

第9章

32位OpenRISC软核结构及应用

9.1 OpenRISC1200处理器核概述

9.1.1 OpenRISC 1000/1200处理器的体系结构

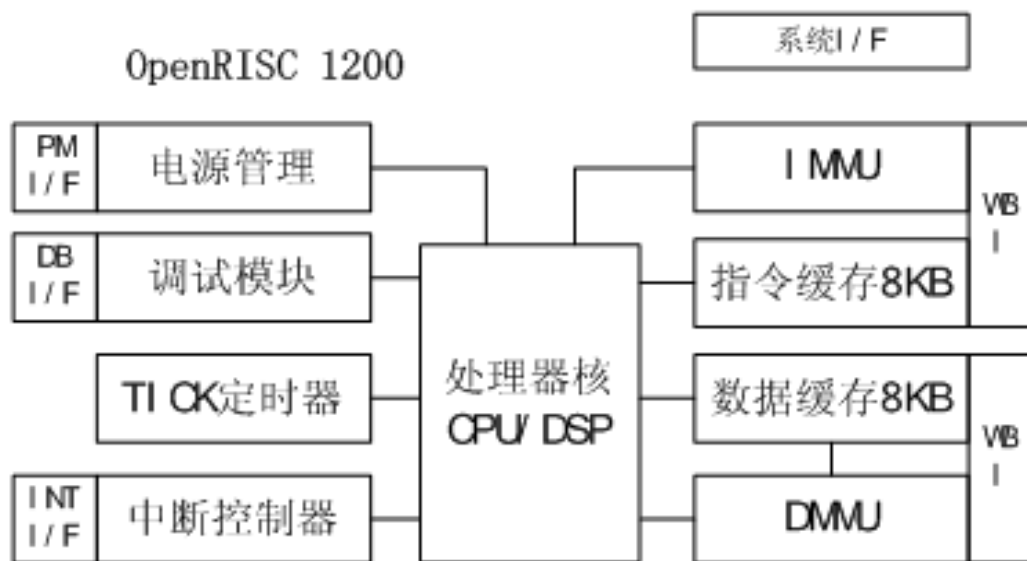


图 9-1 OR1200 核内部结构

9.1 OpenRISC1200处理器核概述

表 9-1 OpenRISC 1000 的主要性能

类型	性能与特性	补充描述
源码开放	免费开源	遵循 GNU LGPL 协议
指令集	基本指令集 ORBIS32/64	32 位宽度，在存储器中以 32 位地址对齐存储，可操作 32/64 位数据
	向量/DSP 扩展指令集 ORVDX64	32 位宽度，可操作 8/16/32/64 位数据
	浮点扩展指令集 ORFPX32/64	32 位宽度，可操作 32/64 位数据
	线性的 32 位或 64 位	在不同的实现中有不同的物理地址空间
逻辑地址空间	1, 寄存器数据加有符号 16 位立即数	得到有效地址
内存地址模式	2, 寄存器数据加一个存放有效地址的寄存器所指向的地址中的有符号 16 位立即数	
	大多数指令为两个寄存器操作数（或一个寄存器和一个立即数）操作，结果写回第三个寄存器	
操作数	支持影子寄存器或者一个 32 项或者一个 16 项的寄存器文件	
寄存器	分支跳转延时槽	保持指令流水线的高效运行
指令流水线	支持独立分开的指令和数据缓存，MMU	哈佛结构
内存管理	支持统一的指令和数据缓存，MMU	标准结构
	支持快速上下文切换	
	支持 Cache 和 TLB 大小可配置	
可伸缩性	动态电源管理	

9.1 OpenRISC1200处理器核概述

9.1.1 OpenRISC 1000/1200处理器的体系结构

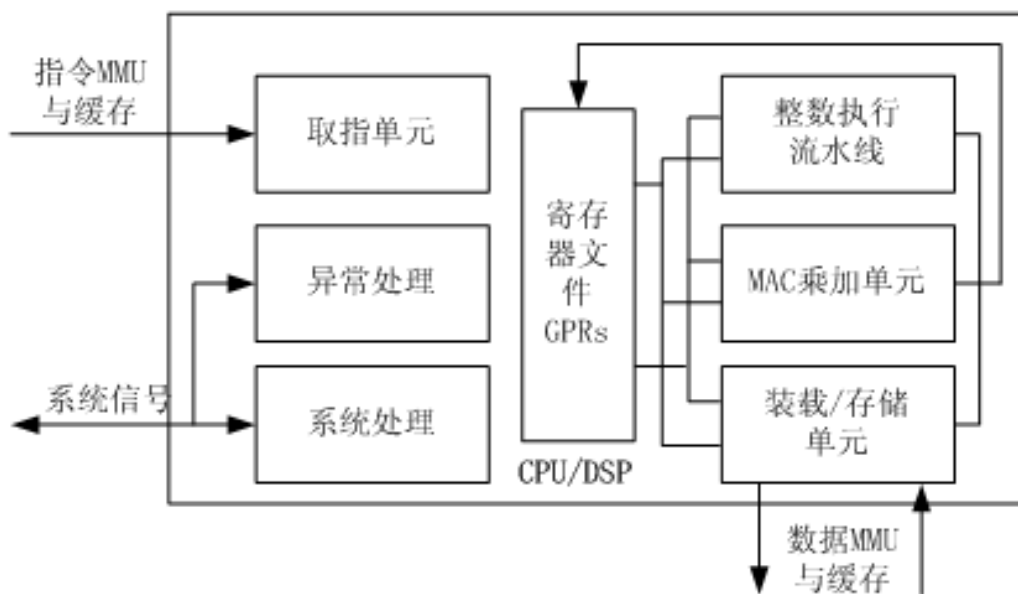


图 9-2 CPU/DSP 内部结构

9.1 OpenRISC1200处理器核概述

9.1.2 OR1200指令集及指令流水线

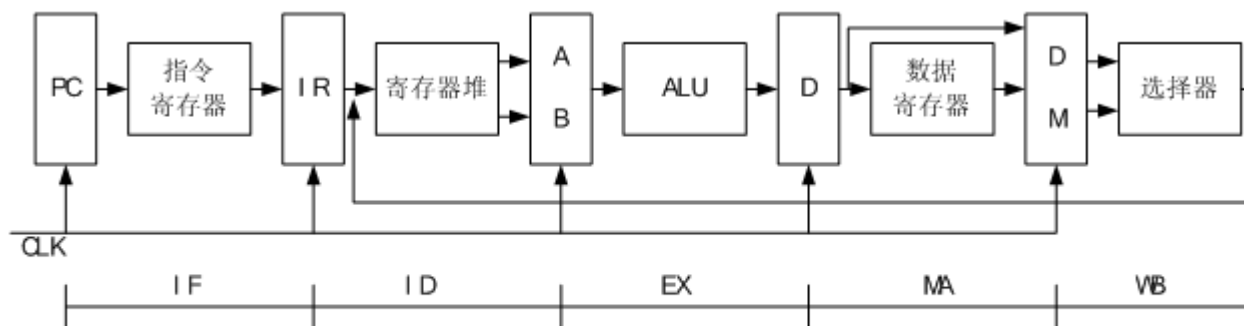


图 9-3 OR1200 的 5 级流水线

9.1 OpenRISC1200处理器核概述

9.1.3 OR1200核的异常模型和可编程中断控制器

表 9-2 OR1200 的异常向量表

异常类型	异常向量 默认配置	异常向量 实际配置	异常产生条件
复位	0x100	0x10	复位输入信号有效
总线错误	0x200	0x20	访问无效的物理地址
数据页失败	0x300	0x30	在页表中发现不匹配的 PTE 或者访存操作中页保护违规
指令页失败	0x400	0x40	在页表中发现不匹配的 PTE 或者在指令预取中页保护违规
Tick 定时器	0x500	0x50	Tick 定时器中断
对齐	0x600	0x60	访存地址不对齐的位置
非法指令	0x700	0x70	在指令流中非法的指令
外部中断	0x800	0x80	外部中断信号有效
D-TLB 未命中	0x900	0x90	在数据 TLB 中没有匹配的条目
I-TLB 未命中	0xA00	0xA0	在指令 TLB 中没有匹配的条目
系统调用	0xC00	0xC0	软件产生的系统调用
断点	0xD00	0xD0	调试单元引起的异常

9.1 OpenRISC1200处理器核概述

9.1.3 OR1200核的异常模型和可编程中断控制器

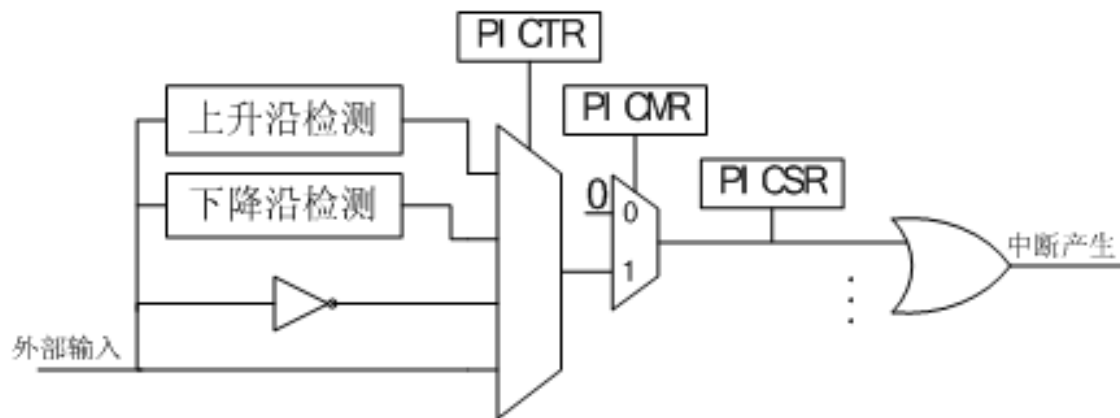


图 9-4 可编程中断控制器结构

9.1 OpenRISC1200处理器核概述

9.1.4 OR1200核的寄存器

表 9-3 特殊功能寄存器组

寄存器组号	寄存器组说明
0	系统控制和状态寄存器
1	数据 MMU (在只有一个统一 MMU 情况下, 组 1 和组 2 作为单个组)
2	指令 MMU
3	数据缓存 (在只有一个统一缓存情况下, 组 3 和组 4 作为单个组)
4	指令缓存
5	MAC 单元
6	调试单元
7	执行计数单元
8	电源管理单元
9	可编程中断控制器
10	Tick 定时器
11	浮点单元
12-23	保留
24-31	用户自定义单元

9.1 OpenRISC1200处理器核概述

9.1.4 OR1200核的寄存器

表 9-4 第 0 组特殊功能寄存器

寄存器编号	寄存器名	说明
0	VR	版本寄存器
1	UPR	单元有效寄存器
2	CPUCFGR	CPU 配置寄存器
3	DMMUCFGR	数据 MMU 配置寄存器
4	IMMUCFGR	指令 MMU 配置寄存器
5	DCCFGR	数据缓存配置寄存器
6	ICCFGR	指令缓存配置寄存器
7	DCFGR	调试单元配置寄存器
8	PCCFGR	电源管理单元配置寄存器
16	NPC	PC 在 SPR 空间下的映射, 下一个 PC
17	SR	超级监管者寄存器
18	PPC	PC 在 SPR 空间下的映射, 前一个 PC
20	FPCSR	FP 控制状态寄存器
32-47	EPCRn	异常 PC 寄存器
48-63	EEARn	异常 EA 寄存器
64-79	ESRn	异常 SR 寄存器
1024-1535	GPRn	GPR 在 SPR 空间下的映射

9.1 OpenRISC1200处理器核概述

9.1.5 OR1200核的Tick定时器

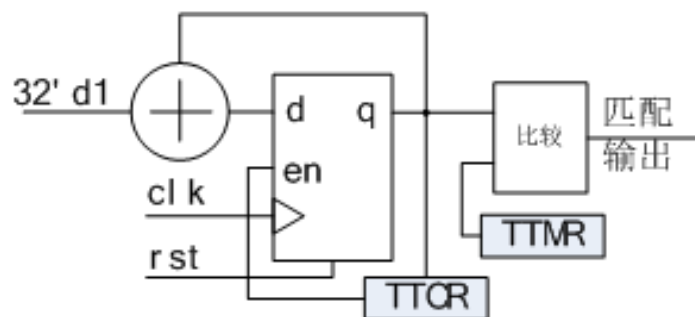


图 9-5 Tick 定时器结构

表 9-5 TTCR 寄存器各位域代表的意义

31	30	29	28	27:0
SR	CR	IE	IP	CNT
单次运行	连续运行	中断使能	中断标志	当前计数值

9.2 WISHBONE 片上总线

9.2.1 WISHBONE总线概述

- 1、支持点对点互联、共享总线互联、交叉互联和基于交换结构的互联。
- 2、支持全功能的数据读写操作，包括单次读写操作，块读写操作，读修改写操作。
- 3、允许从设备进行部分地址译码。有利于减少冗余地址译码逻辑，提高译码速度。
- 4、支持用户自定义标签信号。

9.2 WISHBONE 片上总线

9.2.2 WISHBONE接口信号说明

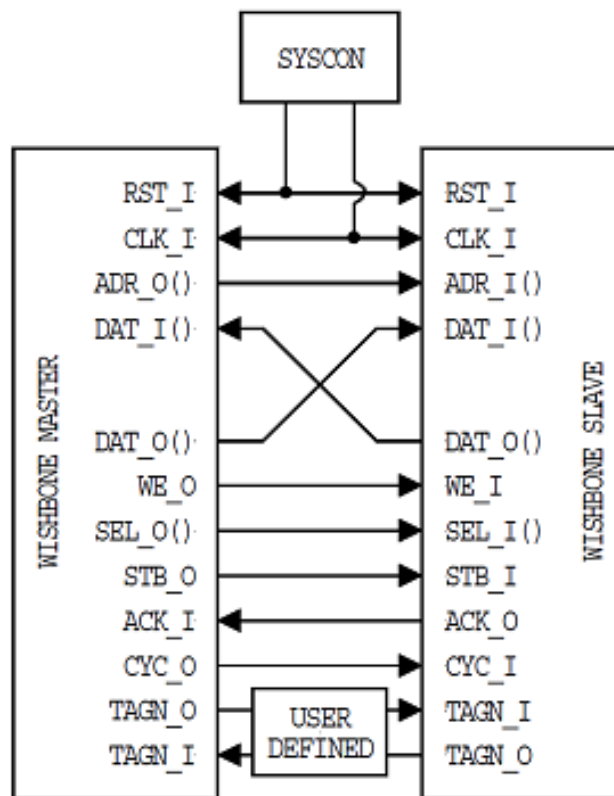


图 9-6 WISHBONE 标准互联图

9.2 WISHBONE 片上总线

9.2.3 WISHBONE总线协议与数据传输

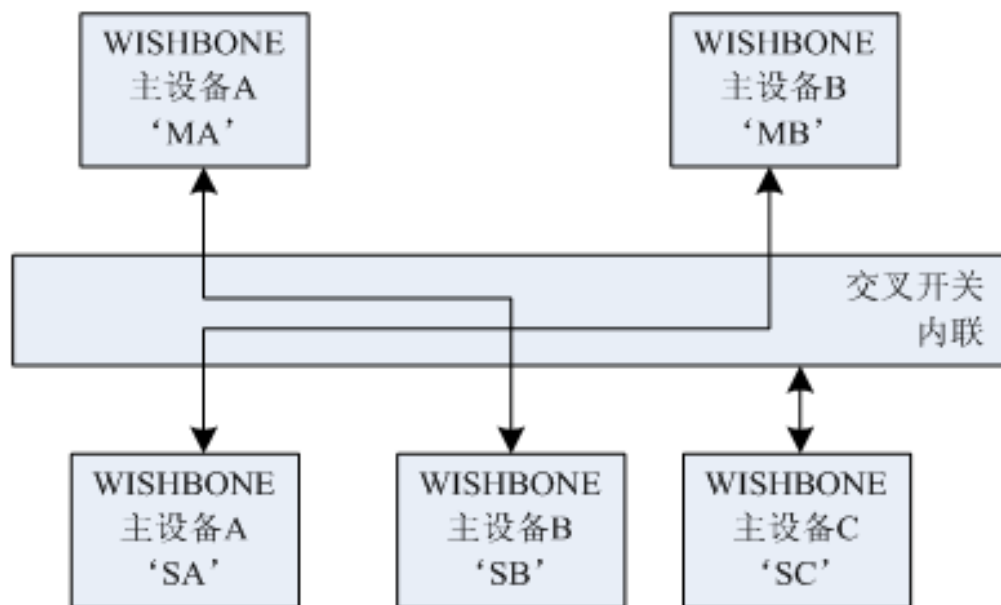


图 9-7 WISHBONE 总线交叉互联方式

9.2 WISHBONE 片上总线

9.2.3 WISHBONE总线协议与数据传输

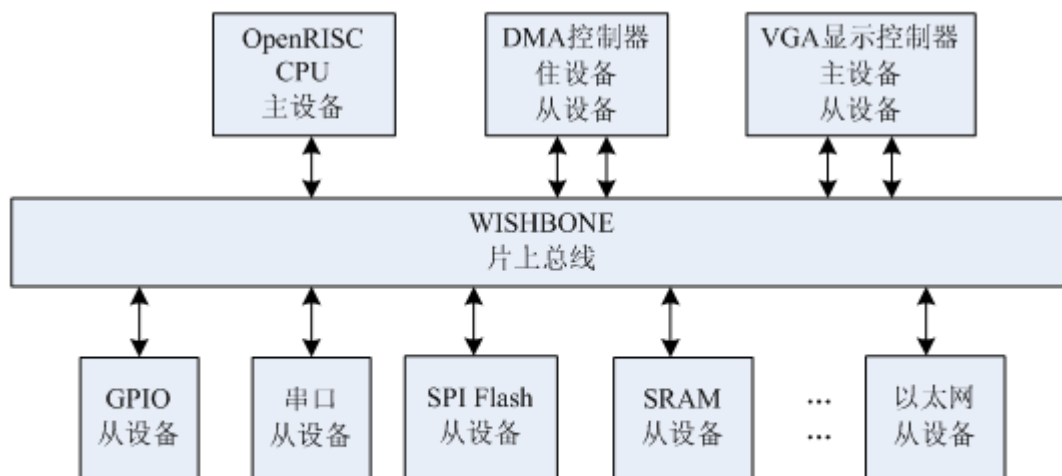


图 9-8 基于 WISHBONE 总线的 OpenRISC 典型应用

9.3 OpenRISC的软件开发环境

9.3.1 OpenRISC的GNU工具链

- **add2line**: 把程序地址转换成文件名和行号;
- **ar**: 建立、修改和提取归档文件;
- **as**: 汇编器, 主要用来编译 GNU C 编译器 `gcc` 输出的汇编文件, 产生的目标文件传递给接器 `ld`;
- **c++filt**: 链接器使用它来过滤 C++ 和 Java 符号, 防止重载函数冲突;
- **gprof**: 显示程序调用段的各种数据;
- **ld**: 链接器, 把目标文件和归档文件链接在一起, 重定位数据并链接符号引用;
- **nm**: 列出目标文件中的符号;
- **objcopy**: 在目标文件之间拷贝内容;
- **objdump**: 显示目标文件信息;
- **ranlib**: 产生归档文件索引, 并将其保存到这个归档文件中;
- **readelf**: 显示 `elf` 格式可执行文件的信息;
- **size**: 列出目标文件每一段及总体大小;
- **strings**: 打印某个文件的可打印字符串;
- **strip**: 删除目标文件中特定符号

9.3 OpenRISC的软件开发环境

9.3.2 使用Makefile管理工程

9.4 一个简单的OR1200核的SOC设计示例

9.4.1 KX_OR1200_SOC概述及设计流程

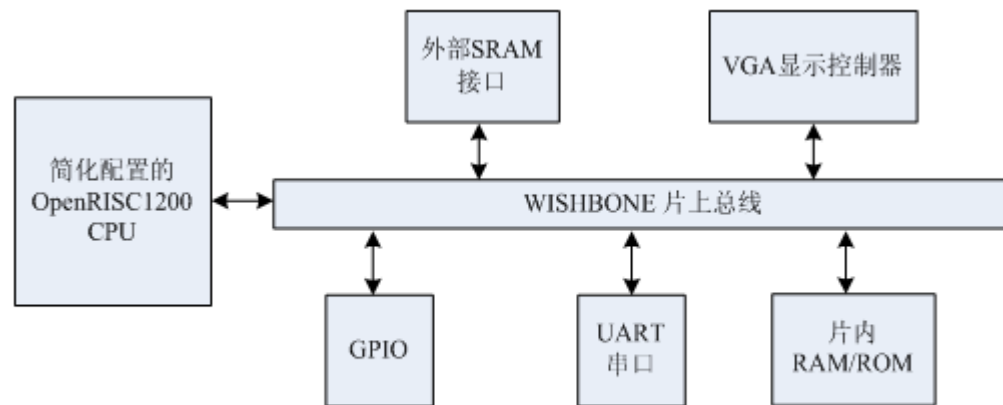


图 9-9 KX_OR1200_SOC 的基本结构

9.4 一个简单的OR1200核的SOC设计示例

9.4.1 KX_OR1200_SOC概述及设计流程

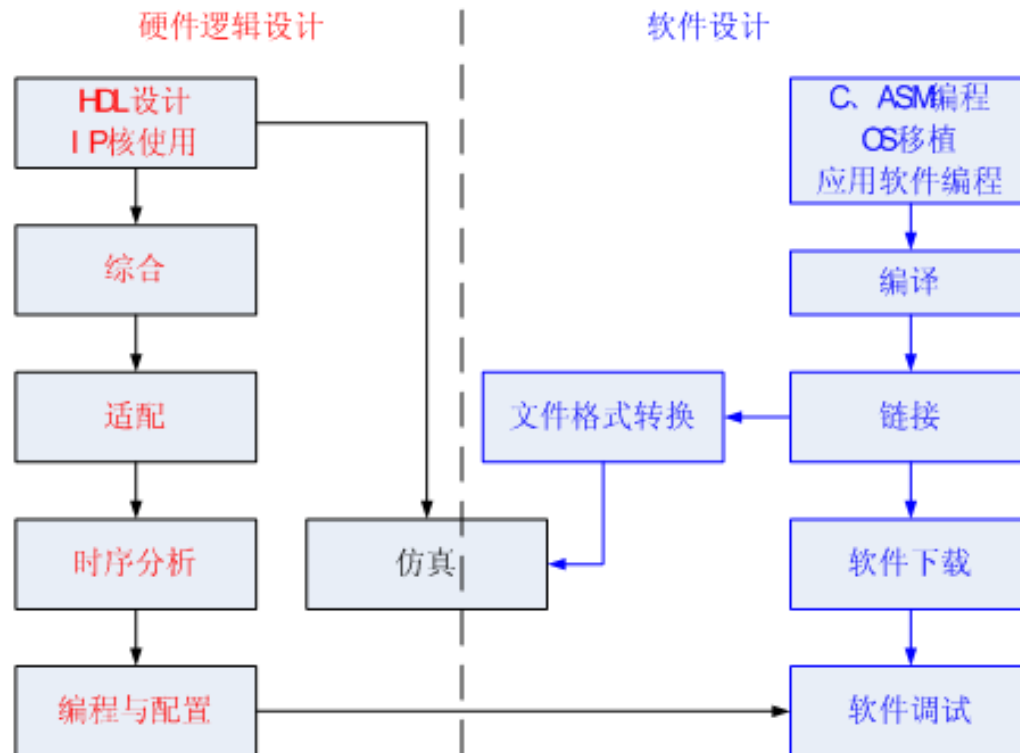


图 9-10 设计流程图

9.4 一个简单的OR1200核的SOC设计示例

9.4.2 KX_OR1200_SOC的存储器结构及初始化

9.4.3 GPIO通用输入输出端口

表 9-6 GPIO IP 核寄存器列表 (参考图 9-11)

寄存器名	地址	长度	说明
GPIO_IN	0x30010000	32 bits	GPIO 输入数据寄存器
GPIO_OUT	0x30010004	32 bits	GPIO 输出数据寄存器
GPIO_OE	0x30010008	32 bits	GPIO 输出使能寄存器
GPIO_INTE	0x3001000C	32 bits	GPIO 输入中断使能寄存器
GPIO_PTRIG	0x30010010	32 bits	GPIO 输入中断触发类型
GPIO_AUX	0x30010014	32 bits	GPIO 端口复用输出寄存器
GPIO_CTRL	0x30010018	2 bits	GPIO 控制寄存器
GPIO_INTS	0x3001001C	32 bits	GPIO 输入中断状态寄存器
GPIO_ECLK	0x30010020	32 bits	GPIO 外部时钟使能寄存器
GPIO_NEC	0x30010024	32 bits	GPIO 外部时钟边沿选择寄存器

9.4 一个简单的OR1200核的SOC设计示例

9.4.3 GPIO通用输入输出端口

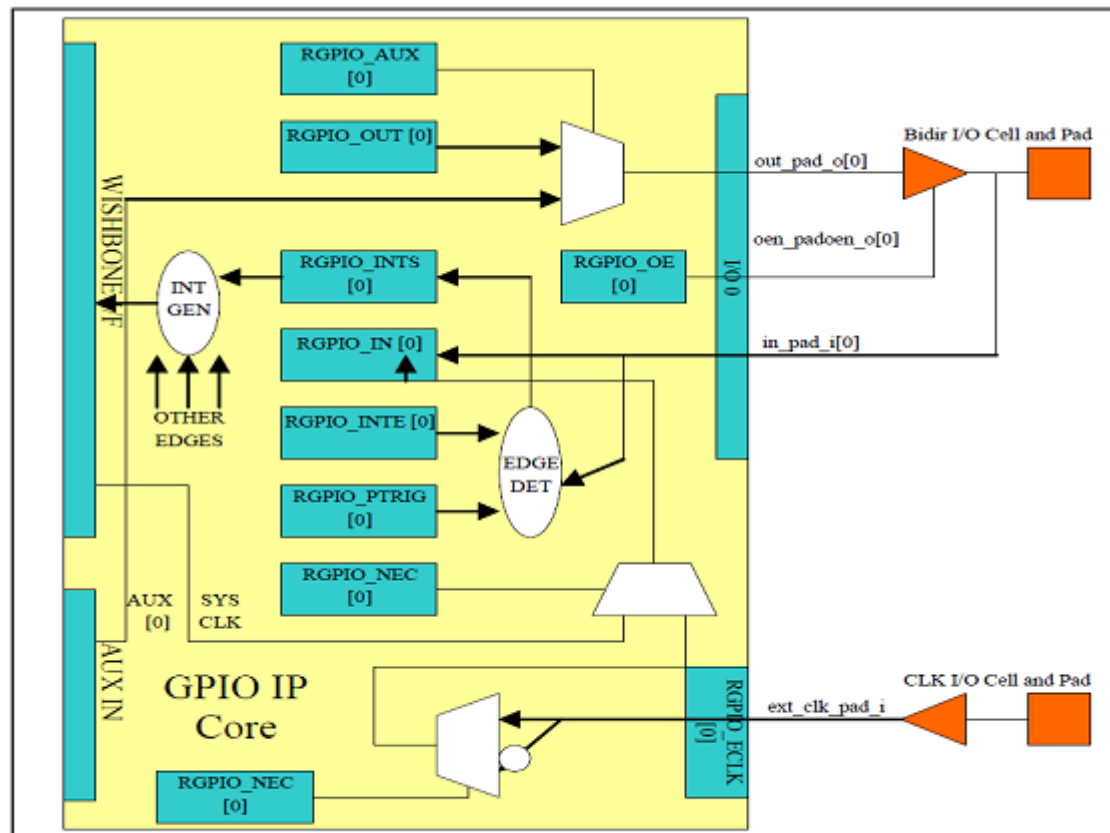


图 9-11 10 GPIO IP 核内部结构

9.4 一个简单的OR1200核的SOC设计示例

9.4.3 GPIO通用输入输出端口

表 9-7 LED 数码管 及按键控制模块寄存器列表

寄存器名	地址	长度	说明
键 2: KEY_IN	0x30020000	4 bits	按键输入数据寄存器
键 3: KEY_INTE	0x30020004	4 bits	按键输入中断使能寄存器
键 4: KEY_PTRIG	0x30020008	4 bits	按键输入中断触发类型
键 5: KEY_INTS	0x3002000C	4 bits	按键输入中断状态寄存器

9.4 一个简单的OR1200核的SOC设计示例

9.4.4 UART16550串行通信模块应用

表 9-8 uart16550 寄存器列表

寄存器名	地址	长度	说明
UART_RBR	0x30000000	8 bits	接收数据缓存寄存器（只读）
UART_THR	0x30000000	8 bits	发送数据保持寄存器（只写）
UART_IER	0x30000001	8 bits	中断时能寄存器
UART_IIR	0x30000002	8 bits	中断类型识别寄存器（只读）
UART_FCR	0x30000002	8 bits	FIFO 控制寄存器（只写）
UART_LCR	0x30000003	8 bits	通信线控制寄存器
UART_MCR	0x30000004	8 bits	调制解调控制寄存器
UART_LSR	0x30000005	8 bits	通信线状态寄存器
UART_MSR	0x30000006	8 bits	调制解调状态寄存器
UART_DLL	0x30000000	8 bits	波特率分频寄存器低字节
UART_DLH	0x30000001	8 bits	波特率分频寄存器高字节

9.4 一个简单的OR1200核的SOC设计示例

9.4.5 VGA/LCD显示控制器设计

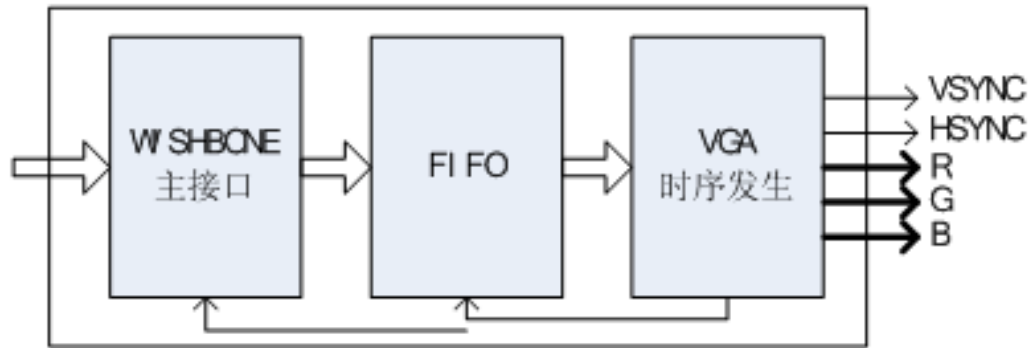


图 9-12 11 VGA/LCD 显示控制器内部结构

表 9-9 VGA 时序

总体时序		
屏幕刷新频率	60 Hz	
行刷新频率	31.46875 kHz	
像素刷新频率	25.175 MHz	
水平刷新时序		
水平扫描阶段	像素个数	时间(us)
可视区域	640	25.422
前沿	16	0.636
同步脉冲	96	3.813
后沿	48	1.907
整行	800	31.778
垂直刷新时序		
垂直扫描阶段	行数	时间(ms)
可视区域	480	15.253
前沿	10	0.318
同步脉冲	2	0.0636
后沿	33	1.049
一帧	525	16.683

9.4 一个简单的OR1200核的SOC设计示例

9.4.6 外设的初始化及系统的启动

```
    mtspr(_SPR_TT_TTMR, (1 << _SPR_TT_TTMR_CR) |
                    (0 << _SPR_TT_TTMR_SR) |
                    (1 << _SPR_TT_TTMR_IE) |
                    (0 << _SPR_TT_TTMR_IP) |
                    (F_CPU / TICKS_PER_SEC)
    );
```

```
#define  UART_DL_VALUE    (F_CPU / (BAUD * 16))
#define  UART_DLL_VALUE  (UART_DL_VALUE & 0x00FF)
#define  UART_DLH_VALUE  (UART_DL_VALUE >> 8)
```

```
UART_DLH = UART_DLH_VALUE;
UART_DLL = UART_DLL_VALUE;
UART_IER = UART_IER_RDA;
```

9.4 一个简单的OR1200核的SOC设计示例

9.4.6 外设的初始化及系统的启动

```
org 0x10
_reset:
l.movhi r3, hi(_startup)
l.ori   r3,r3, lo(_startup)
l.jr    r3

l.nop

.section .text, "ax"
_startup:
/* Set stack pointer */
l.movhi r1,0x1000
l.ori   r1,r1,0x07FC
/* Jump to main */
l.movhi r3, hi(_main)
l.ori   r3, r3, lo(_main)
l.jr    r3

l.nop

l.j     .

l.nop
```

9.4 一个简单的OR1200核的SOC设计示例

9.4.6 外设的初始化及系统的启动



图 9-13 12 直线绘制的显示效果



图 9-14 13 圆形绘制的显示效果

9.4 一个简单的OR1200核的SOC设计示例

9.4.7 KX_OR1200_SOC的 μ C/OS II移植

9.4.8 基于SignalTap-II的硬件实时调试

9.5 基于OR1200的ORPSoC设计

ORPSoC 可以支持下列硬件特性：

- 标准 OpenRISC1200 核
- SDRAM 控制器
- SPI Flash 控制器
- SD 卡控制器
- UART
- USB OTG
- 以太网

ORPSoC 可以支持下列软件特性：

- GCC 4 交叉编译器
- Linux2.6 或 linux3 嵌入式操作系统
- Or1ksim 模拟器（可以模拟仿真 Linux 运行）
- 提供 Virtual Box 虚拟机下配置好 OpenRISC 软件环境的 Ubuntu 系统的硬盘镜像，以方便开发

实验与设计

9-1 基于OR1200的简单SOC系统设计

9-2 基于KX_OR1200_SOC的串口程序设计

```
line x0 y0 x1 y1 [color]           // 从点(x0,y0)到点(x1,y1)绘直线
r   x0 y0 x1 y1 [color]           // 以对角两点(x0,y0),(x1,y1)绘矩形
rf  x0 y0 x1 y1 [color]           // 以对角两点(x0,y0),(x1,y1)绘矩形并填充
c   x0 y0 r0 [color]              // 以点(x0,y0)为圆心, 以 r0 为半径绘圆
cf  x0 y0 r0 [color]              // 以点(x0,y0)为圆心, 以 r0 为半径绘圆并填充
t   x0 y0 x1 y1 x2 y2 [color]     // 以给定的三点绘制三角形
test color                         // 色彩测试
test line color                    // 线条测试
test circle color                  // 圆形测试
help                               // 打印出帮助信息。
```