计算机组成与结构专题实验

# 实验报告

# 第四次 程序计数器PC与地址寄存器AR

## 实验目的

1．掌握地址单元的工作原理；

2．掌握 PC 的两种工作方式，加 1 计数和重装计数器初值的实现方法；

3．掌握地址寄存器 AR 从程序计数器 PC 获得数据和从内部总线 BUS 获得数据的实现方法。

## 实验原理

### 1．采用总线多路开关联接方式

地址单元主要由三部分组成：程序计数器 PC、地址寄存器 AR 和多路开关 BUSMUX。程序计数器 PC用以指出下一条指令在主存中的存放地址， CPU 正是根据 PC 的内容去存取指令的。因程序中指令是顺序执行的，所以 PC 有自增功能。程序计数器提供下一条程序指令的地址，在 T4 时钟脉冲的作用下具有自动加 1 的功能；在 LDPC 信号的作用下可以预置计数器的初值（如子程序调用或中断相应等）。当 LDPC 为高电平时，计数器装入 data[ ]端输入的数据。 aclr 是计数器的清 0 端，高电平有效（高电平清零）； aclr 为低电平时，允许计数器正常计数。

地址寄存器 AR（ DFF\_8 或 273）锁存访问内存 SRAM 的地址。地址寄存器 AR 的地址来自两个渠道，一是程序计数器 PC 的输出，通常是下一条指令的地址；二是来自于内部数据总线的数据，通常是被访问操作数的地址。为了实现对两路输入数据的切换，在 FPGA 的内部通过总线多路开关 BUSMUX 进行选择。 LDAR与多路选择器的 sel 相连，当 LDAR 为低电平，选择程序计数器的输出；当 LDAR 为高电平时，选择内部数据总线的数据。

### 2．采用 PC、 AR 通过三态门 lpm\_bustri 与 BUS 联接

程序计数器 PC 与地址寄存器 AR 结合，产生对存储器 RAM 进行读写的地址。地址单元主要由三部分组成：程序计数器 PC、地址寄存器 AR 和三态门 lpm\_bustri。程序计数器 PC 用以指出下一条指令在主存中的存放地址， CPU 正是根据 PC 的内容去存取指令的。程序计数器提供下一条程序指令的地址，在时钟脉冲 PC\_CLK 的作用下具有自动加 1 的功能；在 LOAD\_PC 信号的作用下可以预置计数器的初值（如子程序调用或中断相应等）。当 LOAD\_PC 为高电平时，计数器装入 data[7..0]端输入的数据。 RST是计数器的清 0 端，高电平有效（高电平清零）； RST 为低电平时，允许计数器正常计数。 地址寄存器 AR采用锁存器 lpm\_latch 结构,锁存访问内存 SRAM 的地址。

## 实验任务

### 实验任务 1

按照电路图编辑、输入电路，实验台选择 0 工作模式。对输入原理图进行编译、引脚锁定、并下载到实验台。硬件实验验证，与仿真波形图比较。

1．用模式键选模式“0”，再按一次系统的复位键；键 2（锁定 PIO12~PIO15)和键 1(PIO8~PIO11),可输入 8 位总线数据 B[7..0] （此值显示于发光管 D1~D8 和数码管 2/1(PIO16~PIO23）)；CLR（键 5,锁 PIO4）按 2 次(0🡪 1🡪 0)，产生一正脉冲，高电平清零； LDAR（键 6 锁 PIO5） =0 时， BUSMUX输出程序计数器 PC 的值； LDAR=1 时， BUSMUX 输出 D[7..0]总线数据。 LDPC（键 7 锁 PIO6）：程序计数器 PC 预置控制端，当 LDPC=1 时，将 D[7..0]总线数据装入程序计数器 PC；当 LDPC=0 时，程序计数器 PC 处于计数自动工作状态，对 T4 进行计数； T4（键 8 锁 PIO7）：程序计数器 PC 的计数时钟 CLK，键 8 按动两次产生一个计数脉冲。

2．通过 D[7..0]设置程序计数器的预加载数据。当 LDPC=0 时，观察程序计数器自动加 1 的功能；当LDPC=1 时，观察程序计数器加载输出情况。示例操作步骤如下：

（1）所有键置 0，键 2/键 1 输入 A5；按键 5🡪 PC 计数器清 0(0🡪 1🡪0)；

（2）连续按动键 8，可以从数码 8/7 上看到 Q 的输出，即 PC 值自动从 0 加不断累加 1；

（3）按键 6，输出高电平 ‘1’，选通直接输出总线上的数据 A5 作为 PC 值，按键 8，产生一个脉冲上升沿，即可看到 Q（显示在数码 8/7）的输出为 A5； 如键 6 再置‘1’，按动键 8，发现，又回到原来初始的计数值开始计数。

（4）使键 7=1，仍选通 PC 计数器输出，这时键 2/1 输入 86，按键 8 产生一个上升脉冲(0🡪 1🡪 0)，即用 LDPC 将 86 加载进 PC 计数器； 再把键 7=0；

（5）连续按动键 8，可以发现 AR 的输出在 86 上累加输出： 86、 87、 88 等。

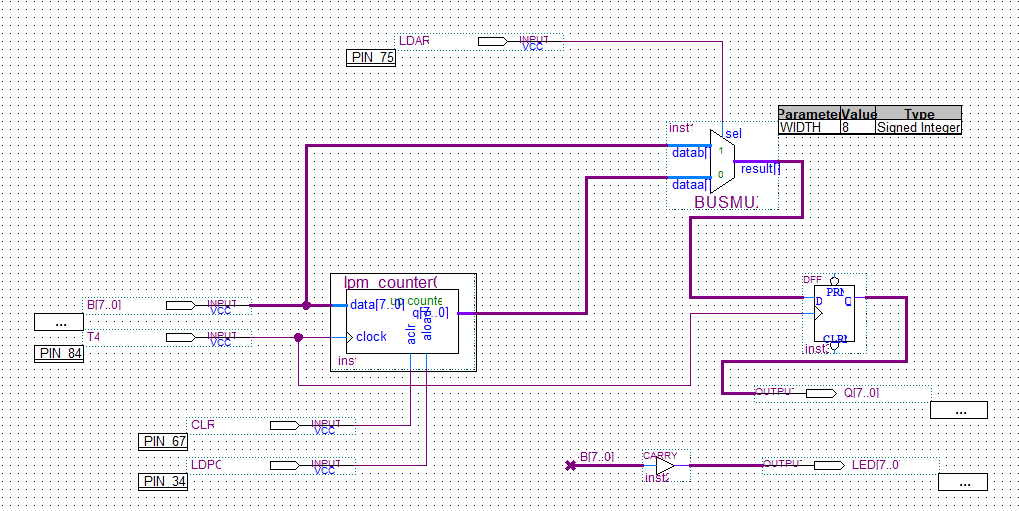
### 实验任务 2

按照程序计数器原理图编辑、输入电路，采用 LPM 库中的元件 lpm\_latch 锁存器、lpm\_counter 计数器和 lpm\_bustri 总线三态输出缓冲器进行设计，对于 KX-CDS 实验系统，建议实验台选择 NO.0 工作模式，按照以上方式完成实验。键 1、键 2 输入 8 位地址数据，键 3～键 8 分别作为 RST、 PC\_CLK、LOAD\_PC、 INPUT\_B、 PC\_B、 AR\_CLK； D1～D8 显示输入的地址数据，数码 1/2 显示地址锁存器的输出数据AR[7..0]，数码 3/4 显示程序计数器的输出数据 PC[7..0]。对输入原理图进行编译、引脚锁定、并下载到实验台进行硬件验证。

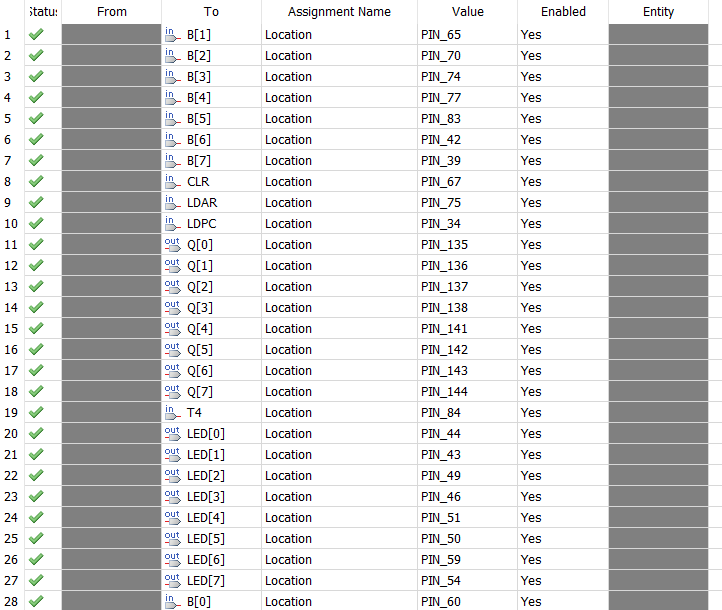
## 实验步骤及结果

### 实验任务 1

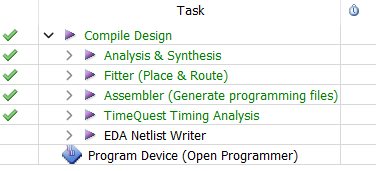
电路图绘制

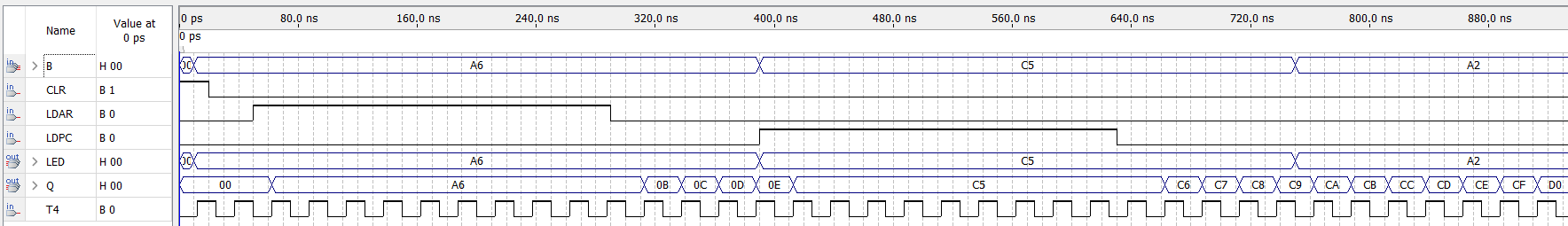


绑定引脚



编译并下载至FPGA，仿真

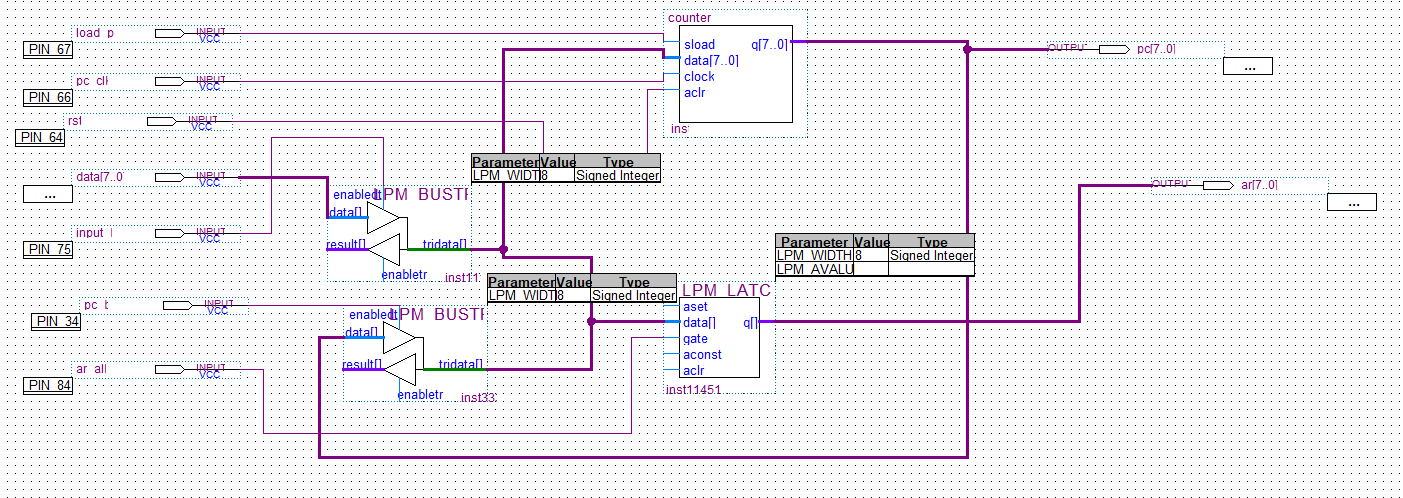




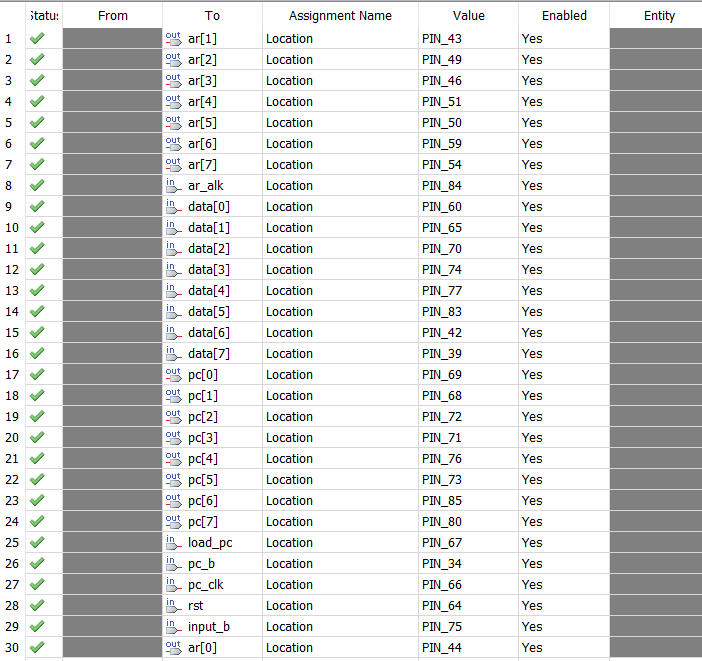
实验结果与预期结果一致。

### 实验任务 2

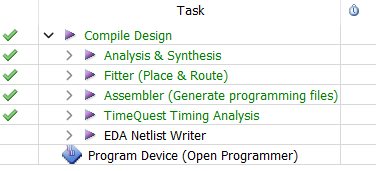
电路图绘制

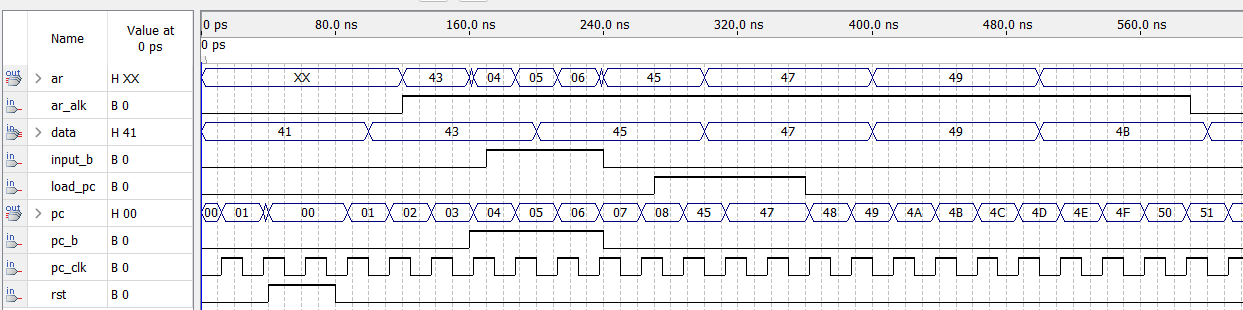


绑定引脚



编译并下载至FPGA，仿真





实验结果与预期结果一致。

## 实验总结及问题分析

**思考题1，执行分支/转移指令与执行普通指令时，对于实验中两种实现方式，地址单元的操作有何区别？**

当执行分支/转移指令时，两种方式都会从总线上读取地址，并将该地址存储到LPM\_COUNTER中，然后继续取指令执行。

而在执行顺序指令时，两种实现方式都是从LPM\_COUNTER中获取PC的值，然后在内存中找到对应地址的指令。接着，它们会将指令中操作数的地址放入输入数据中。对于第一种方式，这意味着提供一个高脉冲给T4；而对于第二种方式，则需要打开input\_b并给gate一个高脉冲。这两种方式在执行跳转指令时，仍然会从总线上读取地址并将其保存到LPM\_COUNTER中，然后继续取指令执行。

### 思考题2，从存储器读取运算数据和取指令操作时，对于实验中两种实现方式，地址单元完成的操作有何不同？

在第一种方式中，当需要取指令时，地址单元会将BUSMUX设置为0，然后通过LPM\_COUNTER输出PC的值，并将其通过DFF传输到AR中。而当需要取数据时，地址单元会将BUSMUX设置为1，然后直接通过DFF将数据放到AR中。

相比之下，在第二种方式中，当需要取指令时，指令会通过LPM\_BUSTRI、LPM\_COUNTER、LPM\_BUSTRI传输到LPM\_LATCH输出。而当需要取数据时，则直接通过LPM\_BUSTRI传输到LPM\_LATCH输出。