



上海大学

SHANGHAI UNIVERSITY

2022-2023 学年夏季学期

课程报告

《计算机硬件综合大型作业报告》

小组序号: 1-9

项目名称: 可变模计数器的设计与实现

指导老师: 张云华

组员学号姓名:

组长: 黎楚凡 21121018

组员 陈楠 21120992

罗鑫洋 21120984

杨佳辉 21121011

吴穹 21120973

计算机工程与科学学院

报告日期 2023 年 6 月 30 日

# 计算机硬件大型作业报告

## ——可变模计数器的设计与实现

### 一、 实践项目意义

我们的实践项目主要目标是设计并实现一个可变模计数器。这不仅是一个传统的计数器，能够顺序地对输入的数字信号进行计数，而且需要实现可以灵活地设置多种模式，以满足不同类型的计数需求。实现可变模计数器具有生产生活实践上的重要意义，体现在：

1.该计数器能够任意设定不同的模进行计数。这意味着，无论我们需要以何种方式进行计数，这个计数器都能够进行调整以满足需求。这在实际应用中，比如生产线的数量统计，考试倒计时等，有着广泛的应用前景。

2.该计数器具有暂停、继续和倒计时的功能，使得用户可以更加灵活地控制计数过程，这在某些特定的情况下，例如监控时间敏感的事件或者需要进行定时操作的情况下，极具价值。

3.该计数器可以实现多种常见的计数，包括 8、10、12、16、24、30、365、1024 等。这意味着该计数器可以适应多种不同的常见计数需求，无论是日常生活中的时间计数，还是科研实验中的数据统计，都能够准确、快速地完成任

务。对我们小组而言，这个项目的重要性不仅仅体现在它的实用性，更重要的是，这个项目将使我们深入理解数字电路的设计和实现过程。我们通过设计并实现这个计数器，锻炼我们的硬件操作技能，提高我们的电路设计综合能力。在实践中，我们可以理解和掌握数字电路的基本原理，体验到从设计到实现的整个过程，这对我们的专业技能和综合素质的提升都有着重要的意义。

### 二、 实践项目原理

本项目的核心原理主要涉及计数器的设计（包括倒计时、暂停等附加功能），进制之间的转换，以及输入和输出的控制。为了实现我们的设计，我们将采用 QuartusII 软件来绘制逻辑电路图。

#### 1、预备知识

##### 1.1 计数器工作原理

计数器是一种常用的数字电路，其基本功能是对输入信号进行计数，可以用来记录事件的发生次数。其基本组成部分是触发器，本项目中，我们将采用 74LS192 芯片，这是一个可预置的四位二进制计数器，它能够进行正向计数和反向计数，非常适合于我们的计数需求。

##### 1.2 8421BCD 码

8421BCD 码是一种用于表示十进制数字的二进制编码系统。它通过使用四位二进制数来表示一个十进制数的每一位数值，其中每一位的权重分别为 8、4、2 和 1。这种编码系统的主要目的是以一种可靠而简单的方式将十进制数字转换为二进制形式。实验箱的数码管接受 8421BCD 码。

##### 1.3 QuartusII

Quartus II 是 Altera 公司的综合性 PLD/FPGA 开发软件，支持原理图、VHDL、VerilogHDL 以及 AHDL (Altera Hardware Description Language) 等多种设计输入形式，内嵌有综合器和

仿真器，可以完成从设计输入到硬件配置的完整 PLD 设计流程。

Quartus II 可以在 XP、Linux 以及 Unix 上使用，提供了完善的用户图形界面设计方式。具有运行速度快，界面统一，功能集中，易学易用等特点。

Quartus II 支持 Altera 的 IP 核，包含了 LPM/MegaFunction 宏功能模块库，使用户可以充分利用成熟的模块，简化了设计的复杂性，加快了设计速度。对第三方 EDA 工具的良好支持也使用户可以在设计流程的各个阶段使用熟悉的第三方 EDA 工具。

此外，Quartus II 通过和 DSP Builder 工具与 Matlab/Simulink 相结合，可以方便地实现各种 DSP 应用系统；支持 Altera 的片上可编程系统 (SOPC) 开发，集系统级设计、嵌入式软件开发、可编程逻辑设计于一体，是一款综合性的开发平台。

用 Quartus II 开发可编程逻辑器件的开发流程可分为四个步骤，它们分别是功能模块设计、项目编译处理、项目分析校验和器件下载编程。

## 2、步骤简述

### 2.1 功能模块设计

Quartus II 的功能模块设计可以通过集成环境的各种设计工具来完成。Quartus II 具有多种设计输入方法，其中包括：原理图输入和符号编辑、硬件描述语言输入、波形设计输入、平面布局编辑、层次设计输入。

Quartus II 集成环境使信息可在各种应用程序间交流，设计者可以在一个工程内直接从某个设计文件切换到其他设计文件，而不必理会它是图形格式、文本格式，还是波形格式。

### 2.2 项目编译处理

Quartus II 处理一个设计项目时由软件编译器读取设计文件，产生用于器件编程、仿真和定时分析的输出文件。由消息处理器产生在编译过程中发现的错误或警告。

### 2.3 项目分析校验

Quartus II 提供的设计校验包括设计仿真和定时分析，以此来测试设计项目的逻辑操作和内部时序。其中设计仿真又分为功能仿真、时序仿真和多器件仿真三种。

### 2.4 器件下载编程

Quartus II 编程器用来生成多种格式的编程文件。对于不同器件所生成的下载编程/配置文件是不同的。由于我们使用的是 Altera 器件，所以我们首先生成针对 Altera 器件的下载编程/配置文件，然后将生成的下载编程/配置文件下载到 Altera 器件中并进行测试，观察其实际输出的结果与时序仿真输出的波形结果是否一致，以验证实际逻辑电路是否正常工作。

## 3、芯片选择

### 3.1 74LS138

74LS138 是一个 3 线-8 线译码器/多路复用器，用于地址解码

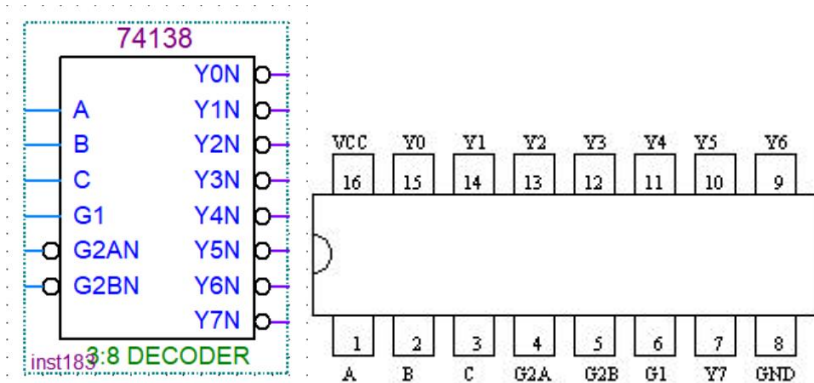


图 1 74LS138

G1	G*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	1	X	X	X	1	1	1	1	1	1	1	1
0	X	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0

表格 1

$$G^*=G2A+G2B;$$

### 3.2 74LS194

74LS194 是一个 4 位双向移位寄存器，用于数据的输入和存储。

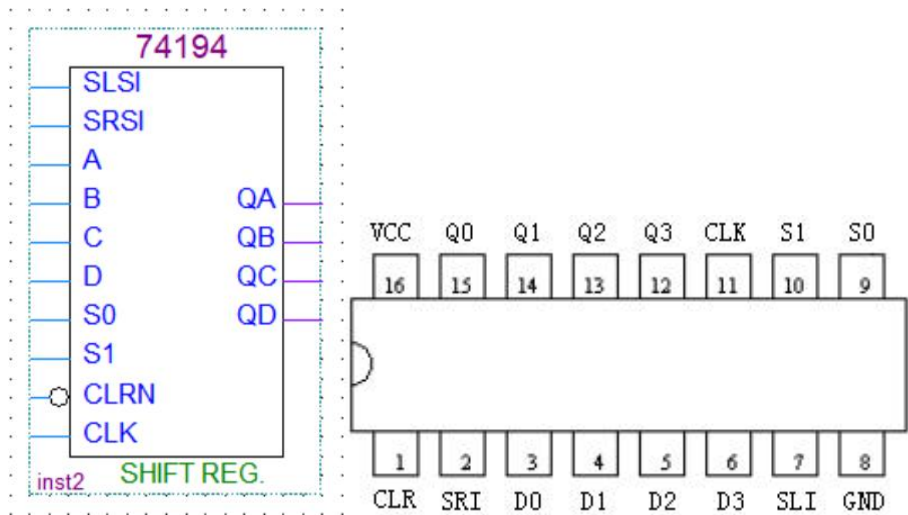


图 2 74LS194

输入										输出				功能
/CR	M1	M0	CP	DSL	DSR	D0	D1	D2	D3	Q0	Q1	Q2	Q3	
0	×	×	×	×	×	×	×	×	×	0	0	0	0	清零
1	×	×	0	×	×	×	×	×	×	Q00	Q10	Q20	Q30	保持
1	1	1	↑	×	×	D0	D1	D2	D3	D0	D1	D2	D3	置数
1	0	1	↑	×	1	×	×	×	×	1	Q0n	Q1n	Q2n	右移
1	0	1	↑	×	0	×	×	×	×	0	Q0n	Q1n	Q2n	右移
1	1	0	↑	1	×	×	×	×	×	Q1n	Q2	Q3n	1	左移
1	1	0	↑	0	×	×	×	×	×	Q1n	Q2n	Q3n	0	左移
1	0	0	×	×	×	×	×	×	×	Q00	Q10	Q20	Q30	保持

74LS194 功能表



表格 2

3.3 74LS192

74LS192 是一个可预置的四位二进制计数器，用于实现计数功能。它是同步十进制可逆计数器，是一款高速 CMOS 四位同步计数器芯片，也被称为 BCD decade up/down counter。它可以实现二进制加法和减法操作，并且输出结果可以采用 BCD 码表示。下面我们详细介绍 74LS192 的引脚图及功能表。

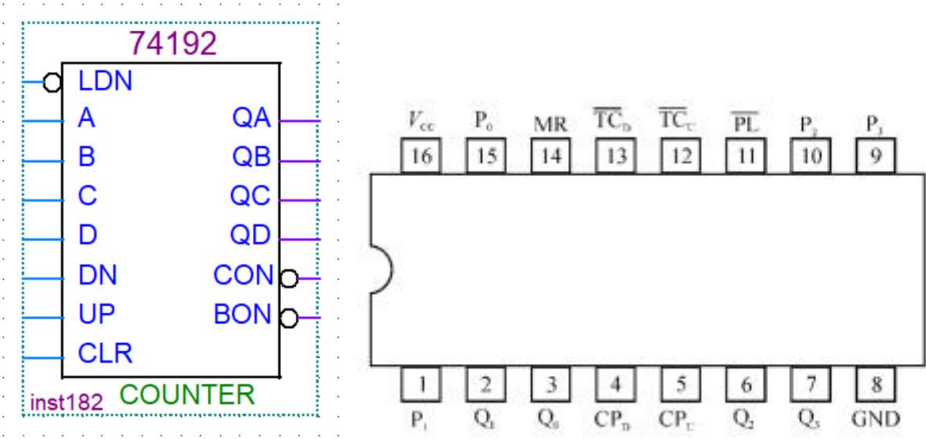


图 3 74LS192

序号	名称	功能
1	LDN	异步预置输入端，低电平有效
2	A	并行数据输入端
3	B	并行数据输入端
4	C	并行数据输入端
5	D	并行数据输入端
6	DN	异步清零输入端，低电平有效
7	UP	加计数时钟输入端
8	CLR	异步清零输入端，低电平有效
9	QA	十进制输出端 A
10	QB	十进制输出端 B
11	QC	十进制输出端 C
12	QD	十进制输出端 D
13	CON/BON	进位/借位输出端，高电平有效

表格 3

输入								输出			
MR	$\overline{PL}$	$CP_{\overline{D}}$	$CP_D$	P3	P2	P1	P0	Q3	Q2	Q1	Q0
1	X	X	X	X	X	X	X	0	0	0	0
0	0	X	X	d	c	b	a	d	c	b	a
0	1	↑	1	X	X	X	X	加计数			
0	1	1	↑	X	X	X	X	减计数			

表格 4

### 3.4 74LS85

74LS85 是一个 4 位全比较器，用于比较数据。

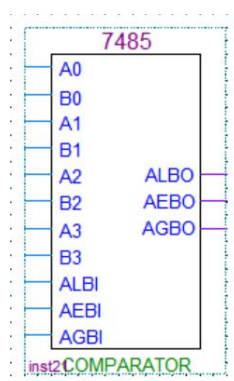


图 4 7485

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A <sub>3</sub> B <sub>3</sub>	A <sub>2</sub> B <sub>2</sub>	A <sub>1</sub> B <sub>1</sub>	A <sub>0</sub> B <sub>0</sub>	I <sub>A&gt;B</sub>	I <sub>A&lt;B</sub>	I <sub>A=B</sub>	O <sub>A&gt;B</sub>	O <sub>A&lt;B</sub>	O <sub>A=B</sub>
A <sub>3</sub> >B <sub>3</sub>	X	X	X	X	X	X	H	L	L
A <sub>3</sub> <B <sub>3</sub>	X	X	X	X	X	X	L	H	L
A <sub>3</sub> =B <sub>3</sub>	A <sub>2</sub> >B <sub>2</sub>	X	X	X	X	X	H	L	L
A <sub>3</sub> =B <sub>3</sub>	A <sub>2</sub> <B <sub>2</sub>	X	X	X	X	X	L	H	L
A <sub>3</sub> =B <sub>3</sub>	A <sub>2</sub> =B <sub>2</sub>	A <sub>1</sub> >B <sub>1</sub>	X	X	X	X	H	L	L
A <sub>3</sub> =B <sub>3</sub>	A <sub>2</sub> =B <sub>2</sub>	A <sub>1</sub> <B <sub>1</sub>	X	X	X	X	L	H	L
A <sub>3</sub> =B <sub>3</sub>	A <sub>2</sub> =B <sub>2</sub>	A <sub>1</sub> =B <sub>1</sub>	A <sub>0</sub> >B <sub>0</sub>	X	X	X	H	L	L
A <sub>3</sub> =B <sub>3</sub>	A <sub>2</sub> =B <sub>2</sub>	A <sub>1</sub> =B <sub>1</sub>	A <sub>0</sub> <B <sub>0</sub>	X	X	X	L	H	L
A <sub>3</sub> =B <sub>3</sub>	A <sub>2</sub> =B <sub>2</sub>	A <sub>1</sub> =B <sub>1</sub>	A <sub>0</sub> =B <sub>0</sub>	H	L	L	H	L	L
A <sub>3</sub> =B <sub>3</sub>	A <sub>2</sub> =B <sub>2</sub>	A <sub>1</sub> =B <sub>1</sub>	A <sub>0</sub> =B <sub>0</sub>	L	H	L	L	H	L
A <sub>3</sub> =B <sub>3</sub>	A <sub>2</sub> =B <sub>2</sub>	A <sub>1</sub> =B <sub>1</sub>	A <sub>0</sub> =B <sub>0</sub>	X	X	H	L	L	H
A <sub>3</sub> =B <sub>3</sub>	A <sub>2</sub> =B <sub>2</sub>	A <sub>1</sub> =B <sub>1</sub>	A <sub>0</sub> =B <sub>0</sub>	H	H	L	L	L	L
A <sub>3</sub> =B <sub>3</sub>	A <sub>2</sub> =B <sub>2</sub>	A <sub>1</sub> =B <sub>1</sub>	A <sub>0</sub> =B <sub>0</sub>	L	L	L	H	H	L

表格 5

## 三、实践项目设计

### 1、预研报告

#### 1.1 项目背景和目标

随着计算机技术的发展, 各种各样的计数器在各个领域发挥着重要的作用。在电路设计、控制系统、通信技术等多个领域, 我们经常会遇到需要进行特定模式计数的需求。为应对不同模的需求, 我们此次硬件大作业的目标是设计并实现一个可变模计数器。这个计数器的主要功能有:

- ① 能够任意设定不同的模实现计数的功能;
- ② 能够暂停、继续、倒计时;
- ③ 能够完成常见的 8、10、12、16、24、30、365、1024 的计数 (独立模式)。

我们希望通过此次设计项目, 深入理解数字电路的设计原理, 提高硬件设计能力, 为未来在计算机领域的进一步研究和工作打下坚实的基础。

## 1.2 技术预研

在我们的可变模计数器设计中, 我们将应用多项技术和方法。以下是主要的技术预研内容:

**数字电路设计:** 计数器是一种典型的数字电路系统, 因此, 我们需要对数字电路的基本理论和设计方法有深入的了解。包括基本的逻辑门电路、触发器、编码/译码器等元件和模块的工作原理和设计方法。

**Quartus II 软件:** 我们将使用 Quartus II 软件进行电路设计。Quartus II 是一款功能强大的 FPGA 设计软件, 能够提供全套的设计方法, 从设计输入、编译、仿真到下载实现等。我们需要了解 Quartus II 的基本操作, 以及如何用它来进行电路设计和仿真。

**集成电路 (IC):** 我们将使用多种集成电路 (IC) 来实现计数器的功能, 包括 74LS138、74LS194、7485 和 74LS192 等。我们需要了解这些 IC 的工作原理和使用方法, 以及如何将它们集成到我们的设计中。

## 1.3 可行性分析

**技术可行性:** 根据我们的技术预研, 我们需要的所有技术和方法都是已经成熟和广泛应用的, 包括数字电路设计、Quartus II 软件以及各种集成电路等。我们的团队成员也有足够的知识和技能来应用这些技术和方法。

**资源可行性:** 我们需要的资源主要包括 Quartus II 软件以及各种集成电路。我们通过老师上传到学习通平台的安装包, 根据说明文档顺利完成了软件的安装, 并到学校实验室将设计好的电路下载到实验箱上, 完成调试和改进工作。

综上所述, 我们有足够的技术、资源和时间来完成这个可变模计数器的设计项目。

## 1.4 实施计划

项目的实施将分为以下几个步骤:



图 5 实施流程图

**预研:** 对项目需求进行深入研究, 选择合适的设计方案和技术。

**设计:** 使用 Quartus II 进行电路设计, 完成计数器模块、控制模块、输入输出模块以及辅助功能模块的设计。

**仿真:** 使用 Quartus II 的仿真功能, 对设计进行仿真测试, 生成相应的波形图, 验证设计的正确性。

**上机调试:** 将电路下载到实验箱上, 进行实际的测试和调试。

**优化:** 根据测试和调试的结果, 对设计进行优化和改进。

**报告撰写:** 整理并撰写项目报告, 包括设计过程、实施过程、测试结果和经验教训等。



### 1.5 预期结果

完成该项目后，我们预期得到一个功能完整的可变模计数器，能实现上述所列的功能，并在此基础上针对可能出现的问题进行优化。此外，我们希望通过这个项目，提升我们的电路设计能力与硬件水平，加深对数字电路基础知识的理解，提高解决实际问题的能力。

## 2、概要设计说明书

### 2.1 设计目标

在本次硬件设计项目中，我们以学习和理解数字电路设计为核心，通过设计和实现具备多种模式计数、正倒计时及暂停继续功能的可变模数计数器，熟练应用 Quartus II 电路设计软件和各类集成电路，实现硬件设计，最后通过撰写项目报告来提高我们的实践能力、理论知识以及技术文档撰写能力。

### 2.2 计数器概述

计数器系统是一个集成了计数、输入输出处理以及辅助功能等多个模块的复杂系统，旨在实现可变模数计数器的设计和操作。

计数器模块：这个模块负责实现计数功能。

控制模块：这个模块负责对整个系统的运行进行控制，包括计数模式（如正向计数、反向计数）、计数速度、开始/停止计数等。

辅助功能模块：提供了输入信号的切换、信号预处理、错误处理等辅助功能，使得计数器系统更加灵活和易用。

输入/输出模块：这个模块负责处理与外部设备或用户的交互，包括输入设备和输出设备。

这些模块通过内部接口相互连接，共同组成一个功能完善的计数器系统。在操作过程中，用户首先通过交互模块设定计数模式，然后通过控制模块启动计数任务。计数模块开始工作，计数结果经过处理后通过数码管显示，用户可以随时通过交互模块查看当前的计数值，或者通过辅助功能模块进行暂停、继续以及倒计时操作。

### 2.3 芯片选择

(1) 74LS192：是一种同步十进制可逆计数器，它具有双时钟输入，并具有清除和置数等功能<sup>1</sup>。在我们的设计中，74192 被用于实现计数器模块的核心部分。

(2) 74LS194：是一种四位双向移位寄存器。在我们的设计中，74LS194 被用于实现输入输出模块的部分功能以及数据的暂存。

(3) 7485：7485 是一个四位比较器，可以比较两个四位二进制数的大小。在我们的设计中，7485 被用于判断当前的计数值与设定值的大小关系从而控制计数器的正向计数，以及纠错电路。

(4) 74LS138：74138 是一个 3 线-8 线译码器，可以根据 3 个输入信号选择 8 个输出线中的一个。在我们的设计中，74138 被用于地址译码，将用户的操作转化为对应的控制信号，从而实现模式切换、控制输入位等功能。

这些集成电路的选择和配置都是基于我们的设计需求，我们将它们有机地组合在一起，形成一个功能强大、稳定可靠的可变模数计数器。

### 2.4 功能需求

(1) 选择模式：我们的计数器允许使用者选择不同的计数模式。这包括自定义模式和



独立模式。当输入端  $d=1$  时即为自定义模式，用户可以任意设置计数器的模，当前位置数值设置完毕后，按 **enter** 跳到下一位，然后开始计数。当输入端  $d=0$  时即为独立模式，通过设定  $cba$  的值选择预设的模式进行计数，这些模式包括 8, 10, 12, 16, 24, 30, 365, 1024 的计数。

(2) 暂停和继续：我们的计数器允许用户在计数过程中通过改变  $pause$  的值实现对计数的暂停与继续。

(3) 正/倒计时模式：我们的计数器允许用户选择正序或倒序计数的不同模式。使用者可以通过改变  $U\_D\_switch$  的输入值来选择不同的计数方式。

### 3、详细设计报告

#### 3.1 系统结构

我们的设计中包括四个主要部分：计数器模块、控制模块、辅助功能模块以及输入/输出模块。

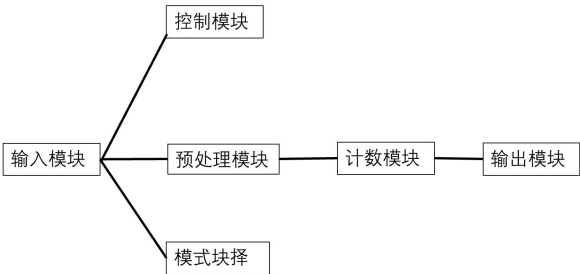


图 6 模块结构

计数模块：计数器的核心模块，负责处理所有的计数任务，包括正计时、倒计时等。通过设定不同的模式，可以实现 8、10、12、16、24、30、365、1024 等多种计数以及自定义模计数。

控制模块：负责协调和管理其他模块的工作，控制计数器实现不同工作模式的转换以及暂停、继续等功能的变化，确保整个系统能够协同工作，满足用户的需求。

辅助功能模块：提供了输入信号的切换、溢出处理等辅助功能，使得计数器系统更加灵活和易用。

输入输出模块：负责处理所有的输入和输出任务。该模块的任务包括接收用户的输入，转换成系统可以处理的信号，以及将系统的状态和计数结果转换成用户可以理解的输出。用户可以通过六位数码管与系统进行交互，其中两位选择模式，四位显示当前状态的计数值。

这些模块通过内部接口相互连接，共同组成一个功能完善的计数器系统。

#### 3.2 模块设计

(1) 计数器模块：计数器模块的核心组成部分是 74LS192。74LS192 是一个 BCD（二进制编码十进制）计数器，它可以对输入的电信号进行向上、向下计数，并且可以通过输入进行预设。对于正计时功能，我们将 74LS192 配置为向上计数模式。对于倒计时功能，我们将 74LS192 配置为向下计数模式。通过控制计数器的  $CLEAR$  和  $LOAD$  引脚，我们可以随时清除当前的计数值或加载新的预置值。

我们使用 74LS192 作为主计数器， $UP$ 、 $DOWN$  都接时钟信号， $CON$  用于进位， $BON$

用于退位。对于正计数、倒计数，不用的位置设为 1，对应需要使用的位置设置上升沿信号。开始计数时，提供一个时钟信号。正计数模式下，每当时钟信号的边缘从低到高时，计数器的值就会加 1；倒计数模式下，每当时钟信号的边缘从低到高时，计数器的值就会减 1 直到计数器的值达到 0。

比较器 7485 则用来保存输入的模，实现 clear 清零功能。74LS194 用作寄存器，在逐位输入数值时，能够保存上一位的内容，74LS194 送出的数值直接连接到 7485 比较器，并且减 1 之后连接到 74192 为倒计数的重置做好准备。倒计数模式下有专门的“减一电路”，以确保倒计数正确循环，避免上升探测值冲突，造成死循环。

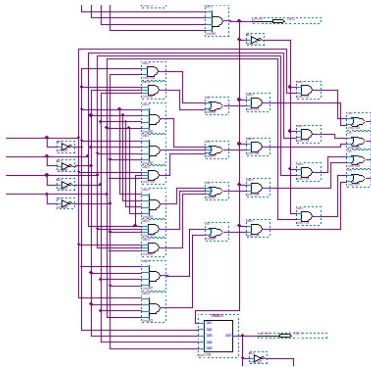


图 7 减一电路（部分）

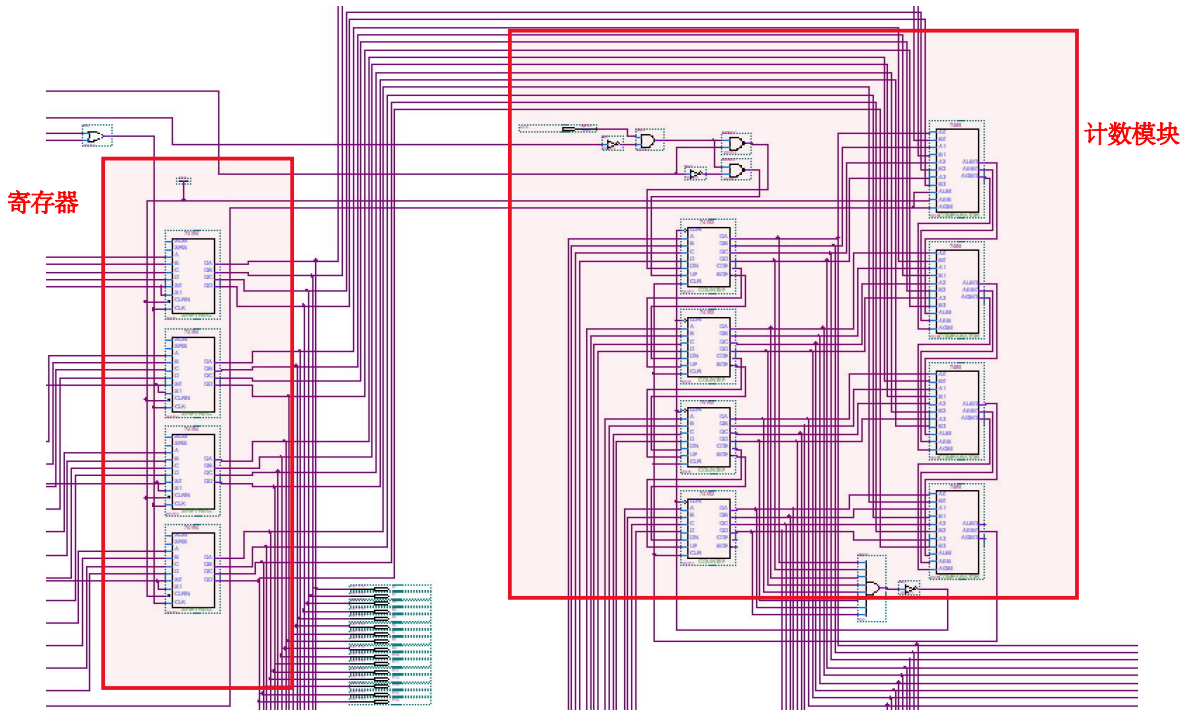


图 8 计数模块

(2) 控制模块：在接收到恰当的信号时，通过改变 pause 的值能够实现对计数的暂停与继续，pause 与时钟信号 CLK 的值相与决定停下或继续计数。

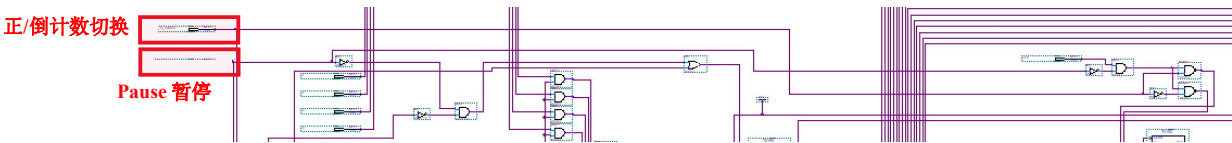


图 9 控制模块（暂停功能）

芯片 74LS138 作为控制模块的一部分，用于选择或控制其他组件，如选择特定的模式

或输出。选择电路中数值在 0~3 间循环，帮助定位正在输入数值的位置。如从低至高四位分别对应 0, 1, 2, 3。同时，74LS138 还用于寄存器屏蔽信号，当计数器在指定状态工作时，产生固定的屏蔽信号，确保一种状态下只有 1 位数字发生变化。

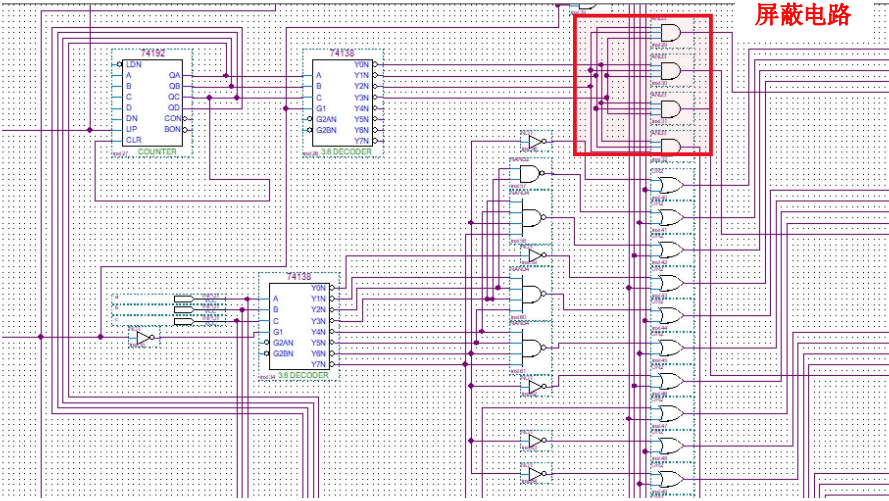


图 10 控制模块

(3) 辅助功能模块：实现溢出处理等辅助功能。为了避免由于输入错误而导致的错误问题，我们应用比较器 7485 专门针对可能出现的溢出问题设计了纠正电路。当输入的 BCD 码数值大于 9（十进制）时，纠正电路会自动将其转为 9，避免溢出对计数造成影响。

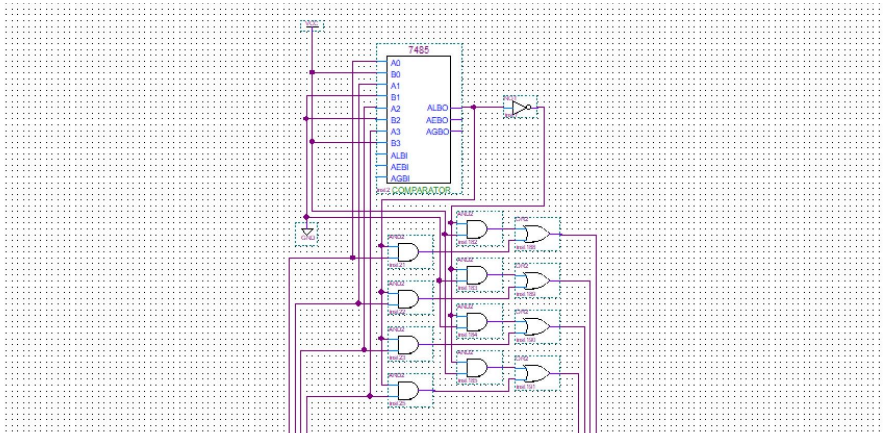


图 11 纠错电路

(4) 输入/输出模块：接受用户的输入（如计数模式和数值）并显示计数值。输入信号接入逻辑电路中，以便在改变输入时触发相应的系统动作。

控制逻辑：接收输入信号并控制数码管显示的电路。根据用户的输入调整计数模式，开始或停止计数，并更新数码管显示。

数码管显示：6 位数码管用来显示计数值和计数模式。2 位用来显示模式。例如，“0”代表正计数，“1”代表倒数计数，“2”代表暂停，“3”代表自定义模式等。在输入状态下，第一个数码管表示状态，第二个数码管显示 0~4，表示当前正在操作哪一个位置的数值，后面四位数码管随输入逐位确定。自定义模式下其余 4 位用来显示计数值。

### 3.3 接口设计

#### (1) 输入设备

计数模式和初始计数值可以通过一组物理开关进行输入。

计数模式输入：一组二进制开关可以用于选择计数模式，直接连接到计数器的控制逻辑。

## (2) 数码管显示

计数器的状态和计数值可以通过一个六位数码管进行显示。

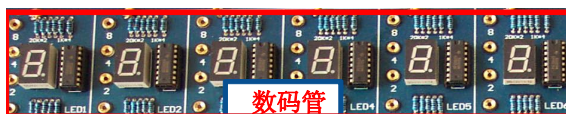


图 12 实验箱数码管

计数模式显示：数码管的前两位可以用来显示当前的计数模式。

计数值显示：数码管的后四位用来显示当前的计数值。数码管接收 8421BCD 码可直接以十进制数显示出来。

在这种设计中, 计数器的状态和值可以直接从数码管上读取, 无需任何其他的用户接口。此外, 这种设计也使得用户可以在任何时候改变计数模式, 从而提供了灵活的用户交互。

## 3.4 计数器的优化

在设计过程中, 我们对可变模计数器进行了三次优化:

### (1) 自定义模式优化

自定义模式在要求的 8 种独立模式的基础上提供了更多的灵活性, 可以通过添加更多的控制开关以及逻辑部件实现更多的模式, 最终使我们的计数器能够完成任意模的计数任务。

在倒计数模式下, 当数码管数值从 0 跳到  $n-1$  时, 0 的显示周期仅为其他的一半。针对这个问题, 我们修改了原本的以 74LS192 上 bon 输出端作为判断的思路, 直接在四位输出端检测 1001 (9) 来判断是否重新置数为  $n-1$ , 顺利解决了显示周期不同步的问题, 最终使得计数器所有数字显示时间相同。

### (2) 输出进制转换优化

我们由最初构想的十六进制修改为直接以十进制显示计数数值。最初我们计划使用 74LS193 芯片实现正、倒计数功能, 初步设计完成后, 结合实验箱输出的要求, 我们决定更换使用 74LS192 直接实现十进制计数功能, 并搭建相应的“减一电路”。由于计数模块的芯片更换, 其他部分也依此作出较大改动。

### (3) 显示优化

通过多次修改逻辑电路图, 我们对可变模计数器的功能操作界面进行了优化, 使 6 位数码管对应显示不同的内容。在不同状态下, 数码管能够切换显示相应的内容。



正计	0	0-8
倒计	1	0-8
暂停	2	0-8
模n输入	3	0-3

图 13 数码管功能示意图

总的来说, 这些优化可以使得计数器更加强大和灵活, 同时也更加易用。

## 4、仿真设计小结

本次课程大作业中我们一共设计了 4 个版本的计数器, 第一个版本我们先完成了基本符合课程要求的模 8 计数器, 因为只需要完成固定模的计数, 所以电路图的连线比较简单, 但是这个版本的计数器为我们接下来的结构设计打下基础。在之后的版本中我们依次增加了 8





如图所示，我们当前选择了独立模式，并输入了 010，即选择模式 3，也就是常见的模 12 计数。O\_5 显示"0000"，表示进行正数计数；O\_4 显示"0010"，即选择了模式 3 的模 12 计数；O\_0、O\_1、O\_2、O\_3 分别显示个位、十位、百位和千位。由于进行的是模 12 计数，所以百位和千位显示为 0，而个位显示 0-9。当个位为 9 时，下一个数会进位到十位并将个位归零。在十位为 1 时，个位依次计数 0-1，再计入 1 后，计数器归零，完成模 12 计数。

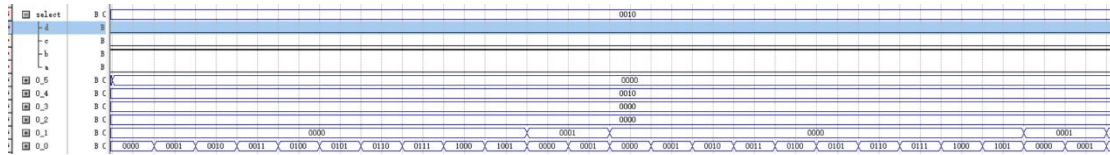


图 15 模 12 波形图

完成对输入信号 d 的测试后，我们对模式位进行了更加详尽的测试。

在暂停状态下，数码管 O\_5 显示为"0011"，表示进入模 n 输入状态。同时，手动连续输入多个 enter 脉冲信号。数码管 O\_4 会循环显示"0000""0001""0010""0011"，其中"0000"、"0001"、"0010"、"0011"分别代表输入 0、1、2、3。

当开始计数后，数码管 O\_5 显示为"0000"，表示进行正数计数。同时，数码管 O\_4 显示为"1000"，表明计数器正处于自定义模式。

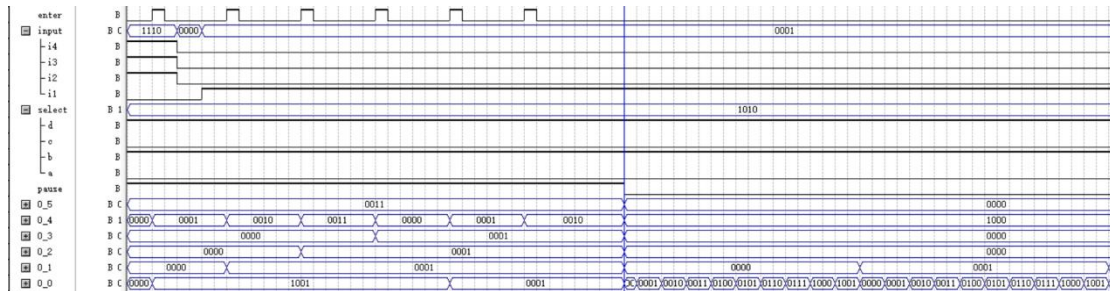


图 16 模 19 波形图

## 1.2. 暂停和继续

在暂停和继续计数的测试中，我们使用波形图观察控制信号 pause 的波形，以确保在暂停和继续计数时信号正确变化。

首先，我们将 pause 设置为 1，并观察计数器是否暂停，以及第一个数码管上显示的数值是否为计数值是否为"2"。

然后，我们将 pause 设置为 0，并观察计数器是否继续计数，以及数码管上的计数值是否按预期变化。

如图所示，当计数暂停时，O\_5 显示为"0010"；当继续计数时，O\_5 显示为"0000"。而且，在继续计数时，计数器从暂停计数的位置开始继续计数。



图 17 暂停功能仿真

## 1.3. 正/倒计数状态

在正/倒计数状态的测试中，我们使用波形图观察输入信号 U\_D\_switch 的波形，以确保在选择计数状态时信号正确变化。

首先，我们将 U\_D\_switch 设置为 0，并观察计数器是否进入正计数状态，以及数码管上的模式位是否正确显示为正计数状态（例如数码管上显示"0"）。

然后，我们将 U\_D\_switch 设置为 1，并观察计数器是否进入倒计数状态，以及数码管上的模式位是否正确显示为倒计数状态（例如数码管上显示"1"）。

Timing diagram for V\_B... showing signals 0\_5, 0\_4, 0\_3, 0\_2, 0\_1, and 0\_0 over time. The diagram shows various logic levels (0000, 0001, 0010, 0011, 1000, 1001, 1100, 1101, 1110, 1111) and transitions marked with 'X'.

图 18 倒数计数仿真

通过以上调试过程，我们逐步验证了硬件电路的功能，确保每个模块都按照预期工作。在发现问题的过程中，我们依靠波形观察、逻辑分析和逐步调试来定位和修复故障，以确保整个计数器系统正常运行。这些调试过程对于验证设计的正确性和可靠性起到了关键作用。

## 2、实验箱调试

当进行实践调试过程时，首先我们参照提前设计好的原理图来定义管脚，进行编译，连线时发现有些位置在实验箱上没有对应的引脚，于是根据 DICE-SEM II 实验箱与 EP1K10、EP1K30 引脚对照表，重新定义了管脚，按照指定的管脚连接到试验箱上面，生成编程下载文件。

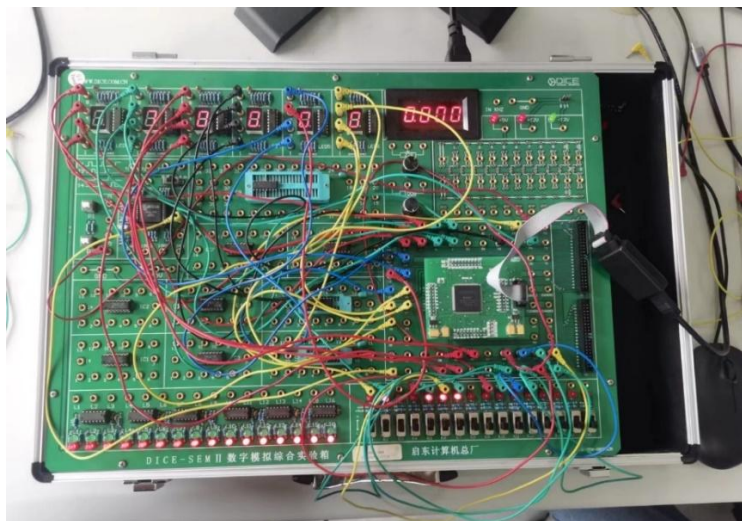


图 19 调试实验箱

第一次调试发现，所需的功能都没能够正常执行，检查发现没有给 CLK 输入，修改完重新编译之后，独立模式下的 8 10 12 16 24 30 365 1024 计数都没出现问题，暂停，继续和倒计时功能也可以正常执行，但是自定义模式下的输入不正确，4 位同时置位。

重新查看试验箱构造发现没有使用脉冲信号发生开关，而是使用下面 16 个开关中的一个进行模拟，猜测这些开关可能导致信号不稳定，没有办法短时间产生一个稳定的信号。修改后重新进行调试，不会出现所有位同时置位，但数码管显示的结果仍与预期不符，当进行数据输入时，每次输入到第三位时，第四位将会一起置位。

发现错误后，我们先去查看了波形仿真和原理图，并未发现错误，但是试验箱上依旧无法正确显示，后面仔细查找问题，发现芯片选择错误，立即对芯片进行了更正，然而问题仍未得到解决。在自定义模式下，手动输入数据到第三位时，第四位数码管上也会同时显示第三位输入的数字。

于是我们针对第四位数码管进行了检查,决定修改第四位数码管的引脚,重新进行接线,再次进行调试。这次,可以通过在第四位直接输入数字,从而手动去除,成功实现自定义模式下的输入,并且暂停,继续和倒计时功能也能正常执行,调试成功。

在调试过程中，我们犯的错误的都是一些非常基本的东西，比如 CLK 输入，芯片的选择等，而非电路出问题，说明我们在实验时需要更加认真仔细，注重微小的细节。调试过程中



遇到问题是正常的，相反，将这些问题看作是学习的机会，通过分析错误的原因和解决方案，从而增长自己的知识和技能。此外，记录每一步的改变和结果是非常重要的，这可以帮助我们回顾整个过程，并在需要时与团队或其他人进行有效的沟通和交流。当发现问题时，不要轻易放弃，而是持续迭代和改进，逐步查找问题的根源，并尝试不同的解决方案，直到找到有效的解决办法。

总之，调试过程是一个需要耐心和细心的过程。通过检查硬件连接、利用调试工具、持续迭代和改进等，并记录和沟通，能够逐步解决问题并实现预期的功能。记住从错误中学习，并将这些经验应用于未来的项目中，以提高自己的技能。

## 五、大型作业的心得与收获（个人）