

# Cortex-MO的IP集成和底层驱动软件编写





- ■背景知识
- ■开发工具简介
- GPIO集成与功能验证
- ■硬件流水灯集成与功能验证
- ■思考



# 背景知识介绍





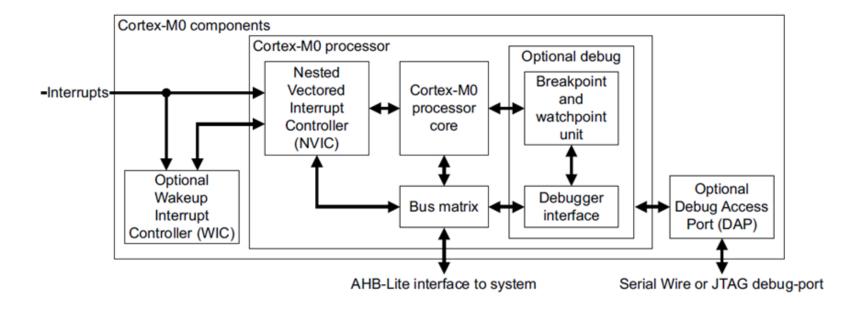
#### ■ 参考文档

- Cortex™-M System Design Kit Technical Reference Manual
- AMBA 3 AHB Lite Protocol Specification
- Multi-layer AHB Overview
- ARMv6-M Architecture Reference Manual
- Cortex-M0 Devices Generic User Guide
- ARM Cortex-M0 Processor Technical Reference Manual
- Simulation Quick-Start for ModelSim-Intel FPGA Edition
- Vivado Design Suite User Guide
- Intel Quartus Prime Pro Edition User Guide



#### Cortex-M0简介

■ Cortex-M0架构图



# Cortex-M0简介



#### ■ 关键信号描述

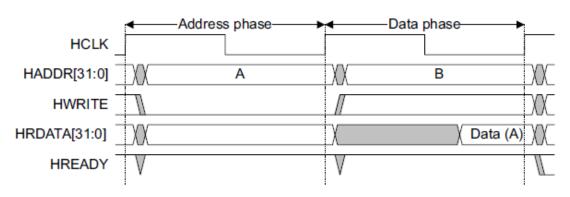
信号名	描述
HXXXX(以H开头的信号)	总线相关,下一节将会详细介绍
vis_rX_o	通用寄存器,通过仿真观察这些寄存器能够知道相关汇编指令是否正常执行(ADD、MOV等)
vis_msp_o	栈指针
vis_r14_o	连接寄存器,程序返回出错时需要检查此寄存器
vis_pc_o	程序计数器,可以用来判断处理器是否在正常运行
vis_ipsr_o	中断状态寄存器,当发生错误的时候,PSR的值会改变,用于判断发生 何种类型错误
LOCKUP	处理器处理NMI或者HardFault时又发生错误后,处理器将会被死锁

#### 硬木課堂 www.emooc.cc

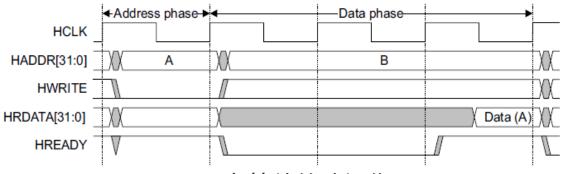
#### ■ 部分关键信号描述

名称	来源	描述
HADDR[31:0]	Master	传输地址
HBURST[2:0]	Master	Burst类型
HSIZE[2:0]	Master	数据宽度 00: 8bit Byte 01: 16bit Halfword 10: 32bit Word
HTRANS[1:0]	Master	传输类型 00: IDLE, 无操作 01: BUSY 10: NONSEQ, 主要的传输方式 11: SEQ
HWDATA[31:0]	Master	核发出的写数据
HWRITE	Master	读写选择 (1: 写, 0: 读)
HRDATA[31:0]	Slave	外设返回的读数据
HREADOUT	Slave	何时传输完成 (通常为1)
HRESP	Slave	传输是否成功 (通常为0)

#### ■ NONSEQ传输时序

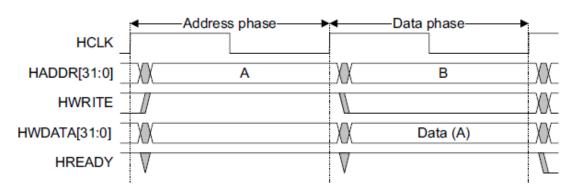


基本读操作

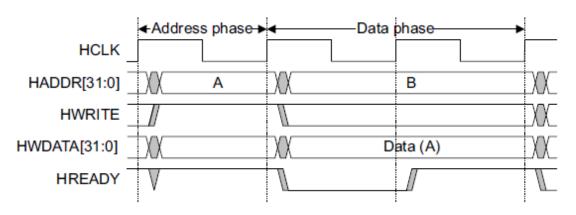


具有等待的读操作





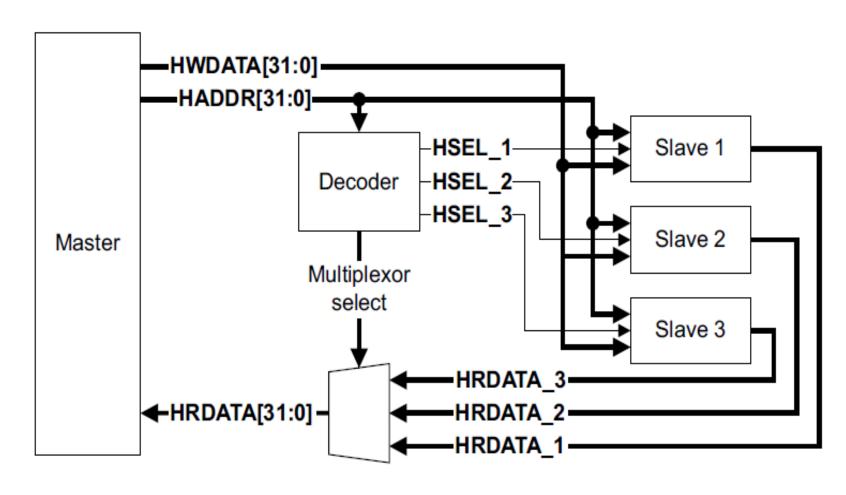
基本写操作



具有等待的写操作

■ AHB-Lite总线拓展

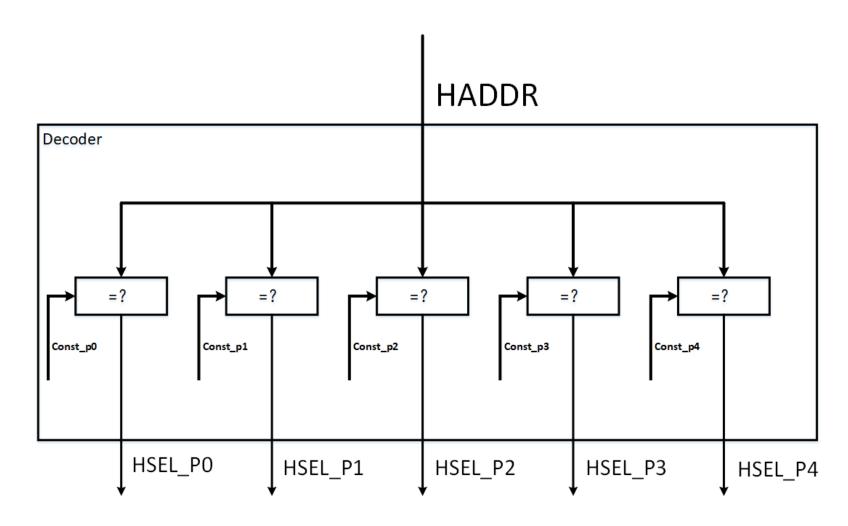




Single Master AHB Interconnect结构

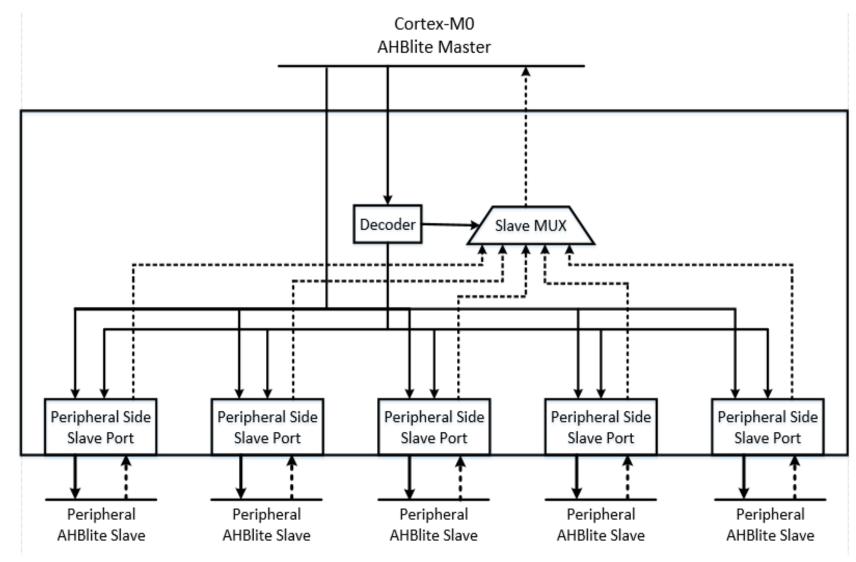
■ Decoder内部结构





■ 总线拓展结构图







# 开发工具简介

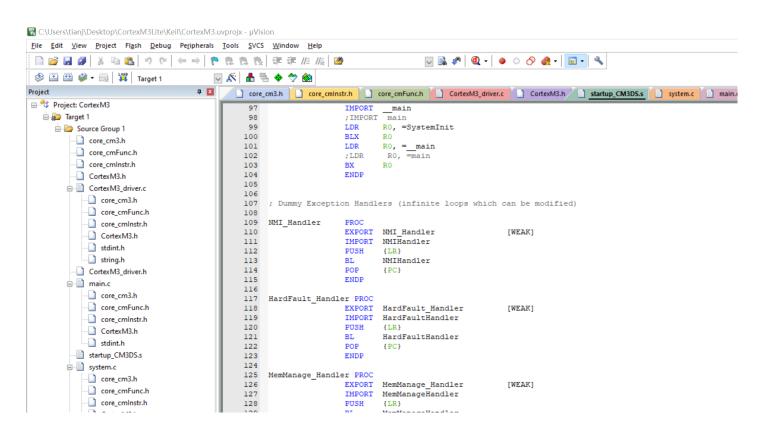
## 开发工具

- 软件编译工具
  - Keil
- 调试工具
  - Keil
- 仿真工具
  - Modelsim
  - Vivado
- 综合/实现工具
  - Vivado
  - Quartus Prime





# 软件编译工具

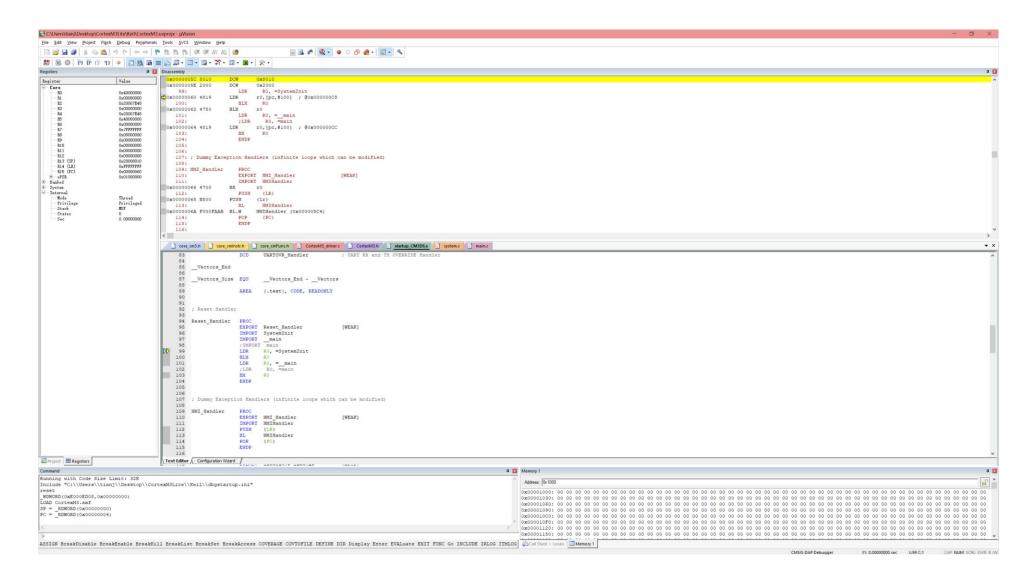






# 调试工具

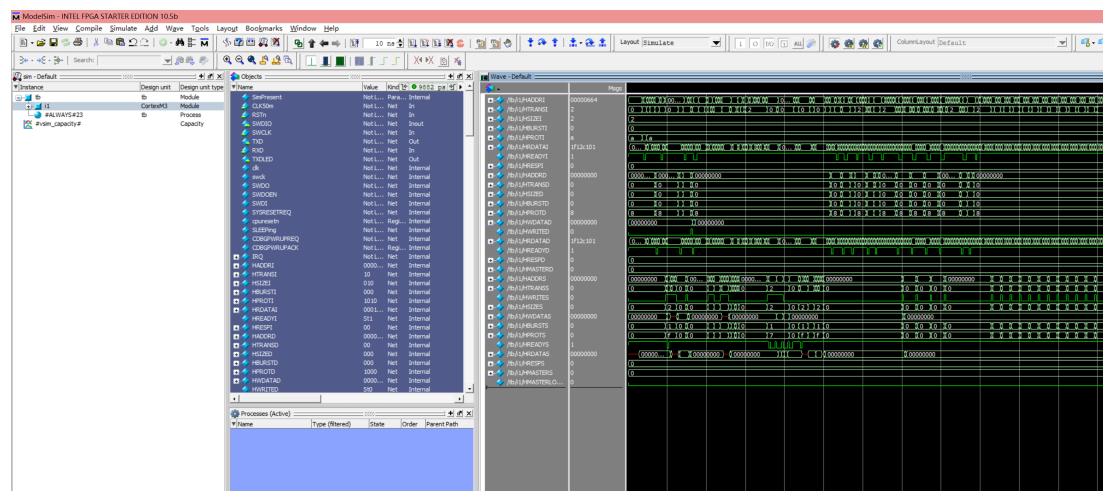
Keil





## 仿真工具

■ Modelsim – INTEL FPGA Starter Edition 10.5b





## 综合/实现工具

- 使用Quartus Prime还是Vivado取决于你的板子
- 使用哪个版本的软件取决于你的FPGA型号
  - Quartus Prime 1x.x Lite
  - Vivado 2018.x





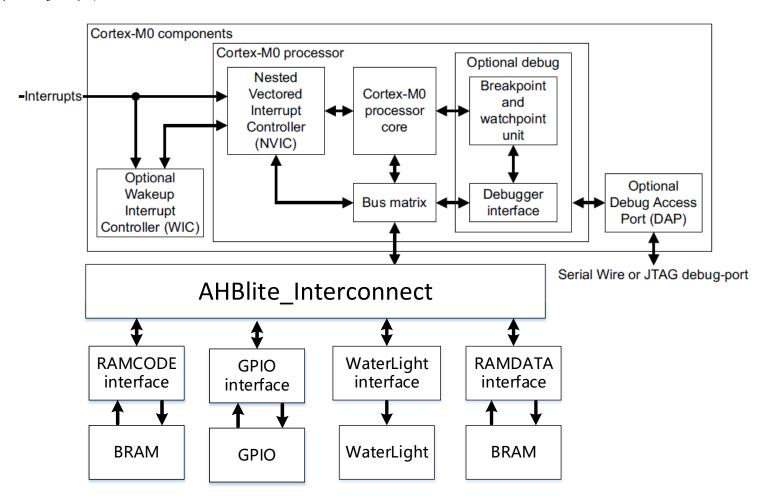


# GPIO模块集成与功能验证



### 本次实验实现的SoC

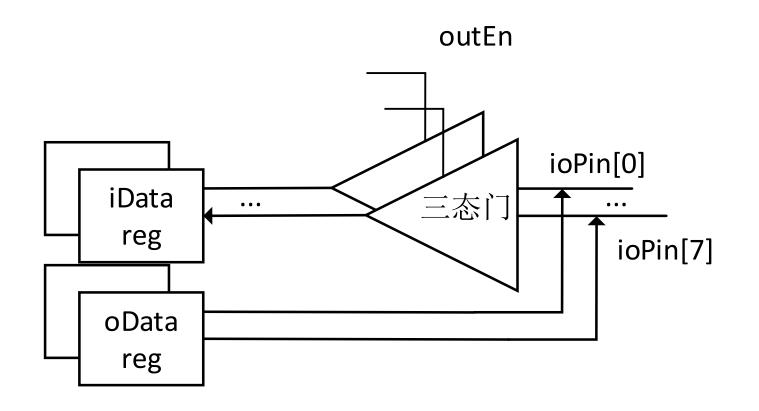
■ 本次实验实现的SoC





#### GPIO模块原理

**■** GPIO原理



- 三个控制端口
- a) outEn
- b) iData\_reg
- c) oData\_reg

#### 第一步添加修改Decoder



#### RAMDATA

```
/*RAMDATA enable parameter*/
parameter Port1 en = 0,
/****************
 改为:
/*RAMDATA enable parameter*/
parameter Port1 en = 1,
/***************
//0x20000000-0x2000ffff
/*Insert RAMDATA decoder code there*/
assign P1 HSEL = 1'b0;
/***********/
改为:
//0x20000000-0x2000ffff
/*Insert RAMDATA decoder code there*/
assign P1 HSEL = (HADDR[31:16] == 16'h2000)?
                                              Portl en :
                                                            1'b0;
/***********/
```

#### 第一步添加修改Decoder



#### GPIO

```
/*GPIO enable parameter*/
parameter Port4 en = 0,
/**************
/* GPIO enable parameter*/
parameter Port4 en = 1,
/**************
/*Insert GPIO decoder code there*/
                                         //0x40000028 OUT ENABLE
assign P4 HSEL = 1'b0;
                                         //0X40000024 IN DATA
/***********/
                                         //0X40000020 OUT DATA
改为:
/*Insert GPIO decoder code there*/
assign P4 HSEL = (HADDR[31:4] == 28'h4000002) ? Port4 en : 1'd0;
/***********/
```

#### 第二步在顶层模块添加连接

/\*\*\*\*\*\*\*\*\*\*\*\*/



```
/* Connect to Interconnect Port 1 */ /* Connect to Interconnect Port 4 */
.HCLK
                    (clk),
                                              .HCLK
                                                                      (clk),
.HRESETn
                    (cpuresetn),
                                              .HRESETn
                                                                      (cpuresetn),
.HSEL
                    (HSEL P1),
                                             .HSEL
                                                                      (HSEL P4),
                                                                      (HADDR P4),
                    (HADDR P1),
.HADDR
                                              . HADDR
.HPROT
                    (HPROT P1),
                                             .HPROT
                                                                      (HPROT P4),
                    (HSIZE P1),
                                                                      (HSIZE P4),
.HSIZE
                                             .HSIZE
.HTRANS
                    (HTRANS P1),
                                             .HTRANS
                                                                      (HTRANS P4),
                    (HWDATA P1),
.HWDATA
                                             .HWDATA
                                                                      (HWDATA P4),
                    (HWRITE P1),
                                                                      (HWRITE P4),
.HWRITE
                                             .HWRITE
.HRDATA
                    (HRDATA P1),
                                             .HRDATA
                                                                      (HRDATA P4),
.HREADY
                    (HREADY P1),
                                             .HREADY
                                                                      (HREADY P4),
                    (HREADYOUT P1),
                                                                      (HREADYOUT P4),
.HREADYOUT
                                             .HREADYOUT
.HRESP
                    (HRESP P1),
                                              .HRESP
                                                                      (HRESP P4),
             (RAMDATA WADDR),
                                                                      (outEn),
.BRAM WRADDR
                                              .outEn
.BRAM RDADDR
                (RAMDATA RADDR),
                                             .oData
                                                                      (oData),
                                             .iData
.BRAM RDATA (RAMDATA RDATA),
                                                                      (iData)
                                          /*********/
.BRAM WDATA (RAMDATA WDATA),
.BRAM WRITE (RAMDATA WRITE)
```

### 第三步Keil编写外设驱动软件



请参看Keil界面,讲解代码

### 第四步Modelsim仿真



- 常见问题1: BLOCK\_RAM.v中RAM未初始化成功,modelsim仿真全是未知态。
- 常见问题2:某些版本的Modelsim会自动优化仿真信号,导致仿真之后object中找不到信号。

## 第五步使用开发板验证IP工程



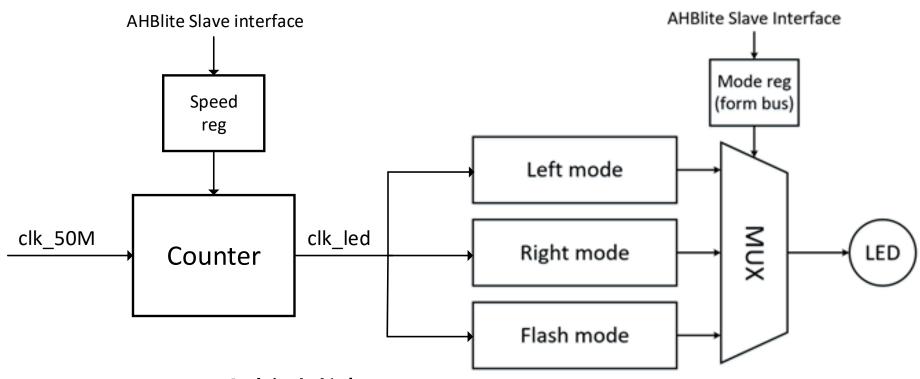
见VIVADO界面



# 硬件流水灯介绍

# 硬件流水灯结构





- 两个控制端口
- a) Speed\_reg
- b) Mode\_reg

#### 硬件流水灯集成与验证流程



- 其流程跟GPIO类似
- 1. 先设计流水灯模块,即IP模块
- 2. 根据IP模块和分配给控制端口的地址,修改总线的decoder和Mux
- 3. 根据IP模块,设计对应的Interface电路
- 4. 将IP模块的Interface 与总线Master相连接
- 5. 编写驱动软件
- 6. 使用Modelsim等工具仿真功能
- 7. 使用FPGA开发板进一步验证功能

### 思考:



■ 软件流水灯和硬件流水灯的关系和优缺点?在实际应用中,软硬件 该如何分工合作?



#### Thank you!