



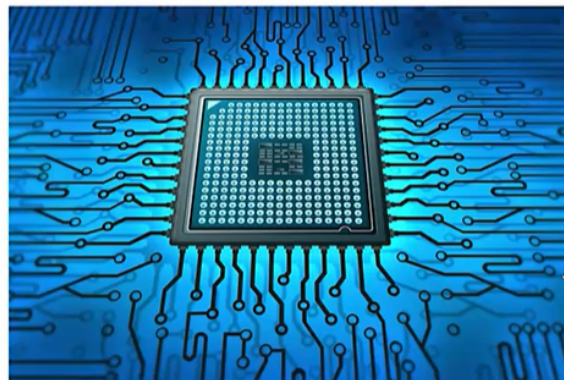
嵌入式系统原理及应用

第5章 GPIO编程

张帆

中南大学自动化学院

第5章 GPIO编程



输入输出
Input output
I/O
IO

5.1 GPIO 功能介绍

5.2 S5PC100 芯片的 GPIO 控制器介绍

5.3 S5PC100 的 GPIO 的实例

思考与练习

5.1 GPIO 功能介绍

GPIO: General-Purpose IO ports, 通用IO口

在嵌入式系统中，有数量众多、结构简单的外部设备/电路，

或需要CPU为之提供控制手段

或需要被CPU用作输入信号。

很多只要求一位，即只要有开/关两种状态

例如

控制LED灯亮与灭

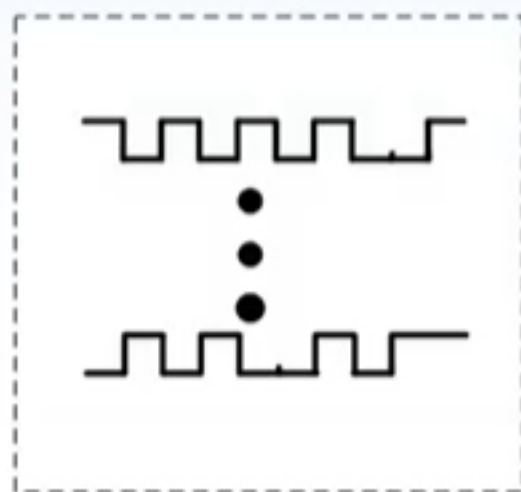
或通过获取管脚的电平属性去判断外围设备的状态

- CPU与外设两者的信号不兼容，在信号类型、功能定义、逻辑定义和时序关系上都不一致。

外设



CPU



- 两者的工作速度不匹配，CPU速度快，外设速度慢。

外设

音频设备

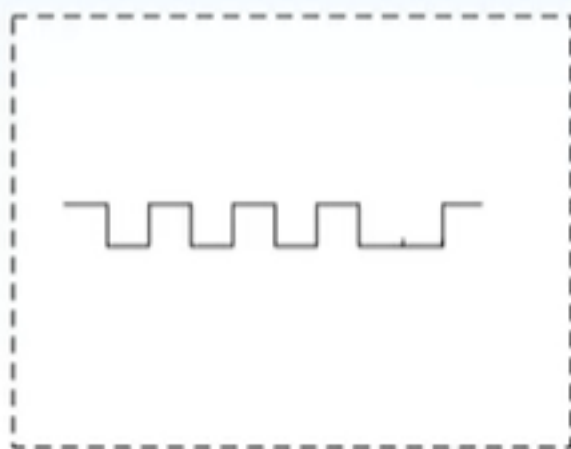
打印机

CPU

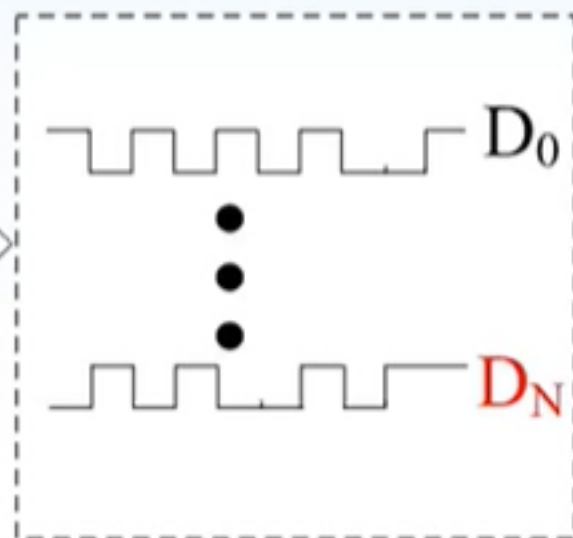
主频高
(ARM9达到400MHz)

- 数据传输方式不同，有并行、串行之分。

外设(串行)



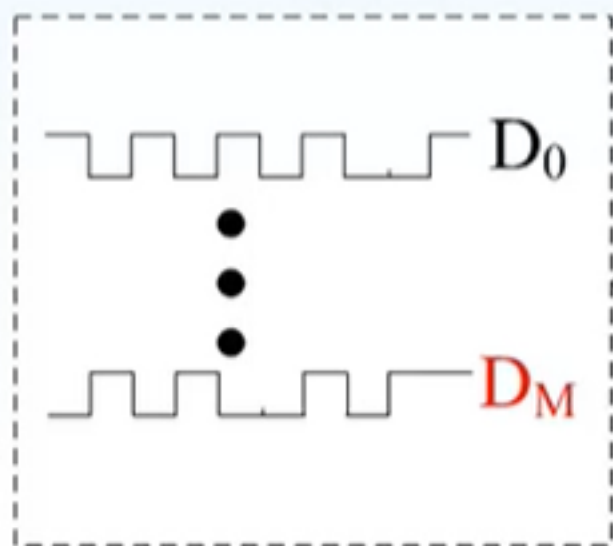
CPU



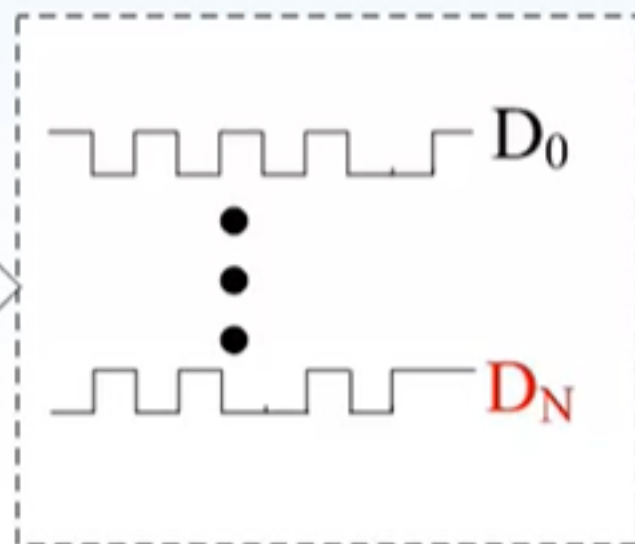
时序关系

- 数据传输方式不同，有并行、串行之分。

外设(并行)



CPU



- CPU与外设两者的信号不兼容，在信号类型、功能定义、逻辑定义和时序关系上都不一致。
- 两者的工作速度不匹配，CPU速度快，外设速度慢。
- 数据传输方式不同，有并行、串行之分。

外设不能与CPU直接相连,必须经过中间I/O接口电路相连。

接口功能

设备选择：译码选择设备、端口

数据类型转换：模/数，数/模

数据收发和格式转换：读/写，串/并、并/串

接收解释执行CPU的命令：控制信号的实现

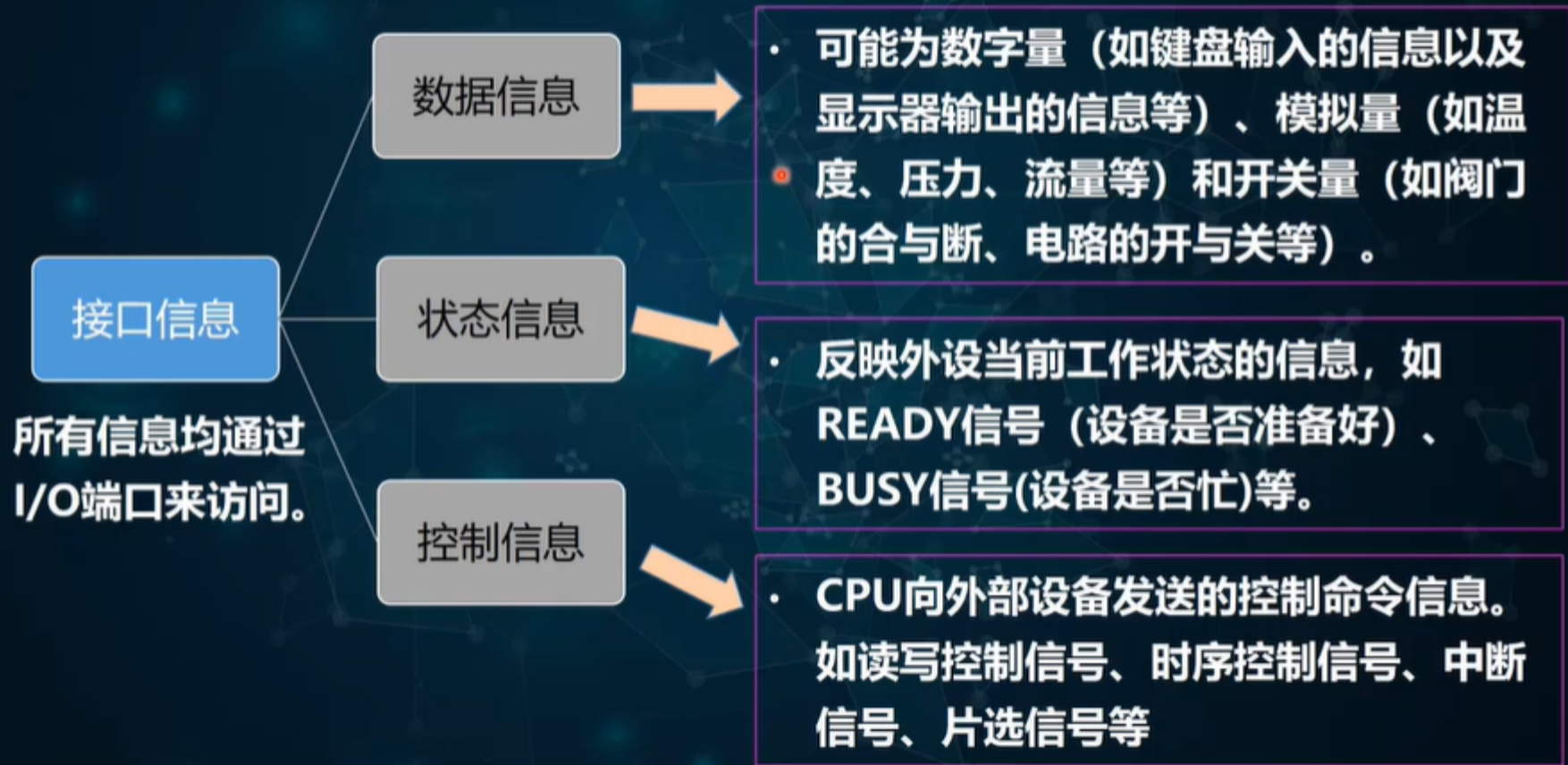
外设状态接收并转发给CPU：状态端口

支持查询、中断、DMA等多种传输控制方式

提供缓冲、暂存、驱动能力：信号驱动

错误检测功能：奇偶校验

复位



存储器映射编址

I/O端口的地址与内存地址统一编址，即I/O单元与内存单元在同一地址空间。

I/O映射编址

I/O端口与内存单元分开编址，即I/O单元与内存单元都有自己独立的地址空间

优点	<ul style="list-style-type: none">•可采用丰富的内存操作指令访问I/O单元•无需单独的I/O地址译码电路•无需专用的I/O指令	<ul style="list-style-type: none">•I/O单元不占用内存空间•I/O程序易读
缺点	<ul style="list-style-type: none">•外设占用内存空间•I/O程序不易读	<ul style="list-style-type: none">• I/O操作指令仅有单一的传送指令• I/O接口需有地址译码电路
举例	ARM系列	Intel 80X86系列，I/O端口地址空间大小为64KB

5.1 GPIO 功能介绍

对这些设备/电路的控制，使用传统的串行口或并行口都不合适。因此在微控制器芯片上一般都会提供一个GPIO。

GPIO接口至少有两个寄存器，即**通用IO控制寄存器**与**通用IO数据寄存器**

控制寄存器：设置某个管脚的属性为输入、输出或其他特殊功能

数据寄存器：各位都直接引到芯片外部，设置为高或者低或者其他

实际的MCU中，GPIO有多种形式。例如数据寄存器可位寻址或不可以，这在编程时就要区分了。比如传统的 8051 系列，就区分成可位寻址和不可位寻址两种寄存器。

很多MCU的GPIO接口除必须具备两个标准寄存器，还提供上拉寄存器。上拉寄存器可以设置 IO 的输出模式是高阻，或带上拉的电平输出，或者是不带上拉的电平输出。用在电路设计中，就可以简化外围电路

5.2 S5PC100 芯片的 GPIO 控制器介绍

5.2.1 特性

S5PC100 的 GPIO 特性包括如下几点。

- (1) 173个通用控制 IO, 141个睡眠不可唤醒中断IO, 32个睡眠可唤醒中断IO
- (2) 130个多功能输入/输出接口。
- (3) 控制引脚状态模式 (除了 GPH0、GPH1、GPH2 和 GPH3) 。

5.2 S5PC100 芯片的 GPIO 控制器介绍

5.2.1 结构图

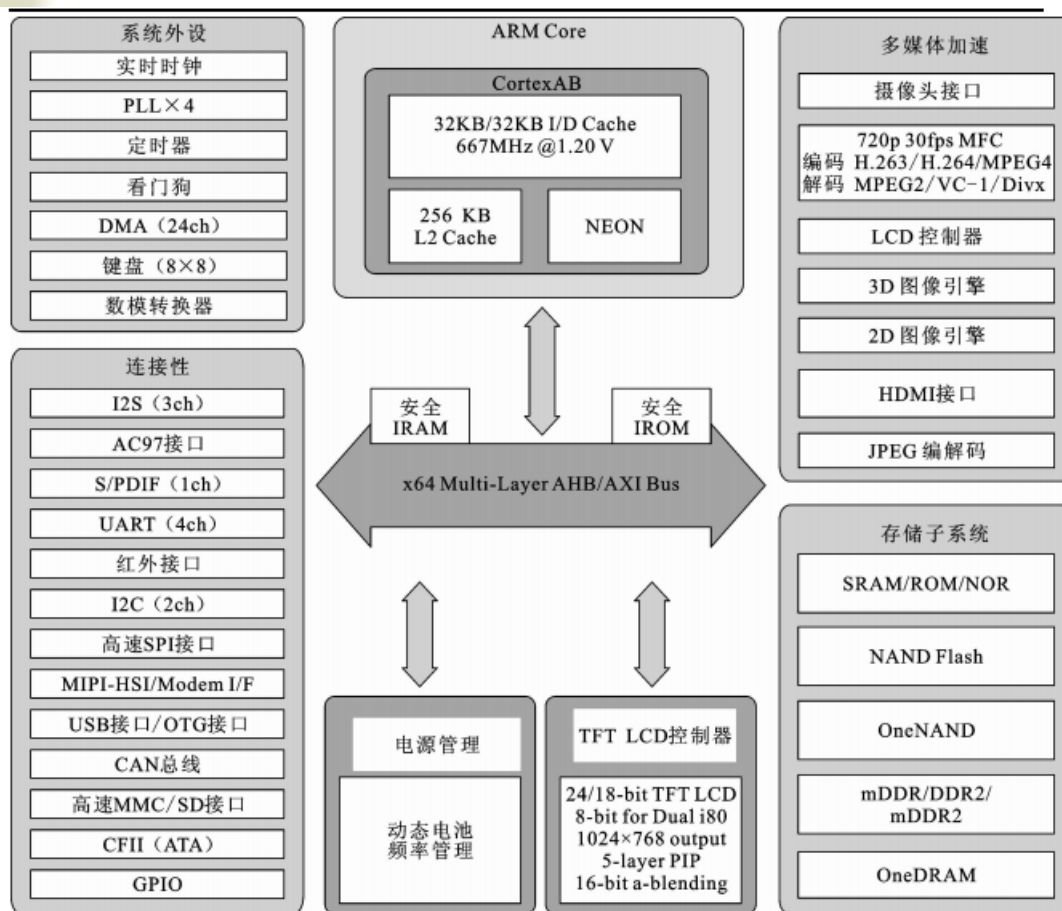
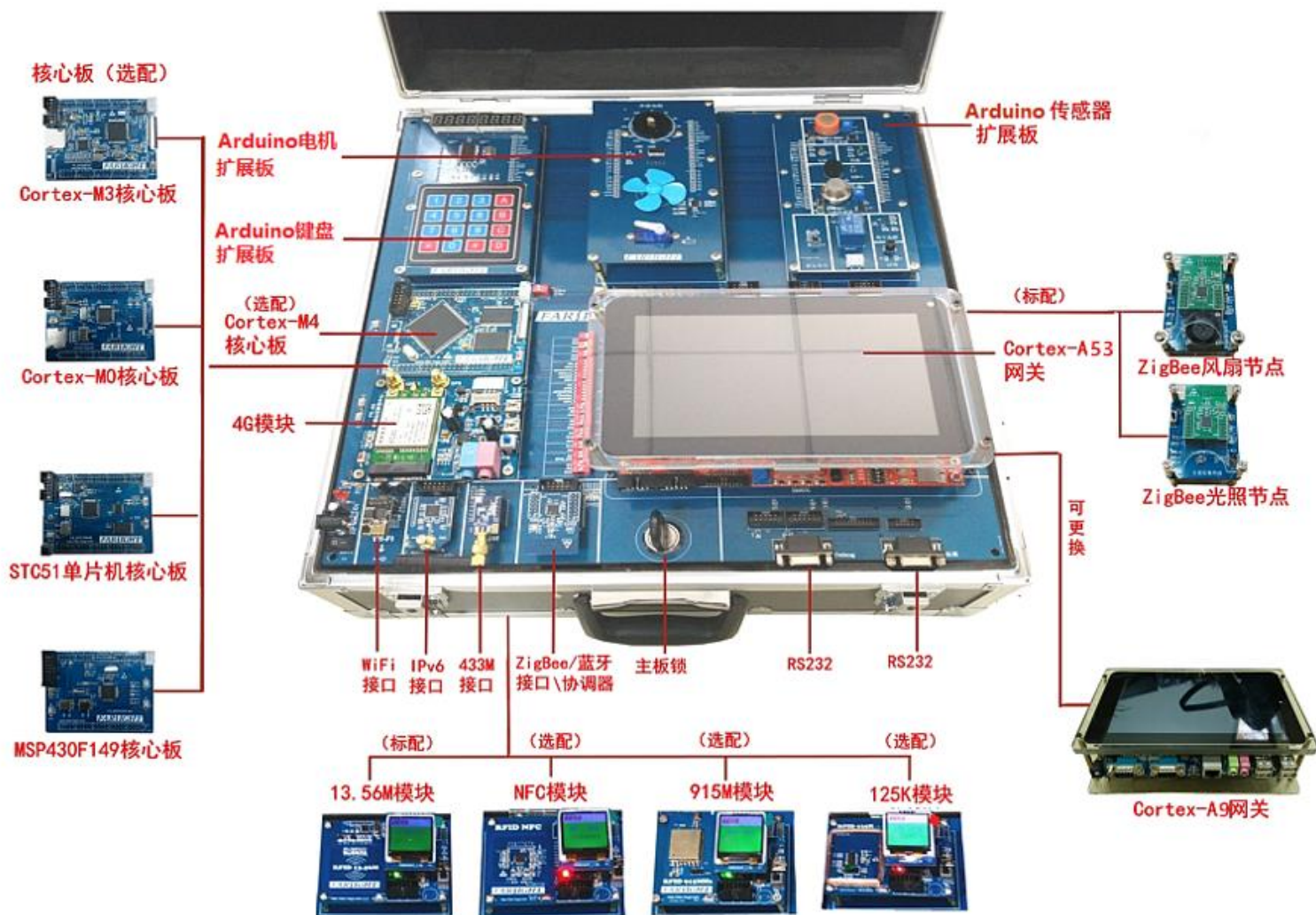


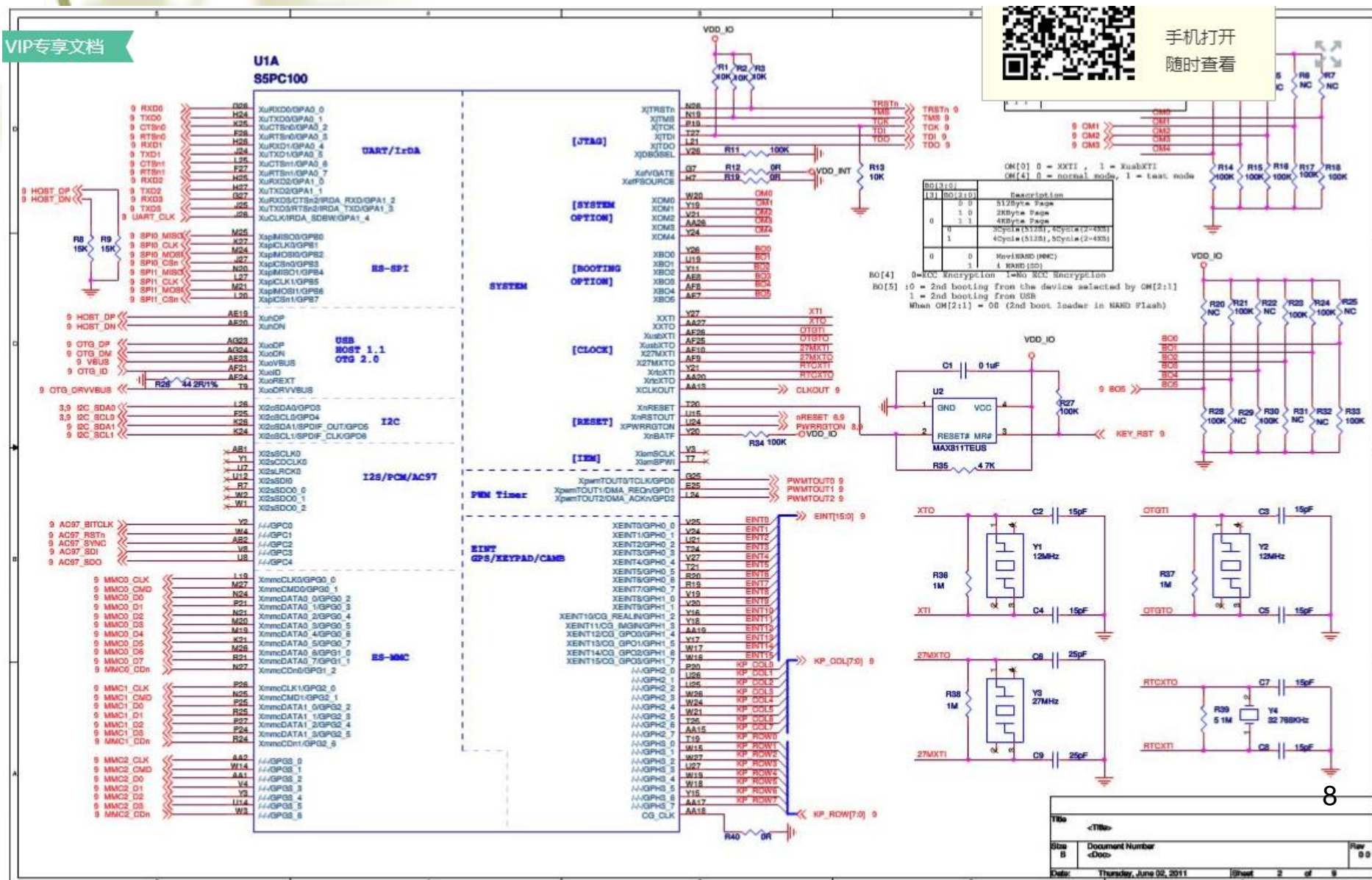
图 2-9 S5PC100 结构框图

5.2 S5PC100 芯片的 GPIO 控制器介绍

外观



VIP专享文档



5.2 S5PC100 芯片的 GPIO 控制器介绍

5.2.2 GPIO 分组预览

- (1) GPA0: 8 in/out pin 2xUART 带控制流。
- (2) GPA1: 5 in/out pin 2xUART 不带控制流 or 1xUART 带控制流, 1x IrDA。
- (3) GPB: 8 in/out pin 2x SPI 总线接口。
- (4) GPC: 5 in/out pin I2S 总线接口, PCM 接口, AC97 接口。
- (5) GPD: 7 in/out pin 2xI2C 总线接口, PWM 接口, External DMA 接口, SPDIF 接口。
- (6) GPE0,1: 14 in/out pin 摄像头接口, SD/MMC 接口。
- (7) GPF0,1,2,3: 28 in/out pin LCD 接口。
- (8) GPG0,1,2,3: 25 in/out pin 3xMMC channel, SPI, I2S, PCM, SPDIF 各种接口。
- (9) GPH0,1,2,3: 32 in/out pin 摄像头通道接口, 键盘, 支持 32 位的睡眠可中断接口。
- (10) GPI: 8 in/out pin PWI 接口。
- (11) GPJ0,1,2,3,4: 33 in/out pin Modem IF, HIS, ATA 接口
- (12) GPK0,1,2,3: 30 in/out pin EBI 控制信号。

5.2 S5PC100 芯片的 GPIO 控制器介绍

5.2.3 S5PC100 的 GPIO 常用寄存器分类

1. 端口控制寄存器 (GPACON-GPHCON)

在 S5PC100 中，大多数的引脚都可复用，因此必须对每个引脚进行配置。端口控制寄存器 (GPNCON) 定义了每个引脚的功能。

2. 端口数据寄存器 (GPADAT-GPHDAT)

如果端口被配置成了输出端口，可以向 GPNDAT 的相应位写数据。如果端口被配置成了输入端口，可以从 GPNDAT 的相应位读出数据。

3. 端口上拉寄存器 (GPBUP-GPHUP)

端口上拉寄存器控制了每个端口组的上拉电阻的允许/禁止。如果某一位为 0，相应的上拉电阻被允许；如果是 1，相应的上拉电阻被禁止。如果端口的上拉电阻被允许，无论在何种状态（输入、输出、DATAn、EINTn 等）下，上拉电阻都起作用。

- 【回忆】

- 1. 按照内存映射 (Memory Map) 中的内存分配方案, 该产品架构支持多少个片内功能模块?
- 2. 我们所用的这款芯片实际内置了多少个片内功能模块?

256

72个

- 【复习】

- 我们这款芯片有多少个普通I/O引脚?

95

- 【思考】

- 对于我们这款芯片, 假设:
 - 片内功能模块有一半都需要通过引脚与片外进行通信,
 - 平均每个与片外通信的功能模块需要3个信号线
- 这种假设条件下, 芯片引脚是否够用?

$(72 \times 50\%) \times 3 = 108$

- 为了优化小型封装的功能，引脚可通过信号多路复用提供多种功能。

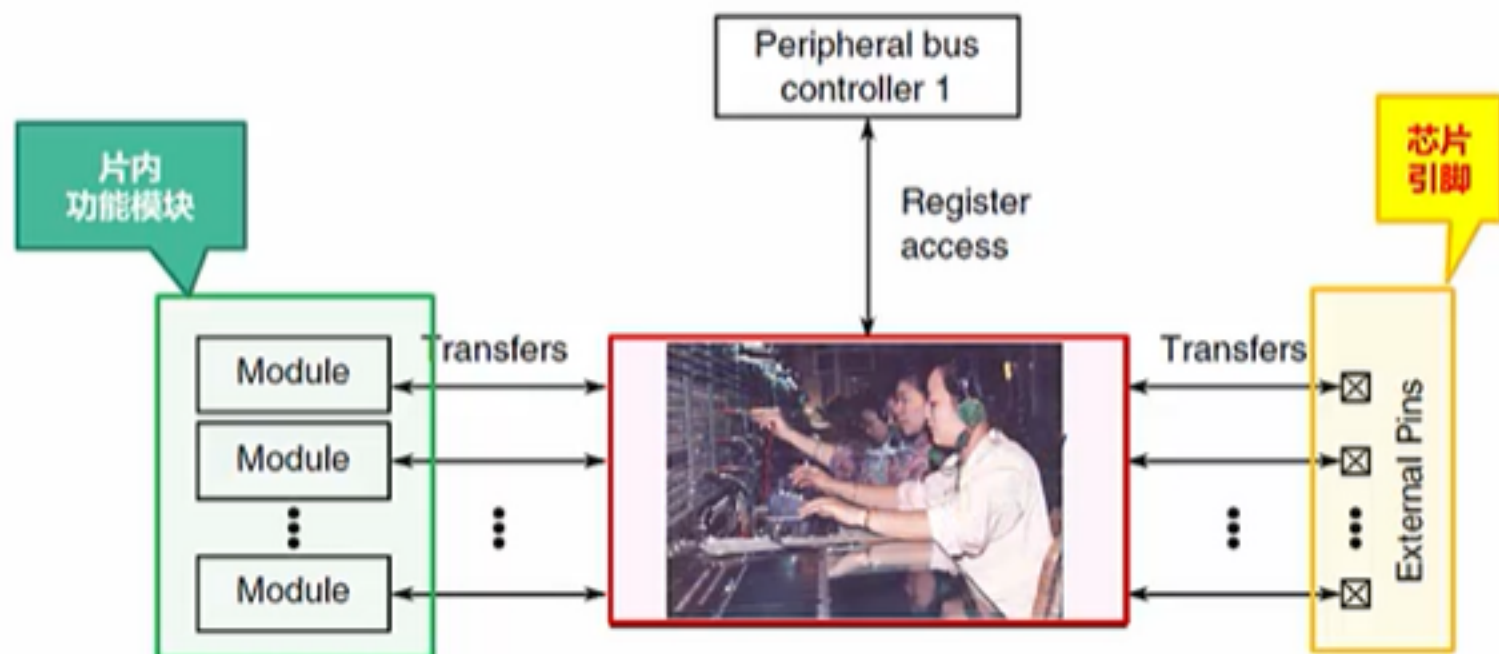


Figure 10-1. Signal multiplexing integration

5.2 S5PC100 芯片的 GPIO 控制器介绍

5.2.4 S5PC100 I/O 接口常用寄存器详解

(1) GPG3CON 寄存器, 决定 GPG3 组的每一位 IO 引脚的工作模式。GPG3 组一共有 6 个 IO 引脚, 每 4 位来决定一位 IO 引脚的工作模式

表 6-1

GPG3 控制寄存器(Address = 0xE030001C0)

GPG3CON	位	描 述	初始状态
GPG3CON[0]	[3:0]	0000 = 输入, 0001 = 输出, 0010 = SD_2_CLK, 0011 = SPI_2_CLK, 0100 = I2S2_SCLK, 0101 = PCM_0_SCLK, 1111 = NWU_INTG14[0]	0000
GPG3CON[1]	[7:4]	0000 = 输入, 0001 = 输出, 0010 = SD_2_CMD, 0011 = SPI_2_nSS, 0100 = I2S2_CDCLK, 0101 = PCM_0_EXTCLK, 1111 = NWU_INTG14[1]	0000
GPG3CON[2]	[11:8]	0000 = 输入, 0001 = 输出, 0010 = SD_2_D[0], 0011 = SPI_2_MISO, 0100 = I2S2_LRCK, 0101 = PCM_0_FSYNC, 1111 = NWU_INTG14[2]	0000
GPG3CON[3]	[15:12]	0000 = 输入, 0001 = 输出, 0010 = SD_2_D[1], 0011 = SPI_2_MOSI, 0100 = I2S2_SDI, 0101 = PCM_0_SIN, 1111 = NWU_INTG14[3]	0000
GPG3CON[4]	[19:16]	0000 = 输入, 0001 = 输出, 0010 = SD_2_D[2], 0011 = Reserved, 0100 = I2S2_SDO, 0101 = PCM_0_SOUT, 1111 = NWU_INTG14[4]	0000
GPG3CON[5]	[23:20]	0000 = 输入, 0001 = 输出, 0010 = SD_2_D[3], 0011 = Reserved, 0100 = Reserved, 0101 = SPDIF_0_OUT, 1111 = NWU_INTG14[5]	0000
GPG3CON[6]	[27:24]	0000 = 输入, 0001 = 输出, 0010 = SD_2_CDn, 0011 = Reserved, 0100 = Reserved, 0101 = SPDIF_EXTCLK, 1111 = NWU_INTG14[6]	0000

5.2 S5PC100 芯片的 GPIO 控制器介绍

(2) GPG3DAT 寄存器，用来决定每一个引脚输出电平的状态的寄存器

表 6-3

GPG3DAT 寄存器 (Address = 0xE03001C4)

GPG3DAT	描 述	决 定 位
GPG3DAT 的[0]	用来决定 GPG3 的第 0 位引脚高低电平状态	GPG3_0
GPG3DAT 的[1]	用来决定 GPG3 的第 1 位引脚高低电平状态	GPG3_1
GPG3DAT 的[2]	用来决定 GPG3 的第 2 位引脚高低电平状态	GPG3_2
GPG3DAT 的[3]	用来决定 GPG3 的第 3 位引脚高低电平状态	GPG3_3
GPG3DAT 的[4]	用来决定 GPG3 的第 4 位引脚高低电平状态	GPG3_4
GPG3DAT 的[5]	用来决定 GPG3 的第 5 位引脚高低电平状态	GPG3_5
GPG3DAT 的[6]	用来决定 GPG3 的第 6 位引脚高低电平状态	GPG3_6
GPG3DAT 的[31:7]	没有使用，保留	

相应位为 1 时：输出高电平，即等于 VCC_IO 的电压。

相应位为 0 时：输出低电平，即等于 0 V。

5.2 S5PC100 芯片的 GPIO 控制器介绍

(3) GPG3PUD 寄存器，用来决定每一个引脚是否接上拉电阻的功能

表 6-4

GPG3PUD 寄存器 (Address = 0xe03001C8)

GPG3PUD	描 述	决定位
GPG3PUD 的[1:0]	用来决定 GPG3 的第 0 位引脚上拉/下拉电阻的状态	GPG3_0
GPG3PUD 的[3:2]	用来决定 GPG3 的第 1 位引脚上拉/下拉电阻的状态	GPG3_1
GPG3PUD 的[5:4]	用来决定 GPG3 的第 2 位引脚上拉/下拉电阻的状态	GPG3_2
GPG3PUD 的[7:6]	用来决定 GPG3 的第 3 位引脚上拉/下拉电阻的状态	GPG3_3
GPG3PUD 的[9:8]	用来决定 GPG3 的第 4 位引脚上拉/下拉电阻的状态	GPG3_4
GPG3PUD 的[11:10]	用来决定 GPG3 的第 5 位引脚上拉/下拉电阻的状态	GPG3_5
GPG3PUD 的[13:12]	用来决定 GPG3 的第 6 位引脚上拉/下拉电阻的状态	GPG3_6

相应位的控制功能如下：

00 = Disables Pull-up/down 禁止上拉电阻和下拉电阻的功能；

01 = Enables Pull-down 使能下拉电阻的功能；

10 = Enables Pull-up 使能上拉电阻的功能；

11 = Reserved 保留，没有使用。

5.3 S5PC100的GPIO的实例

利用S5PC100的4个I/O引脚控制4个发光二极管，可以对其亮或灭进行控制

5.3.1 电路原理

LED4, LED1~LED3 分别与 GPG3_0~GPG3_3 相连, 通过 GPG3_0~ GPG3_3 引脚的高低电平来控制三极管的导通性, 从而控制 LED 的亮灭

(高电平: 亮; 低电平: 灭) why?

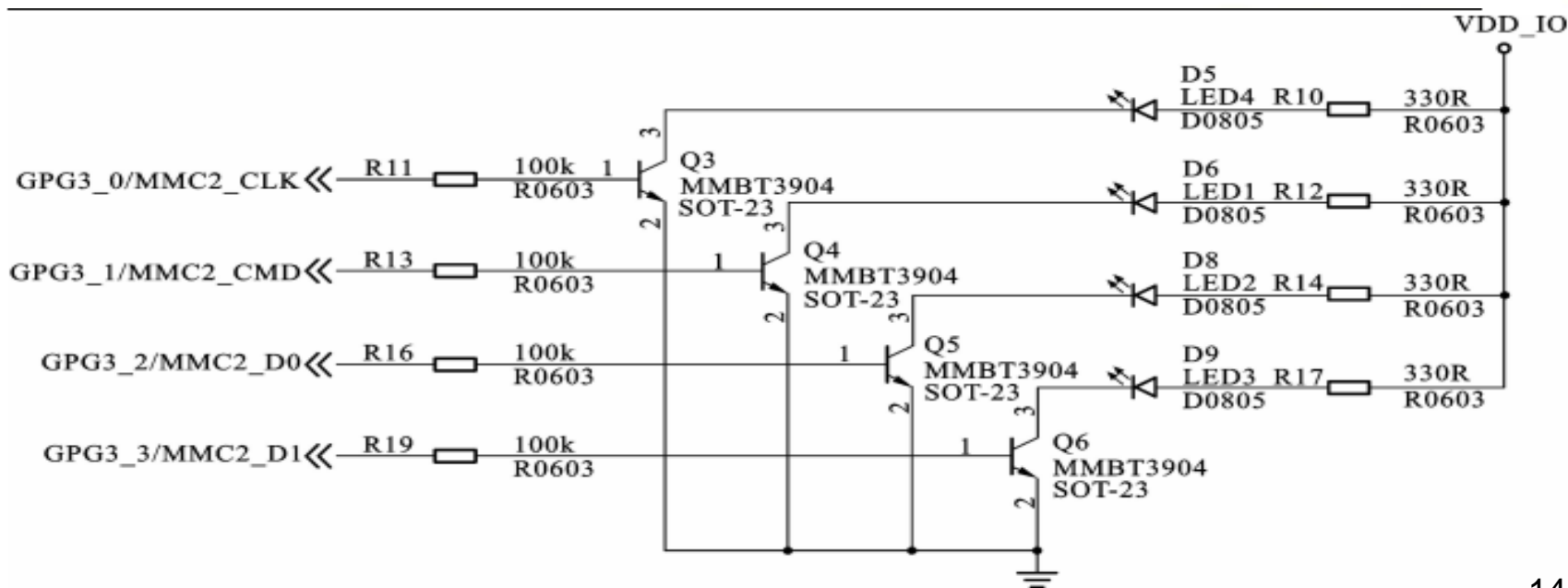


图 6-1 LED 电路原理

5.3 S5PC100的GPIO的实例

5.3.2 寄存器设置

为了实现控制 LED 的目的，需要通过配置 GPG3CON 寄存器将 GPG3_0、GPG3_1、GPG3_2、GPG3_3 设置为输出属性。

通过设置GPG3DAT寄存器实现点亮与熄灭4个LED。

对于本例来说，GPG3PUD 默认是使能上拉，因此不用设置。

5.3 S5PC100的GPIO的实例

5.3.3 程序编写

```
#include "s5pc100.h"
int main()
{
    int i;
    GPG3.GPG3CON = (~ (0xf<<4)&GPG3.GPG3CON) | (0x1<<4);
    while(1)
    { GPG3.GPG3DAT = 0x2;
      for(i = 0; i <= 1000000; i++);
      GPG3.GPG3DAT = 0x0;
      for(i = 0; i <= 1000000; i++);
    }
    return 0;
}
```

5.3 S5PC100的GPIO的实例

/* 以下是 GPG3 结构体在 s5pc100.h 中的定义*/

/* GPG3 */

```
typedef struct {  
    unsigned int GPG3CON;  
    unsigned int GPG3DAT;  
    unsigned int GPG3PUD;  
    unsigned int GPG3DRV;  
    unsigned int GPG3PDNCON;  
    unsigned int GPG3PDNPUD;  
} gpg3;  
#define GPG3 (* (volatile gpg3 * )0xE03001C0 )  
*/
```

实验过程与结果如下：

- (1) 将程序编译后获得的.elf 文件，通过仿真器下载并运行在目标板上；
- (2) 观察实验结果，可以看到 LED1 闪亮的现象。

思考与练习

1. 什么是GPIO?
2. S5PC100有几组GPIO端口?
3. 如何实现利用S5PC100的GPD4控制LED? 请画出原理图, 并编程实现。

思考与练习

如何实现利用S5PC100的GPD4控制LED？请画出原理图，并编程实现。

1、原理图找到硬件连接图，找到相关信号

GPD4_0 -> led4

GPD4_1 -> led1

GPD4_2 -> led2

GPD4_3 -> led3

2、找数据手册，查看信号的控制模块

GPD4 -> GPIO Pad Control

3、GPG3寄存器说明

GPD4CON 0x***** 选择管脚功能，相应的位设置为b0001则表示输出

GPD4DAT 0x***** 位对应相应管脚的状态

GPD4PUD 0x***** 内部上拉下拉控制

4、编写程序

make生成目标文件

通过loadb下载到开发板上运行程序，查看现象

ARM 集成开发环境搭建

1. FS-JTAG 仿真器介绍

硬件特点。

- ① USB 特性：USB2.0 全速接口、USB 电源供电。
- ② JTAG 特性：IEEE 1149.1 标准。

软件特点。

- ① Eclipse 集成开发环境：提供实时调试功能，如单步、全速运行、复位、软 / 硬断点、跳转动态查看寄存器和存储器、变量观察。
- ② 源码级别调试器 Openocd，开源，并且提供良好的交互界面。



图 5-1 FS-JTAG 仿真器

ARM 集成开发环境搭建

2. 开发工具的安装

Eclipse for ARM 是借用开源软件的 Eclipse 的工程管理工具，嵌入 GNU 工具集，使之能够开发ARM公司Cortex-A系列的CPU

1. ARM-GCC 编译工具
2. GNU make 工具
3. java运行环境
4. Eclipse
5. FS-JTAG

ARM 集成开发环境搭建

3. 创建新工程

1. 汇编或C源文件
2. Makefile 编译
3. 调试命令文件 连接仿真器并发送调试命令

ARM 集成开发环境搭建

4. 调试工程

1. 配置FS-JTAG调试工具
2. Eclipse 配置
3. 选择led工程 (elf)
4. debug调试 (断点, 单步, 全速, 寄存器变化, 程序运行指示)

谢谢！