

计算机组成原理实验

授课老师: 吴炜滨

大纲



- > 同步时序逻辑设计
 - 概述
 - 16位并行加载寄存器
 - 4位BCD计数器
 - 码表计数器

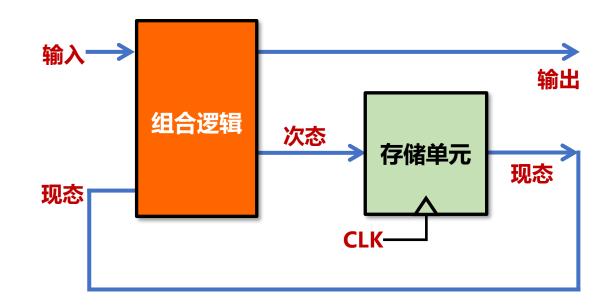
大纲



- ▶ 同步时序逻辑设计
 - 概述

同步时序逻辑电路概述

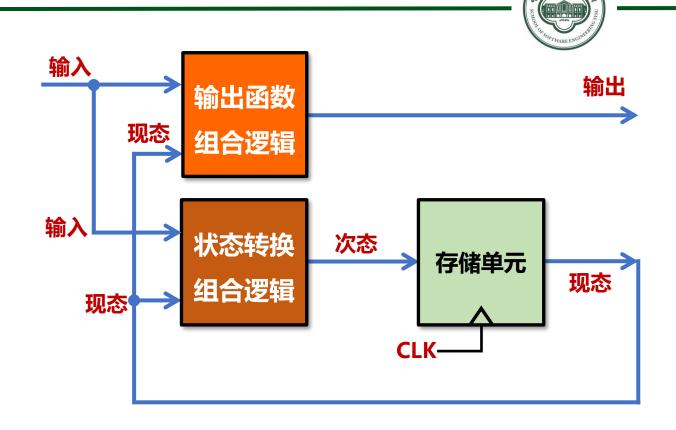
- 同步时序电路特征
 - 由组合逻辑和存储单元构成
 - 电路存在反馈回路
 - 公共时钟进行同步
- 通用的同步时序逻辑电路的模型
 - 存储单元输入是次态,输出为现态
 - 次态、输出由外部输入和现态决定



同步时序逻辑电路概述

■ 通用的同步时序逻辑电路的模型

- 将组合逻辑部分分解为输出函数和状态 转换两部分
- 状态转换部分: 现态和次态之间的转换
- 输出函数部分: 输出信号的产生
- 同步时序逻辑设计流程
 - 根据功能需求,构建状态图
 - 填写状态转换表(真值表),构建状态 转换组合逻辑
 - 构建输出函数组合逻辑 (真值表)
 - 连接存储单元和组合逻辑电路,实现最终电路



实验目的



■ 实验目标

- 理解同步时序逻辑电路设计的基本流程
- 熟练利用Logisim构建同步时序逻辑电路

■ 实验任务

- 构建16位并行加载寄存器 (存储数据)
- 设计4位BCD计数器 (单数码管计时)
- 实现码表计数器 (4数码管计时)

大纲



- ▶ 同步时序逻辑设计
 - 16位并行加载寄存器

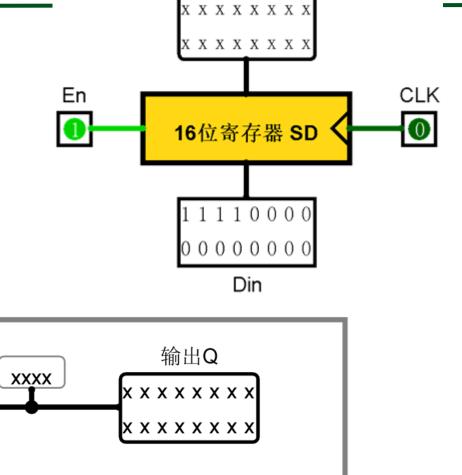
16位并行加载寄存器

■ 输入: 16位输入Din , 使能信号En , 时钟信号

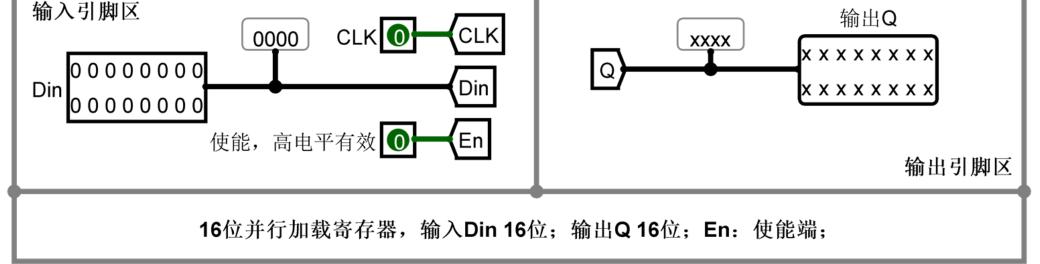
■ 输出: 16位输出Q

■ 功能: Q ← Din (遇到时钟上升沿时,将输入传播到输出)

■ 约束: 由4位寄存器模块构成



Q



4位并行加载寄存器

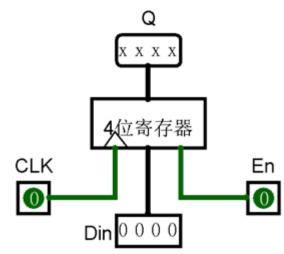
SCHOOL WAS DISCOURS OF THE PARTY OF THE PART

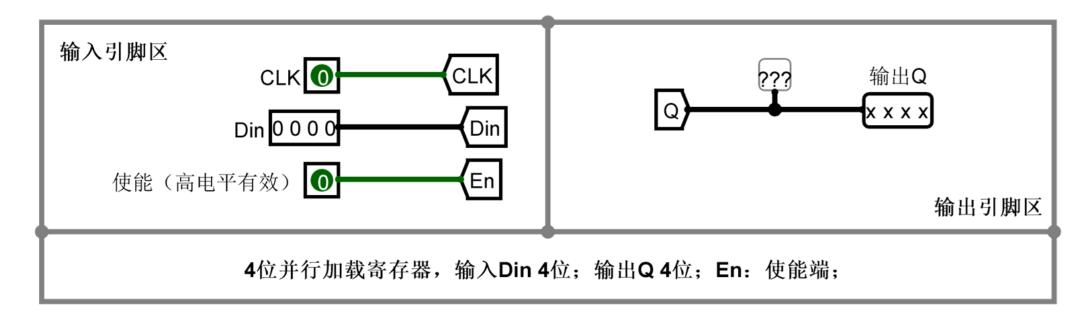
■ 输入: 4位输入Din, 使能信号En, 时钟信号

■ 输出: 4位输出Q

■ 功能: Q ← Din (遇到时钟上升沿时,将输入传播到输出)

■ 约束:使用D触发器构成

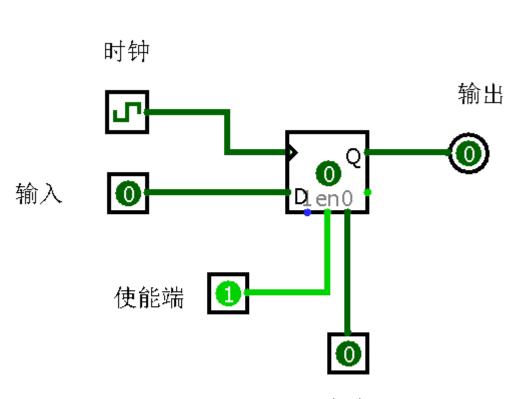




D触发器 (delay flip-flop)



- 使能端
 - 输入为1或不连接时触发器正常工作
- 异步清零
 - 输入为1时,输出为0
- ■功能
 - 触发时(时钟上升沿),输入传播到输出, 否则输出保持原来值不变



异步清零

4位并行加载寄存器



■ 4个D触发器并发 En Den0 Dien0 0 Den0 Den0

16位并行加载寄存器



■ 4个4位寄存器并发 4位寄存器 4位寄存器 4位寄存器 4位寄存器

大纲



- ▶ 同步时序逻辑设计
 - 4位BCD计数器

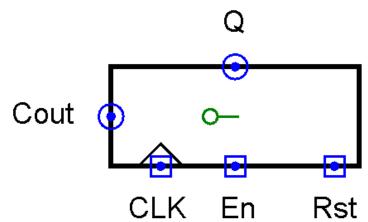


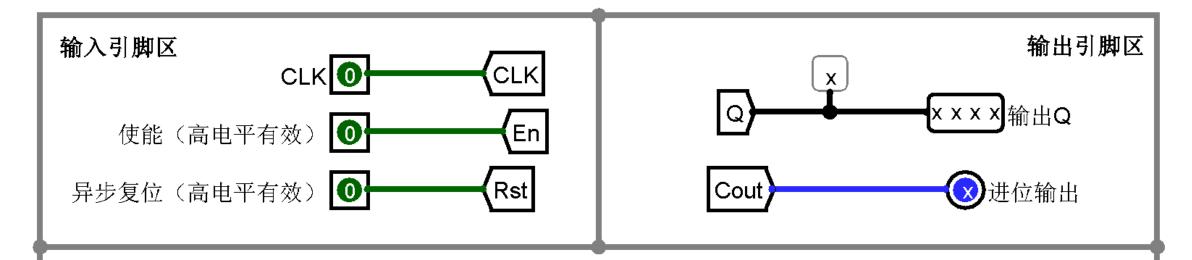
■ 输入: 时钟信号, 使能信号En, 异步复位Rst

■ 输出: 4位BCD计数器值Q, 进位输出信号Cout

■ 功能: Q ← (Q==9)? 0: Q+1

Cout=(Q==9)? 0: 1





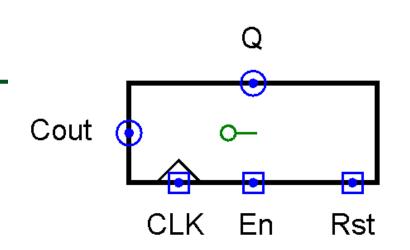
4位BCD计数器,输出Q在0-9之间循环计数,EN:使能端,Rst:异步复位,Cout=(Q==9)?0:1;

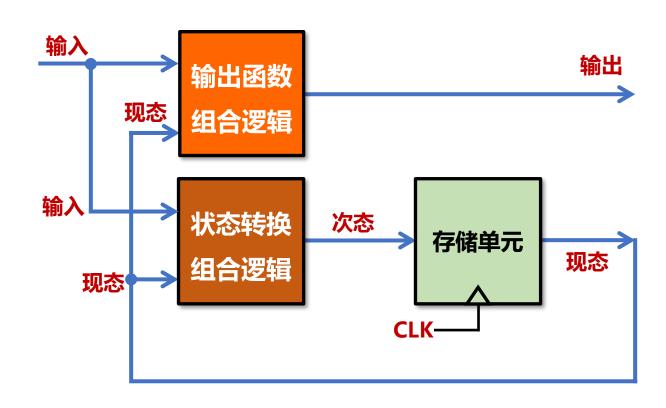
家件工 SOUTH ARREST OF THE ARRE

- BCD码
 - 用4位二进制代码表示一位十进制数
 - 1010-1111这6种代码是无效的

十进制数	BCD码			
0	0000			
1	0001			
2	0010			
3	0011			
4	0100			
5	0101			
6	0110			
7	0111			
8	1000			
9	1001			

- 不需要外部输入
- 使能、复位、时钟端
 - 使用D触发器的使能、复位、时钟端
- 实现计数功能
 - BCD码 (4位二进制数) 来表示一个 十进制数
 - 4位二进制数 → 4个状态位 → 4个D 触发器
 - 4位BCD计数器值Q: 现态的值
 - 进位输出信号Cout: 输出值





BCD计数器状态转换



■ 状态转换逻辑

• 4位计数器值Q ← (Q==9)?0:Q+1

• $0 \rightarrow 1 \rightarrow \cdots \rightarrow 9 \rightarrow 0 \rightarrow \cdots$

■ 定义真值表

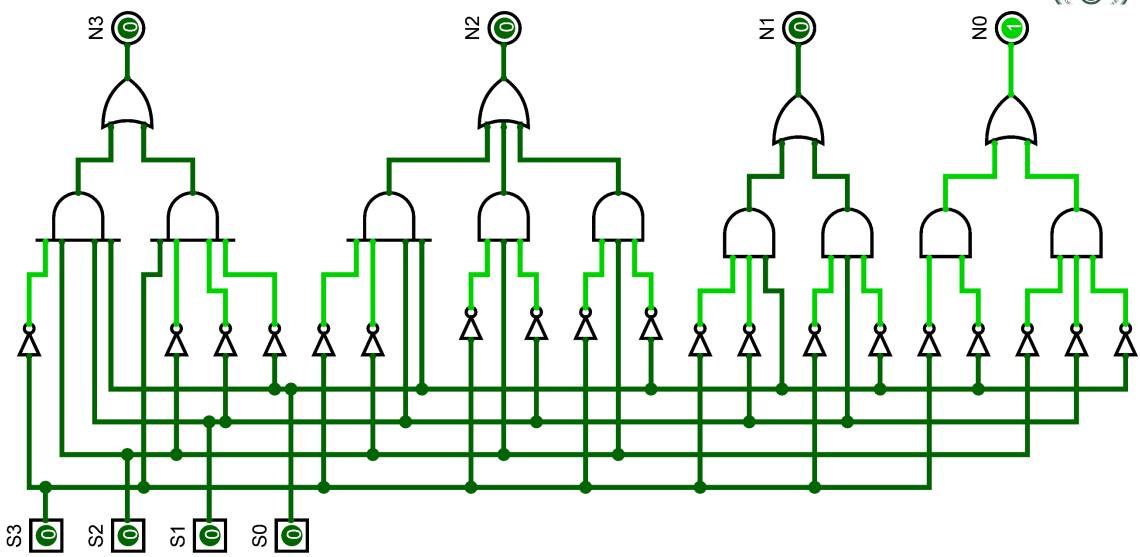
• 输入: 现态 S₃S₂S₁S₀

• 输出: 次态N₃N₂N₁N₀

S 3	S2	S1	SO	N3	N2	NI	NO
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	0

BCD计数器状态转换





BCD计数器输出函数



■ 输出函数

• 进位输出信号Cout=(Q==9)?0:1

■ 定义真值表

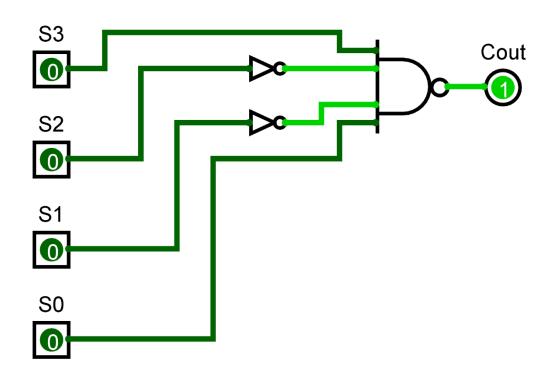
• 输入: 现态 S₃S₂S₁S₀

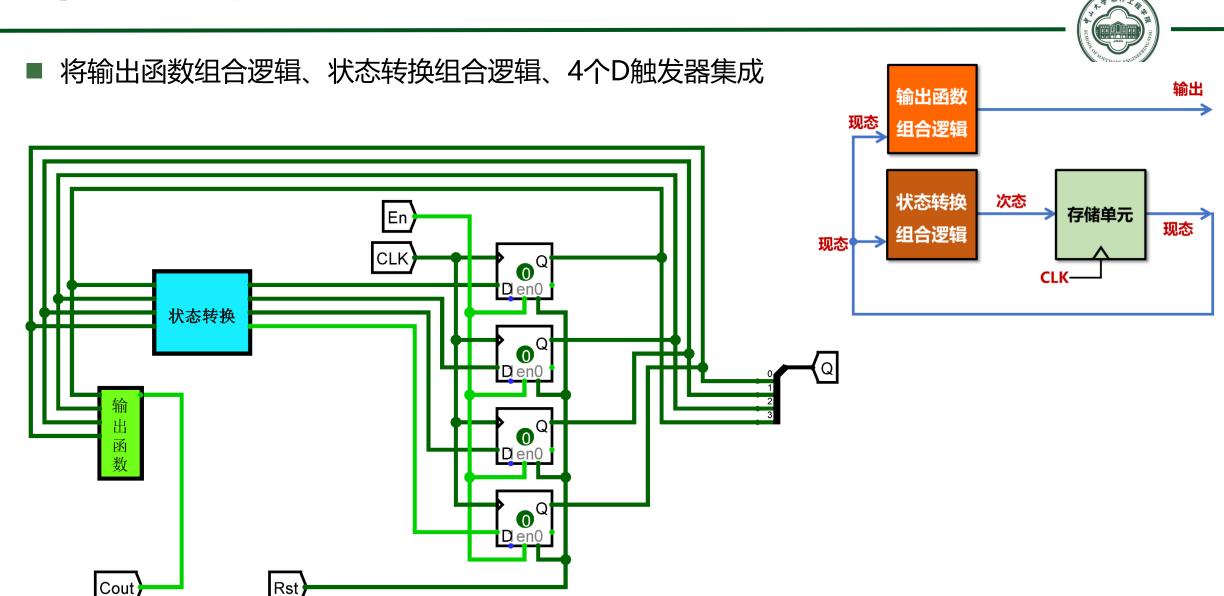
• 输出: 进位输出信号Cout

S3	S2	S1	SO	Cout
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

BCD计数器输出函数



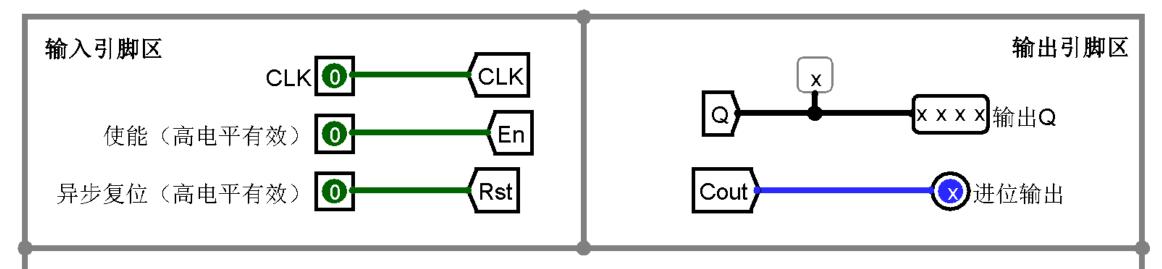






■功能测试

• 使能端为1,复位端为0,不断点击时钟输入端,观察能否正常计数



4位BCD计数器,输出Q在0-9之间循环计数,EN:使能端;Rst:异步复位;Cout=(Q==9)?0:1;

大纲



- ▶ 同步时序逻辑设计
 - 码表计数器

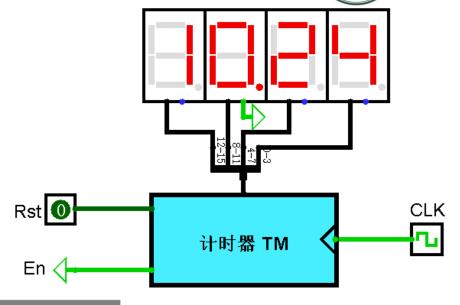
码表计数器

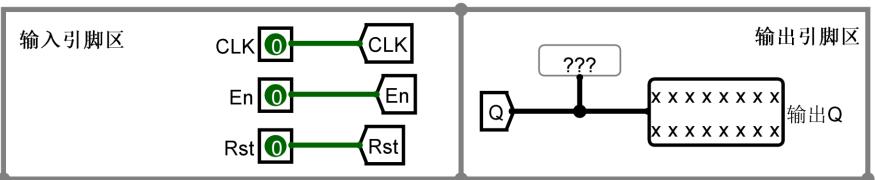
■ 输入: 时钟信号CLK, 使能En, 异步复位Rst

■ 输出: 16位计数器值Q

■ 设计约束:利用BCD计数器级联而成

■ 功能: 低位计数器从9到0时, 高位计数器加1



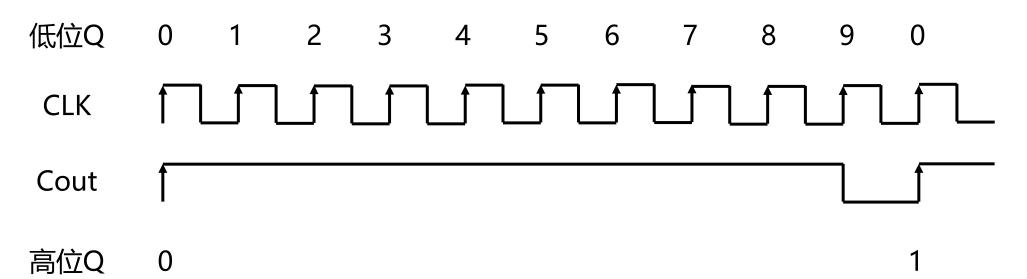


码表计数器,16位输出,包括4个BCD计数器,分别对应10秒、1秒、1/10秒、1/100秒。低位计数到9时,相邻高位在时钟到来时加1。 En:使能端;Rst:异步复位;

Ctrl+K驱动时钟自动测试

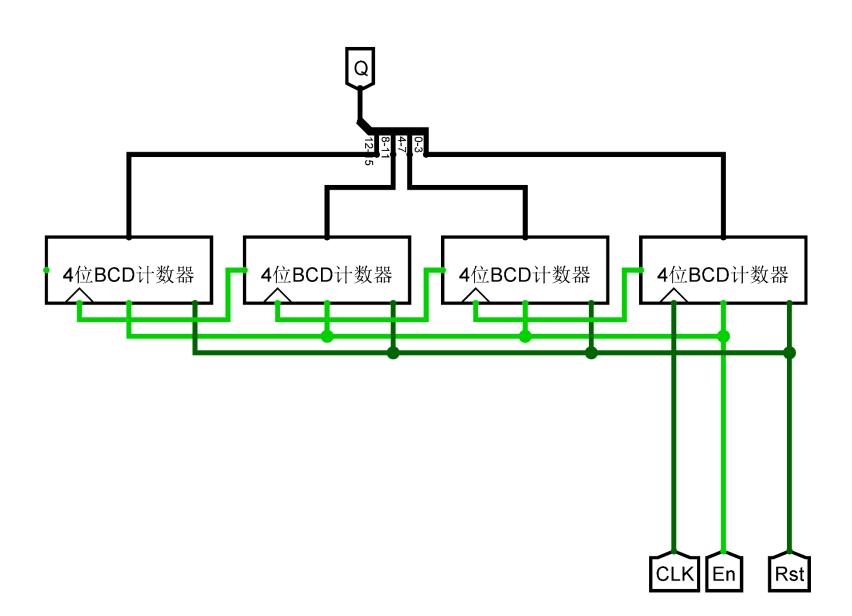
码表计数器

- 功能: 低位计数器从9到0时, 高位计数器加1
 - 将低位BCD计数器的进位输出信号Cout作为相邻高位BCD计数器的时钟输入
 - 上升沿触发,低位BCD计数器从9到0时
 - 高位计数器加1
 - 进位输出信号Cout产生上升沿: 0 → 1
 - Cout=(Q==9)?0:1



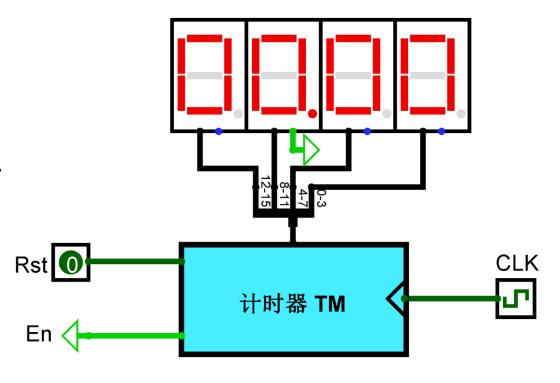
码表计数器





码表计数器自动测试

- Ctrl+K驱动时钟自动测试
- 问题:仿真的时钟频率没有100Hz这个选项
 - 将时钟频率设为1KHz
 - 增加分频电路
 - 增加一个4位BCD计数器作为千分之一秒
 - 将干分之一秒计数器的进位输出作为百分之一秒计数器的时钟输入



Ctrl+K驱动时钟自动测试



谢谢!