**东南大学电工电子实验中心**

**实 验 报 告**

**课程名称： 数字电路实验**

**第 4 次实验**

实验名称： FPGA时序逻辑设计

院 （系）：电气工程学院专 业：电气工程及其自动化

姓 名： 王皓冬 学 号： 16022627

实 验 室: 401 实验组别：

同组人员： 实验时间：

评定成绩： 审阅教师：

**一、实验目的**

1、 进一步学习时序逻辑电路的分析和设计方法，掌握状态机设计方法

2、 了解可编程数字系统设计的流程，掌握 Quartus II 软件的使用方法

3、 掌握原理图输入方式设计时序逻辑电路的方法和流程

**二、实验原理**

**0 输入信号与输出信号编码**

输入信号

进行动态验证时，由于该电路以预置的形式产生了信号，而未采用input形式，因此该实验中***无输入信号***。而需要手动加载数据时，采用CLK按键与拨码开关作与逻辑的方式进行信号输入。故

***Input：***手动加载的序列信号，用拨码开关实现。

为了同一两种检测方式，引入选择功能，采用拨码开关实现。即

***Model：***模式选择。为“1”时手动输入信号进行验证，为“0”时检测预置的信号“1011011100100”。

输出信号

***OD：***Order，输出预定信号“111001”部分模块的实际输出。

***Y：***输出的待测信号“1011011011100100”的实际输出。

***Z：***代表“待测信号中是否被检测到含有预定信号”，“0”代表未检测到，“1”代表检测到。

***CLKo：***输出时钟信号，用于拓展功能，便于验证。o代表out。

***Modelo：***输出Model，直观显示当前检测模式。用于拓展功能，便于验证。o代表out。

***X：***对每一个CLK内的OD与Y异或，用于表示OD与Y信号是否相等。“0”代表不相等，“1”代表相等。用于拓展功能，便于验证。

时钟信号

用***CLK***代表时钟信号。实验中的时钟信号是唯一的，仿真时，CLK的周期为20ns。由于利用FPGA单步验证时，其按钮按下产生低电平，与通常的CLK信号相反，因此电路设计时采用了CLKi(i代表in)作为输入时钟信号的标识，CLK则仍作为接入各芯片的时钟信号标识。

其他标识说明

***LD：***二进制码实现序列检测器时接入161LD端的导线逻辑。LD为“0”时，161实现加载数据的功能。

***Nand：***One-Hot码中接回D0触发器、用作自启动的导线逻辑。Nand为“1”，即与非的项中含有逻辑“0”时，D0触发器传递1，输出0，第一个译码器开始工作。

**1、 实验内容：**

**(1) 必做：**用二进制编码设计一个“111001”序列检测器，对串行输入序列进行检测，当连续检测到 6 个码元符合检测码“111001”时，检测器输出为 1

**选做：**用 One-Hot 编码设计一个“111001”序列检测器，对串行输入序列进行检测，当连续检测到 6 个码元符合检测码“111001”时，检测器输出为 1 注意：“111001”序列的最后一个码元不能作为当前待测序列的第一个码元

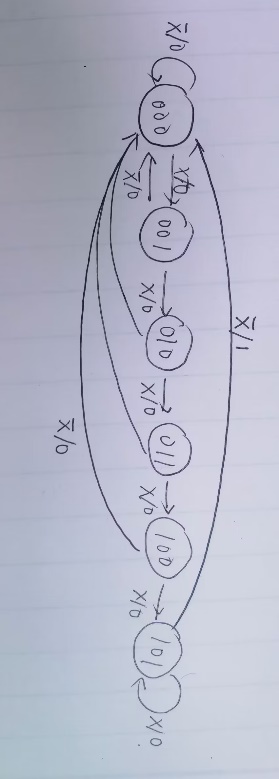
**(2) 用静态（单步）的测试方法对实验结果进行验证**

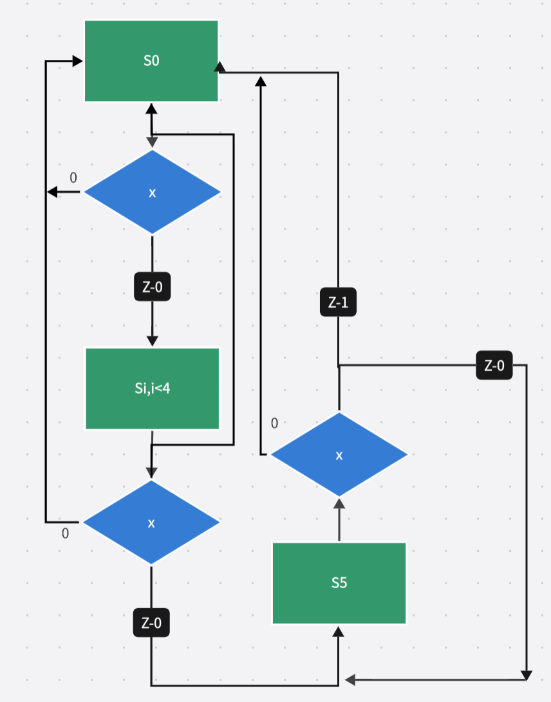
**(3) 自行设计一个“1011011011100100”序列发生器用于对实验结果进行动态验证**

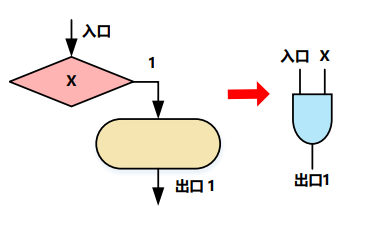
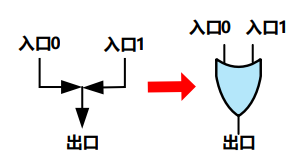
**(4) 用双踪示波器观察并记录动态验证结果。**

**2、 实验要求：**

(1) 根据设计要求划分设计层次、单元模块和接口信号，在报告上记录设计过程，绘制系统框图，每个模块的状态转移图和 ASM 图，并设计验证方案。







X的判断是通过真值表写出逻辑函数式实现的。

(2) 用原理图输入法设计所有单元模块并编译，分析编译时产生的错误和警告信息

(3) 对所有的单元模块进行功能仿真，并记录和分析全部仿真结果

(4) 在顶层文件中连接全部单元模块并编译、综合、分配管脚和适配。

(5) 对整个系统进行时序仿真，并记录和分析仿真结果。

(6) 将仿真正确的设计下载到实验箱上，连接输入输出设备和示波器进行板级验证

**1. 状态图**

**该工程包含的模块如下：**

(1) **待测信号产生模块**。本质是序列发生器，包括手动输入模块与预置信号(1011011011100100)模块。

(2) **判别相等模块**，判断每一位信号是否相等。

(3) **预定信号模块**，即“111001”。本质是序列发生器。

(4) **序列检测器模块**。

**判别相等：**

判别数值是否相等的功能可以通过同或逻辑实现。由于器件中没有同或门，采用异或+非的逻辑实现。将OD与Y异或后得到X，即判断两信号是否相等。

表 1 判断相等信号

|  |  |  |
| --- | --- | --- |
| **OD** | **Y** | **X** |
| 0 | 0 | 1 |
| 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 |

**序列检测器(Sequential Detector，SD)：**

**6位SD(二进制实现)：**

表 2 二进制码实现6位序列检测器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **文字描述** | | | | **二进制编码** | | | | | | **74161集成计数器** | | | | | | |
| **现态** | **次态** | **条件** | **输出** | **现态** | | | **次态** | | | **功能** | **端口** | | | | | |
| Q2 | Q1 | Q0 | Q2 | Q1 | Q0 |  | EN | LD | D3 | D2 | D1 | D0 |
| S0 | S0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 保持 | 0 | 1 |  |  |  |  |
| S1 | X | 0 | 0 | 0 | 1 | 计数 | 1 | 1 |  |  |  |  |
| S1 | S0 |  | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 置数 | 0 | 0 | 0 | 0 | 0 | 0 |
| S2 | X | 0 | 0 | 1 | 0 | 计数 | 1 | 1 |  |  |  |  |
| S2 | S0 |  | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 置数 | 0 | 0 | 0 | 0 | 0 | 0 |
| S3 | X | 0 | 0 | 1 | 1 | 计数 | 1 | 1 |  |  |  |  |
| S3 | S0 |  | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 置数 | 0 | 0 | 0 | 0 | 0 | 0 |
| S4 | X | 0 | 1 | 0 | 0 | 计数 | 1 | 1 |  |  |  |  |
| S4 | S0 |  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 置数 | 0 | 0 | 0 | 0 | 0 | 0 |
| S5 | X | 0 | 1 | 0 | 1 | 计数 | 1 | 1 |  |  |  |  |
| S5 | S0 |  | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 置数 | 0 | 0 | 0 | 0 | 0 | 0 |
| S5 | X | 0 | 1 | 1 | 0 | 保持 | 0 | 1 |  |  |  |  |

**6位SD(One-Hot码实现)：**

表 3 One-Hot码实现6位序列检测器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **文字描述** | | | | **One-Hot码** | | | | | | | | | | | |
| **现态** | **次态** | **条件** | **输出** | **现态** | | | | | | **次态** | | | | | |
| Q5 | Q4 | Q3 | Q2 | Q1 | Q0 | Q5 | Q4 | Q3 | Q2 | Q1 | Q0 |
| S0 | S0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| S1 | X | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| S1 | S0 |  | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| S2 | X | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| S2 | S0 |  | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| S3 | X | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| S3 | S0 |  | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| S4 | X | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| S4 | S0 |  | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| S5 | X | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| S5 | S0 |  | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| S5 | X | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

**待测信号：**

由于手动输入待测信号逻辑简易，在这里给出用序列发生器产生“1011011100100”的真值表。用74161+2片74138的组合获得16位序列发生器。2片74138组装为4线-16线译码器，将对应地址与非输出即可得到待测信号“1011011011100100”。由于74161恰为16进制计数器，故无需特意改装，正常计数即可。

表 4 待测信号4线-16线译码器组装

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **信号** | **高位芯片** | | | | **低位芯片** | | | |
| **QD** | **EN1** | **EN2A** | **EN2B** | **功能** | **EN1** | **EN2A** | **EN2B** | **功能** |
| 0 | 0 | x | x | 封锁 | 1 | 0 | 0 | 工作 |
| 1 | 1 | 0 | 0 | 工作 | x | 1 | 1 | 封锁 |

表 5 待测信号输出Y

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **地址输出** | | | | | | | | | **信号输出** |
| **Y0** | **Y2** | **Y3** | **Y5** | **Y6** | **Y8** | **Y9** | **Y10** | **Y13** | **Y** |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

**预定信号：**

用74161+74138的组合获得6位序列发生器，原理同上。其中，由于是将预定信号与待测信号逐位比较，应在比较结果为“不相同”时重新从预定序列的开端开始比较。即，假如比较序列结果为“假”，预定信号应直接重新产生“111001”的序列。因此，置数端LD的接线需要特别说明。

表 6 预定信号输出OD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **地址输出** | | | | **信号输出** |
| **Y0** | **Y1** | **Y2** | **Y5** | **OD** |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

**预定序列ASPRG模块的LD端：**

由于X信号已经代表了当前Y与OD是否相等，且X输出为“0”时代表判断序列不相等，LD端低电平有效。因此，X可直接作为LD端的接入。考虑到预定序列的ASPRG本身是一个模6计数器，在从0计数到5后应当返回0计数，这两个条件是“或”的关系。即：序列不相等，或序列完整输出了，则重新计数。因此，将二者(X与)作与运算即可。

**2. 状态转移方程**

根据状态转移表求得状态转移方程如下。

**2.1 SD：**

SD(二进制码)：

其中，代表（Q2Q1Q0）组成的最小项的现态。

SD(One-Hot码)：

**2.2 ASPRG：**

自动加载数据“101…”时，用于产生待检测信号的16位ASPRG中，138组装4线-16线译码器，高位芯片与低位芯片的使能端分别如方程组（1）（2）所示：

(1)

(2)

而手动加载数据时，待测信号的输出仍用Y指代，则有：

**2.3 各组件连接部分**

由上文分析，对预定序列序列发生器(二进制码)LD脚(记为LDasprg)，有：

其中，是该序列发生器74138的脚输出。

同样，对预定序列序列发生器(One-Hot码)LD脚，有：

产生X的异或逻辑为：

对于待测信号，记Model=0时的Y为Y0，Model=1时的Y为Y1，则总电路的待测信号应有：

实际上，由于手动输入的信号在FPGA板上是通过拨码开关代替的，考虑到直接同上式接线会导致Input无视CLK信号，在CLK低电平期间改变拨码开关状态同样会直接导致Y的改变。因此，使Input经过了一个D锁存器的寄存，锁存器的时钟信号与CLK同步。(验证表明，这是必不可少的！！因为没按CLK的时候换Input开关状态也计入信号检测，直接导致信号判断为不相等！)

**3. 电路原理图**

首先给出Model=0（预置信号模式）的电路图。

**预定序列序列发生器**：



**待测信号序列发生器：**

****

**SD(二进制码实现)：**

****

**SD(One-Hot码实现)：**

****

**待测信号产生合块(动态+手动)：**

****

综合可得，总电路如下：

**总电路(二进制码实现)：**

****

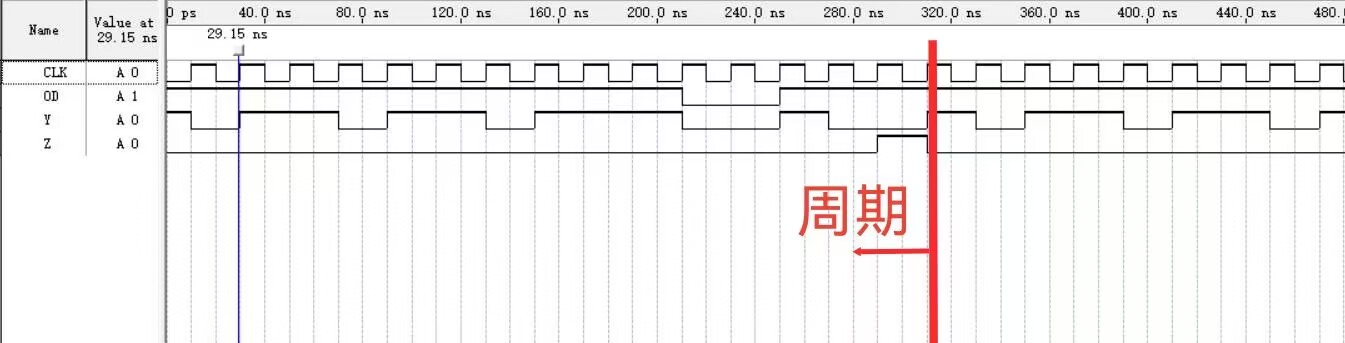
**总电路(One-Hot码实现)：**

**4. 仿真**

在此说明，仿真均是利用Model=0时的功能仿真。方便起见，直接采用了未合并的电路（即为加入Model与Input信号的电路）进行仿真。

4.1 二进制码实现

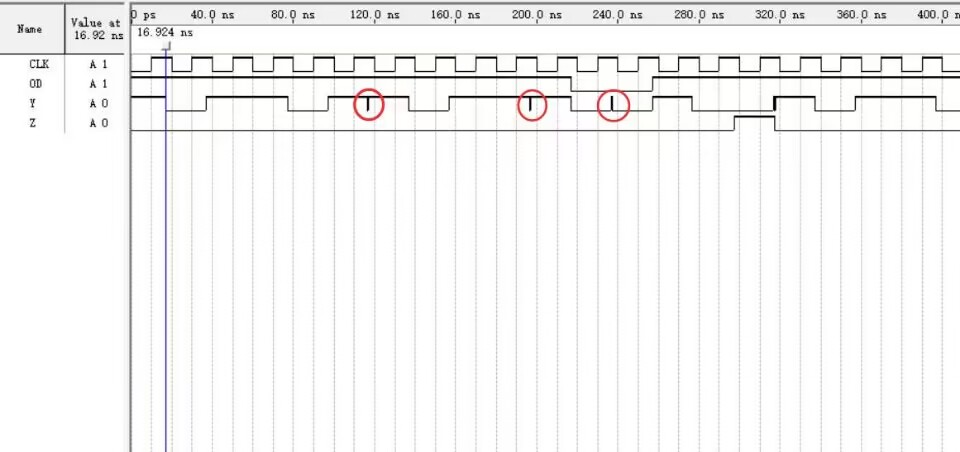
功能仿真：

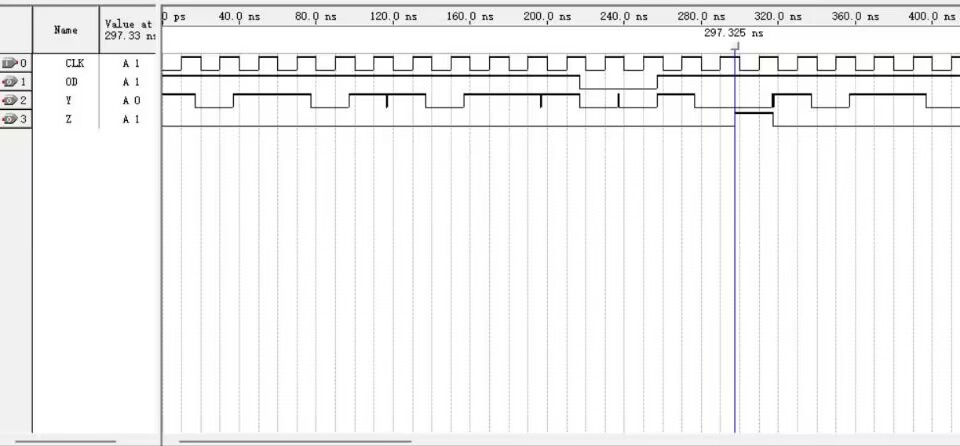




可以观察到，在相同波形（111001）后的一个周期后，Z跃变为1。延迟了一个周期是由输出端的D触发器引起的。功能仿真正确。

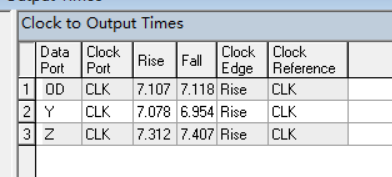
时序仿真：





可以观察到，Y信号约延迟CLK信号6.924ns，Z信号约延迟CLK信号7.325ns。OD、Z均无毛刺产生，毛刺主要集中在Y的波形上。以第一个毛刺为例，是在138低位芯片从Y5转到Y6时形成的，而此时高位芯片的Y6脚为低。观察发现毛刺出现时，不工作的芯片上均有工作芯片对应脚相反的信号。故推测毛刺的形成是两片138芯片之间使能端间不同步，产生竞争-冒险导致的。也因而，同样用于序列发生器的OD波形上没有毛刺产生。

同时，对照延迟信息：

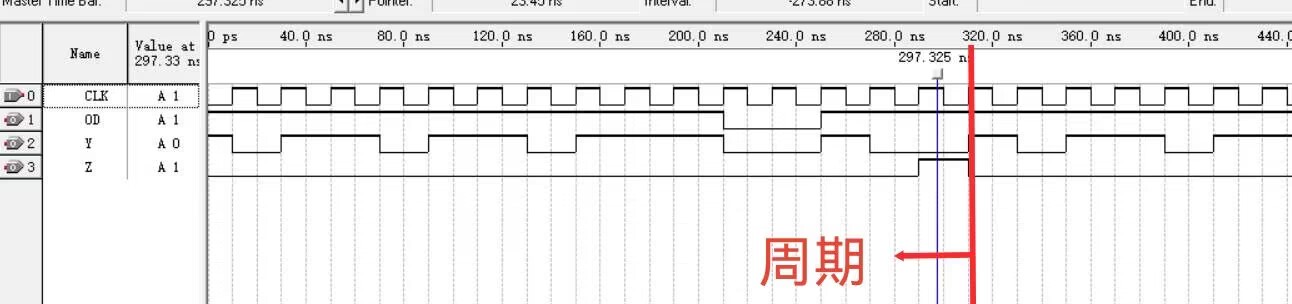


发现均与全部毛刺的产生时间无法对应。因此可以排除CLK产生毛刺的假设。

同样可以发现，毛刺的产生对Z的判断没有影响。

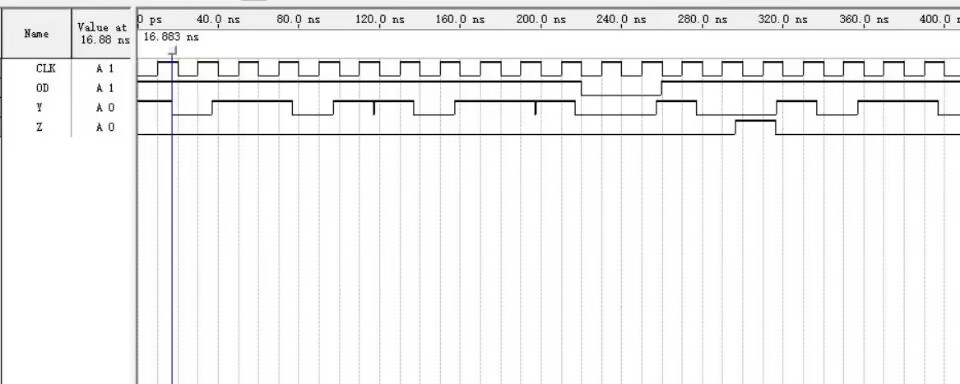
4.2 One-Hot码实现

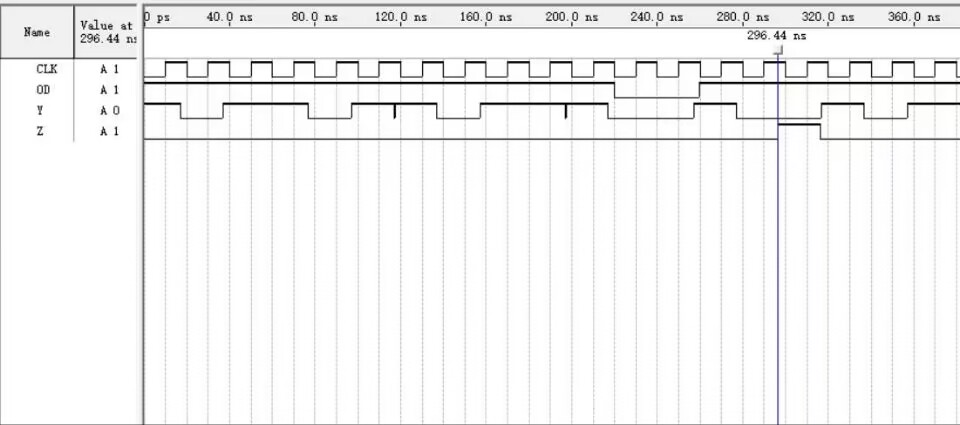
功能仿真：



分析同二进制码电路，功能正确。

时序仿真：

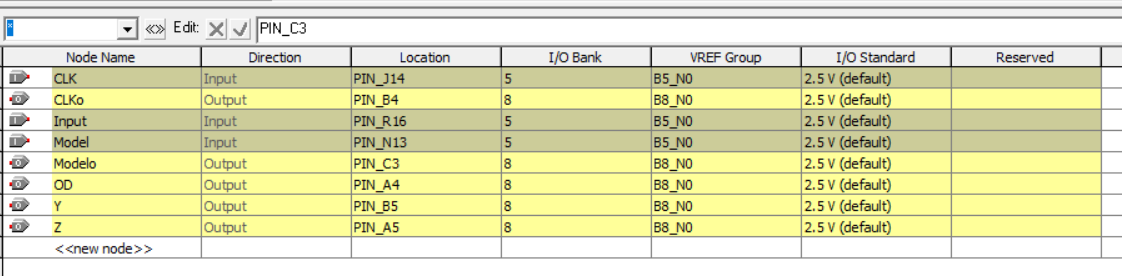




可以观察到，Y信号约延迟CLK信号6.883ns，Z信号约延迟CLK信号6.44ns。毛刺的分析同二进制码电路。

**5. 管脚分配**

管脚分配如下。其中，为便于实现静态验证，CLK接到按键J14上。



**6. 验证方案**

预置序列验证（1011011011100100）时，仅需将开关SW7断开，按动按钮KEY0产生单次静态脉冲作为CLK验证。手动输入序列验证时，将开关SW7闭合，同时改变开关SW0的状态后按动按钮产生单次静态脉冲。SW0断开时按动按钮等效于输入“0”，闭合时按动按钮等效于输入“1”。

手动输入序列验证时，输入不同的序列信号，观察是否能正常实现功能。

**三、实验记录**（记录实验具体步骤、原始数据、实验过程、实验中遇到的故障现象、排除故障的过

程和方法等）

实验步骤：

遇到的问题及解决方法：

**四、实验仪器**

**五、实验小结**（总结实验完成情况，对设计方案和实验结果做必要的讨论，简述实验收获和体会）

**六、参考资料**