

当前位置： 首页 > NEWS >正文

北航计算机学院-计算机组成原理课程设计-2020秋

PreProject-Logisim-Logisim时序逻辑电路

本系列所有博客，知识讲解、习题以及答案均由北航计算机学院计算机组成原理课程组创作，解析部分由笔者创作，如有侵权联系删除。

从本节开始，课程组给出的教程中增添了很多视频讲解。为了避免侵权，本系列博客将不会搬运课程组的视频讲解，而对于文字讲解也会相应地加以调整，重点在于根据笔者自己的理解给出习题的解析。因此带来的讲解不到位敬请见谅。

SR锁存器

通过前面的学习，我们已经学会使用Logisim搭建简单的组合电路。但是，这些电路的输出仅仅取决于电路的当前输入值。如果我们想搭建一个电路，使其输出结果不仅与当前的输入有关，还要与电路之前的输入有关（即电路能记住之前的状态），那么该怎么做呢？

我们先来讲解一种简单的电路——SR锁存器（SR latch），它由两个交叉耦合的或非门（或者，等价地，两个反置输入的与非门）组成，整个电路的状态可以由S（Set）和R（Reset）输入来决定，对应得到两个相反的输出Q和~Q，它的真值表如下：

$\overline{S_D}$	$\overline{R_D}$	Q^n	Q^{n+1}	功能
0	1	0	1	置1：次态为1
0	1	1	1	
1	0	0	0	置0：次态为0
1	0	1	0	
1	1	0	0	保持：次态不变
1	1	1	1	
0	0	0	约束条件， $\overline{S_D}$ 和 $\overline{R_D}$ 不能同时为0，即必有： $\overline{S_D} + \overline{R_D} = 1$	
0	0	1		


<https://blog.csdn.net/JeremyZhao1998>

其中SD和RD为是电路输入的两个端口，Qn表示电路当前的输出，Qn+1电路下一个状态的输出。由真值表可以看出，这个电路的输出不仅和当前输入有关，也和上一轮的输出有关。显然这个电路就是一种能记住自己之前状态的电路。其实，这就是一个最简单的时序电路，我们称它为SR锁存器。

观察它的功能一栏，可以看到，通过改变SD和RD为合适的值.我们可以改变电路的输出，而当SD和RD均为0时，电路会一直保持原来的输出不变，这看上去很像U盘之类的设备（通电时能够修改存储的内容，断电时保持内容不变）。事实上，通过配合合适的外部电路，我们就可以使用这个电路来存储整个电路的状态，从而搭建起更复杂的时序电路。

那么这个电路到底是怎么搭建的呢？常见的有两种方法:


最新文章

- 

第六章在web中


2023/8/19 19:38:1
- 

OpenWrt 刷机后

2023/8/19 19:38:1
- 

海 信 IP906H- 5

安-50T1版-当页

2023/8/19 19:38:1
- 

TX2使用 sdkma

题记录

2023/8/19 19:38:1
- 

TX2刷机 Jetpac

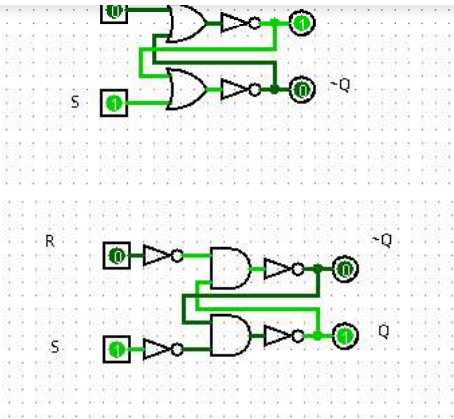
亲测记录

2023/8/19 19:38:1
- 

openwrt dhcp 无

何安装 Openwrt

2023/8/19 19:38:1



你可以在Logisim中动手搭建一下，看效果是不是和真值表一致。对SR锁存器更详细的介绍可以参考这个网页

D锁存器（D latch） 和 **D触发器（D Flip-Flop）** 的内容分析方法类似，请同学们自行学习（你可以尝试在Logisim中使用SR锁存器来搭建它们）。有兴趣的同学可以在logisim上用基本门电路搭建出一个Memory的库，并和logisim内置的Memory库进行比较，思考优缺点。

思考题：为什么上面两个电路是等价的，有没有从电路图上直观判断的方法？（回忆离散数学相关知识，并参考《数字设计和计算机体系结构2nd Edition》的第2.5.2节 bubble pushing）

由德摩根律， $\sim A \& \sim B = \sim(A | B)$ ，不难看出上面两个电路是等价的。

时序电路的相关部件

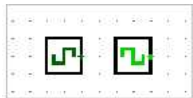
常用的时序电路部件有：

- 时钟
- register
- RAM
- ROM

相关的功能以及使用说明：

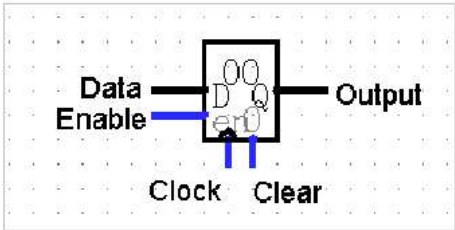
时钟：

在Logisim中，当时钟使能端打开后，时钟按照一定的频率输出高频低频信号，所有的时钟都按照相应的频率输出信号。

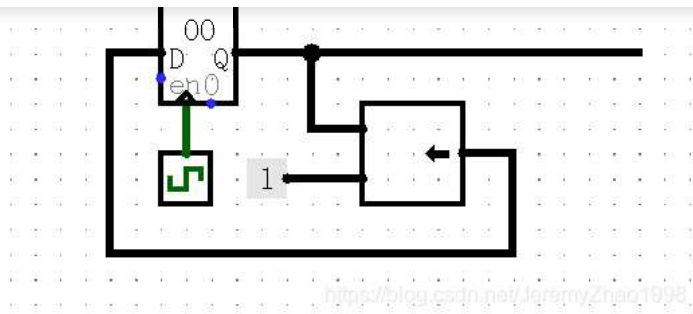


上图即是不同状态的Clock元件。时钟作为我们日后仿真测试中必要的元器件，必须掌握其相关的仿真技巧以及时钟频率的调节，在simulate栏目中有对时钟频率的具体调节方式，请大家先自行探索。

寄存器：



思考题：若只使用clock，一个寄存器，和其他的基础元器件，如何实现第i个周期，寄存器的值被更新为 2^i 。



如上图，将寄存器的输出端信号连接一个移位器，每次左移一位，再将结果作为输入返还到寄存器的输入端口，就可以实现每一个周期寄存器中的值是前一个周期的2倍，从而实现了第*i*个周期寄存器的值被更新为 2^i 。但需要注意，第0个周期，也就是初始的寄存器值应当是 $2^0 = 1$ ，也就是寄存器需要用其他方法预置1，这可以通过Plexers等器件进行一个判断，或者其他更巧妙的方法来实现，在上图的电路中并没有表现出来；并且数据的位数是有上限的，若干个周期之后数据会溢出，寄存器的值会重新回到0。若要实现不间断地循环左移，可以搭配判断的电路，判定此时寄存器输出是否为0，如果为0则下一个周期将其置1。

RAM存储器：

RAM是一个可读可写的存储器，在我们的实验中，我们采用的是读与写相互分离的类型，所以在选择RAM时，请将数据接口选择为“Separate load and store ports”。

ROM存储器：

ROM是一个只读类型的存储器，顾名思义，在使用过程中只能对其进行读取操作，而不能进行写操作，所以ROM在创建时，必须一次性将所有信息全部导入，之后不可再进行更改。

对于RAM和ROM的数据的导入：

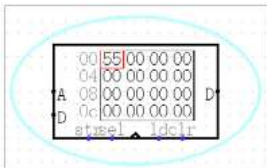
1. 手动导入：可手动在数据区域选择内存直接进行更改
2. 文件导入：可编写相应的数据文件，进行一次性导入，文件头需要增加一行“v2.0 raw”字样才可以正确导入。

对于RAM，ROM数据位宽，在我们的实验中，请选择32位，这与MIPS的指令长度相匹配，至于地址位宽，请确保位宽长度能够容纳实验要求的指令数量。下一节将用一个视频来讲解Memory的操作流程，帮助大家对其进行巩固。

Memory

手工键入数据：

鼠标确定一块内存位置，显示红色矩形框，这时便可以输入相应的数据。



注意，这里输入数据时，是以**16进制**进行表示的，这里选择的数据的位数为8位，键入的模式为后四位前移，前四位去除的模式，并且Backspace不能消除数据，若要消除一块数据，需要键入00。

对于数据框位置的选择，不可使用上下左右键进行选择，其中Enter表示下一行，Backspace表示上一块选择数据。

选择地址位置时，上述选择方式依然适用，也可以直接输入地址位置，进行跳转。

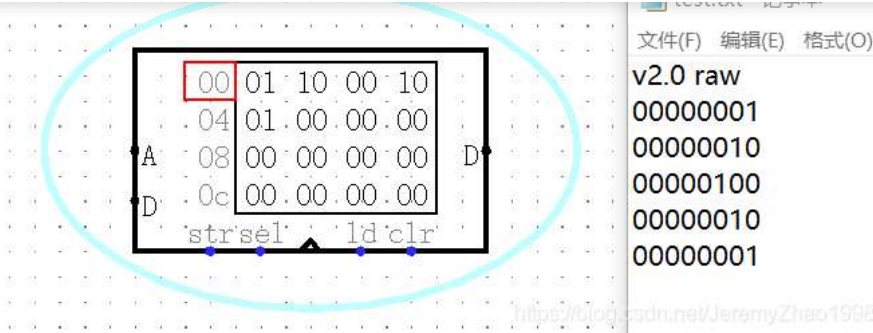
文件输入模式：

- 文件头：v2.0 raw
- 输入数据要和存储器的数据位宽相匹配，否则会出现数据截断

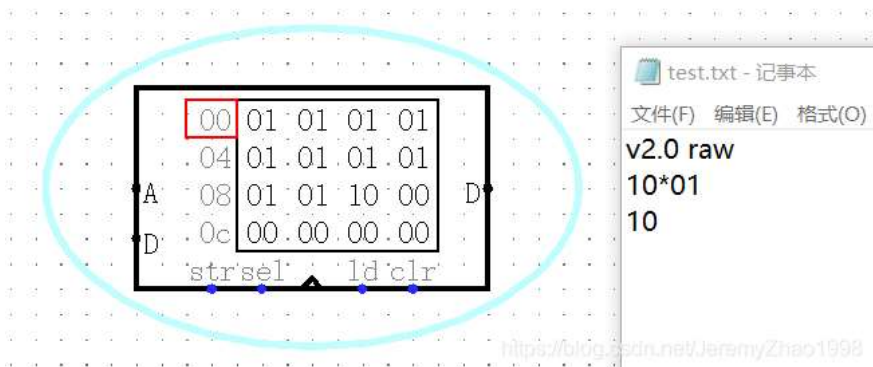
思考题：如果对存储器分别输入以下文本内容，最终存储器的数据分别应为怎样：

```
1 v2.0 raw
2 00000001
3 00000010
4 00000100
5 00000010
6 00000001

1 v2.0 raw
2 10*01
3 10
```



第一种情况对应的结果如上图，Logisim的文件输入是一行一个十六进制数输入进去，当输入进去的数据超过两位十六进制的表示范围时，Logisim会对其进行截断处理。上图中文件里第1、2、4、5行分别是0x01、0x10、0x10、0x01，对应存储器中的位置都是正确的，但第3行内容是0x100，超过了表示范围，截断后是0x00。



第二种情况向我们展示了可以使用乘号来进行简化表示多个相同数据，乘号前面的数是十进制，表示乘号后面的数重复出现的次数。该数据输入到Logisim的结果是10个重复的0x01，最后跟一个0x10。

有限状态机

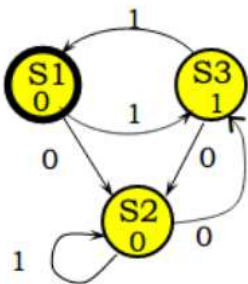
注意：本文所有内容假设阅读者具有基本的有限状态机相关理论知识！如果你不具备相应的理论知识，请在课堂上学习，或者可以通过下载相应的材料进行自学！

中文材料（来自理论课课件）

英文材料（来自MIT 6.004x）

先期准备

时序电路中最常用的一类元件就是有限状态机（Finite State Machine，FSM），因为许许多多比较复杂的时序行为我们都可以通过有限状态机来进行建模，并将其电路化。在Logisim中搭建有限状态机前，我们有必要明确一点，我们需要在开始搭建具体电路前，对状态机进行设计，这一步工作往往才是整个电路搭建工作的难点。其中需要确定的就有状态编码位数k，输入位数n，输出位数m，以及整个状态转移逻辑和输出逻辑。在将这些设计层面的工作完成，转化为状态转移表，或者状态转移图，之后我们才可以着手搭建工作。



状态存储与状态转移电路

有限机中的状态进行编码后，就是一些普通的二进制数，只是会随着时钟周期的进行发生改变，因此对于状态的存储我们只需要使用相应的寄存器部件就可以完成我们的需求了。

0	0	S ₁	S ₂	0	0	0	0	0	x
0	0	S ₂	S ₁	0	0	0	1	0	x
0	0	S ₁	S ₂	0	0	1	0	x	x
0	0	S ₂	S ₁	0	0	1	1	0	0
0	1	S ₁	S ₂	0	1	0	0	x	x
0	1	S ₂	S ₂	0	1	0	1	x	x
0	1	S ₁	S ₂	0	1	1	0	x	x
0	1	S ₂	S ₂	0	1	1	1	0	x
1	0	S ₁	S ₁	1	0	0	0	x	x
1	0	S ₂	S ₂	1	0	0	1	x	1
1	0	S ₁	S ₁	1	0	1	0	0	x
1	0	S ₂	S ₁	1	0	1	1	x	x
1	1	S ₁	S ₁	1	1	0	0	0	x
1	1	S ₂	S ₁	1	1	0	1	x	0
1	1	S ₁	S ₂	1	1	1	0	x	x
1	1	S ₂	S ₂	1	1	1	1	0	x

状态转移电路是有限状态机的核心，一般的有限状态机的下一状态取決当前状态和输入，因此对于k位状态编码以及n位的输入来说，这就是一个n+k位输入，k位输出的组合电路，我们可以通过真值表等方法之前设计好的状态转移表较为容易地得到它。

输出电路

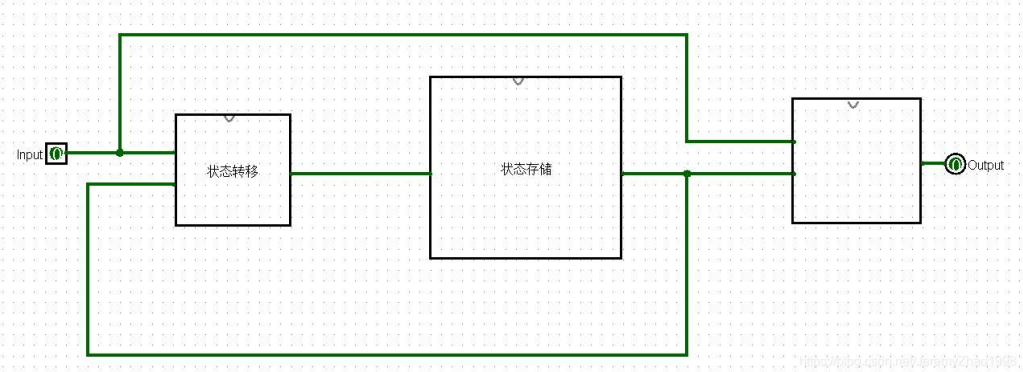
输出电路同样是FSM重要组成部分，输出部分的不同产生了两类有限状态机：

- 输出仅由当前状态决定的Moore Machine
- 输出由当前状态和输入决定的Mealy Machine

从具体电路的视角看来，这两种状态机的差别就在于输出电路的输入位数不同。我们应该根据具体应用场景来进行相应的设计。

整体视图

综上，一个典型的Mealy型FSM应该在Logisim中有如下图的结构，可以拆分成三个较为明确的子电路再去完成相应的功能。



从左至右的三个模块分别是：状态转移，状态存储，输出

如此，我们可以通过组合电路相关工具和自己的设计，来搭建FSM。

状态机浅析

心情电路

设计一个有限状态机来跟踪电子设计实验室里4名学生的心情。学生的心情有HAPPY（开心，电路正常工作），SAD（忧愁，电路烧坏），BUSY（忙碌，正在设计电路），CLUELESS（愚笨，被电路所困扰），ASLEEP（睡觉，趴在实验桌上睡着）。请问

1、这个有限状态机需要多少个状态？

答案：625

4名学生，每名学生可能有5个状态，总状态数自然是5*5*5*5=625个

2、至少需要多少位来代表这些状态？

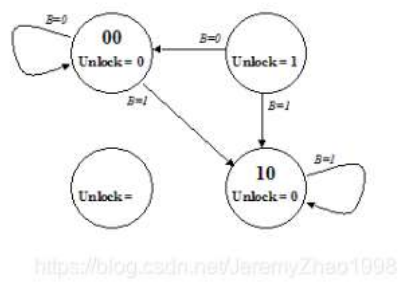
答案：10

数字锁

ACME公司最近收到了Wiley E. Coyote先生的订单，这个订单是全数字的令人费解的挂锁(Perfectly Perplexing Padlock)。这种锁有两个按钮(0和1) 且当按钮按下时控制锁的FSM进入一个新的状态。这两个按钮都在B信号上编码(按钮0: B=0, 按钮1: B=1)。当FSM将UNLOCK输出信号置为1时表示锁此时是打开的。然而这种锁的设计说明是不完整的，现在请你根据下面不完整的状态转移图中的线索和上面所描述的规范补全真值表。

- a)状态转移图中的每个状态都由2位二进制数字编码，称为S1S0。下一状态称为S1'S0'
- b)为每一个状态指定UNLOCK的输出值

注意：Unlock的值对应于当前状态S1S0



S1	S0	B	S1'	S0'	Unlock
0	0	0	0	0	答案：0
0	0	1	1	0	答案：0
0	1	0	1	1	0
0	1	1	1	0	答案：0
1	0	0	0	1	答案：0
1	0	1	答案：1	答案：0	答案：0
1	1	0	答案：0	答案：0	1
1	1	1	答案：1	答案：0	1

此题只要了解状态机的概念即可。

以本人浅薄的理解，状态机实际上就是一个状态向下一个状态自动转换的体系：当前状态和输入共同地唯一决定下一个状态，转换的过程自动地持续进行。

以上题为例，状态图中告诉我们一共有四个状态，使用2位二进制进行编码，其中一些状态代表门锁打开，另一些代表门锁关闭，例如状态00和状态10都代表门锁关闭（Unlock = 0）；状态图中还可以看出，当前状态和输入信号B共同决定下一个状态，因此这是一个Mealy型状态机，例如如果当前是00，而输入B为0，那么下一个状态依旧是00；如果当前是00，而输入B是1，则跳转到10状态。上图的状态图和状态表显然都不完整，我们的任务就是对照图表将二者补充完整。

首先对照图表可以填补表格的最后一列：图中告诉我们状态00和状态10都代表门锁关闭（Unlock = 0），因此最后一列第1、2、5、6行均为0；表格第3行告诉我们状态01同样代表门锁关闭（Unlock = 0），因此最后一列第4行也必定是0，因为其状态同样为01。表格最后两行告诉我们状态11对应Unlock = 1，因此状态图中右上角的Unlock = 1必然对应状态11，同时我们也可以知道左下角的状态只剩下01，其对应Unlock = 0。

接下来补充第4、5两列的后三行，这三行空格其实分别是在询问，状态10输入B为1的下一个状态是谁？状态11输入B为0的下一个状态是谁？状态11输入B为1的下一个状态是谁？查看状态图很容易得到答案，分别是10，00和10。根据表格也很容易将状态图中左下角缺失的两个转移补全，不再赘述。

斐波那契数列

见如下链接：

北航计算机组成原理课程设计-2020秋 PreProject-Logisim-斐波那契数列问题(简单迭代法+矩阵乘法的快速幂)

文章来源:<https://blog.csdn.net/JeremyZhao1998/article/details/108987580>

本文来自互联网用户投稿，该文观点仅代表作者本人，不代表本站立场。本站仅提供信息存储空间服务，不拥有所有权，不承担相关法律责任。如若转载，请注明出处：<https://www.rstt.cn/news/478186.html> 如若内容造成侵权/违法违规/事实不符，请联系我的编程生涯网邮箱：809451989@qq.com进行投诉反馈，一经查实，立即删除！

相关文章：

计算机二进制编码原理,浅谈计算机信息的二进制编码

视频教程-【深入理解计算机网络】讲师解读计算机网络原理视频课程（上）-计算机网络

YOLOv5原理分析及部署教程

计算机网络原理视频学习教程

计算机网络6版,计算机网络教程（第6版）

名校计算机课程百度云,浙江大学计算机类专业视频课程百度云网盘

计算机原理与结构 实验1《实验工具使用》

计算机图形学实用教程苏小红,计算机图形学实用教程（第4版）

遥感、GIS、计算机视频教程

计算机组成原理的基本知识

【福利】计算机公开课合集（视频教程）

视频教程-【深入理解计算机网络】讲师解读计算机网络原理视频课程（中）-计算机网络

计算机处理器基础原理笔记

【MySQL约束】数据管理实用指南

计算机组成原理第五章(2)---中断

图欧学习资源库 <https://tuostudy.com/>

点云库PCL学习

软件开发学习资料大全

学习资料分享平台系统

学习资料链接地址

5个常用的机器学习python库

深度学习，mxnet库的安装

图欧学习导航<https://tuostudy.upnb.top>

关于如何查找和利用PCL库学习资源的一些心得

哈尔滨工业大学考研 网络与空间安全 837 资料库

动态库学习

6大最常用的Java机器学习库一览

Python学习之Turtle库

Python学习资料篇

5种Python深度学习库和资料

智能合约学习资料

机器学习资料整理

学习资料推荐网站

各种学习资源库，非常好的收藏汇总！！！！

迁移学习 Transfer Learning(可能是目前最全的迁移学习资源库？)

本人常用资源整理(ing...)

常见的API接口漏洞总结

深度学习第J8周:Inception v1算法实战与解析

Linux内核配置选项详解

(转) 电脑内外接口全程图解

VGA的焊接方法

堡垒之夜显示连接不上服务器,堡垒之夜匹配提示未能连接到服务器 | 手游网网页游戏攻略大全...

电脑常见的接口大全

详解嵌入式系统知识和接口技术总结

HDMI、DVI、VGA、RGB、分量、S端子)高清接口图片说明

音视频接口 (全)

DisplayPort接口全总结

15针VGA公头焊接示意图

VGA信号详解

【转载】VGA视频信号详解

VGA视频信号详解

VGA线序及接线图

计算机主机箱连接线路,怎样连接电脑所有配件? 电脑主机与显示器接线安装步骤...

lvds接口屏线安装图解_液晶显示器LVDS接口工作原理及改制方法

vga 3+6线序

自制hdmi线一头改vga图_最全VGA、DVI、PS/2、USB等接头详解焊接方法

VGA线材说明与鉴别详解

vga焊接线顺序_VGA线的焊接方法, 15线与7线分别如何焊接?

免焊vga3加6接线图_高清 VGA免焊接头3+6+9 VGA快速接头 免焊公头 3排15针 15针插头...

自制hdmi线一头改vga图_VGA连接线接口定义及引线焊接教程, VGA线不够长时可用网线代替? ...

vga焊接线顺序_VGA线的焊接方法 (VGA接头焊接方法)

VGA线的接法

Open3D 点云包围盒

主成分分析法 (PCA方法) 计算OBB包围盒

AABB与OBB包围盒子

判断包围盒是否相交

cad生成最小包围盒lisp_cad.net 葛立恒凸包算法和面积最小包围盒

项目杂识-包围盒

PCA构建点云包围盒

PCL ——最小包围盒

Unity3D AABB包围盒效果

PCL八叉树的包围盒研究

PCL 最小包围盒

OBB包围盒

AABB包围盒

包围盒算法基本知识 (一)

上海交大教授: 何援军——包围盒 (包容体/包围盒子)

PCL包围盒 (详细介绍)

Bounds(包围盒)详解-【AABB包围盒、Sphere包围球、OBB方向包围盒、FDH固定方向凸包】

马上五一了, 带大家玩一下五子棋——C语言

【Python】狂肝两万字带你学会——类与对象

FreeCAD-中文设置

cad怎么把图层英文变成中文_CAD图层标准中英文互译

CAD二次开发之ArxDbg的安装

Scan2CAD pro中文版

Scan2CAD中文版

CAD中 OLE不能旋转_AutoCAD小秘密042: 光栅图像和OLE图像, 究竟如何选择

CAD中 OLE不能旋转_AutoCAD中光栅图像和OLE图像, 究竟如何选择

[cad怎么把图层英文变成中文_cad图层英文](#)

[Android 进阶——Binder IPC之Binder 驱动基础理论和协议详解（十）](#)

[FPGA经典设计：再读正点原子SDRAM控制器](#)

[SDRAM 控制器（七）——控制模块](#)

[比脑力更强大的DDR SDRAM控制器](#)

[SDRAM控制器说明/altera/northwest logic](#)

[SDRAM随机读写控制器](#)

[SDRAM控制器（初始化）](#)

[我的编程生涯介绍](#) [商务合作](#) [免责声明](#)

CopyRight © 我的编程生涯版权所有