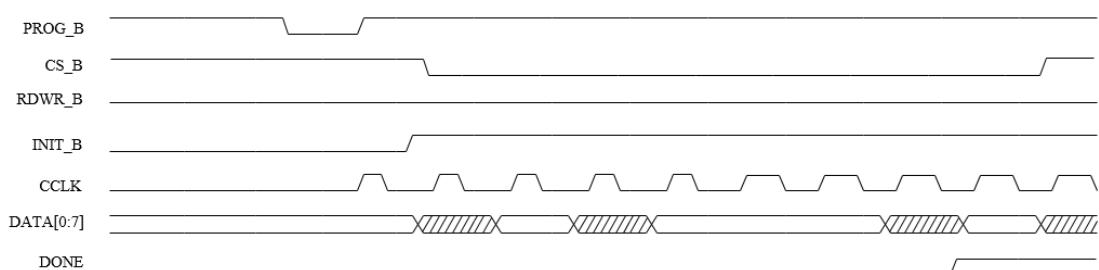


图 63 被动并行配置时序图

说明：如果无需回读，可将 RDWR_B 固定为 0。

建议：使用非压缩码流，CCLK 最大给 20MHz，建议给 10MHz。

8.3.5 JTAG 模式

在 JTAG 配置模式下，通过 JTAG 接口完成配置，配置流程如图 64 所示。

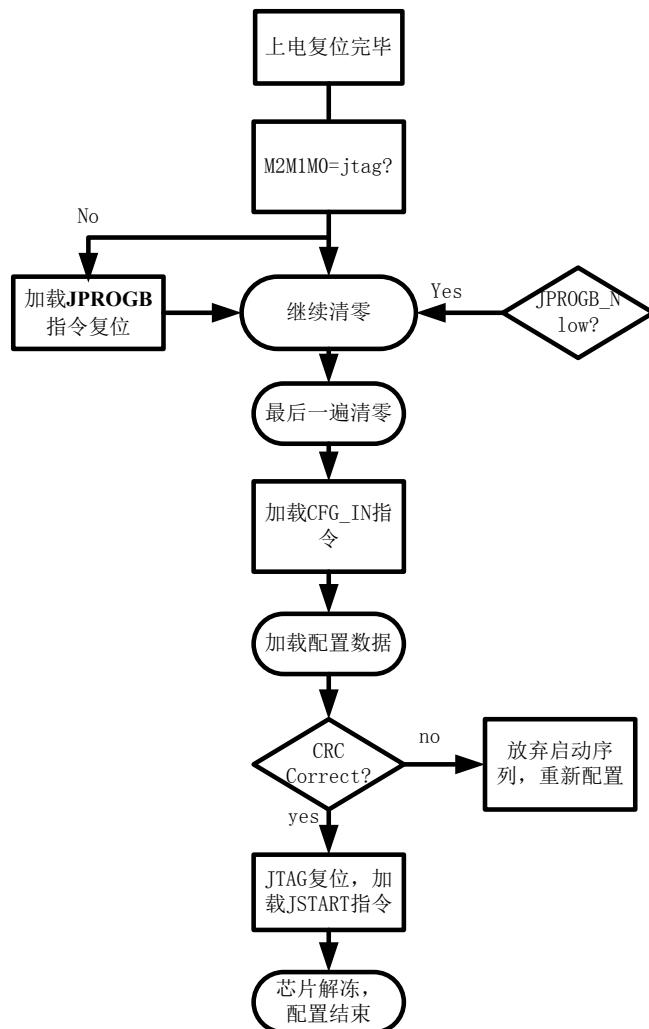


图 64 JTAG 配置流程

在实际应用中，JTAG 配置模式一般使用专用开发软件通过下载线直接对芯片进行配置。在特定情况下也可采用配置芯片通过 JTAG 接口对器件进行配置，多用于产品的开发调试。

JTAG 配置模式拥有高于其他配置模式的优先级，无论处于何种配置模式，都可向 FPGA 发送特定的 JTAG 命令以进行系统调试。

8.3.6 SPI Master 模式

SPI 全称为串行外设接口（Serial Peripheral Interface），是由 Motorola 提出的一种全双工同步串行通信接口，通信波特率可以高达 5Mbps，但具体速度大小取决于 SPI 硬件。SPI 接

口具有全双工操作，操作简单，数据传输速率较高的优点，但是也存在没有指定的流控制，没有应答机制确认是否接收到数据的缺点。

SPI 串行接口有四个 pin 组成,SCK 是移位时钟输入(SLAVE)或者输出(MASTER),MOSI 是对 master 是串行数据输出,对 slave 是串行数据输入,MISO 对 master 是串行数据输入,对 slave 是串行数据输出,CS#是低电平有效的选择信号。

FPGA 的配置码流可能存在外部的 MCU 或者 Flash 中，如果配置码流保存在外部的 MCU 中，则需要 MCU 通过 SPI 接口往 FPGA 内部写入；如果配置码流保存在外部的 Flash 中，则需要 FPGA 自主通过 SPI 接口去读取配置码流。

在 SPI Master 模式下，通过 SPI 接口主动去外部 download 码流：

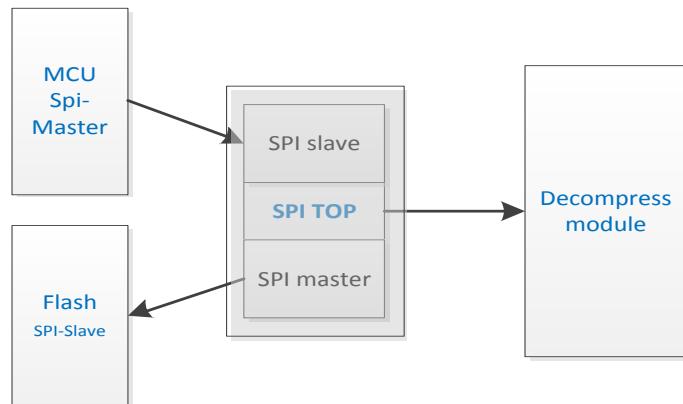


图 65 SPI 接口框图

下图所示为 SPI Master 的配置示意图：

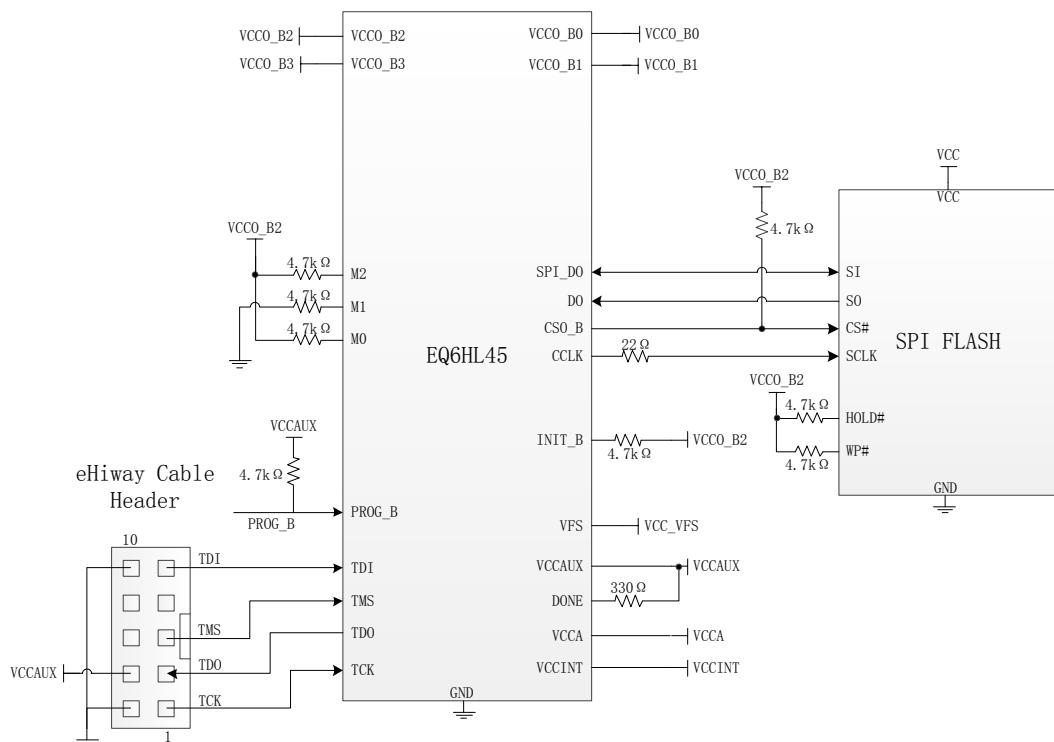


图 66 SPI Master X1&X2 配置模式示意图

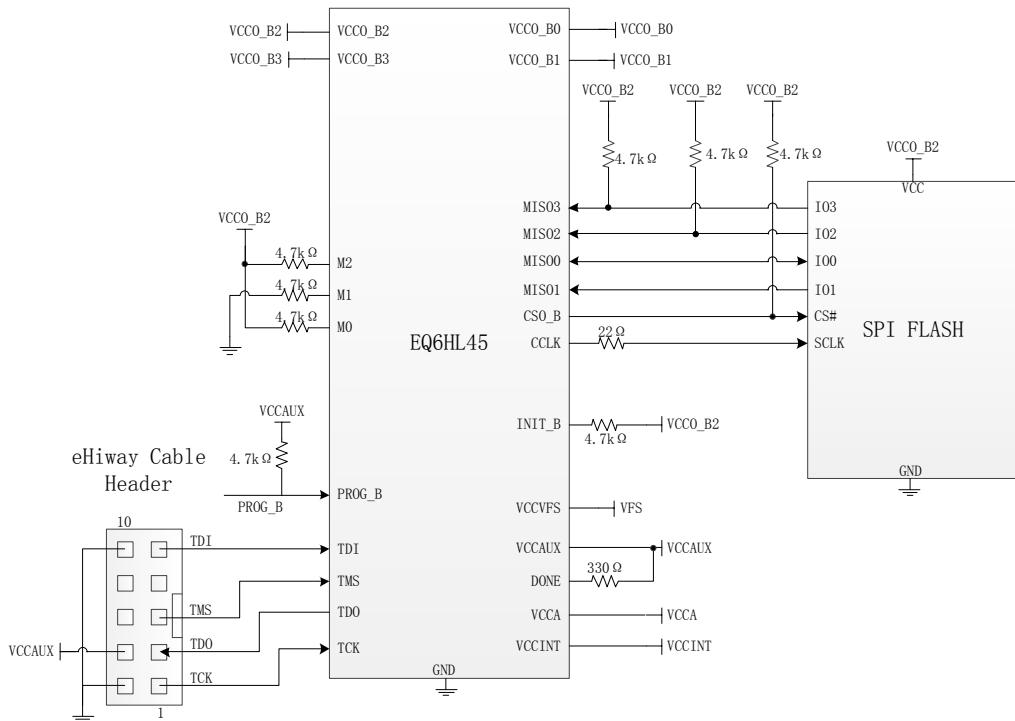


图 67 SPI Master X4 配置模式示意图

下图所示为 SPI Master 的读时序图：

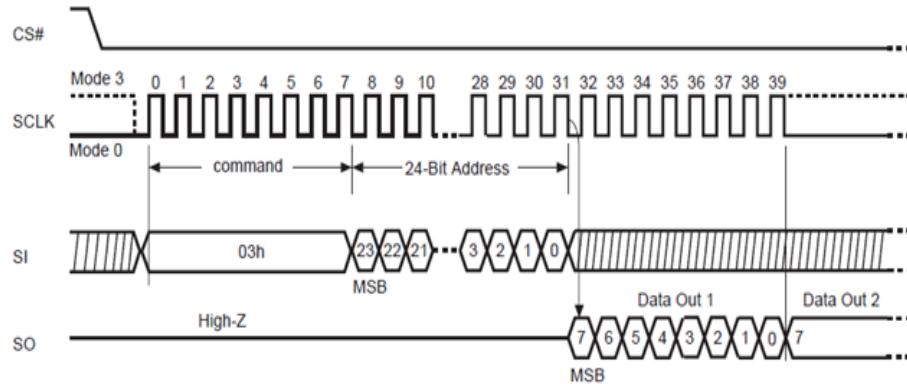


图 68 SPI Master 接口时序

快速读数据命令与读数据命令类似，但是需要在 24bit 地址之后插入 8 个 dummy 时钟，dummy 时钟给内部电路提供了额外的时间用于地址建立，从而可以工作在更高的频率 F 下，在 dummy 时钟期间，SO 上的数据没有意义。下图所示为 SPI Master 的 fast read 的时序图：

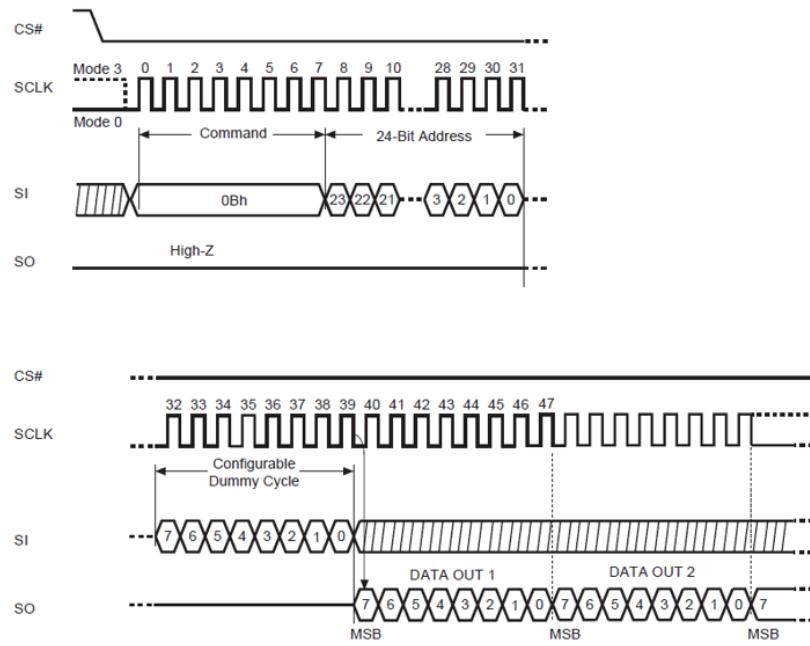


图 69 SPI Master fast read 接口时序

下图为 SPI Master Dual read 的时序图:

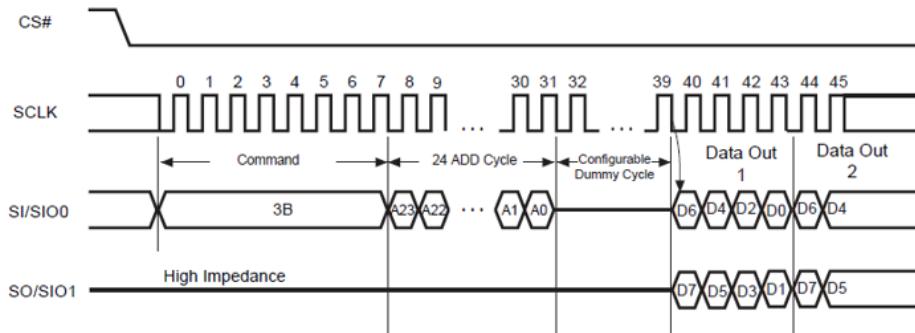


图 70 SPI Master Dual read 接口时序

下图为 SPI Master Quad read 的时序图:

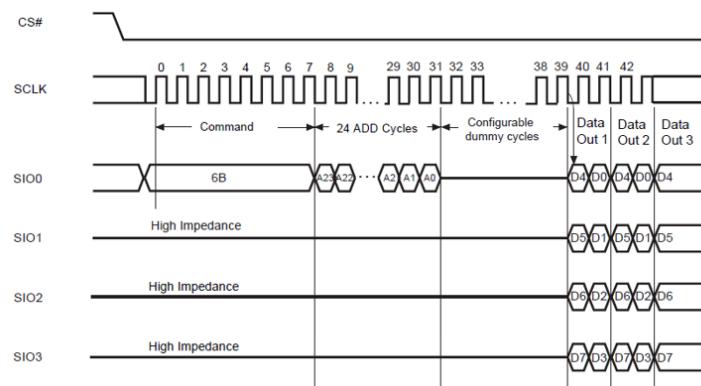


图 71 SPI Master Quad read 接口时序

8.3.7 SPI Slave 模式

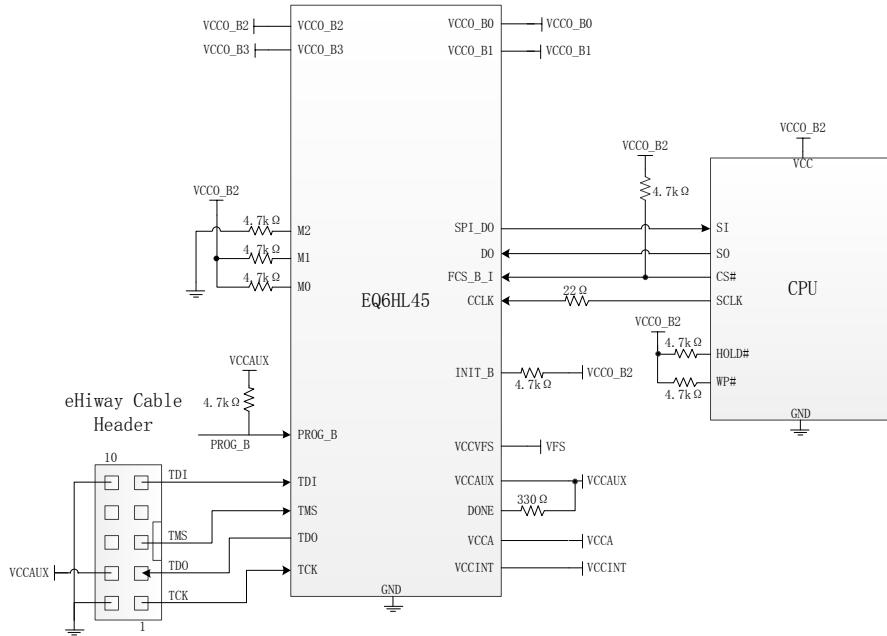


图 72 SPI Slave 配置模式示意图

在 SPI Slave 模式下，外部 CPU 通过 SPI 接口访问内部的寄存器，SPI 访问 Register,采用 command(1) + address bytes(3) + data bytes(4)方式,支持 read 和 write 两种方式, 其中, address bytes 的 A23-A8 对应了 FPGA 内部的 register address 的低 16 位; 而 A7-A0 则是 dummy byte, 主要是为了让 ASIC 有足够的准备时间做读写操作。各种方式定义如下:

表 23 SPI Command Description

Instruction Name	Byte 1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Bytes 8
Read	0x03	A23-A16	A15-A8	A7-A0	D31-D24	D23-D16	D15-D8	D7-D0
Write	0x02	A23-A16	A15-A8	A7-A0	D31-D24	D23-D16	D15-D8	D7-D0

SPI 的 write 为单次写的方式, master 会在 CS#由高到低后的第一个 SCK 的 rising edge, 从 MOSI 输出 command。以 slave 看, 其顺序为 CS#拉低→write command(02H)→3byte address →4byte data→CS#拉高。

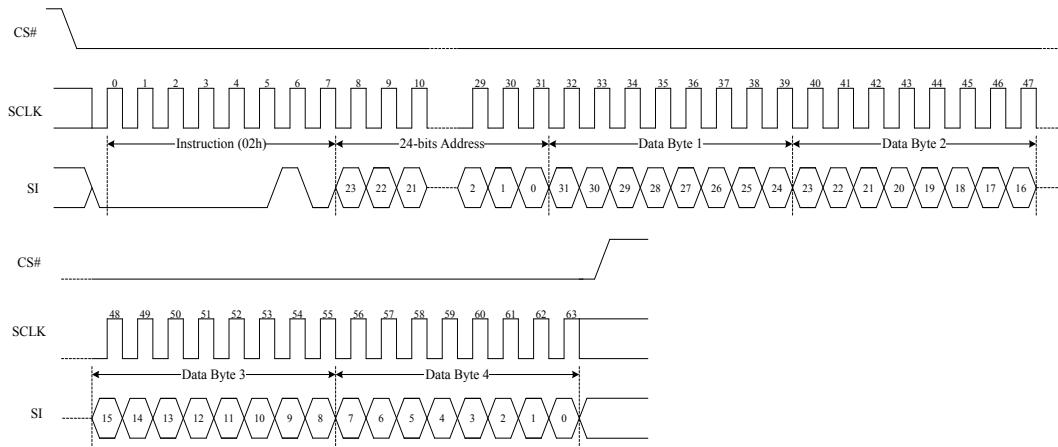


图 73 SPI Slave write 接口时序

Read 也为单次的方式，master 会在 CS#由高到低后的第一个 SCK 的 rising edge，从 MOSI 输出 command。Slave 在拿到 3byte 的地址后以下降沿推出 data。以 slave 看，其顺序为 CS#拉低→read command(03H)→3byte address→4byte data→CS#拉高。Slave 以下降沿推 data。

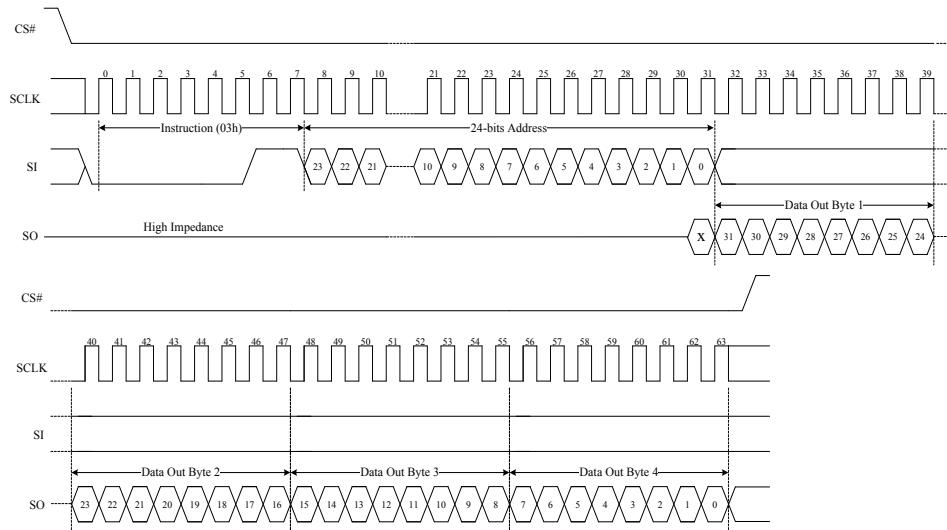


图 74 SPI Slave read 接口时序

8.3.8 Daisy-chain 模式

Daisy-chain 模式目前主要有两种实现方式，一种为串行的 Daisy-chain，一种为 SPI 的 Daisy-chain，串行的 Daisy-chain 如下所示：

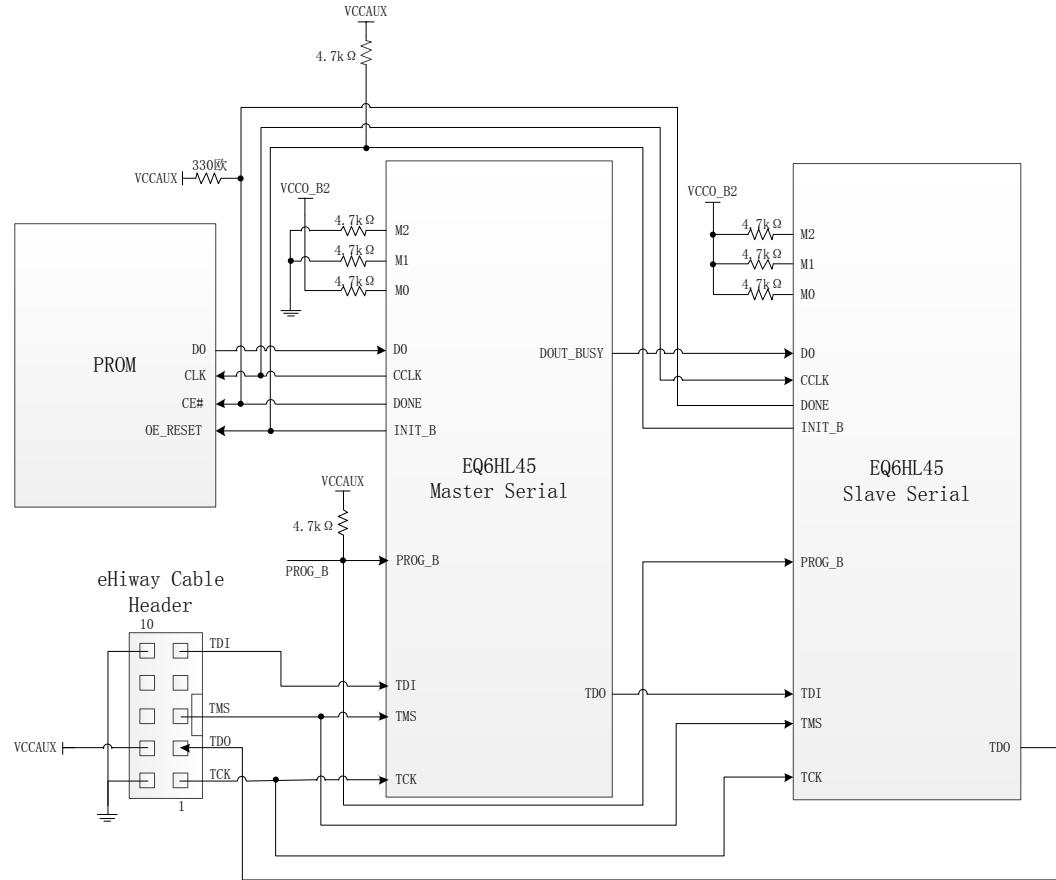


图 75 串行 Daisy-chain 框图

SPI 的 Daisy-chain 如下所示：

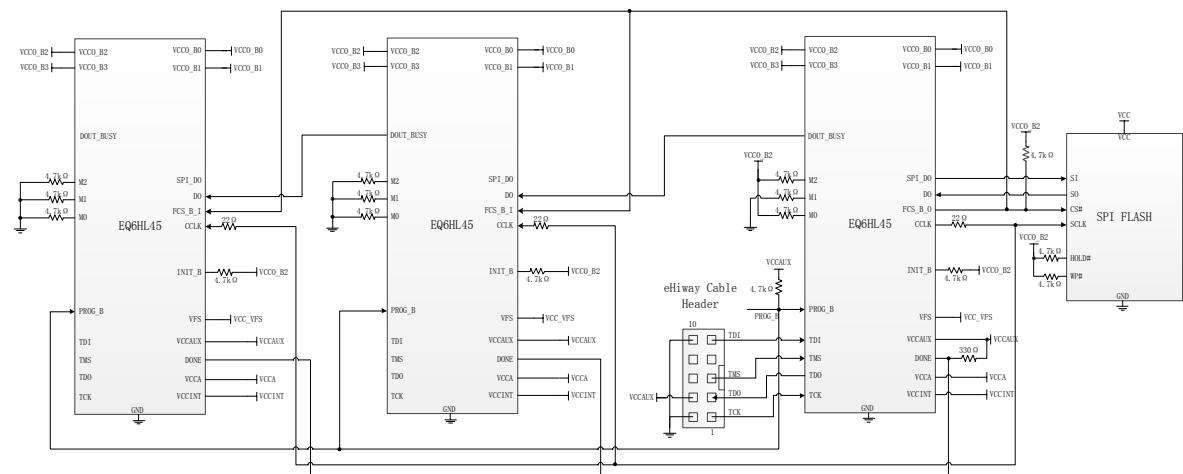


图 76 SPI Daisy-chain 框图

如上图所示，第一级为 SPI Master 模式，第二级和第三级为 SPI Chain slave 模式，将所有的 CS 以及 SCK 接到一起，Flash 中存放的为三段 FPGA 码流信息。

8.4 回读模式

8.4.1 被动并行回读模式

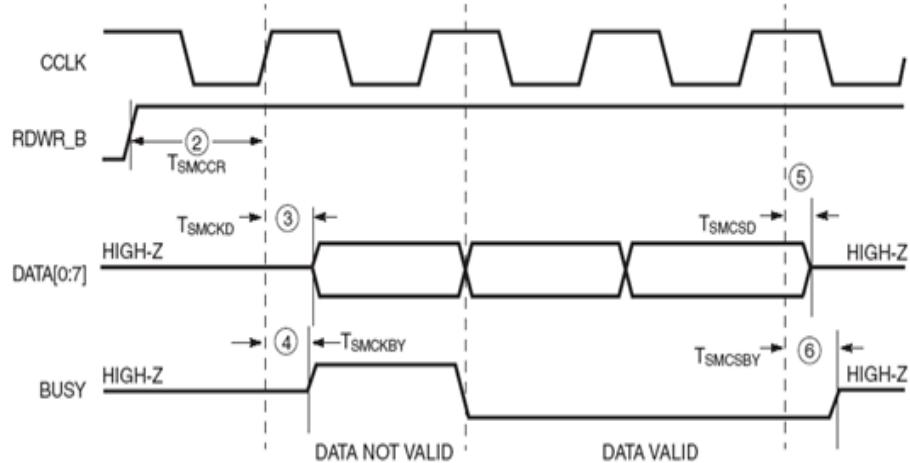


图 77 被动并行回读时序图

被动并行模式下的回读时序如图 77 所示。

8.4.2 JTAG 回读模式

JTAG 回读需先就加载 CFG_IN 指令，将各种回读命令写入相应配置寄存器，然后加载 CFG_OUT 指令，即可回读出相应的数据。回读流程如图 78 所示。

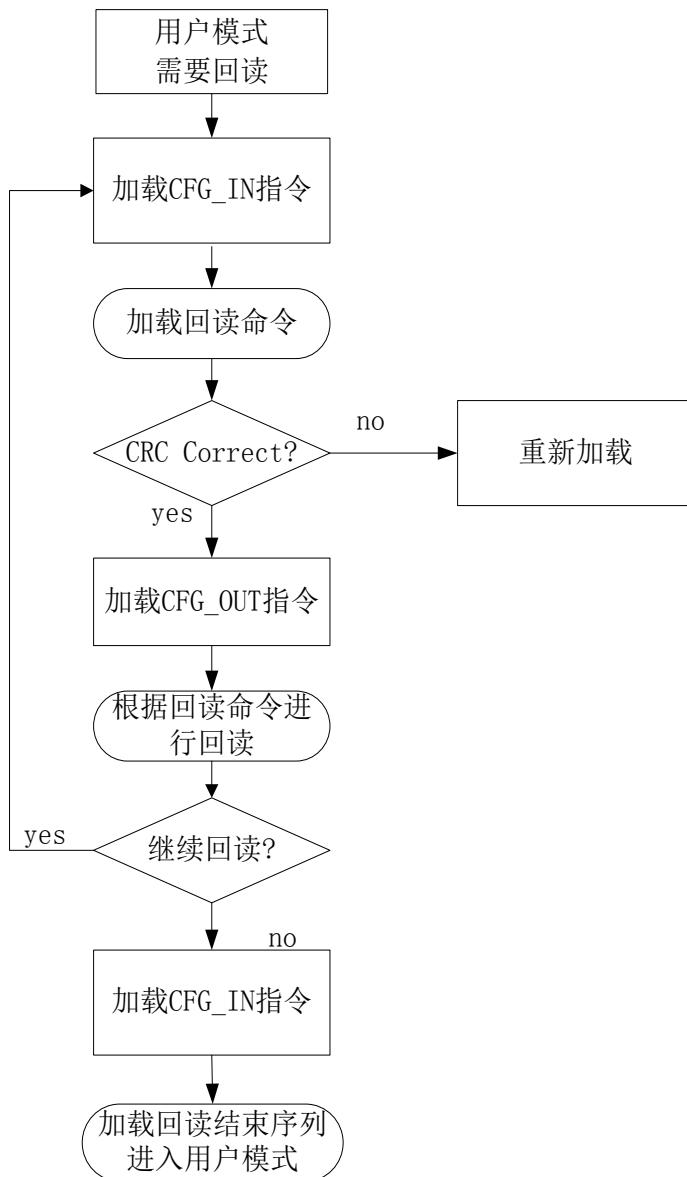


图 78 JTAG 回读流程

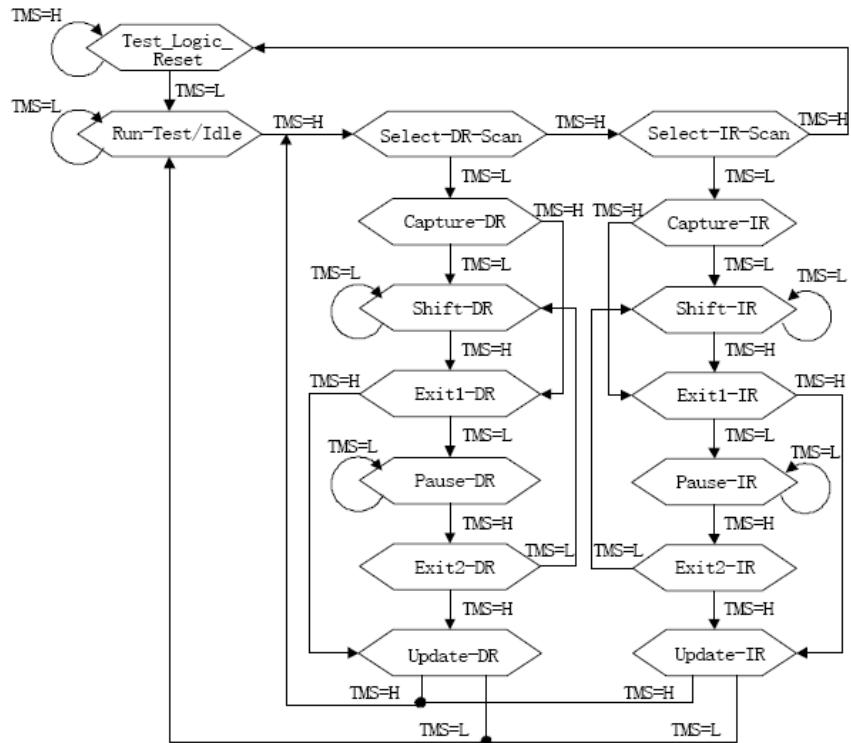


图 79 JTAG 状态转换图

表 24 JTAG 回读操作流程

TAP 状态跳变描述		TDI	TMS	TCK
1	JTAG 复位, TMS 为高电平, 保持五个时钟周期, 确保状态机进入 TLR(Test-Logic-Reset)状态	X	1	5
2	进入 RTI 状态	X	0	1
3	进入 Select-IR-State	X	1	2
4	进入 Shift-IR 状态	X	0	2
5	开始加载 CFG_IN 指令, 最低位先加载	00101	0	5
6	加载 CFG_IN 指令最高位, 同时进入 EXIT1-IR	0	1	1
7	进入 Select-DR-Scan 状态	X	1	2
8	进入 Shift-DR 状态	X	0	2
9	开始加载相应的回读命令	Bit1~Bitn	0	Bitn-1
10	加载最后一位回读命令, 同时进入 EXIT1-DR 状态	Bit0	1	1
11	同步复位进入 Test-Logic-Reset 状态	X	1	5
12	进入 RTI 状态	X	0	1
13	进入 Select-IR-State 状态	X	1	2
14	进入 Shift-IR 状态	X	0	2
15	开始加载 CFG_OUT 指令, 最低位先加载	00101	0	5
16	加载 CFG_OUT 指令最高位, 同时进入 EXIT1-IR 状态	0	1	1
17	进入 Select-DR-Scan 状态	X	1	2
18	进入 Shift-DR 状态, 开始回读数据	X	0	2
19	TDO 接收回读数据	X	0	N (Dout)

8.5 Multiboot

8.5.1 概述

FPGA 所具有的多重启动 (MultiBoot) 特性是指，允许用户在不掉电重启的情况下，根据不同时刻的需求，有选择地从存储于 Flash 中的多个配置文件中加载其中的一个，对 FPGA 在系统重配置，实现不同的功能。MultiBoot 特性使得用户可以在某些场合选择较少逻辑资源的 FPGA，实现需要较大逻辑资源且较昂贵的 FPGA 或 ASIC 才能实现的功能，从而提高器件利用率及系统安全性、降低系统成本。

在远程更新的时候，有时候需要双镜像来保护设计的稳定性。在进行更新设计的时候，只更新一个镜像，另一个镜像在部署之前就测试过没问题，所以一般不再更新。当更新出错时，通过正常配置成功的镜像进行一些操作来远程烧写 Flash，可以将更新失败的数据重新写入 Flash。这样即使更新出错，也能保证设计至少可以被远程恢复。

这里主要分两部分来加以介绍，一部分是通过 ICAP 的方式来实现 Multi-Boot 的操作，一部分是通过码流来实现 Multi-Boot 的操作。

8.5.2 Multi-boot 码流配置

上电后默认都是从 flash 地址 0x000000 位置开始数据的配置，默认一直加载数据到结束，当识别到 Multiboot 启动标志后，会根据设置的 Multiboot 地址进行跳转操作；当找到配置码流的时候，会进行正常的码流加载配置。当出现 CRC 错误或者 watchdog 超时等问题时，将根据设置的 Golden 地址进行跳转来实现 Fallback，从而实现 Golden 码流的加载和配置。码流在 flash 中的放置位置和配置流程图如下图所示。

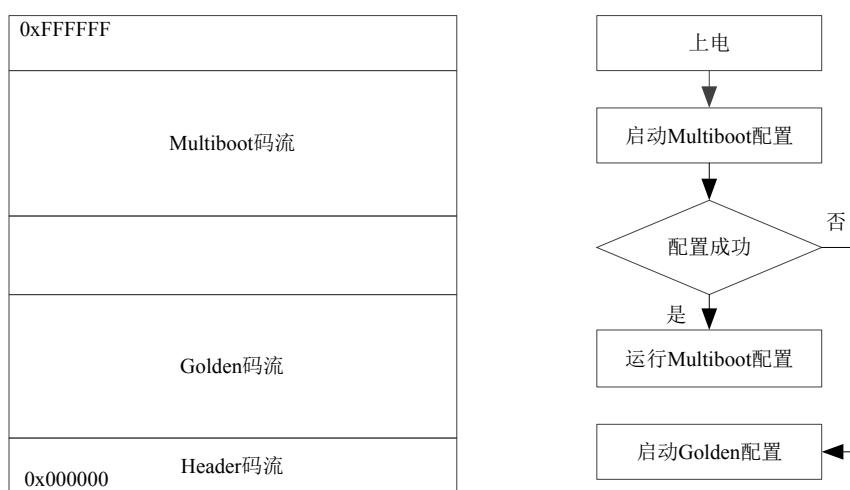


图 80 Multiboot 配置流程图

CRC error 指的是当发生码流校验的 crc 错误；watchdog error 是指在开始配置时，当超过一定的时间还未配置成功，则进行 fallback，这个时间可以在 EDA 软件中进行设置。

8.5.3 Multi-boot ICAP 配置

Multiboot ICAP 配置主要包括 ICAP 软核和 FPGA 芯核处理模块。系统电路结构如图 81 所示。

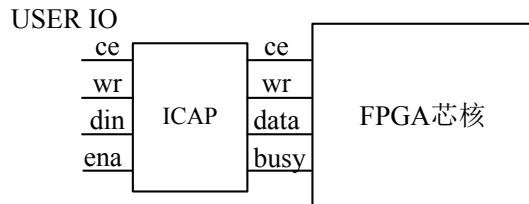


图 81 系统结构图

用户需要例化 ICAP 软核，通过 user IO 按照协议向 ICAP 软核发送相关数据，实现在线配置 Multiboot 启动地址和 Golden 启动地址，灵活控制 FPGA 重新配置加载的码流功能。

8.5.3.1 接口信号说明

表 25 ICAP 输入输出接口

Port_Name	Width	Direction	Description
CLK	1	输入	系统时钟
CE	1	输入	输入数据使能
WRITE	1	输入	读写控制信号
ICAP_ENA	1	输入	ICAP 模块使能信号
I	8	输入	8bit 输入数据
O	8	输出	8bit 输出数据
BUSY	1	输出	读取寄存器状态标志

- (1) CLK: ICAP 模块时钟输入。
- (2) CE: 输入数据使能, 低有效。
- (3) WRITE: 读写控制信号, 0=write, 1=read。
- (4) ICAP_ENA: ICAP 模块使能信号, 高有效, 需要使用 ICAP 功能时, 必须拉高。
- (5) I: 8bit 输入数据, 由用户输入 ICAP 协议数据, 来控制 golden 地址和 multiboot 地址。
- (6) O: 8bit 输出数据, 在回读寄存器状态时, 寄存器的值由这个接口输出。
- (7) BUSY: 读取寄存器状态标志, 高有效, 在写入数据时为低。

8.5.3.2 接口协议说明

表 26 ICAP 的接口协议

配置数据	说明
FFFF	无效数据
BBA	同步头
5566/5577	同步头, 5566 表示 3byte 寻址, 5577 表示 4byte 寻址
3261	Multi-boot 低 16 位地址同步头
xxxx	Multi-boot 起始地址[15:0]
3281	Multi-boot 高 16 位地址同步头
xxxx	Multi-boot 起始地址[31:16]
32A1	Fallback Golden 低 16 位地址同步头
xxxx	Fallback Golden 起始地址[15:0]
32C1	Fallback Golden 高 16 位地址同步头
xxxx	Fallback Golden 起始地址[31:16]
30A1	IPROG CMD 同步头
000E	IPROG CMD
2000	END

8.5.3.3 接口时序说明

icap 的接口时序说明如图所示。

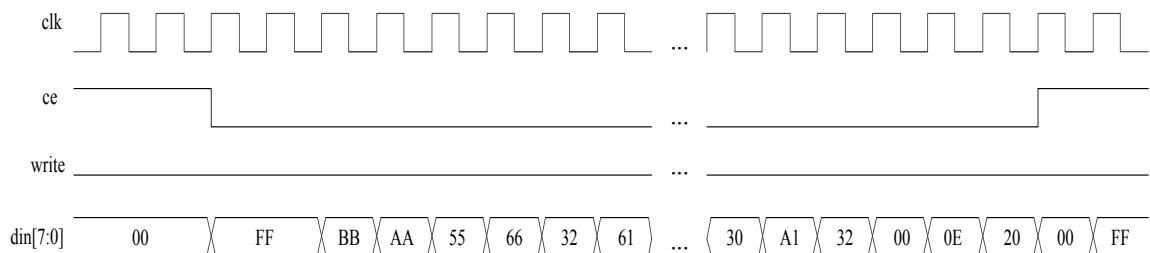


图 82 ICAP 接口时序图

8.5.4 状态寄存器

在 ICAP 中，通过 ICAP 接口可以读取 Multi-Boot 状态寄存器的状态，这样当发生问题的时候，可以知道是什么原因导致的问题。

在 EQ6HL45 中，定义的 8bit 如表所示：

表 27 JTAG 输入输出接口

	Reserved	Reserved	CRC_ERROR	ID_ERROR	WTO_ERROR	IPROG	FALLBACK	VALID
Status_1	0	0	0	0	0	0	0	1
Status_2	0	0	0	0	0	1	0	1

一共存在如下几种情况，以及每种情况下，分别对应的状态寄存器值如图 83 所示：

//following is the situation happened	Curt_status_0	Next_status_1
//1) bitload first done,	8'b0000_0001	8'b0000_0000(8'h0)
//2) bitload first fail, fallback as crc error	8'b0000_0*11	8'b0010_0101(8'h25)
//3) bitload first fail, fallback as id error	8'b0000_0*11	8'b0001_0101(8'h15)
//4) bitload first fail, fallback as timeout error	8'b0000_0*11	8'b0000_1101(8'h0d)
//5) bitload first done, icap success	8'b0000_0*01	8'b0000_0101(8'h05)
//6) bitload first done, icap fail, fallback as crc error	8'b0000_0*11	8'b0010_0101(8'h25)
//7) bitload first done, icap fail, fallback as id error	8'b0000_0*11	8'b0001_0101(8'h15)
//8) bitload first done, icap fail, fallback as timeout error	8'b0000_0*11	8'b0000_1101(8'h0d)

图 83 状态寄存器示意图

下图所示为 ICAP 回读状态寄存器的波形，通过 ICAP 接口的 cs, wr, din 信号来表示回读哪个状态寄存器，din[7:0]为 0 的时候表示回读状态寄存器 0，din[7:0]为 1 的时候表示回读状态寄存器 1。当回读成功时，busy 信号会拉高，外部用户可以通过采样 busy 信号的上升沿来获取数据。

两次回读中间，一定要将 cs 信号拉高。时序图如下图所示。

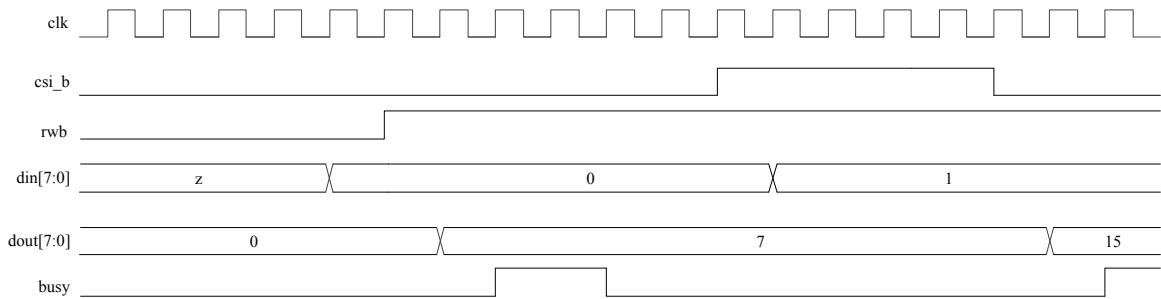


图 84 状态寄存器回读时序图

8.6 配置流程

配置流程由四个主要的步骤组成：清零（Clearing Configuration Memory）、初始化（Initialization）、加载配置数据（Loading Configuration Data）和芯片启动（Device Startup）。

（1）清零

芯片上电之初且电源电压尚未达到足够高的电压值时，POR 信号为低。随着电源电压的升高，POR 信号也逐渐升高。电源电压达到规定值后，POR 为配置控制器提供全局异步复位信号，对全部触发器和 CRAM 进行全局清零操作。POR 上的高电平保持一段时间（约 50ms）后，下降为 0，上电清零操作结束。PROG_B 是低电平有效的复位信号，在配置过程中从片外将其拉低会将配置控制器复位，重新开始新一轮配置过程。

（2）初始化

在初始化阶段，FPGA 将 INIT_B 释放为高阻，对 M2、M1、M0 进行采样，准备加载数据。如果从片外强行将 INIT_B 拉为低电平，则配置始终停留在初始化阶段，加载数据被推迟，直至 INIT_B 被释放到高且配置控制器在输入码流中检测到同步头为止。

（3）加载配置数据

配置控制器从码流中解析出配置信息，并将它们送到相应的配置存储器。通过 COR 寄存器中的 CRC_BYPASS 控制位设定是否对输入的码流进行 CRC 校验。如果选择进行 CRC 校验，则配置控制器在处理码流的同时会对码流中的有效数据进行 CRC 校验。如果校验通过，则配置正常进行。如果不通过，则中止配置并将 INIT_B 拉低。

(4) 芯片启动

芯片完成配置后，DONE 信号释放为高阻，IO 管脚完成解冻，用户电路开始工作。

EQ6HL45 的完整配置流程如图 85 所示。

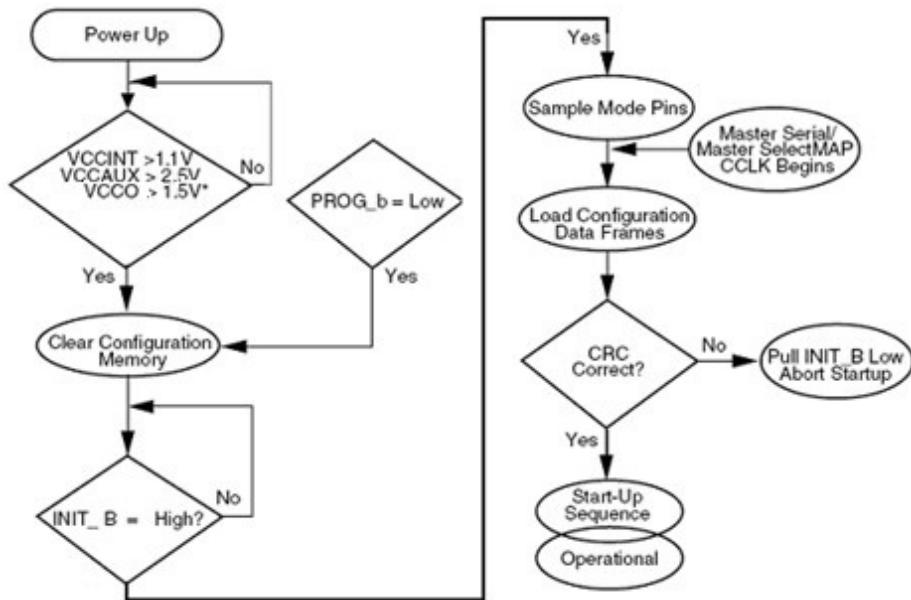


图 85 配置流程图

EQ6HL45 完整配置过程的时序如图 86 所示。

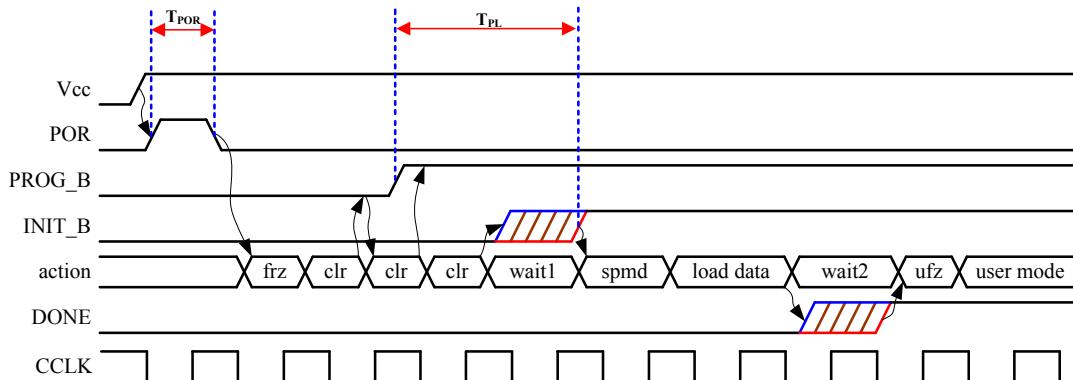


图 86 配置过程的时序图

表 28 上电配置时序参数

参数	说明	参数值		单位	备注
		最小值	最大值		
T_{POR}	上电后 POR 模块输出高电平的时间	8	12	ms	典型值 10ms
T_{PL}	从 PROG_B 上升沿到 INIT_B 上升沿的时间	6	—	ms	—
$T_{PROGRAM}$	将 PROG_B 拉低进行复位时，PROG_B 保持为低电平的时间	3	—	us	—

8.7 配置芯片

在选择配置芯片时，应综合考虑存储容量、电压和封装形式。在一般性商业应用中，建议使用型号如：华邦 W25Q128FVSQ、W25Q128JVSIQ、W25Q256JVFIQ、兆易创新 GD25VQ40、GD25Q80CSIG、GD25Q64CSIG、GD25Q127CSIG、GD25Q256DFIG、GD25Q256CFIQ 等 Flash 芯片作为 EQ6HL45 的配置芯片，如果需要使用主串、被串、主并、被并的下载方式，建议选择 XCF32P 型 PROM 作为 EQ6HL45 的配置芯片。

9 电特性参数

9.1 静态电特性参数

表 29 3.3V—LVTTL 标准

符号	引用标准	条件 (SJ/T10741-2000) -55°C≤TA≤125°C	极限值		单位
			最小值	最大值	
V _{DDIO}	3.3V— LVTTL 标准 (EIA/JEDEC Standard JESD8-B)		3.0	3.6	V
V _{IH}		V _O ≥V _{OH(min)}	2	V _{DDIO} +0.3	V
V _{IL}		V _O ≤V _{OL(max)}	-0.3	0.8	V
V _{OH}		V _{DDIO} =3.0V, I _{OH} =-2mA	2.4		V
V _{OL}		V _{DDIO} =3.0V, I _{OL} =2mA		0.4	V
I _{IH}		V _{DDIO} =3.6V, V _i =3.6V		±5	μA
I _{IL}		V _{DDIO} =3.6V, V _i =0V		±5	μA
I _{OH}				±24	mA
I _{OL}				±24	mA
I _{DD}			10	50	mA

表 30 3.3V—LVCMOS 标准

符号	引用标准	条件 (SJ/T10741-2000) -55°C≤TA≤125°C	极限值		单位
			最小值	最大值	
V _{DDIO}	3.3V— LVCMOS 标 准 (EIA/JEDEC Standard JESD8-B)		3.0	3.6	V
V _{IH}		V _O ≥V _{OH(min)}	2	V _{DDIO} +0.3	V
V _{IL}		V _O ≤V _{OL(max)}	-0.3	0.8	V
V _{OH}		V _{DDIO} =3.0V, I _{OH} =-1mA	V _{DDIO} -0.2		V
V _{OL}		V _{DDIO} =3.0V, I _{OL} =1mA		0.2	V
I _{IH}		V _{DDIO} =3.6V, V _i =3.6V		±5	μA
I _{IL}		V _{DDIO} =3.6V, V _i =0V		±5	μA
I _{OH}				±12	mA
I _{OL}				±12	mA
I _{DD}			10	50	mA

表 31 2.5V—I/O 标准

符号	引用标准	条件 (SJ/T10741-2000) -55°C≤TA≤125°C	极限值		单位
			最小值	最大值	
V _{DDIO}	2.5V—I/O 标 准 (EIA/JEDEC Standard EIA/JESD8- 5)		2.3	2.7	V
V _{IH}		V _O ≥V _{OH(min)}	1.7	V _{DDIO} +0.3	V
V _{IL}		V _O ≤V _{OL(max)}	-0.3	0.7	V
V _{OH}		V _{DDIO} =2.3V, I _{OH} =-2mA	1.7		V
V _{OL}		V _{DDIO} =2.3V, I _{OL} =2mA		0.7	V
I _{IH}		V _{DDIO} =2.7V, V _i =2.7V		±15	μA
I _{IL}		V _{DDIO} =2.7V, V _i =0V		±15	μA
I _{OH}				±16	mA
I _{OL}				±16	mA
I _{DD}			10	50	mA

表 32 1.8V-I/O 标准

符号	引用标准	条件 (SJ/T10741-2000) -55°C≤TA≤125°C	极限值		单位
			最小值	最大值	
V _{DDIO}	1.8V-I/O 标 准 (EIA/JEDEC Standard EIA/JESD8- 7A)		1.65	1.95	V
V _{IH}		V _O ≥V _{OH(min)}	0.65V _{DDIO}	V _{DDIO} +0.3	V
V _{IL}		V _O ≤V _{OL(max)}	-0.3	0.35V _{DDIO}	V
V _{OH}		I _{OH} =-2mA	V _{DDIO} -0.45		V
V _{OL}		I _{OL} =2mA		0.45	V
I _{OH}				±8	mA
I _{OL}				±8	mA
V _{OL}		I _{OL} =2mA		0.45	V

9.2 动态电特性参数

9.2.1 基本逻辑单元

表 33 LE 开关参数

参数	说明	参数值		单位	备注
		最小值	最大值		
t _{SU}	LE register setup time before clock	0.130		ns	
t _H	LE register hold time after clock	-0.090		ns	
t _{CO}	LE register clock-to-output delay	0.137	0.145	ns	
t _{LUT}	LE combinatorial LUT delay for data-in to data-out	0.271	0.388	ns	
t _{CLR}	Minimum clear pulse width	0.5		ns	
t _{PRE}	Minimum preset pulse width	0.5		ns	

表 34 Control Logic 开关参数

参数	说明	参数值		单位	备注
		最小值	最大值		
t _{CLK0}	CLK0 的延迟时间		0.61	ns	
t _{CLK1}	CLK1 的延迟时间		0.61	ns	
f _{CLK0}	CLK0 的频率		500	MHz	
f _{CLK1}	CLK1 的频率		500	MHz	
f _{CLKINDUTY}	输入时钟占空比 tolerance	40	60	%	
t _{NCLR0}	CLR0 的延迟时间		0.61	ns	
t _{NCLR1}	CLR1 的延迟时间		0.61	ns	
t _{ALOAD}	异步置数控制信号延迟时间		0.213	ns	
t _{ADDORSUB}	加减法控制信号延迟时间		0.213	ns	
t _{SLDORCLR}	同步置数延迟		0.213	ns	
t _{SYNCLR}	同步清零延迟时间		0.213	ns	

9.2.2 互连资源开关参数

表 35 互连资源开关参数

参数	说明	参数值		单位	备注
		最小值	最大值		
t_4	Delay for a 4 line with average loading; covers a distance of four LABs	—	0.250	ns	
t_{16}	Delay for a 16 line with average loading; covers a distance of four LABs	—	0.375	ns	
t_{LOCAL}	Local interconnect delay	—	0.2	ns	
t_{CB}	Connection Box delay	—	0.2	ns	
t_{DLINK}	快速直连线延迟	—	0.15	ns	

9.2.3 时钟网络和 PLL

时钟网络主要性能指标如表 36 所示。

表 36 时钟网络开关参数

参数	说明	参数值		单位	备注
		最小值	最大值		
f_{MAX}	Maximum frequency the clock tree can support for clocking registered logic	—	500	MHz	
t_{DCD}	Duty Cycle Distortion	—	2.49%	—	
t_{CKSKEW}	Clock Tree Skew	—	233	ps	
$t_{CKDELAY}$	Clock Tree Delay	—	2.03	ns	

表 37 PLL 参数

参数	说明	参数值			单位	备注
		最小值	典型值	最大值		
F_{in}	参考时钟频率	15		200	MHz	IO 限制 MAX
F_{vco}	VCO输出频率	600		1500	MHz	
M	环路分频数	2		64		Step=1
N	前置分频器分频数	1		64		
D_M	后置分频器分频数	2		64		
$T_{pha_shf_prg}$	相移设置	0.125			T _{vco}	支持
$T_{duty_cyc_prg}$	占空比设置	1			T _{vco}	占空比 C_n/C_m, C_m 为分频数, C_n 小于 C_m 且不大于 32
T_{locked}	锁定时间		200		T _{ref}	
$T_{pj}(pk-pk)$	周期抖动峰-峰值			≤±100	ps	输出频率大于 100MHz
t_{ARESET}	最小复位时间	10	—		ns	

9.2.4 I/O 通道

表 38 IOE 内部开关参数

参数	说明	参数值		单位	备注
		最小值	最大值		
t_{IOESU}	IOE register setup time before clock	0.102		ns	
t_{IOEH}	IOE register hold time after clock		-0.025	ns	
$t_{PIN2COMBOUT}$	input pin to IOE combinatorial output	0.272	1.213	ns	
$t_{COMBIN2PIN}$	IOE data input to combinatorial output pin	0.282	1.194	ns	
t_{IOECLR}	Minimum IOE clear pulse width	0.26		ns	

表 39 可编程延迟开关参数

参数	说明	后端仿真结果		单位	备注
		最小值	最大值		
可编程延迟时间	零延时	0.310	0.362	ns	
	细调 1	0.437	0.487	ns	
	细调 2	0.561	0.616	ns	
	细调 3	0.702	0.753	ns	
	细调 4	0.824	0.880	ns	
	细调 5	0.971	1.010	ns	
	细调 6	1.098	1.132	ns	
	细调 7	1.232	1.266	ns	
	细调 8	1.332	1.376	ns	
	粗调 1	1.163	1.213	ns	
	粗调 2	1.981	2.032	ns	
	粗调 3	2.817	2.876	ns	

9.2.5 嵌入式存储器 BRAM

表 40 M4K 模块开关参数

参数	说明	参数值		单位	备注
		最小值	最大值		
T_{BCKO}	BRAM 中时钟 CLK 到 DOUT 输出（同步）		0.87	ns	
T_{BCKOA}	BRAM 中时钟 CLK 到 DOUT 输出（异步）		2.25	ns	
T_{MKACK}	BRAM 中 ADDR 输入的建立时间	-0.22		ns	
T_{MKCKA}	BRAM 中 ADDR 输入的保持时间		0.29	ns	
T_{MKDCK}	BRAM 中 DIN 输入的建立时间	-0.51		ns	
T_{MKCKD}	BRAM 中 DIN 输入的保持时间		0.58	ns	
T_{MKWCK}	BRAM 中 WREN 输入的建立时间	-0.26		ns	
T_{MKCKW}	BRAM 中 WREN 输入的保持时间		0.33	ns	
T_{MKRCK}	BRAM 中 RDEN 输入的建立时间	-0.26		ns	

参数	说明	参数值		单位	备注
		最小值	最大值		
T_{MKCKR}	BRAM 中 RDEN 输入的保持时间		0.33	ns	
T_{MKBYCK}	BRAM 中 BYTEN 输入的建立时间	-0.36		ns	
T_{MKCBY}	BRAM 中 BYTEN 输入的保持时间		0.43	ns	
t_{MKCLR}	BRAM 中清零信号最小脉宽	0.51		ns	

9.2.6 DSP

表 41 DSP 开关参数

参数	说明	参数值		单位	备注
		最小值	最大值		
T_{DSPSU}	DSP 输入寄存器、pipeline 寄存器、输出寄存器的建立时间	1.27		ns	
T_{DSPH}	DSP 输入寄存器、pipeline 寄存器、输出寄存器的保持时间		-2.31	ns	
T_{DSPCO}	DSP 输入寄存器、pipeline 寄存器、输出寄存器 clk-out 延迟		2.42	ns	
T_{PIPE9}	$9 \times 9\text{-bit}$ 模式下，输入寄存器到 pipeline 寄存器的延迟		0.79	ns	
T_{PIPE18}	$18 \times 18\text{-bit}$ 模式下，输入寄存器到 pipeline 寄存器的延迟		1.24	ns	
$T_{PIPE2ADD}$	2 乘加模式下，pipeline 寄存器到输出寄存器的延迟		1.32	ns	
$T_{PIPE4ADD}$	4 乘加模式下，pipeline 寄存器到输出寄存器的延迟		1.44	ns	
T_{PD9}	$9 \times 9\text{-bit}$ 模式下，输入到输出的组合延迟		1.98	ns	
T_{PD18}	$18 \times 18\text{-bit}$ 模式下，输入到输出的组合延迟		3.10	ns	
T_{PD36}	$36 \times 36\text{-bit}$ 模式下，输入到输出的组合延迟		4.23	ns	
T_{DSPCLR}	DSP 时钟信号最小高电平脉宽	1.27		ns	
$T_{DSPCLRL}$	DSP 时钟信号最小低电平脉宽	1.07		ns	
T_{DSPHL}	DSP 最小清零信号脉宽	1.01		ns	

10 板级电接口设计参考

10.1 电源部分设计

10.1.1 电源电压需求

管脚排布和 Xilinx XC6SLX45 兼容，可参考 Xilinx 所提供参考设计或者成熟的已有设计，所需电压如表 42。

表 42 电平标准参数说明

Symbol	Parameter	Min	Typ	Max	Unit
VCCINT	芯核供电电压	1.05	1.1	1.155	V
VCCAUX	专用 I/O 供电电压	2.375	2.5	2.625	V
		3.135	3.3	3.465	V
VCCA	PLL 电源供电电压	2.375	2.5	2.625	V
		3.135	3.3	3.465	V

注：虽然器件本身对上电顺序不敏感，但建议芯核电源 Vccint 不晚于 VCCIO 和 VCCAUX 上电以保证 IO 在上电期间保持三态。

10.1.2 PLL 电源部分设计

EQ6HL45 中有 PLL 这样的模拟器件，为了降低数字部分产生的噪音对模拟器件的影响，需要隔离电源管脚与地管脚。

可以使用两种隔离 V_{CCA} 的方式：

方法一：使用一个 LDO 单独输出一个电源层 PLANE 给 V_{CCA} 供电，100mA 足够。

方法二：在 V_{CCAUX} 上面隔离出来一块，或者采用一条较宽的 V_{CCA} 线，至少 20mil。同时使用一个磁珠，对每个 V_{CCA} 进行滤波，磁珠选择对 50Mhz 或者更高频率呈现高阻，同时在电源输入处用 10uF 电容进行滤波。

其中任何一种方式，都需要用如图 87 方框中所示的去耦电路，在靠近 V_{CCA} 管脚的地方，用 0.1uF 以及 0.001uF 的陶瓷电容进行去耦。

可以把 GNDA 直接连接到电源数字地上面。

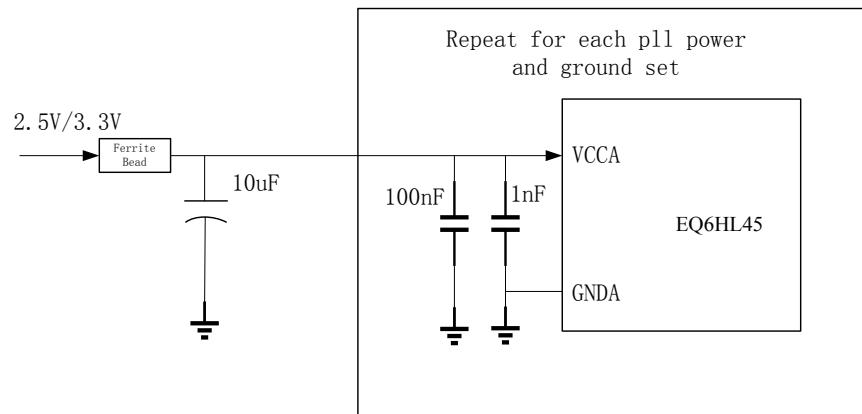


图 87 PLL 电源部分设计

附录 A
缩写对应表

编号	缩写	含义
001	BLE	基本逻辑单元
002	BLK	逻辑模块
003	BRAM	嵌入式可编程存储器
004	CL	控制逻辑模块
005	CP/LF	电荷泵/环路滤波器
006	CRAM	配置存储器
007	DCI	数字控制终端电阻
008	DICE	传统的双互锁
009	DSP	数字信号处理/嵌入式乘法器
010	EDAC	纠错编码
011	HSBS	创新的后台刷新
012	IM	输入信号选择电路
013	IOE	可编程 IO 单元
014	LC	逻辑模块
015	LUT	4 入查找表
016	MAC	乘累加模式
017	OD	漏极开路
018	PFD	鉴频鉴相器
019	PLL	可编程锁相环
020	SEU	抗单粒子翻转
021	SRAM	静态存储器
022	Switch	连线通道交叉开关
023	TILL	创新的三互锁
024	Track	连线通道
025	VCO	振荡器

免责声明

本文档并未授予任何知识产权的许可。除中科亿海微在其产品的销售条款和条件中声明的责任之外，中科亿海微概不承担任何法律或非法律责任。中科亿海微对文档中包含的文字、图片及其他内容的准确性和完整性不承担任何法律或非法律责任，中科亿海微保留修改文档中任何内容的权利。