

# 智能电网无人值守工作站

## 一、 引言

随着网络技术和电子技术的高速发展，以及城市规模的扩大，智能电网以其众多的优点得到了巨大的发展，通过为每个用电用户配置智能化的信息采集系统，把分散的电网用户数据整合起来，再通过网络等途径传输给城市用电的中央控制机构，便于对各处的用电情况进行分析和调配，使得整个电网的管理能够更加的智能化和实时化。在这其中重要的一环就是各个用电单位的电力数据的初级汇总，一般在一个小的区域内（比方说一座住宅楼或一个车间）配置有多台采样系统，它们的采样需要一个统一的控制，同时也需要把数据汇总起来后，用一个发送节点传送到总电网数据系统，根据这样一种工作模式，选用主从工作站的结构是最合适的，即若干从站受控于一台总站，由总站汇总数据并上报。我在上学期学习网络基础课程时曾了解过智能电网的相关内容，在学习了数字电子技术基础这门课程之后，想到其实可以利用我们已经学过的知识初步地实现这样一种主从工作站，因此尝试这设计了这样一个系统，一方面是对已学知识的一种巩固，另一方面也可以锻炼自己综合各方面信息的能力，但是由于水平所限，设计难免会有很多欠考虑之处，而且不能够轻易付诸实用，还望谅解。

## 二、 设计目标

- 1、设计一个主从工作站通信网络，其中包括一台主机和若干台从机，为了设计之便，取从机数为 8，分别编号为 M0~M7，主机代号为 C。为了节约架线成本，主从之间的通信只是用 2 条传输线，采用串行通信的方式，两条线分别称为 DB（数据总线）和 CB（控制总线），DB 是从机向主机传送数据的通道，CB 则是主机向从机发送控制指令的通道，实际中可以把这两条线做成双绞线直接铺设。
- 2、从机负责利用自带的数字电能表采集记录用户用电数据，进行实时更新，这里设计的用电量上限为 255，即电能数据用一个 8 位二进制数保存记录。同时，从机负责响应主机的呼叫请求，当收到主机发来的请求信号，并且判明信号的作用对象是自己时，从机会把当前采集到的电能数据通过 DB 发给主机。
- 3、主机在计时器的帮助下进行全年 365 天计时，每天的 0:00，主机自动通过 CB 依次向 M0~M7 从机发送请求信号，信号的内容各个从机编号的 8 位高低电平表示，比如 M3 的编号为 3，发送内容就是 0000,0100，0 代表低电平，1 代表高电平。每发送完一个控制信号，主机等待从机的反馈数据，取得数据后再发送下一台从机的控制信号，同时把对应用户的电能表数据（存放在 RAM 中）更新。全部数据更新之后，主机重新进入待机状态。
- 4、由于采用串行通信，事先约定主从机之间的通信协议，要求所有串行数据用 2 位高电平作为起始信号，起始信号过后是 8 位的二进制信息，之后用 2 位的低电平作为截止信号，传输线上空闲电平为低电平。

## 三、 设计的模块化

为了能够实现自底向上，逐步求精的设计思想，需要事先把主从机的电路设计进行模块化划分，以便能够分块实现。下面的 FIG1 给出了主机和从机的电路模块图。

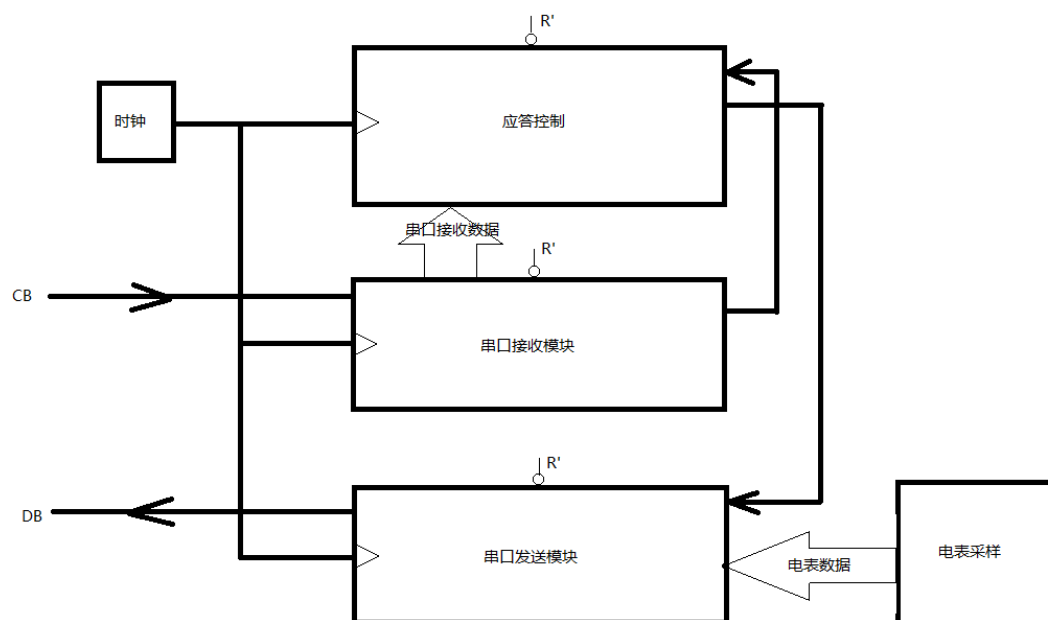
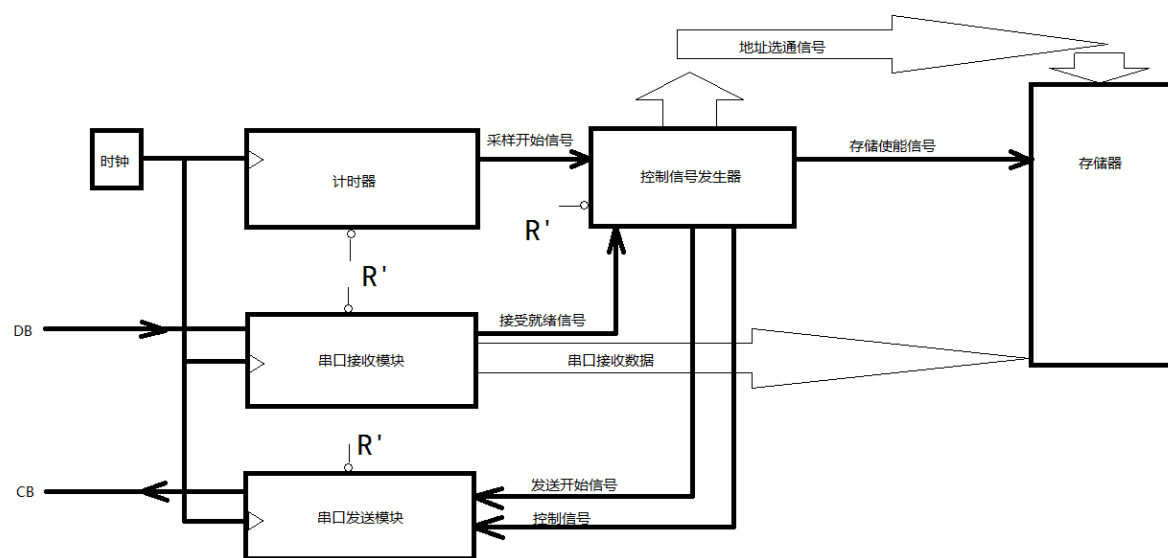


FIG1 电路设计的模块图  
(a) 是主机的设计 (b) 是从机的设计

以下对各个模块进行一个概略的功能描述，以便根据其功能要求进行具体设计。

- 1、时钟模块：提供系统工作必须的时钟信号，这里假设时钟频率为 100Hz，实际上可以使用晶振根据频率进行适当分频得到。
- 2、串口接收模块：负责接收串行数据，自动检测信号线 X 上的电平，每当检测到起始位的出现，在接下来的 8 个时钟周期内不断把信号线上的各位数据移入接收移位寄存器中，在接受完最有一个数据位后发出一个持续 1 个周期高电平的接受就绪信号 RX。从串口接收模块的并行输出端口 D0~D7 可以读取接收移位寄存器中的数据。
- 3、串口发送模块：负责发送串行数据，当把待发送的 8 位数据位传给串口发送模块的 D0~D7 并行输入端口时，再给一个发送信号 TX，在接下来的 12 个时钟周期内，串口发送模块就开始按位发送串行信息的起始位、数据位和截止位。
- 4、控制信号发生器 (C)：用来产生选通各个从机的控制信号。当 C 的计时器计时到一天的 0:00 时，传给控制信号发生器一个持续 10ms (1 个时钟周期) 的高电平信号 ST，控制信号发生器向串口发送模块发送一个发送信号 TX，以及对应于 M0 从机的控制信号 0000,0000，同时输出一个控制存储器的地址信号 A0~A2，选通存储器里存储 M0 数据的一个字节，并用存储使能信号允许向 RAM 里写入新数据；在串口接收模块接收到 M0 的回应数据后，发出 RX 信号通知控制信号发生器一次请求--响应结束，控制信号发生器开始对 M1 进行同样的操作，以此类推，直到全部 8 台从机都完成响应，控制信号发生器才重新转入待机状态。
- 5、存储器 (C)：负责对从机发送回来的电能数据进行记录，还可以在外部控制电路的控制下读出等。
- 6、应答控制 (Mi)：是从机模块，负责控制从机响应主机的请求。当从机的串口接收模块接收到一次串行数据后，向应答控制模块发送一个接受就绪信号 RX，应答控制模块自动比对接收到的数据与本机的编号，当且仅当比对成功时，应答控制模块向从机的串口发送模块发送一个发送控制信号 TX，控制串口发送模块把从机的电能表数据通过 DB 发送给主机，之后转入待机状态。

## 四、 分块设计

下面开始分块设计各个模块的电路结构。

1、时钟模块：略

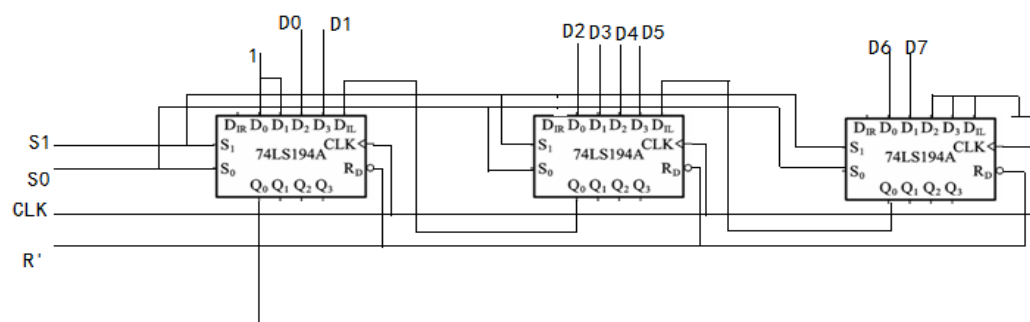
2、串口发送模块：

串口发送模块有两部分组成，一是发送移位寄存器，二是串口发送控制器，后者负责接收发送控制信号 TX，并在接到信号后控制发送移位寄存器把并行数据转换成串行数据发送出去。

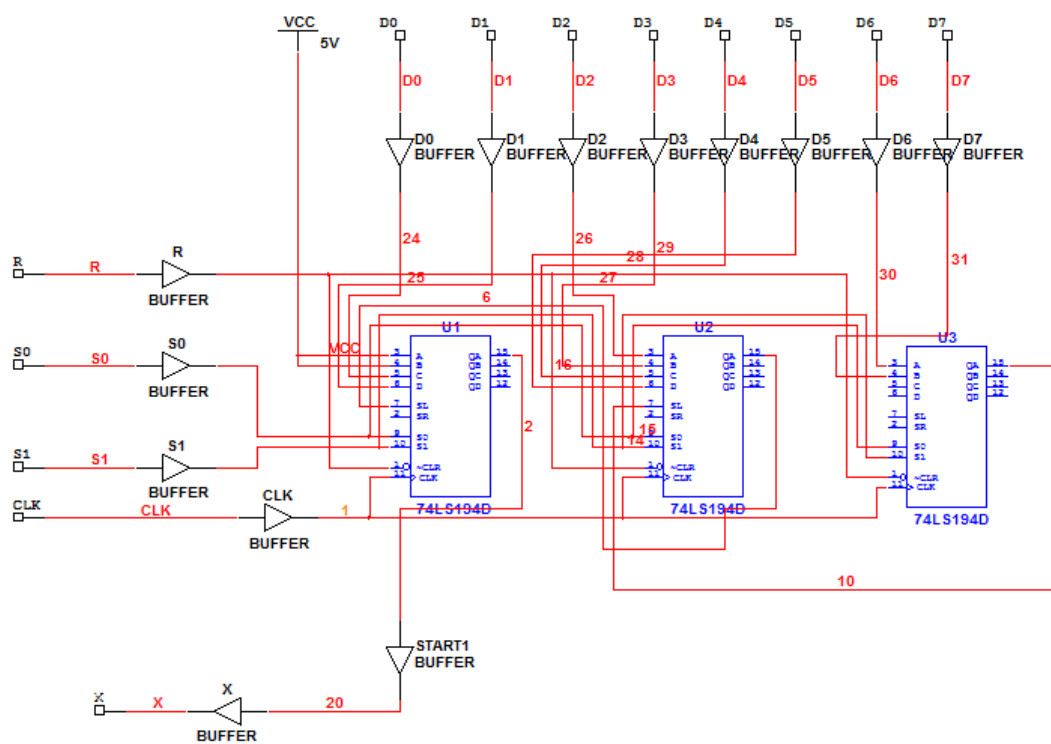
(1) 发送移位寄存器：

由于主要是用来实现串并转换的，可以使用成型产品 74LS194A 级联实现。

以下是设计电路图和 Multisim 10 上搭建的电路图。



(a)



(b)

FIG2 发送移位寄存器设计电路图

(2) 发送控制器:

根据电路的设计要求, 给出状态转换图, 其中  $S_i$  代表电路各个状态, 而 TX 代表发送控制信号,  $S_1, S_0$  是给移位寄存器的控制信号。

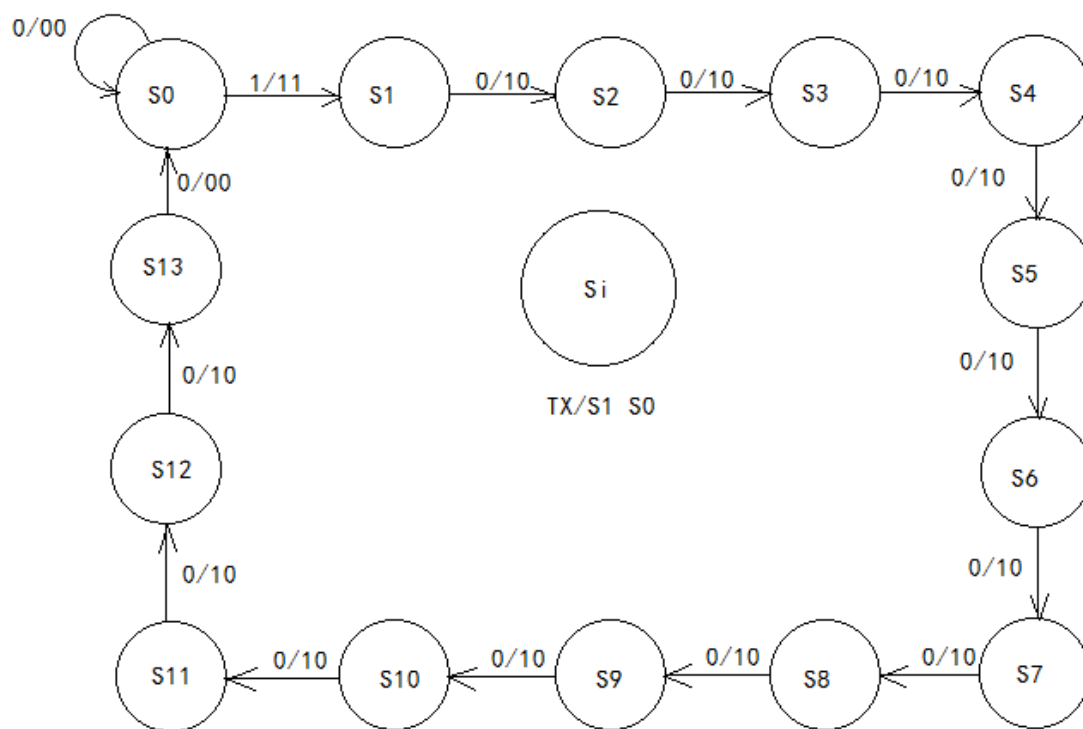
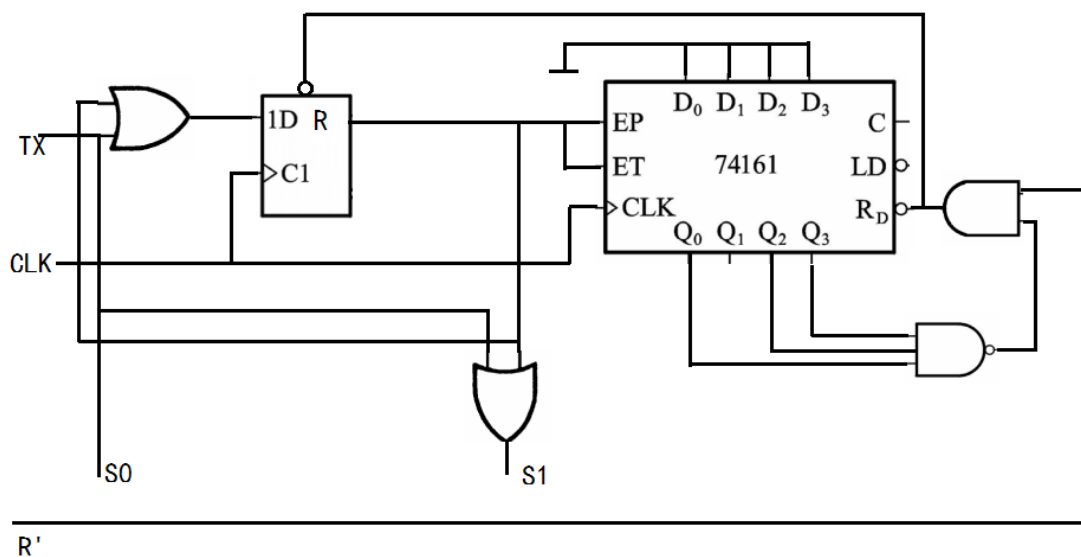


FIG3 发送控制器的状态转换图

根据电路的设计要求, 考虑使用 MSI 来搭建电路, 这就要求概括出电路的逻辑功能特点, 和已有的器件功能相匹配。下面的 FIG4 是设计的电路原理图。



(a)

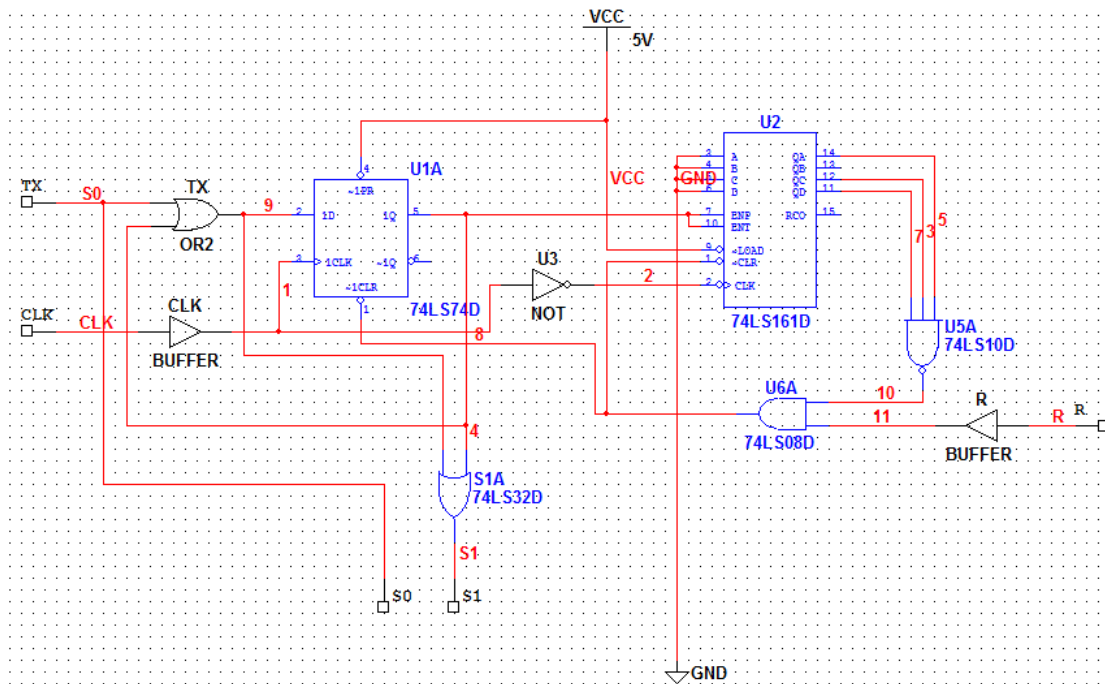
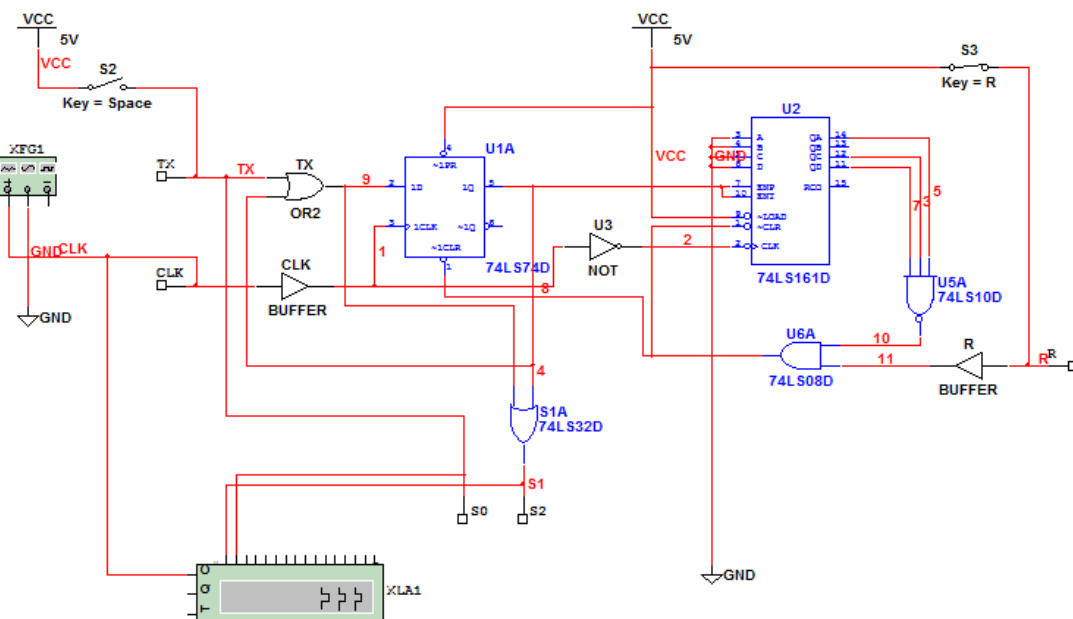


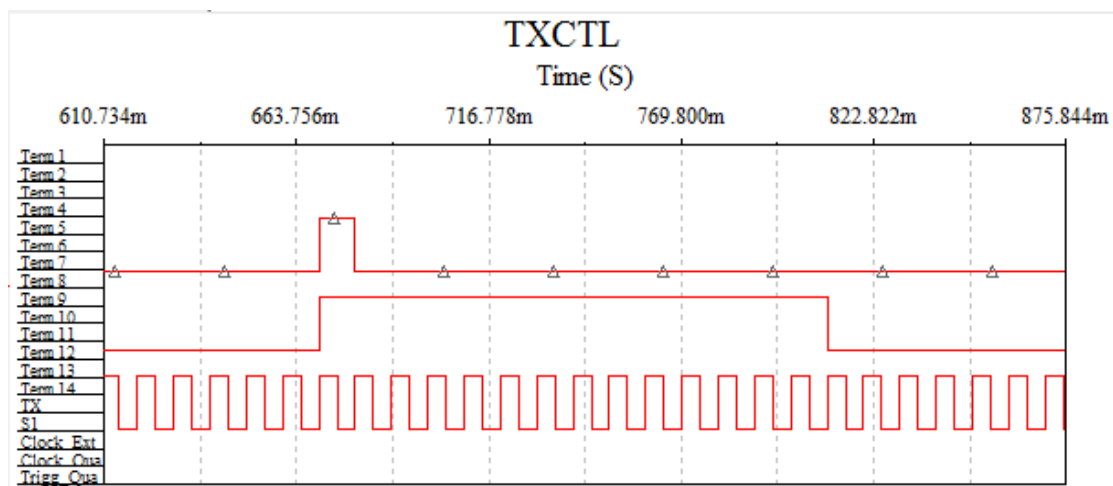
FIG4 发送控制器的设计电路图

可以简述设计思路：当TX信号到来时，由于TX只能持续1个周期，因此直接利用TX作为S0输出即可，而在接下来的13个时钟周期中，显然电路的状态在进行某种计数过程，只有计到某个特殊值时才会退出技术循环，而TX就是计数的启动信号。因此选择了计数器74LS161来计数13个时钟周期，在此期间，利用D触发器和或门组成的结构把TX信号锁存起来，直到计数到13采用异步复位把其清除。而且，从状态图上来看，只要没有退出计数循环，S1始终为1，而计数尚未开始的启动时刻，S1也要置1，因此取S1为TX信号与计数器计数使能信号的或运算。设计完毕。

下面是Multisim 10中对发送控制器的仿真功能验证：



(a)



(b)

FIG5 发送控制器的功能仿真，其中 (a) 是仿真用电路图；(b) 是得到的功能时序图

从图中可以看到，每当接收到一个发送信号 TX 时，首先把 S0 和 S1 都置位，一个时钟周期后让 S0 复位，S1 再继续保持 13 个时钟周期，促使发送移位寄存器进行左移动作，完成一次发送控制。

(3)

串口发送模块组装：

把串口发送模块的两部分组装起来就得到了完整的模块设计电路，下面是在 Multisim 10 中把两部分电路作为模块组装的电路图。

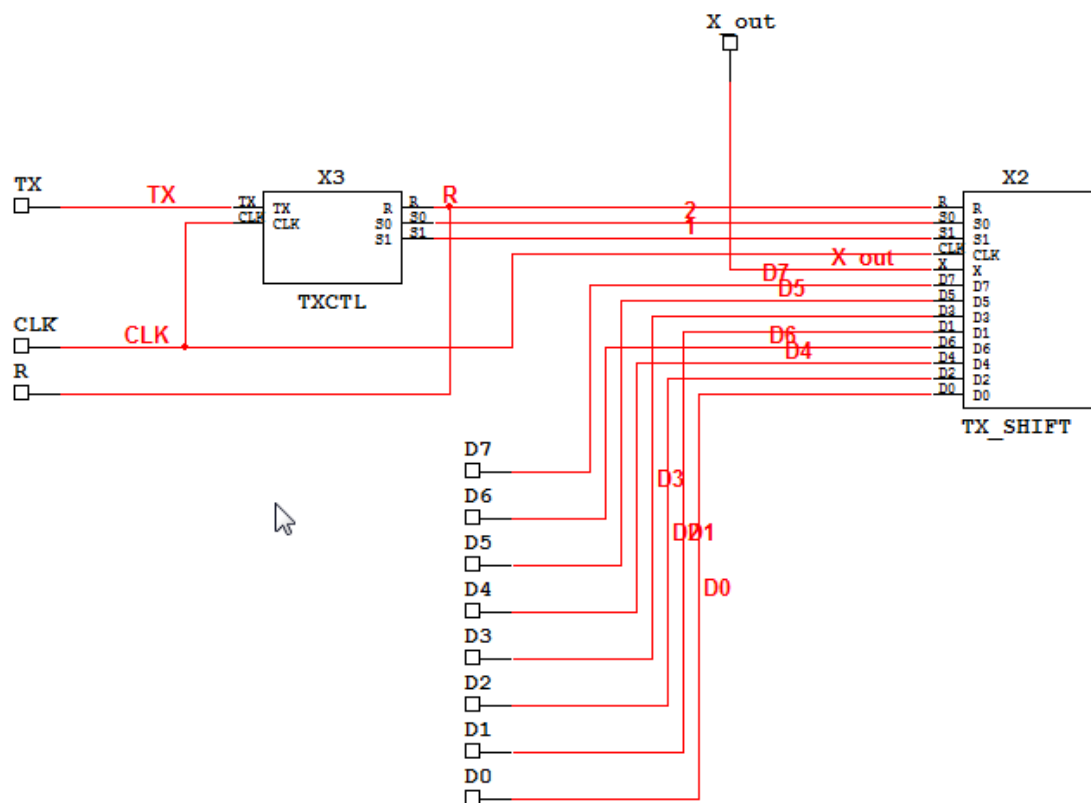


FIG6 串口发送模块组装图，其中 D0~D7 为待发送数据的并行输入端；X\_out 为串行数

据输出端口；R 为异步复位信号输入端（低电平有效）。

下面是串口发送模块的功能仿真。

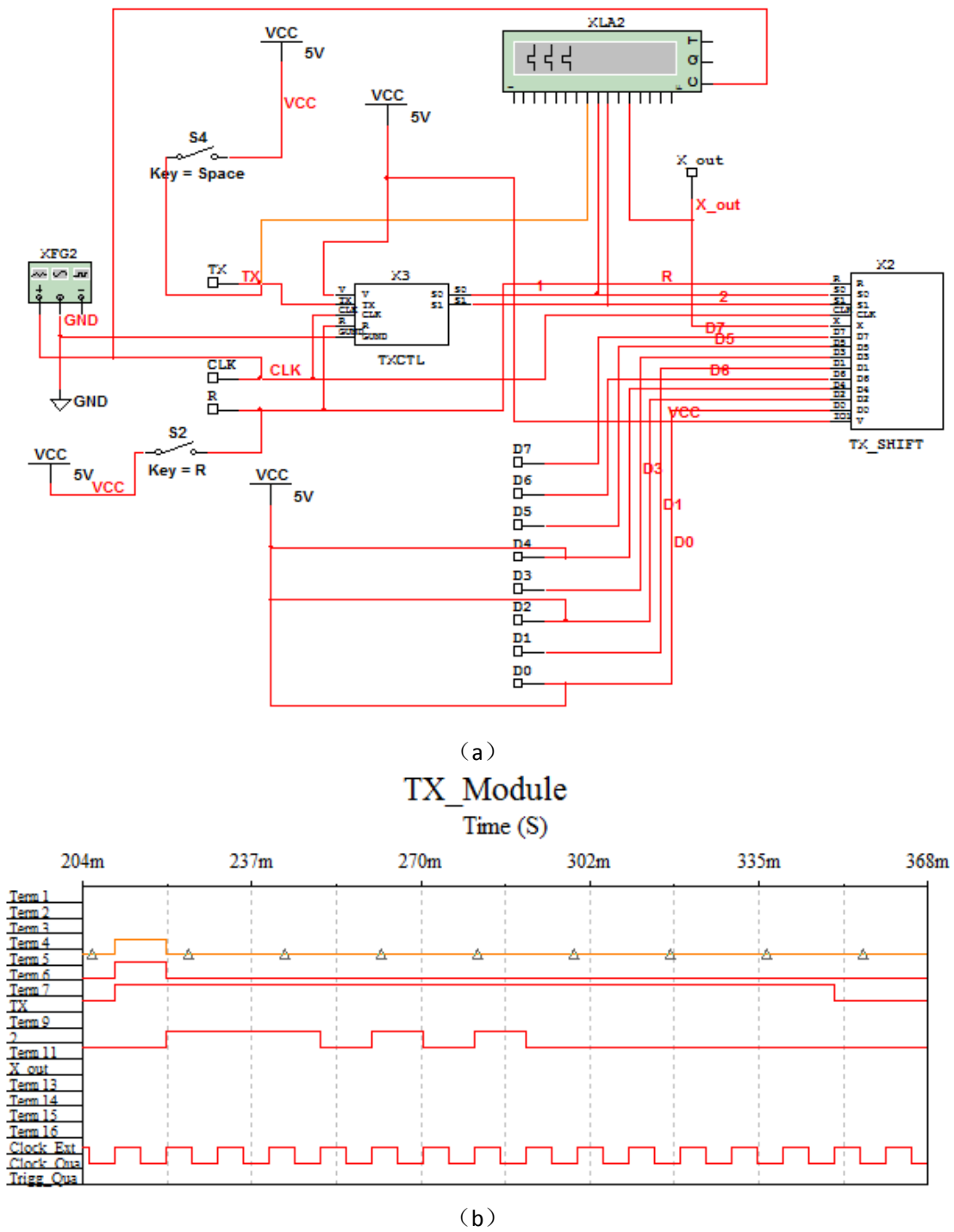


FIG6 串口发送模块的功能仿真，其中（a）是仿真用电路图；（b）是得到的功能时序图

从仿真结果可以看到，每当串口发送模块接收到发送控制信号 TX 时，就在接下来的若干个周期内把并行输入数据（这里是 0x15）加上起始位的 2 位高电平，通过串行数据线输出出去。



### 3、串口接收模块

串口接收模块同样由两部分电路组成，一是接收移位寄存器，二为接收控制器。后者负责自动检测串行数据线上的信号，发现串行数据的起始位后，在接下来的 8 个时钟周期内自动控制移位寄存器接受数据线上的 8 位数据，实现串并转换，接收完毕后发出接收就绪信号 RX，并转入待机状态。

(1)

接收移位寄存器：

同样，由于只需要实现简单的串并转换，直接选用双向移位寄存器 74LS194A 来实现，设计电路图和 Multisim 10 上搭建的仿真图见 FIG6

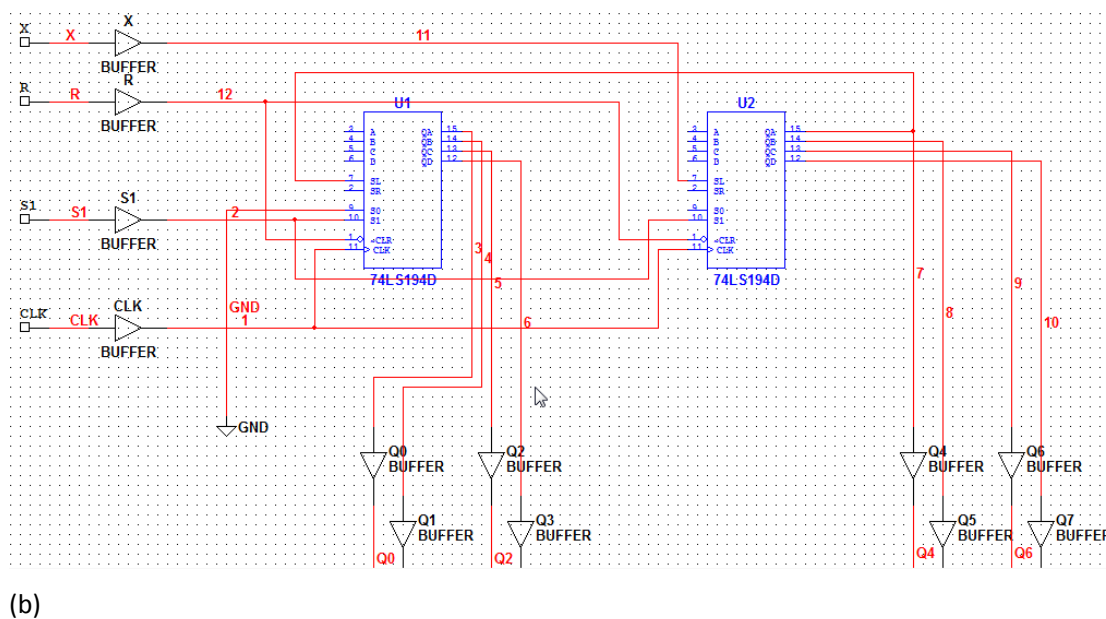
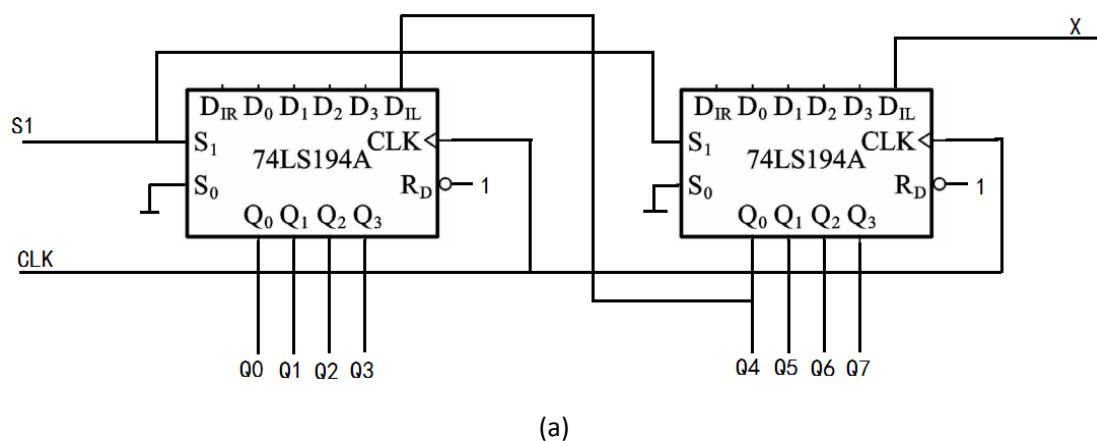


FIG7 接收移位寄存器的设计电路图

(2) 接收控制器：

根据接收控制器的功能要求，可以给出它的状态转换图如 FIG7 所示。

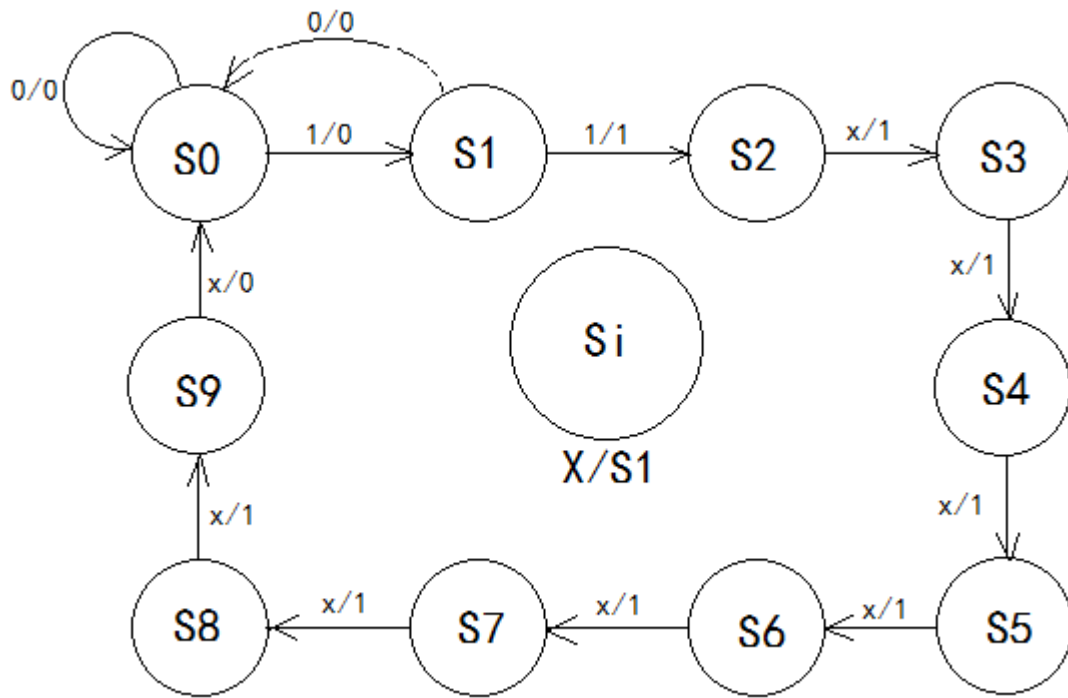


FIG8

接收控制器的状态转换图

假如使用 MSI 来搭建电路，不需要对电路的状态方程进行抽象研究，但是需要准确把握电路的工作特点，结合具体元件来实现。比如说，分析电路的功能，发现其中很重要的一个功能是检测串行数据起始信号（两个连续的高电平），这样的功能可以借助移位寄存器来实现，即移位寄存器不断地把串行数据读入，检测并行输出结果中是否有两个连续的高电平，就可以很轻松地实现检测目的，而且逻辑清晰。个人感觉这就是 MSI 和 SSI 搭建电路的不同设计特点。

FIG8 是接收控制器的电路设计。

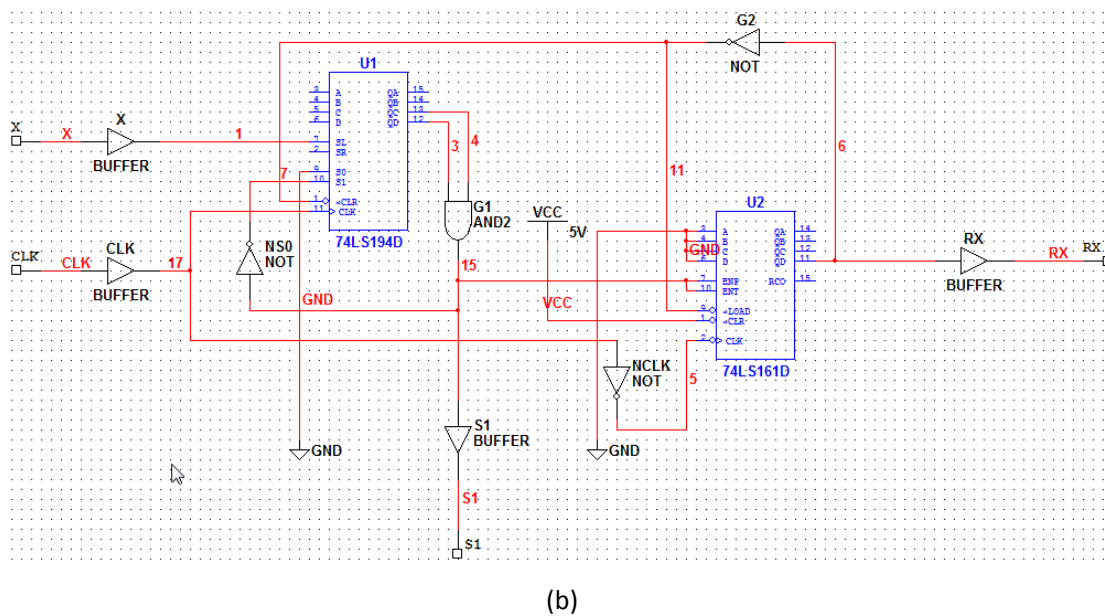
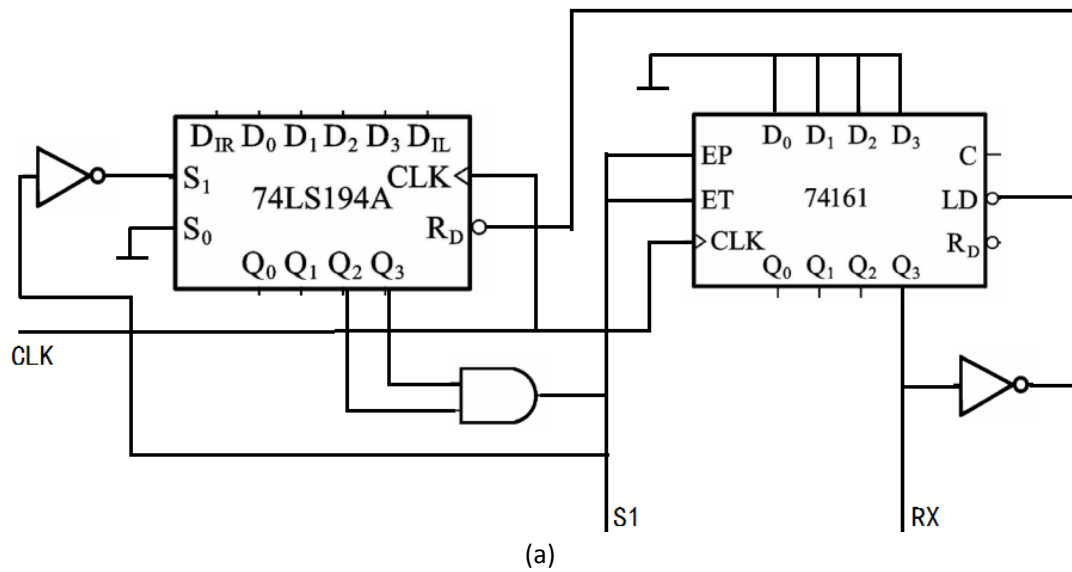


FIG9 接收控制器电路的设计图

下面是接收控制器的功能仿真结果。



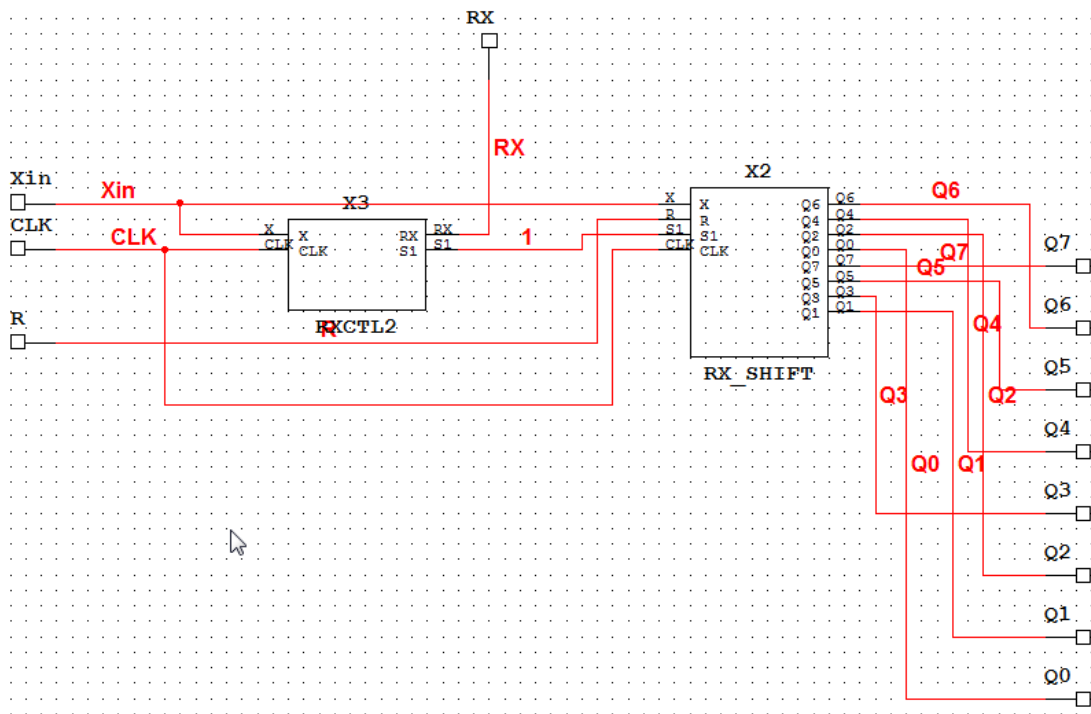
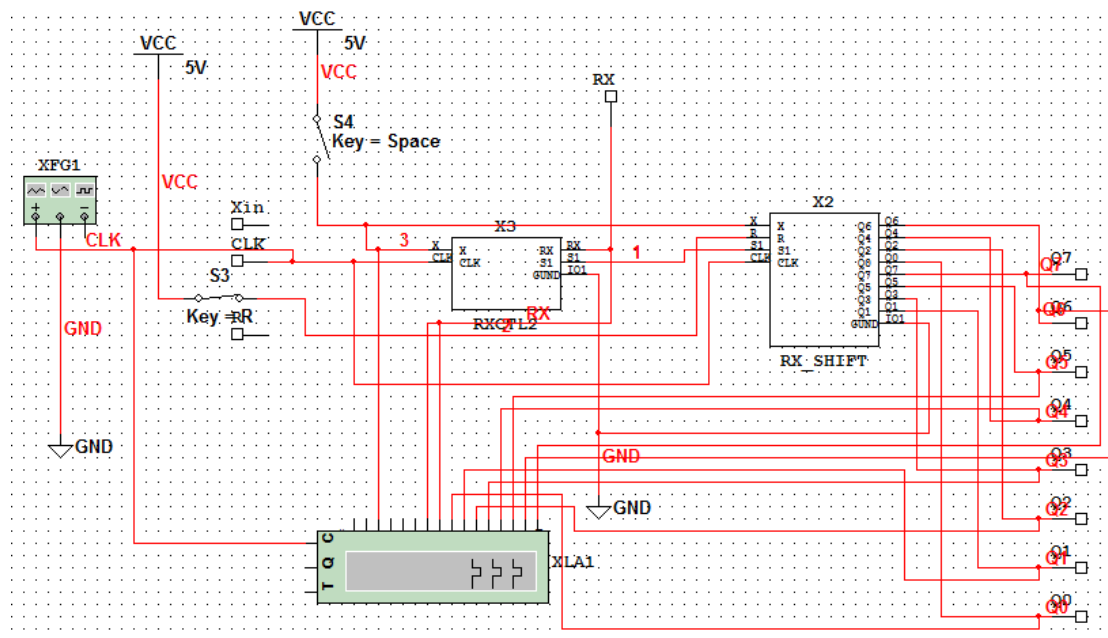
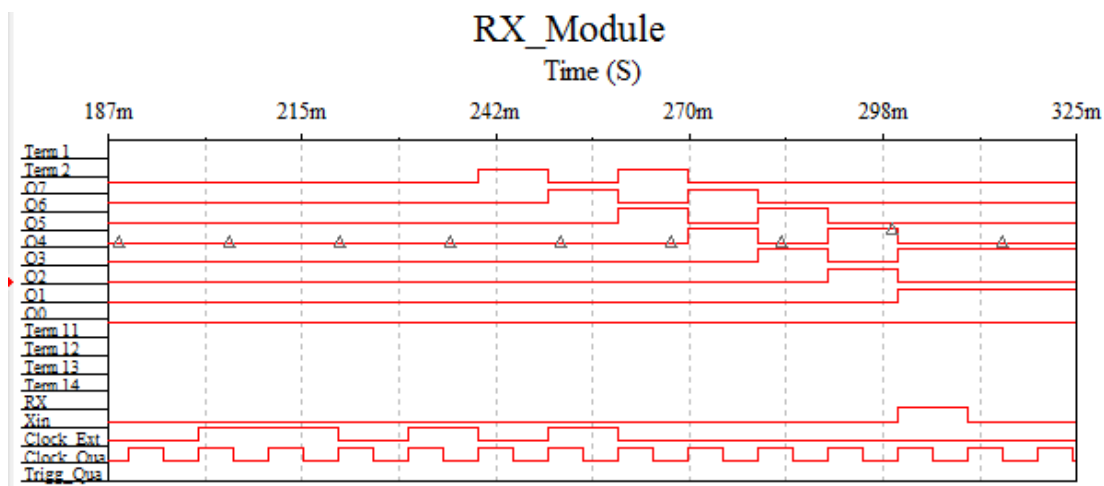


FIG11 串口接收模块的组装图，其中 Xin 是待接收的串行数据线，R 是复位信号输入端，低电平有效，RX 是接受就绪信号输出端，Q0~Q7 是接收数据并行输出端。

下面是整个串口接收模块的功能仿真。



(a)



(b)

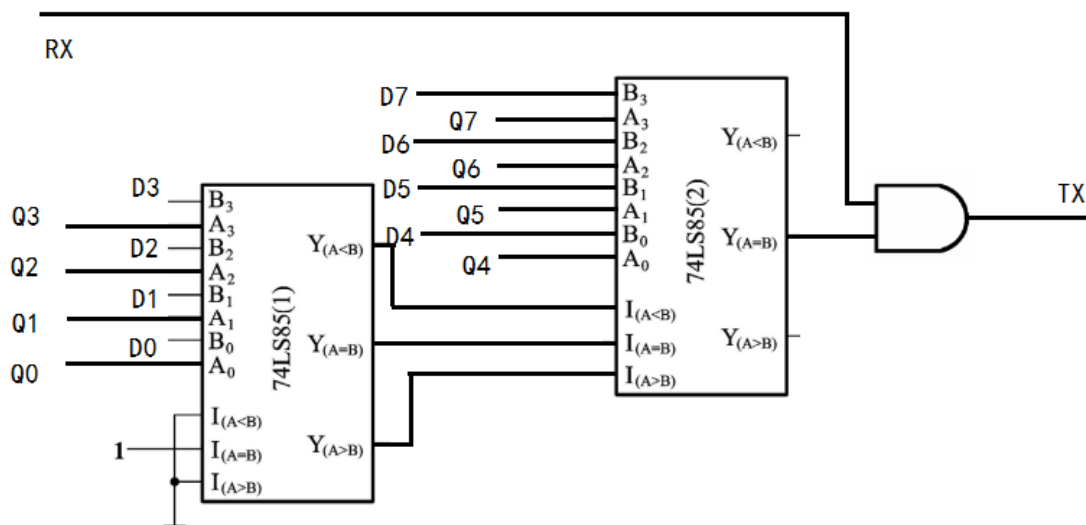
FIG12 串口接收模块的功能仿真，其中 (a) 是仿真用电路图；(b) 是得到的功能时序图。

在仿真时，通过控制开关开合，在串行数据线上产生串行数据，再通过逻辑分析仪观察并行输出和 RX 的输出信号。实验中发送的串口数据帧内容是 0x0A，当 RX 正脉冲产生时，可以看到除了 Q1 和 Q3 为高电平外，其他各位并行输出为低电平，表明并行输出为 0x0A，可以进行数据读取。

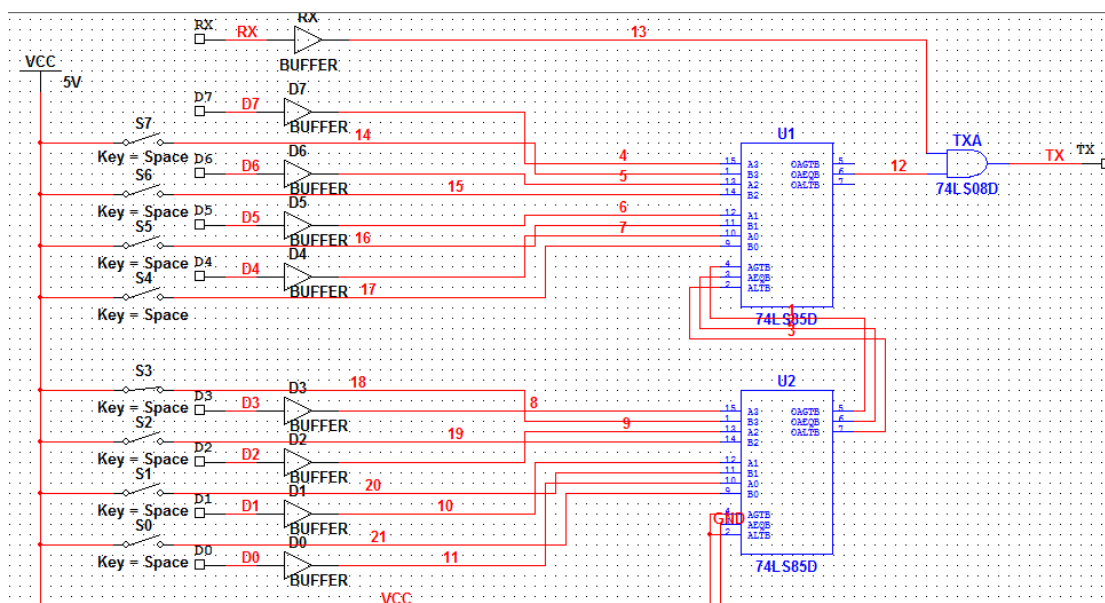
#### 4、从机应答控制模块：

从机的应答控制模块要实现这样的功能：即接收到串口就绪信号 RX 后，从串口接收模块的并行输出端取得串行数据 Q0~Q7，将其与从机的编号进行比较，假如相等，发出发送控制信号 TX，命令从机的串口发送模块将当前电能数据发送给主机。这里涉及到了两个 8 位二进制数据的比较，因此很自然想到使用数据比较器来实现，这里选择使用两片 4 位数值比较器 74LS85 实现电路。对于从机的编号信息，可以使用 8 位的 EPROM 来提供，只要在需要时给 EPROM 写入所需的编号即可。

下面是从机应答控制模块的设计电路图和 Multisim 10 上的仿真电路图。



(a)



(b)

FIG13 从机应答控制模块的电路设计，注意在 Multisim 10 的仿真电路中，使用开关连接电源的方法替代 ROM 来给从机编号，这里的编号是 3，即三号从机。

## 5、主机的控制信号发生器模块：

主机的控制信号发生器有两大功能，一是进行自动的计时，二是在计时到达特定时刻时产生发送启动信号，启动对从机序列进行查询的流程，据此，可以分为两部分进行主机控制模块的设计。

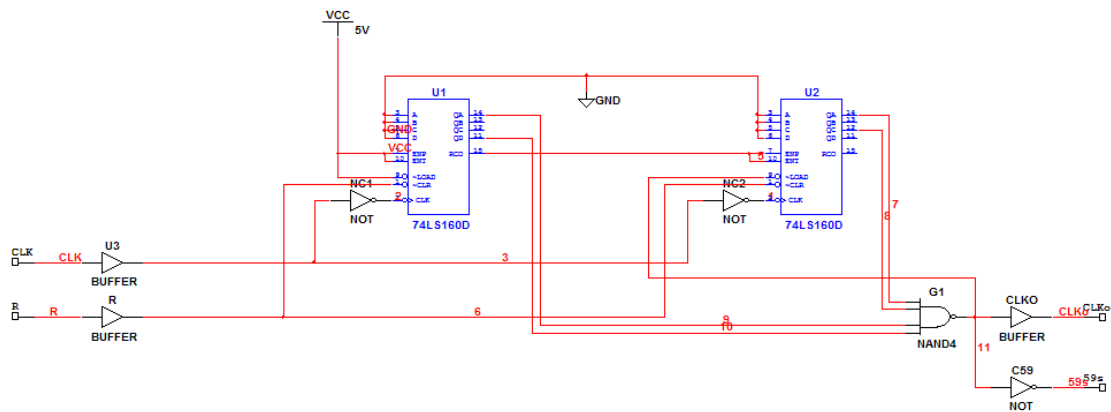
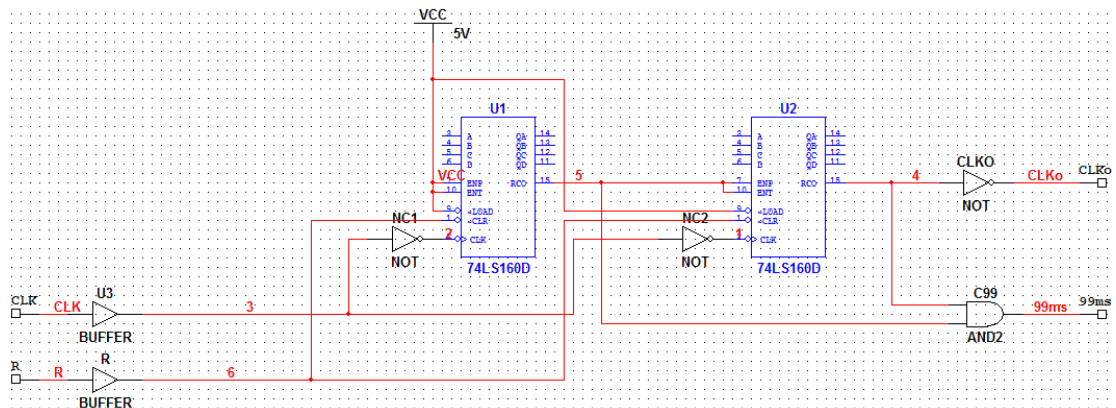
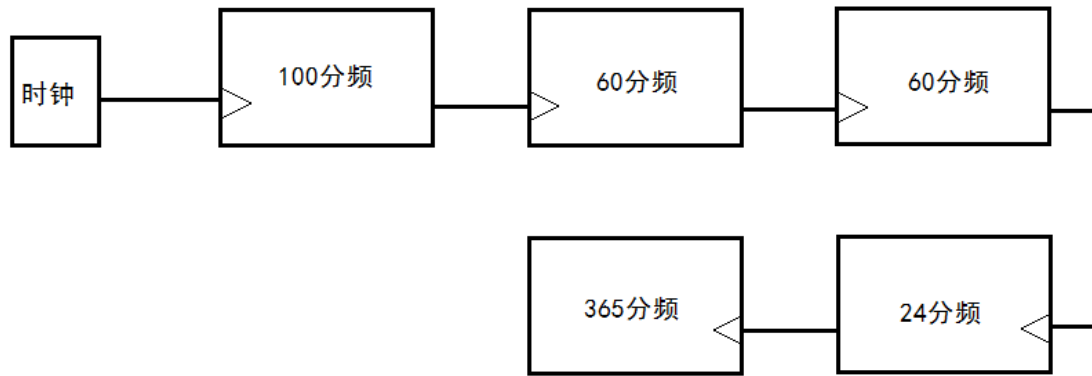
### (1) 主机的计时电路模块：

主机的计时电路负责进行计时，计时范围定为一年 365 天，每天的 0:00 发出发送启动信号。

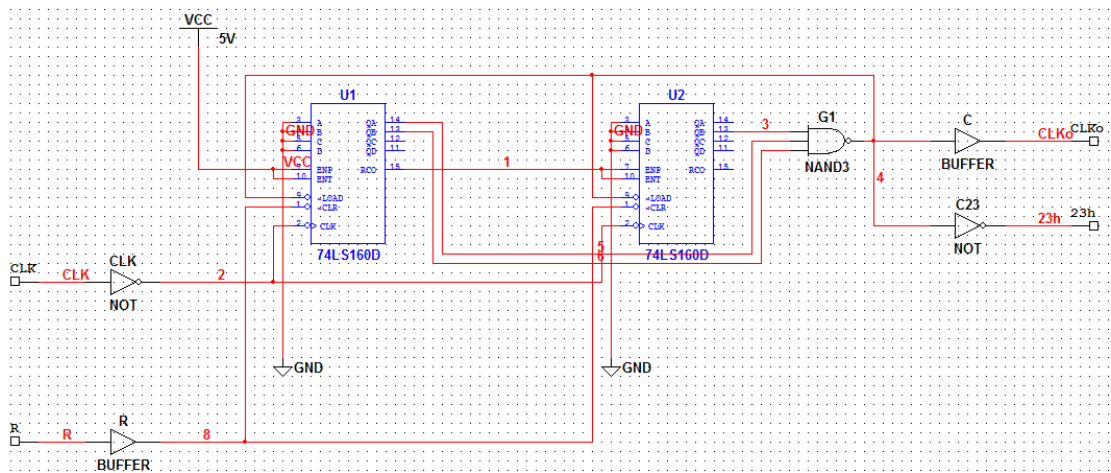
由于系统统一使用 100Hz 的时钟源，因此为了进行 365 天计时，可以分若干个级别对时钟信号进行分频：首先是 100 分频，得到周期为 1s 的信号；再进行两次 60 分频，分别得到周期为 1min 和 1h 的信号；之后进行 24 分频，得到周期为 1 天的信号；最后对周期为 1 天的信号进行计数，计数进制为 365，即可实现电路功能。

下面是电路的设计图，由于分频器的设计利用现成的计数器芯片，设计方法单一，比较简单，因此没有给出各个电路模块的具体设计电路，可以参考附件中 Multisim 10 各个仿真电路图。

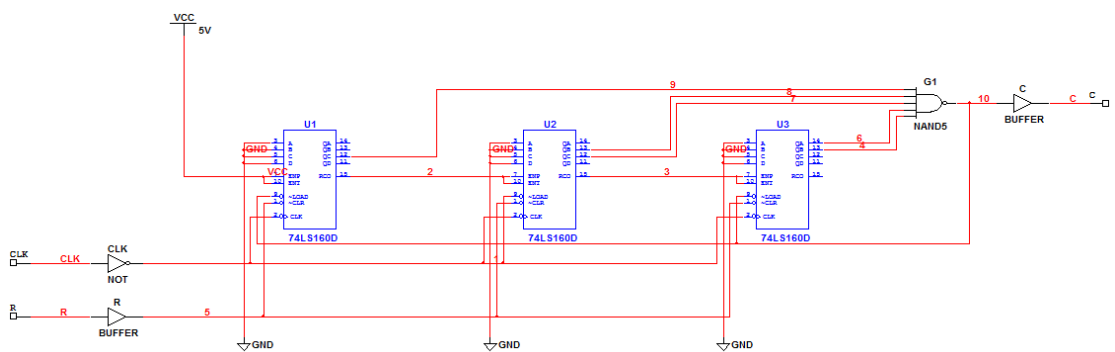
值得注意的是，由于主机的计时电路除了计时功能之外，还负责在特定时刻发出发送启动信号，而且对这个信号的要求是持续时间只能是 1 个时钟周期（即 0.01s），因此简单地使用 24 分频器的进位输出来充当这个启动信号是不行的，因为它的持续时间将长达 1h。这里选择使用各个分频器的进位临界信号做与操作，得到的结果作为启动信号。所谓临界指的是在分频器将要进位之前的那个计数值对应的译码信号，这样一来，启动信号的持续时间就是 23h59min59s990ms 开始到 24h00min00s000ms 的 10ms，即 0.01s，恰好可以满足要求。



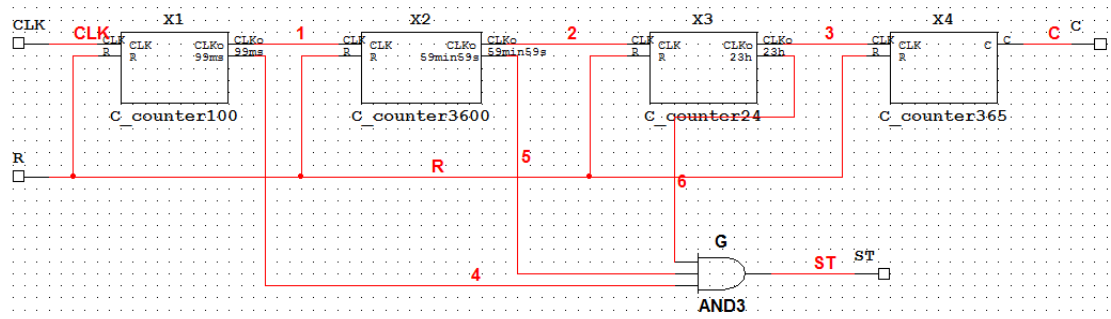




(d)



(e)



(f)

FIG14 主机计时电路的设计

(a) 是电路设计的总体结构图

(b) 到 (e) 分别是电路模块中 100、60、24 和 365 分频器的仿真电路图，注意到在仿真图中设置的临界信号读出端口 C99、C59 和 C23 是为产生发送启动信号 ST 准备的。

(f) 是计时电路的组装图，其中 ST 是发送启动信号，来自与门 G 对各个临界信号做与的结果，其中的 C\_counter3600 是把两个 60 分频器串联得到的 3600 分频器，其临界信号 C59min59s 是把两个 60 分频器的临界信号 C59 相与得到的。

(2) 控制信号发生器：

控制信号发生器的功能要求如下：在接到发送启动信号  $ST$  后，向串口发送模块发出发送控制信号  $TX$ ，同时向其并行输出从机选通控制信息，通过  $CB$  发送给从机，在接到从机在  $DB$  上发送的回馈信号后，再启动下一个从机的选通与接收过程，如此反复，直到完成全部从机的查询工作为止。在对每台从机进行查询的同时，选通存储器模块中与从机电能数据存储相对应的存储空间，把从机的反馈结果存储到其中更新电能数据。

先给出控制信号发生器所需要的状态转换图，如 FIG12 所示。

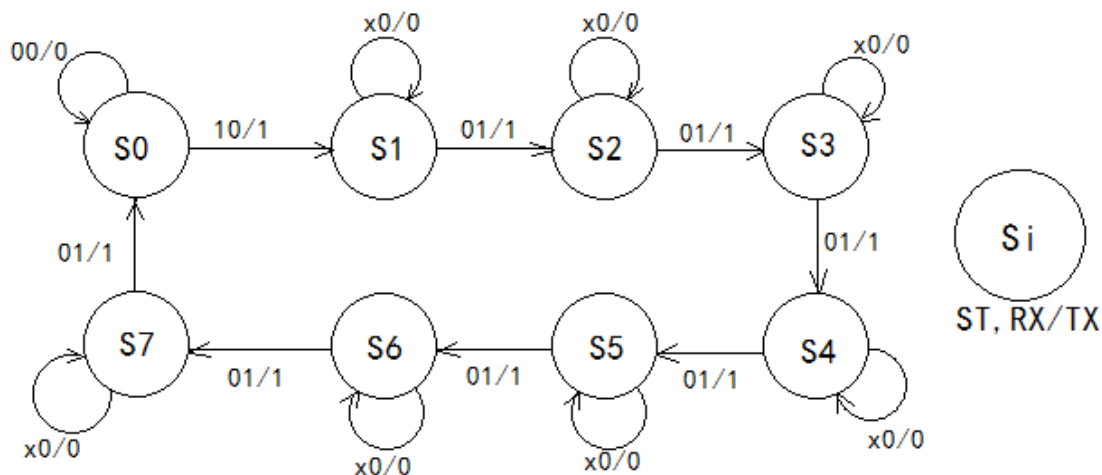


FIG15 主机控制信号发生器状态转换图

假如考虑使用 SMI 来实现电路设计，需要根据状态转换图抽象出电路的状态方程。首先，由于电路只有 8 个有效状态，可以考虑使用 3 个触发器构成主体，列出状态转换卡诺图。假设 3 个触发器的输出分别为  $Q0 \sim Q2$ ，那么电路里有  $Q0 \sim Q2$ 、 $ST$  和  $RX$ ，5 个输入变量，有一个输出变量  $TX$ 。如果把输入变量全部列在状态转换卡诺图中的话，就是 5 变量卡诺图，不光画起来很麻烦，而且化简等操作也比较不习惯。因此，这里选择把输入变量  $ST$  单独拿出来，由于  $ST$  对电路的状态转换的影响相对是最少的，因此可以分别对  $ST=0$  和  $1$  两种情况列出卡诺图，分别化简得到  $Q^* = F_0(Q)$  和  $Q^* = F_1(Q)$  两个状态方程，然后用  $Q^* = F(Q) = ST' \cdot F_0(Q) + ST \cdot F_1(Q)$  得到完整的状态方程。

		Q1Q0			
		00	01	11	10
RXQ2	00	000	001	011	010
	01	100	101	111	110
	11	101	110	000	111
	10	xxx	010	100	011

(a)

		Q1Q0			
		00	01	11	10
RXQ2	00	001	001	011	010
	01	100	101	111	110
	11	101	110	000	111
	10	xxx	010	100	011

(b)

FIG16 控制信号发生器电路的状态转换卡诺图，其中(a)是 ST=0 时的结果，(b)是 ST=1 时的结果。

从卡诺图分别写出两种状况下的状态方程，可以得到：

$$\begin{cases} Q_2^* = (RXQ_1Q_0)Q_2 + (RXQ_1Q_0)'Q_2 \\ Q_1^* = (RXQ_0)Q_1 + (RXQ_0)'Q_1 \\ Q_0^* = RXQ_0 + RX'Q_0 \end{cases} \quad (ST=0)$$

以及：

$$\begin{cases} Q_2^* = (RX'Q_1'Q_0 + RXQ_1Q_0)Q_2' + (RXQ_1Q_0)'Q_2 \\ Q_1^* = (RXQ_0)Q_1' + (RXQ_0)'Q_1 \\ Q_0^* = RXQ_0' + RX'Q_0 \end{cases} \quad (ST=1)$$

由此可得：

$$\begin{cases} Q_2^* = (ST \cdot RX'Q_1'Q_0 + RXQ_1Q_0)Q_2' + (RXQ_1Q_0)'Q_2 \\ Q_1^* = (RXQ_0)Q_1' + (RXQ_0)'Q_1 \\ Q_0^* = RXQ_0' + RX'Q_0 \end{cases}$$

同时，容易知道，电路的输出方程为  $TX = RX + Q_2'Q_1'Q_0' \cdot ST$

根据状态方程的形式可知，使用 JK 触发器比较方便，因此可以选用 3 个 JK 触发器来搭建电路，电路设计如下图。

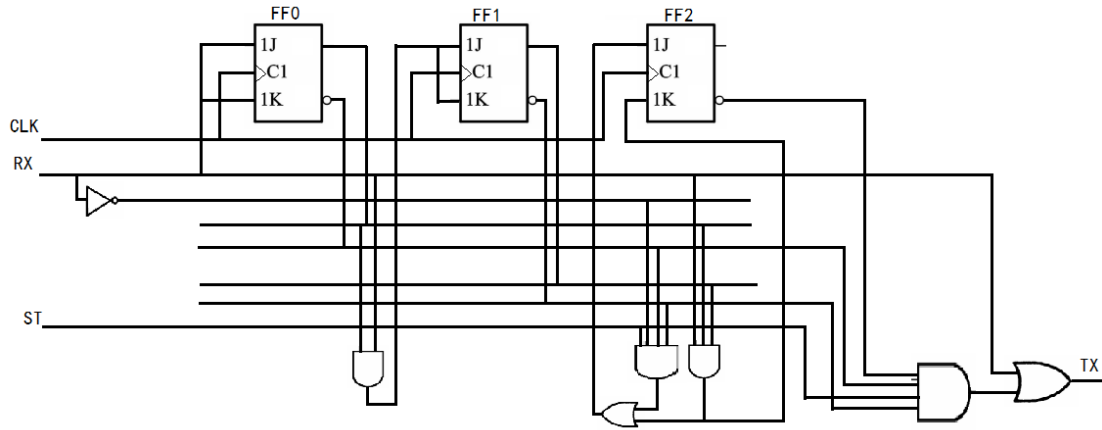


FIG17 主机控制信号发生器的电路设计图

如本段开头所述，控制信号发生器除了负责根据反馈信息发出发送控制信号控制整个查询流程之外，还负责提供主机串口发送模块要发送到控制信息，也就是各个从机的编号信息，同时还要负责给存储器模块发出寻址控制信号，以便把从机发来的电能数据存储起来，这两部分功能可以在控制信号发生电路的基础上附加一些模块来实现。比如说，从机的编号可以用 3 线-8 线译码器，把触发器的计数值  $Q_2 \sim Q_0$  进行译码得到，而存储器模块的寻址信号可以直接利用  $Q_2 \sim Q_0$  来充当。这样得到的完整电路仿真图如 FIG15 所示。

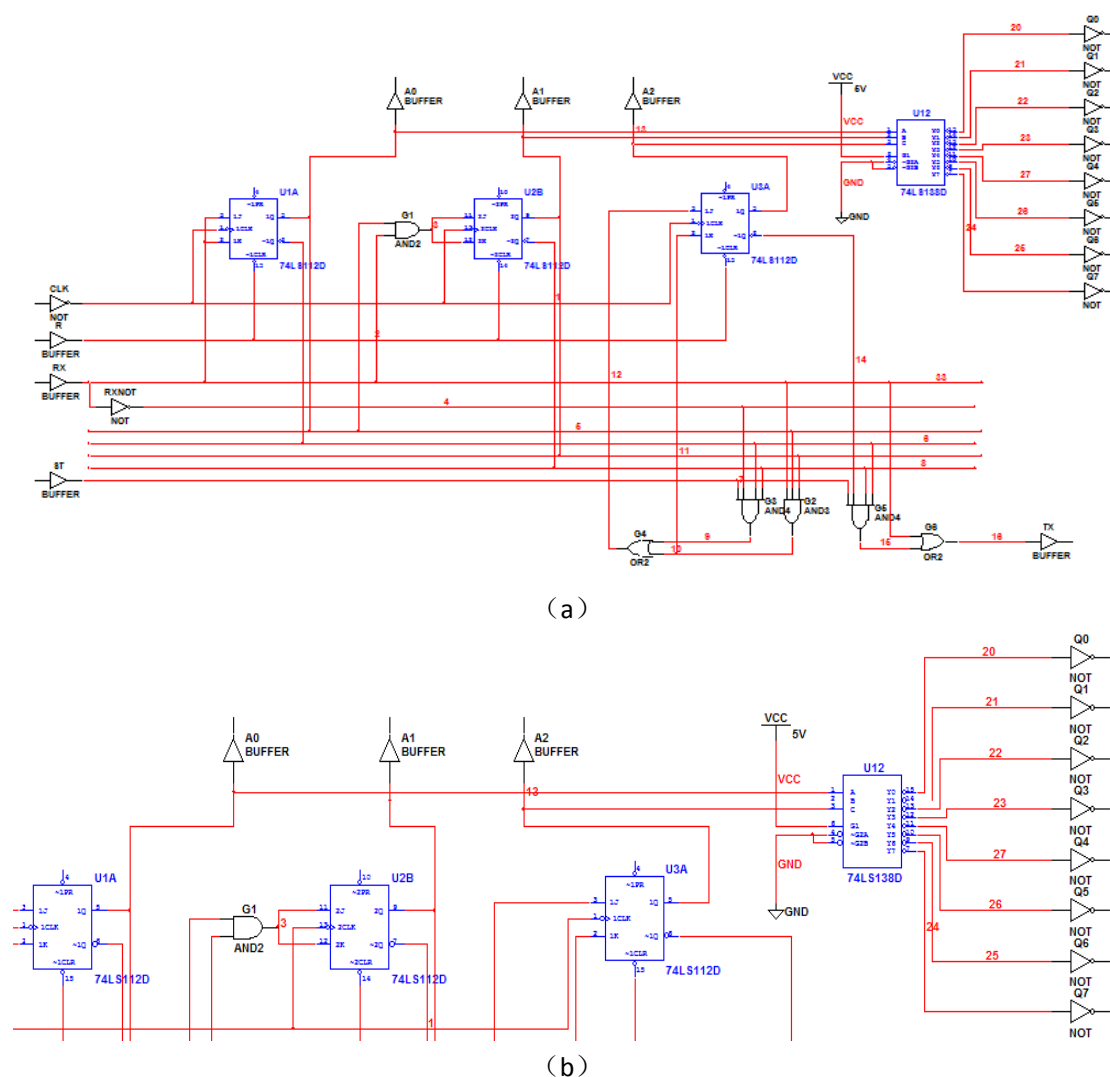
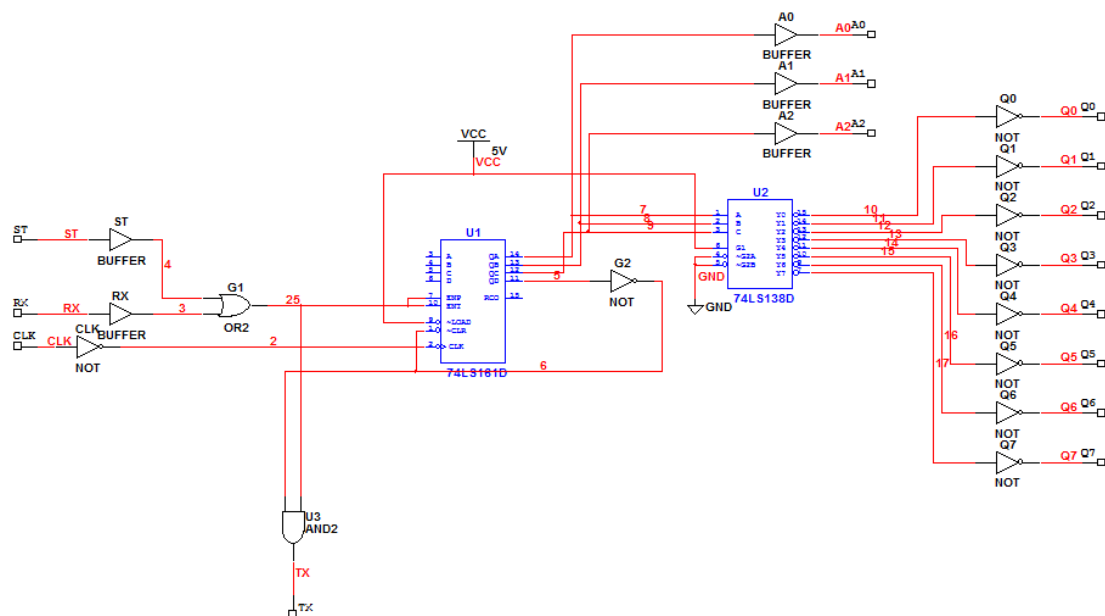
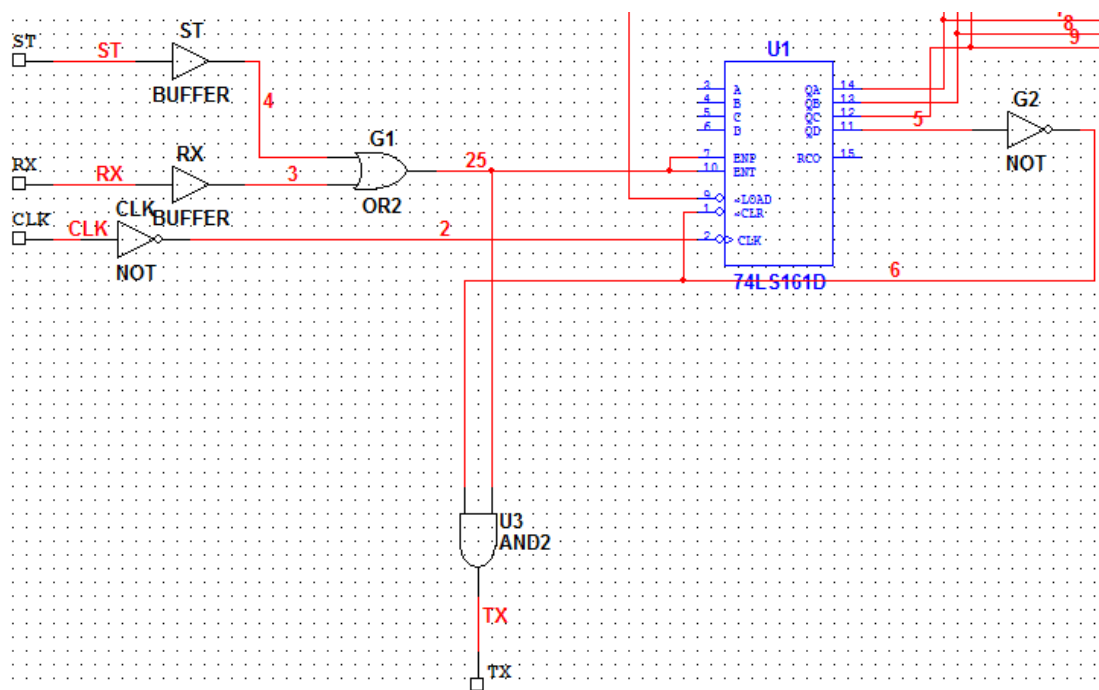


FIG18 主机控制信号发生器电路的完整设计，其中（a）是电路全图（A2~A0 是存储器模块所需的寻址控制信号；Q7~Q0 是发给串口发送模块的从机控制信号；TX 是发给串口发送模块的发送控制信号；R 是复位信号输入端），（b）是存储器寻址信号和从机控制信号的产生电路的局部放大图。

当然，除了使用 SSI 来搭建电路之外，还可以考虑使用 MSI 来实现，这样可以省去求解状态方程、化简卡诺图的麻烦，但是需要对各个常用 MSI 产品的功能有相当的熟悉程度，下面给出了用 MSI 芯片 74LS161 为主体设计的另一种电路，同样可以完成设计的电路功能。



(a)



(b)

FIG19 主机控制信号发生电路的 MSI 设计图，其中 (a) 是整体设计图，各个输入输出变量的定义同 SSI 设计；(b) 是控制信号发生部分即主体部分的设计电路局部放大图。

这里采用的设计思路可以概述如下：首先，观察电路的功能图可见，只有当 ST 或 RX 至少有一者为 1 时，电路的状态才发生转换，联想到计数器的特点，可以考虑把 ST+RX 作为计数器的计数使能信号，这样可以确保在没有接到 RX 时计数器不发生状态转换；其次，要求电路在从状态 0 转换到状态 7 时，接下来再次转换状态时回到状态 0，因此电路主体应当是一个 8 进制的计数器；最后，TX 信号总是在电路进行状态转换时等于 1，状态保持时等于 0，而在状态归零时回复 0，因此可以把 TX 取为计数器计数使能信号和归零信号函数，由此就可以得到上面的电路图了。

最后是主机的存储器模块的设计，存储器模块负责存储主机收到的从机采集数据，并可以在需要时把这些数据调用出来向电力管理部门发送等操作，这里只设计了主机控制存储器的读写电路，所用的存储器是 Multisim 10 中的 2K\*8bit 的 RAM 芯片。电路的读写控制通过三态门构成，仿真电路如 FIG16 所示。

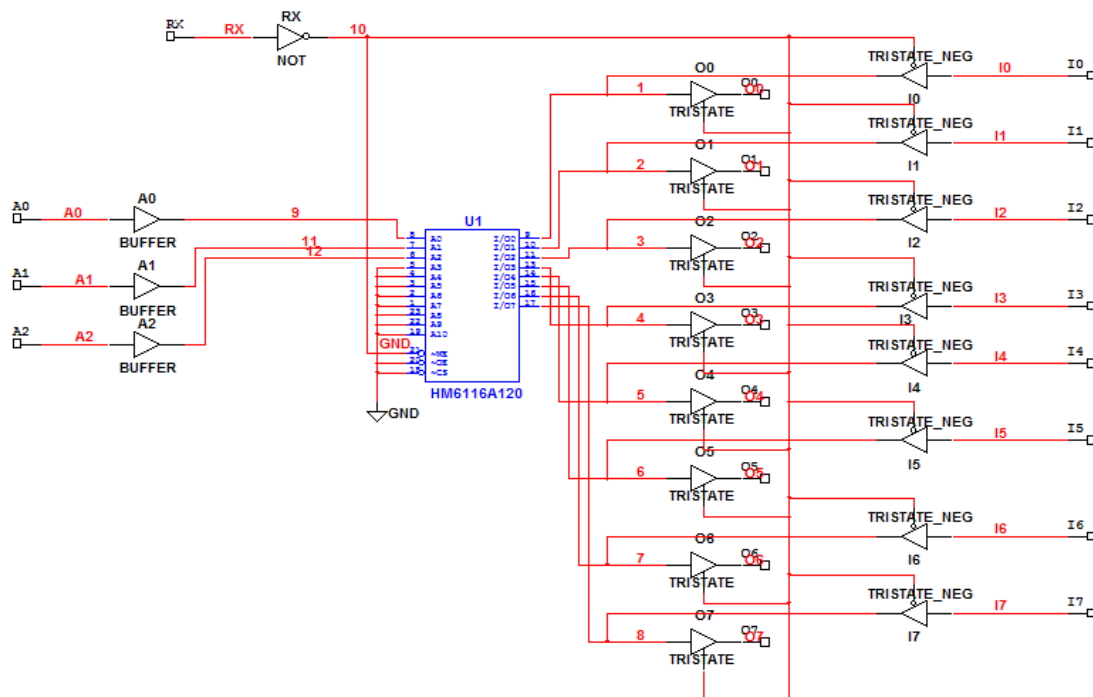


FIG20 主机存储器模块的电路图

电路的总体设计完成了,纵观整个设计过程,所使用的器件和知识没有超出课堂上学习的内容,但是通过结合实际应用进行设计,把学习到的知识相互综合起来应用,却能得到一个几乎可以用于实际之中的实用电路,不得不赞叹数字电路设计技术的强大。

总结一下在设计过程中得到的收获,最大的感受就是数字电路的设计方法多种多样,虽然都可以达到目的,但是不同的方法有不同的适用之处,要谨慎选择。举个例子,在设计电路时,一方面可以使用 SSI 器件,用各种基本的门电路和触发器来搭建,另一方面可以使用 MSI 器件,用各种成型的产品芯片来搭建。这两种方法各有优劣,使用 SSI 器件,假如设计合理,电路可以很精巧,而且能够减少无效态或者裁减掉不少无用部分,成本也较低,但是需要付出的代价就是设计比较复杂,需要对电路的基本工作流程进行抽象,还要做很多的化简工作,以及结合已有的 SSI 芯片对逻辑函数式作必要的形式转化等,而且有的时候由于化简工作没有做好或者形式转化不当,还会增加所用芯片的种类和数量;而使用 MSI 芯片来实现,电路的逻辑可以非常直观地展现在设计者和用户面前,设计起来比较方便,例如设计串行数据的起始位检测电路时,使用移位寄存器来检测就比使用门电路和触发器来检测更容易设计,电路也很简洁明了。但是使用 MSI 要求对各种芯片产品的功能和使用方法有深入的了解,需要深厚的经验和知识储备,而且使用的 MSI 芯片不肯能是完全为了特定的设计而生产的,总会有一些功能和状态未用到,产生无效态问题,也需要妥善解决。这两种设计方式的对比相信可以从本文中所

讲的主机控制信号发生器电路的两种设计方法的对比中看到。

再者，即便是一种设计方法，在实际搭建电路时也有很多具体的细节需要考虑，主要就是器件的种类选择，比如同一个逻辑函数，可以用与非门来实现，也可以用或非门等其他门电路实现，在函数化简时就有简洁与繁琐之分，需要谨慎选择；再比如同样使用 SSI 芯片触发器搭建计数器电路，使用 JK 触发器和 D 触发器，效果就会与不同，有些时候由于电路的状态方程符合 JK 触发器的特性方程，选用 JK 触发器就比较方便，使用 D 触发器就比较繁琐，而有时却又相反，这也是设计时需要考虑的问题。

总的来说，这样的设计过程确实教给了我很多课本上没有的或者未深入说明的知识，使我受益匪浅。

但是，如前所述，由于水平所限，自己设计的电路依然存在不少瑕疵，比如说，在设计过程中几乎没有考虑时序电路的竞争冒险现象，因此得到的电路只是在理想的时序逻辑下才能够正常工作，在实际中，如此众多的芯片和门级联在一起，不同来源的信号延时肯定是不可忽略的，很有可能就会产生诸多的时序错误，这是受限于所学知识而造成的，但同时也让我们看到了一个实用的时序逻辑电路的设计是多么的不易，至少，假如让我们仔细分析电路每条通道的信号的传输延时和竞争冒险现象，恐怕是不敢想象的大工程。

再者，电路的容错性能（或者说鲁棒性）不是很好，一方面，通过 MSI 设计电路，不可避免地会出现很多的无效态和无关项，如何妥善地处理这些状态就是一个很大的问题，比如说在正常的时序逻辑中，串口发送信号 TX 只需要持续一个时钟周期，但是假如考虑到传输延迟和干扰噪声等因素，有可能 TX 的持续时间更长或者在不该出现时产生了一个短脉冲，而这样的状态很有可能就会引起错误的操作，如何避免这样的问题又是一门很大的学问，必须承认，在这一方面本电路的设计还做得不够。另一方面，对于通信系统来说，干扰和噪声几乎就是与生俱来的，因此传输线上的电平不可能总是完美的矩形波，空闲电位不可能总是 0，不可能期望在传输中没有任何干扰信号，也不敢保证不会产生突然的脉冲干扰（比如雷电等的影响），这些因素都要求通信系统能够充分屏蔽这些干扰信号，做到通信的可靠性，最不济也要能够及时发现错误，以便为后续的纠错或再发送提供信息。参考了计算机内部使用的 UART 模块的原理和结构后，可以发现串口通信通常是通过信号的每一位进行多次测量，最后表决确定信号电平的方法排除意外干扰，并且设置了奇偶校验功能来对收到的信息进行校验。由于对这方面知识的欠缺和能力所限，本次设计中没有考虑这些功能，是一大遗憾，也是日后需要改进之处。

最后，还是要说，在这样的课程设计过程中我学到了很多，对数电的基本设计方法有了更深入的了解，也培养了自己的动手能力和独立思考的能力，另外还要在学期结束之时感谢老师和助教一学期来对我们的帮助和指导，让我们一学期的学习画上了圆满的句号。