



# 实验报告

课程名称:	计算机组织与结构II
学 院:	信息科学与工程学院
实验名称:	processor 设计
学 号:	04022212
姓 名:	钟源
时 间:	2024.2.27

## 一、实验目的

本实验的目的是使用 Vivado 软件，通过 Verilog 语言编写设计和仿真并行输出控制器（POC, parallel output controller），它充当系统总线和 printer 之间的接口。

## 二、实验任务

- 1.设计并行输出控制器（POC）、处理器和 printer 模块，联合进行仿真，完成三者之间的数据交互与传输；
- 2.POC 模块必须同时支持查询方式和中断方式，可以根据需要切换选择两种方式中的一种；
- 3.使用功能仿真，验证 POC 设计的逻辑正确性。

## 三、实验原理

### 3.1 查询模式打印数据流程

#### 1) 初始化阶段：

Processor 和 POC 初始化，SR（POC 状态寄存器）的 SR0 位一直保持为 0，表示查询模式。Printer 初始化，RDY（表示 printer 准备好）置为 1，表示已准备好接收数据。

#### 2) Processor 查询 SR 状态：

Processor 通过地址选中 POC 的 SR 寄存器，查询 SR7（准备位）的状态。如果 SR7=1，表示 POC 处于“准备好”状态，可以接收新数据。

#### 3) Processor 写入数据：

Processor 选中 BR（POC 缓存寄存器）的地址，将要打印的数据写入 BR，写入完成后，再将 SR7 置为 0，表明 processor 已经写入新数据给 POC 且 POC 尚未处理。

#### 4) POC 检测到 SR7=0：

POC 检测到 SR7 被置为 0，开始与 printer 进行握手操作。

#### 5) POC 与 printer 握手：

Printer 检测到 RDY=1，表示 printer 准备好接收新的数据。POC 将数据从 BR 发送到 PD（并行数据端口），并通过 TR（传输请求）发送脉冲。Printer 检测到 TR 脉冲后，将 RDY 置为 0，接收 PD 端口的数据并开始打印。

#### 6) 打印完成：

打印完成后，printer 将 RDY 重新置为 1，表示 printer 再次准备好。POC 将 SR7 重新置为 1，回到“准备好”状态，等待下一次数据传输。

### 3.2 中断模式打印数据流程

#### 1) 初始化阶段：

Processor 和 POC 初始化，SR 的 SR0 位一直保持为 1，表示中断模式。Printer 初始化，RDY 置为 1，表示 printer 已准备好接收数据。

## 2) POC 检测 SR 状态:

POC 不断检测 SR7 的状态。如果 SR7=1, POC 向 processor 发送 IRQ (中断请求信号), 低电平有效。

## 3) Processor 响应中断:

Processor 收到 IRQ 信号后, 由于是中断模式, 不会查询 SR7, 直接选中 BR 的地址。Processor 将要打印的数据写入 BR 寄存器。写入完成后, processor 将 SR7 置为 0, 表明 processor 已经写入新数据给 POC 且 POC 尚未处理。

## 4) POC 检测到 SR7=0:

POC 检测到 SR7 被置为 0, 开始与 printer 进行握手操作。

## 5) POC 与 printer 握手:

Printer 检测到 RDY=1, 表示 printer 准备好接收新的数据。POC 将数据从 BR 发送到 PD 端口, 并通过 TR 发送脉冲。Printer 检测到 TR 脉冲后, 将 RDY 置为 0, 接收 PD 端口的数据并开始打印。

## 6) 打印完成:

打印完成后, printer 将 RDY 重新置为 1, 表示 printer 再次准备好。POC 将 SR7 重新置为 1, 回到“准备好”状态。由于 SR0=1 且 SR7=1, POC 再次发出中断请求 IRQ 信号, 等待下一次数据传输。

## 3.3 POC 状态机设计

使用变量 state 表示 POC 工作状态。

### 1) state = 2'b00:

如果此时 SR0=1, 表明为中断方式, 则 IRQ 置 0, 表示发送中断请求; 如果此时 SR0=0, 表明为查询方式, 则 IRQ 置 1, 表示该信号无效。

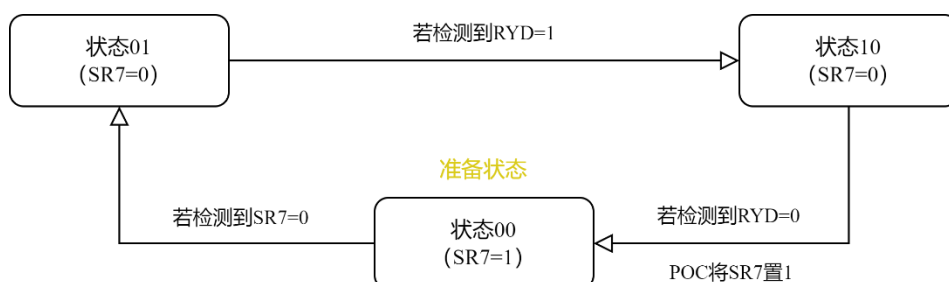
POC 将停留在此状态, 直到检测到 SR7=0, 表明 processor 已经写入新数据给 POC 且 POC 尚未处理, POC 将开始与 printer 握手, 跳转至状态 01 (state <= 2'b01)。

### 2) state = 2'b01:

POC 将停留在此状态, 直到检测到 RDY=1, 表明 printer 已经准备好接收新的数据, 此时 POC 要给打印机发 TR 脉冲, 并将 BR 中存储的数据传给并行数据端口 PD[7:0], 然后跳转至状态 10 (state <= 2'b10)

### 3) state = 2'b10:

POC 将停留在此状态, 直到检测到 RDY=0, 表示打印机已检测到 TR, 并开始打印, 打印机打印时, POC 将 TR 置零, SR7 置 1, 回到状态 00 (state <= 2'b00)。



## 四、设计思路

### 4.1 系统原理图

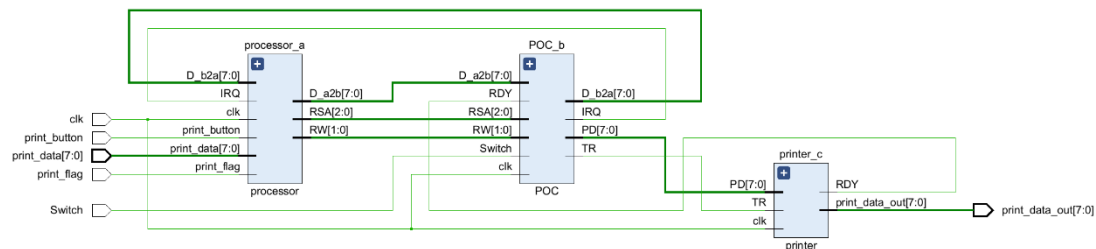


图 2. 系统原理图

### 4.2 Processor 模块

输入端口：clk（时钟信号），print\_data（要打印的数据），print\_flag（打印机开关），print\_button（开始打印按钮），IRQ，D\_b2a（POC 传输给 processor 的数据）。

输出端口：RW（读写控制），D\_a2b（processor 传输给 POC 的数据），RSA（寄存器选择地址信号）。

### 4.3 POC 模块

输入端口：clk，Switch（传输方式，0:查询方式，1:中断方式），RW，D\_a2b，RSA，RDY（Ready 信号，1:printer 已准备好，0:printer 未准备好）。

输出端口：IRQ，D\_b2a，TR（给 printer 的打印请求）PD（并行输出端口）。

### 4.4 Printer 模块

输入端口：clk，TR，PD，RDY。

输出端口：print\_data\_out（printer 打印的数据）

### 4.5 Top 模块

将以上三个模块封装例化的一个综合模块，是系统的桥梁。

## 五、仿真结果

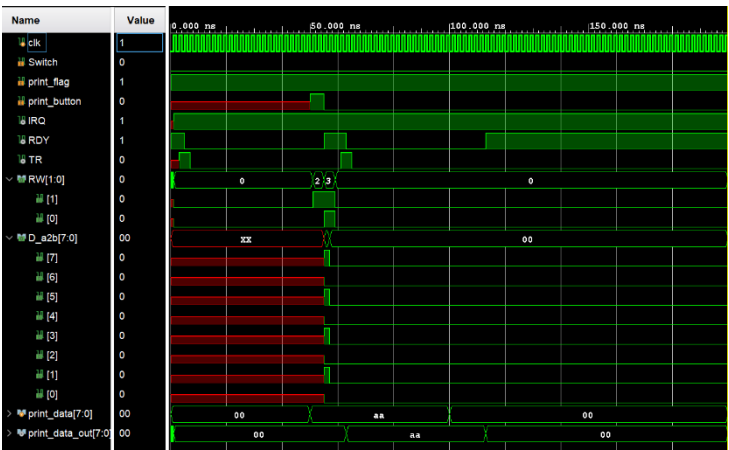


图 3. 查询模式仿真结果

#### 1. 查询模式仿真：

0-50ns: print\_flag=1, 表示打印机已开机，但并没有要输入的数据，故 print\_data=0 表示数据清零。

50-100ns: 按下按钮，产生了一段 print\_button 正脉冲，同时输入第一组数据，一段时间后 printer 顺利输出要打印的数据

100-200ns: 数据清零。

观察上述仿真结果，可见 IRQ 一直为高电平，中断请求无效，而 RW 在进入 11（写状态）前，会先进入 10（读状态），可见此时正在查询 SR7 状态，可见其查询模式的特点。（注：程序中设计再按下按钮再进入查询状态，而不是定时查询）。

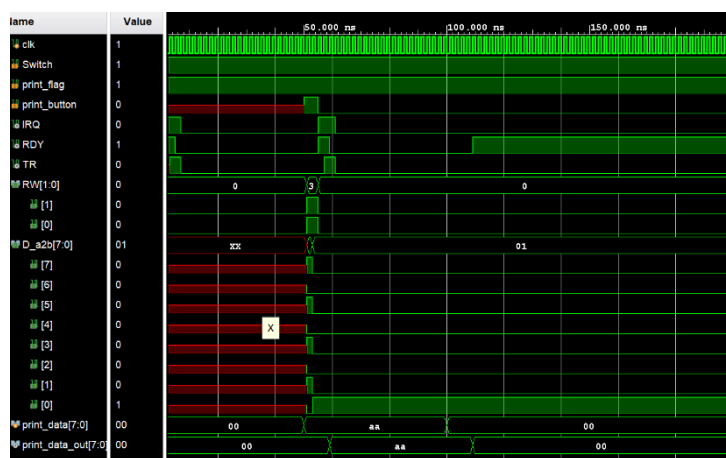


图 4. 中断模式仿真结果

观察上述仿真结果，可见 IRQ 除 SR7 为 0（POC 在与打印机握手）的时间外，均有低电平有效，且 RW 会直接进入 11（写状态），可见其中断模式的特点。

### 3. 综合仿真：

综合仿真一共仿真了 3 阶段 6 组数据，第一阶段为查询模式（0-300ns），共输出 2 组数据；第二阶段为中断模式（300-700ns），共输出 3 组数据；第三阶段为查询模式（700-850ns），输出 1 组数据。结果如下图所示，可见仿真结果符合预期。

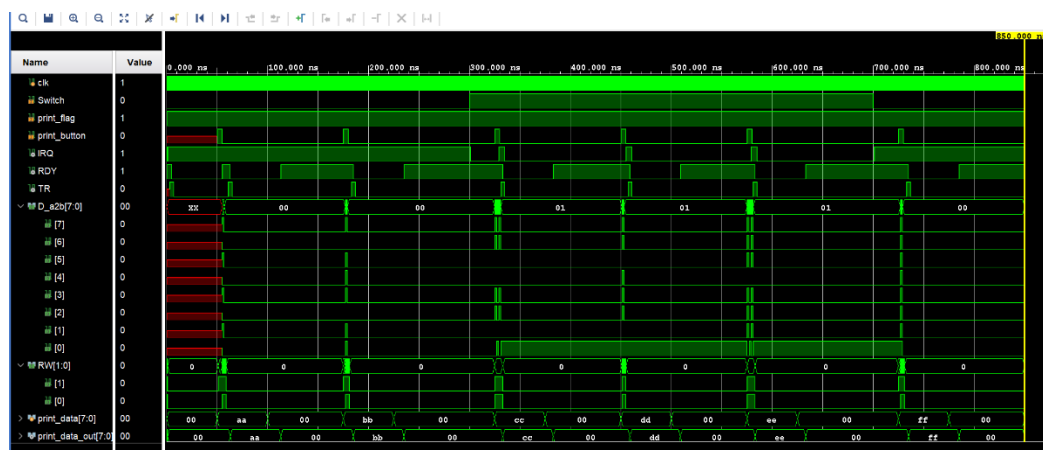


图 5. 综合仿真结果

附：具体代码可见：[https://github.com/AlbertCouchy0/POC\\_Design\\_Based\\_on\\_FPGA](https://github.com/AlbertCouchy0/POC_Design_Based_on_FPGA)