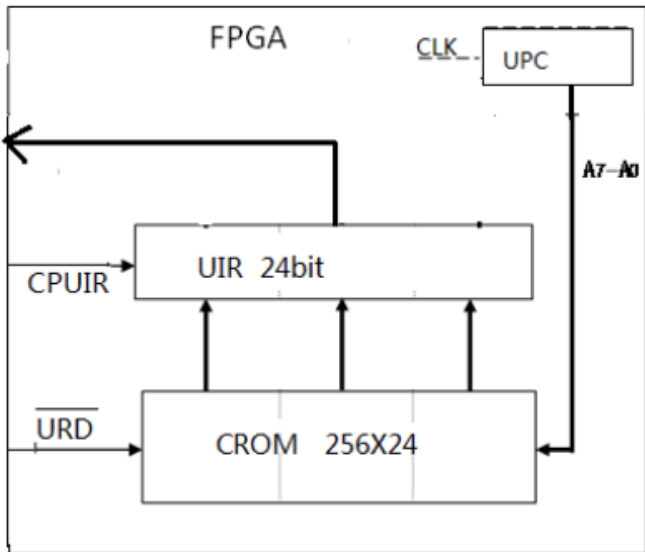


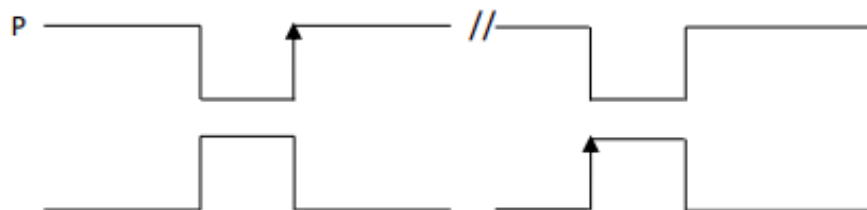
计算机组成与设计 课程实验报告

学号：202000130143	姓名： 郑凯饶	班级：2020 级 1 班
实验题目： 控制器实验		
实验学时：2	实验日期： 2022-5-29	
实验目的： 采用微程序方法设计控制器。		
实验软件和硬件环境： 软件环境： QuartusII 软件 硬件环境： 1. 实验室台式机 2. 计算机组成与设计实验箱		
实验原理和方法： 1. 微程序控制器电路结构如下，有控制存储器 CROM、微程序 uPC 计数器和伪指令寄存器 uIR 构成：		
 <p>图 13-1 微程序控制器框图</p> <p>->UPC 提供 8 位微地址 ->在控存信号的作用下，读出一条长 24 位的微指令代码 ->在打入命令 CPUIR 的作用下送入 UIR</p>		
2. 时钟信号控制原理		

每当按一次脉冲键便产生一个负脉冲，该脉冲的作用是：

- 作为读控存的命令 $\overline{\mu RD}$ 。
- 负脉冲当作 $CP_{\mu IR}$ 将读出的微指令打入微指令寄存器 μIR 。
- 负脉冲的上升沿使 $\mu PC+1$ 形成下一条微指令的地址。
- 负脉冲反相后的上升沿作为寄存器打入脉冲。

微程序时序如图 13-2。



3. UPC 设计

(1) 微程序计数器 μPC 的设计，完成 8 位具有加 1 功能和清除功能的计数器设计并封装，如图 13-3。

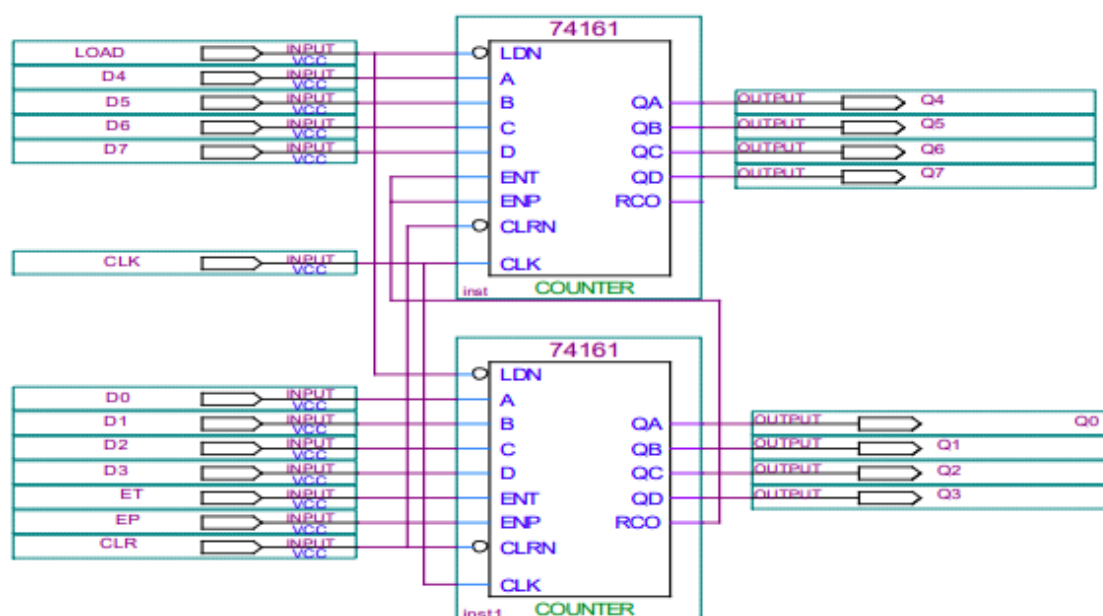


图 13-3 UPC 的设计图

其中，CLR：清零端，低电平有效；CLR=0 时，Q7Q6Q5Q4Q3Q2Q1Q0=00000000；

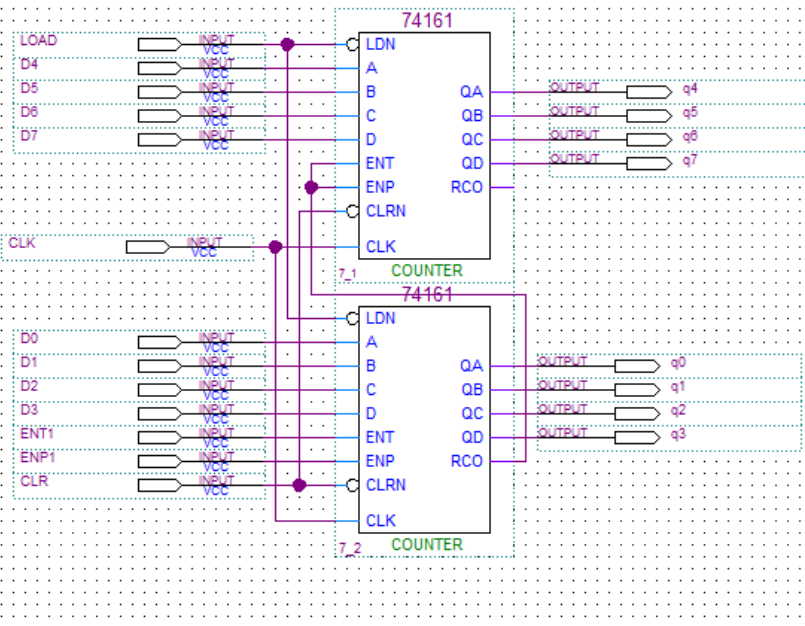
LOAD：置数端，低电平有效；LOAD=0 时，在 CLK 的上升沿，

Q7Q6Q5Q4Q3Q2Q1Q0=D7D6D5D4D3D2D1D0；

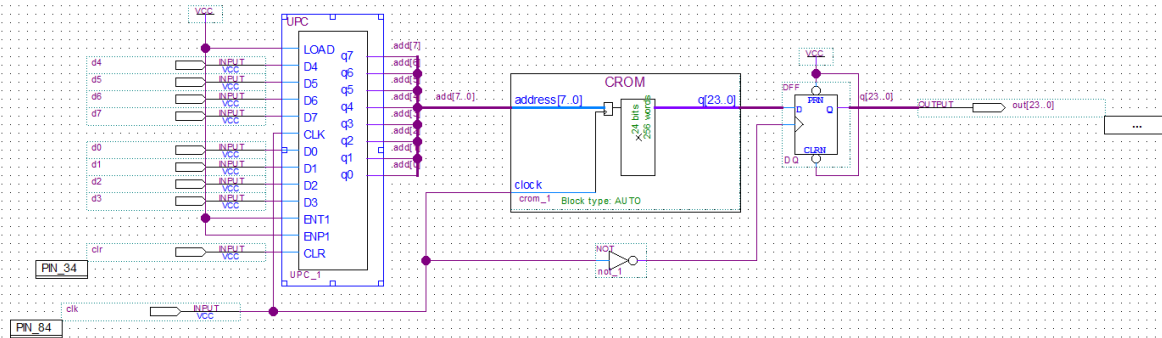
当 CLR=1，LOAD=1，ET=1，EP=1 时，对 CLK 进行增 1 计数。

注意：本实验使用时，只有 clk、CLR 两引脚引出，其它引脚，ET、EP、LOAD 接高电平。

实验步骤：
 连接电路原理图：
 UPC：



控制器整体：



引脚分配：

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair
altera_reserved_tck	Input				PIN_16	2.5 V (default)		8mA (default)		
altera_reserved_tdi	Input				PIN_15	2.5 V (default)		8mA (default)		
altera_reserved_tdo	Output				PIN_20	2.5 V (default)		8mA (default)	2 (default)	
altera_reserved_tms	Input				PIN_18	2.5 V (default)		8mA (default)		
clk	Input	PIN_84	5	85_NO	PIN_84	2.5 V (default)		8mA (default)		
dr	Input	PIN_34	2	82_NO	PIN_34	2.5 V (default)		8mA (default)		
d0	Input				PIN_30	2.5 V (default)		8mA (default)		
d1	Input				PIN_121	2.5 V (default)		8mA (default)		
d2	Input				PIN_88	2.5 V (default)		8mA (default)		
d3	Input				PIN_89	2.5 V (default)		8mA (default)		
d4	Input				PIN_42	2.5 V (default)		8mA (default)		
d5	Input				PIN_77	2.5 V (default)		8mA (default)		
d6	Input				PIN_43	2.5 V (default)		8mA (default)		
d7	Input				PIN_124	2.5 V (default)		8mA (default)		
out[23]	Output	PIN_144	8	88_NO	PIN_144	2.5 V (default)		8mA (default)	2 (default)	
out[22]	Output	PIN_143	8	88_NO	PIN_143	2.5 V (default)		8mA (default)	2 (default)	
out[21]	Output	PIN_142	8	88_NO	PIN_142	2.5 V (default)		8mA (default)	2 (default)	
out[20]	Output	PIN_141	8	88_NO	PIN_141	2.5 V (default)		8mA (default)	2 (default)	
out[19]	Output	PIN_138	8	88_NO	PIN_138	2.5 V (default)		8mA (default)	2 (default)	
out[18]	Output	PIN_137	8	88_NO	PIN_137	2.5 V (default)		8mA (default)	2 (default)	
out[17]	Output	PIN_136	8	88_NO	PIN_136	2.5 V (default)		8mA (default)	2 (default)	
out[16]	Output	PIN_135	8	88_NO	PIN_135	2.5 V (default)		8mA (default)	2 (default)	
out[15]	Output	PIN_125	7	87_NO	PIN_125	2.5 V (default)		8mA (default)	2 (default)	
out[14]	Output	PIN_128	8	88_NO	PIN_128	2.5 V (default)		8mA (default)	2 (default)	
out[13]	Output	PIN_114	7	87_NO	PIN_114	2.5 V (default)		8mA (default)	2 (default)	
out[12]	Output	PIN_120	7	87_NO	PIN_120	2.5 V (default)		8mA (default)	2 (default)	
out[11]	Output	PIN_105	6	86_NO	PIN_105	2.5 V (default)		8mA (default)	2 (default)	
out[10]	Output	PIN_113	7	87_NO	PIN_113	2.5 V (default)		8mA (default)	2 (default)	
out[9]	Output	PIN_100	6	86_NO	PIN_100	2.5 V (default)		8mA (default)	2 (default)	
out[8]	Output	PIN_101	6	86_NO	PIN_101	2.5 V (default)		8mA (default)	2 (default)	
out[7]	Output	PIN_80	5	85_NO	PIN_80	2.5 V (default)		8mA (default)	2 (default)	
out[6]	Output	PIN_85	5	85_NO	PIN_85	2.5 V (default)		8mA (default)	2 (default)	
out[5]	Output	PIN_73	5	85_NO	PIN_73	2.5 V (default)		8mA (default)	2 (default)	
out[4]	Output	PIN_76	5	85_NO	PIN_76	2.5 V (default)		8mA (default)	2 (default)	
out[3]	Output	PIN_71	4	84_NO	PIN_71	2.5 V (default)		8mA (default)	2 (default)	
out[2]	Output	PIN_72	4	84_NO	PIN_72	2.5 V (default)		8mA (default)	2 (default)	
out[1]	Output	PIN_68	4	84_NO	PIN_68	2.5 V (default)		8mA (default)	2 (default)	
out[0]	Output	PIN_69	4	84_NO	PIN_69	2.5 V (default)		8mA (default)	2 (default)	
<<new node>>										

测试、调试：

初始化 ROM 数据文件如下：

Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	0	7	14	21	28	35	42	49#*1
8	56	63	70	77	84	91	98	105	8?FMT[b
16	112	119	126	133	140	147	154	161	pw~.....
24	168	175	182	189	196	203	210	217
32	224	231	238	245	252	259	266	273
40	280	287	294	301	308	315	322	329
48	336	343	350	357	364	371	378	385
56	392	399	406	413	420	427	434	441
64	448	455	462	469	476	483	490	497
72	504	511	518	525	532	539	546	553
80	560	567	574	581	588	595	602	609
88	616	623	630	637	644	651	658	665
96	672	679	686	693	700	707	714	721
104	728	735	742	749	756	763	770	777
112	784	791	798	805	812	819	826	833
120	840	847	854	861	868	875	882	889
128	896	903	910	917	924	931	938	945
136	952	959	966	973	980	987	994	1001
144	1008	1015	1022	1029	1036	1043	1050	1057
152	1064	1071	1078	1085	1092	1099	1106	1113
160	1120	1127	1134	1141	1148	1155	1162	1169



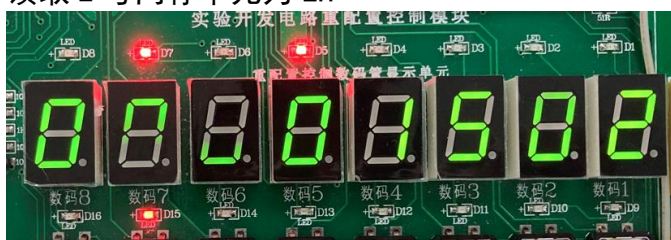
读取 0 号内存单元为 0H



读取 1 号内存单元为 7H



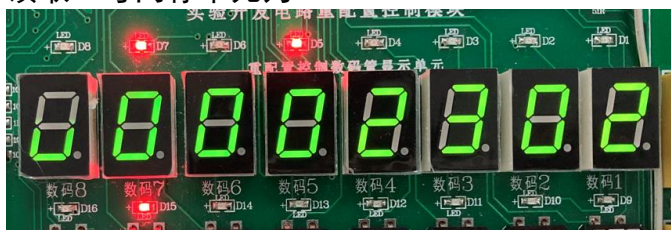
读取 2 号内存单元为 EH



读取 3 号内存单元为 15H



读取 4 号内存单元为 1CH



读取 5 号内存单元为 23H

验证通过！

结论分析与体会：

这次实验我们实践了简单的控制器，最让我印象深刻的是控制信号的设计，按键之后首先作为读控存信号，读出当前微指令，同时其反相作为寄存器打入脉冲，将微指令“显示”，而后其上升沿使 $UPC+1$ 形成下一条微指令的地址。控制单元的设计中节拍安排是一个

比较有意思的问题，希望在课设中深入理解。