电子线路设计与测试（二）

有限状态机与数字钟

(1) P241 步进电机脉冲分配器设计

(2) P250 多功能数字钟电路设计

有限状态机与数字钟

P241实验任务1

步进电机脉冲分配器设计

p209/250设计课题1

多功能数字钟设计

- 能显示小时、分钟、秒钟（时、分用显示器，秒用LED）
- 能调整小时、分钟的时间

检查《电子线路设计、测试与实验（二）》单元八测验结果

提高：

（1）任意闹钟

（2）小时实现12/24进制可切换

（3）报正点数（几点钟LED闪烁几下）

学习要求

- 掌握用verilog HDL描述数字逻辑电路与系统的方法;
- 有限状态机概念;
- 掌握用verilog HDL描述有限状态机的方法
- 掌握分层次电路设计方法;
- 熟练掌握数字钟的设计与调试方法

（一）有限状态机建模（P239）

◆一般情况下，触发器的数量是有限的，其状态数也是有限的，故称为有限状态机（Finite State Machine，简称为FSM）。

◆状态机中所有触发器的时钟输入端被连接到一个公共时钟脉冲源上，其状态的转换是在同一时钟源的同一脉冲边沿同步进行的，所以它也被称作时钟同步状态机。

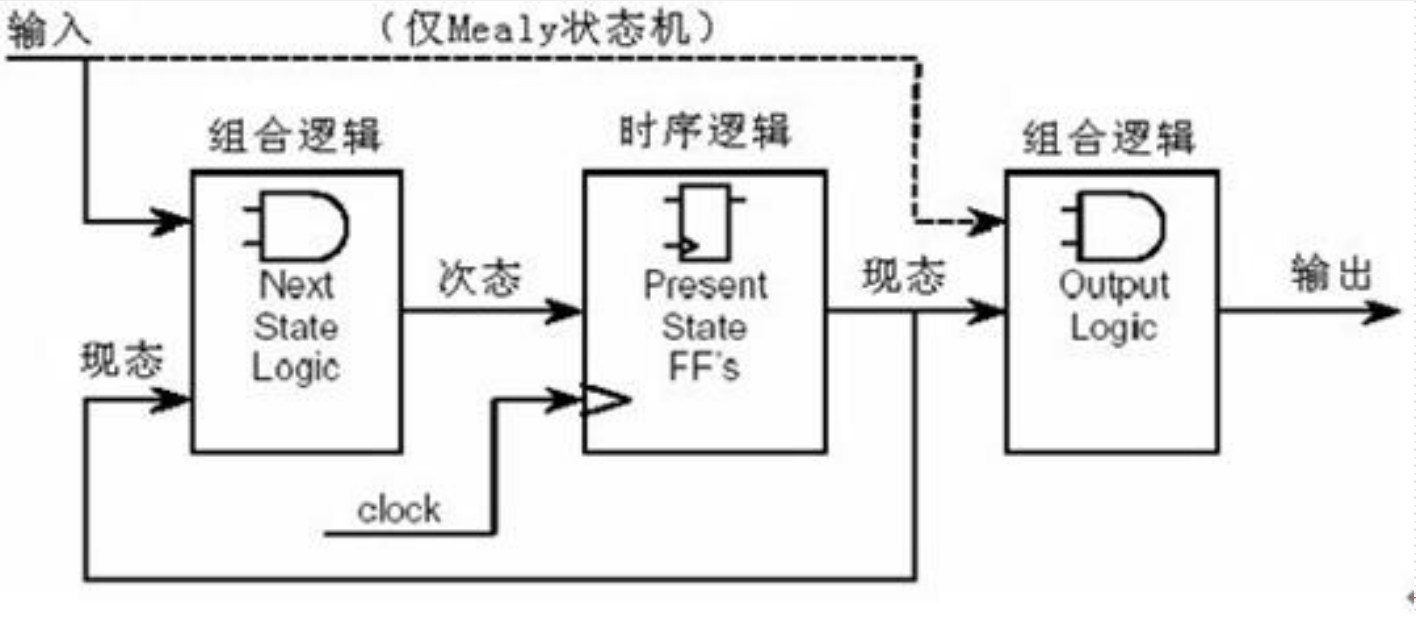
（一）有限状态机建模（P239）

有限状态机的标准模型如图，由三部分组成：

(1) 次态组合逻辑电路，

(2) 存储当前状态的时序逻辑电路，

(3) 输出组合逻辑电路。



（一）有限状态机建模（P239）

根据电路的输出信号是否与电路的输入有关，可以将状态机分为两种类型：

◆米利型（Mealy）状态机，电路的输出信号不仅与电路当前的状态有关，还与电路的输入有关；

◆穆尔型（Moore）状态机，电路输出仅仅取决于各触发器的状态，而不受电路当时的输入信号影响或没有输入变量。

（一）有限状态机建模（P239）

例：设计一个序列检测器电路。

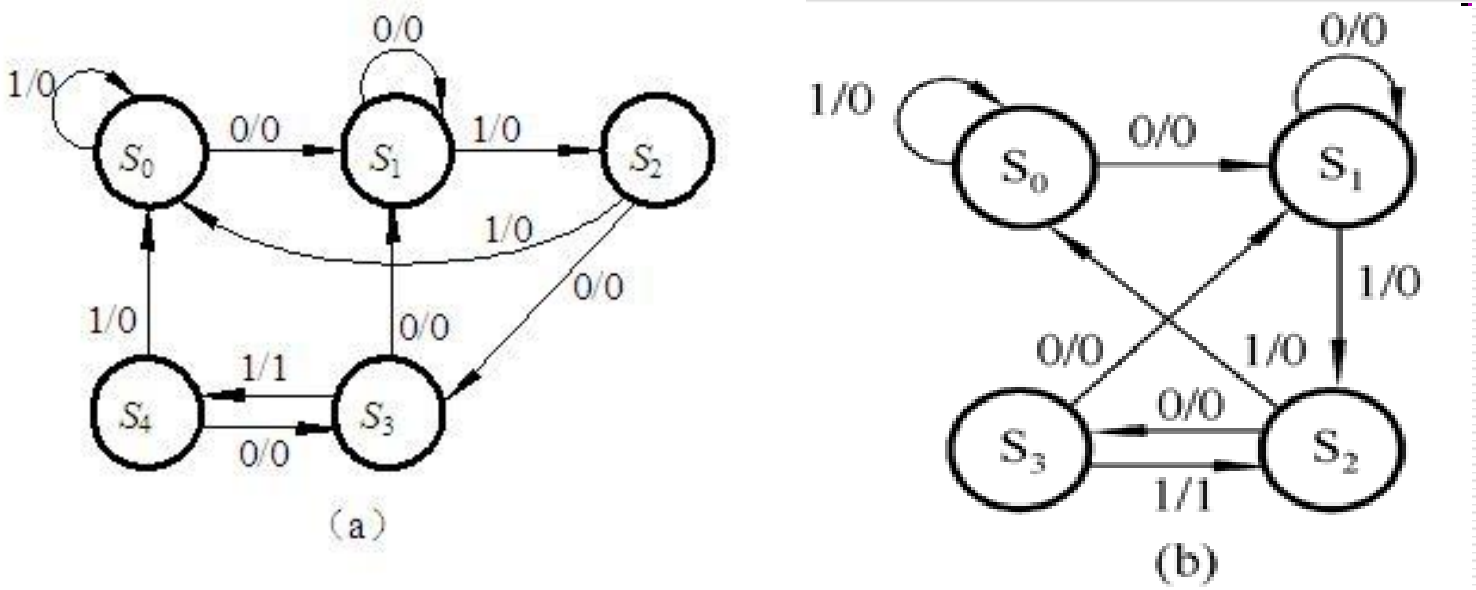
功能是检测出串行输入数据*Sin*中的4位二进制序列0101（自左至右输入），当检测到该序列时，输出*Out*=1；没有检测到该序列时，输出*Out*=0。（注意考虑序列重叠的可能性，如010101，相当于出现两个0101序列）。

（一）有限状态机建模（P239）

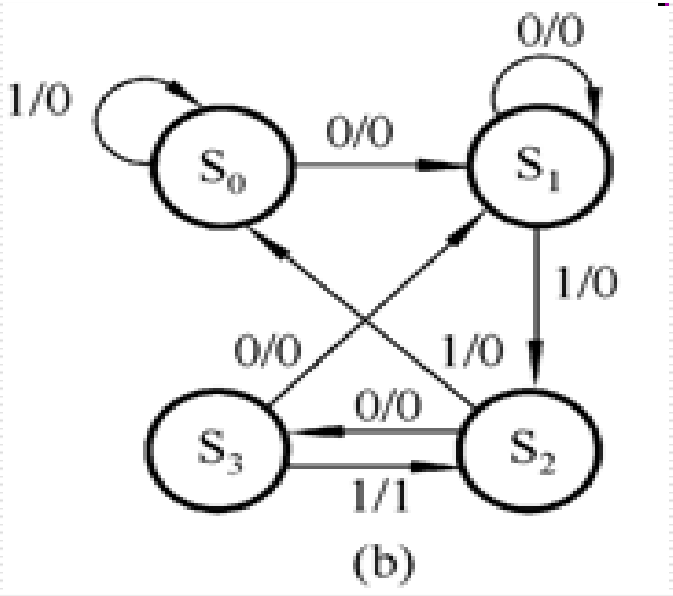
解：首先，确定采用米利型状态机设计该电路。因为该电路在连续收到信号0101时，输出为1，其他情况下输出为0，所以采用米利型状态机。

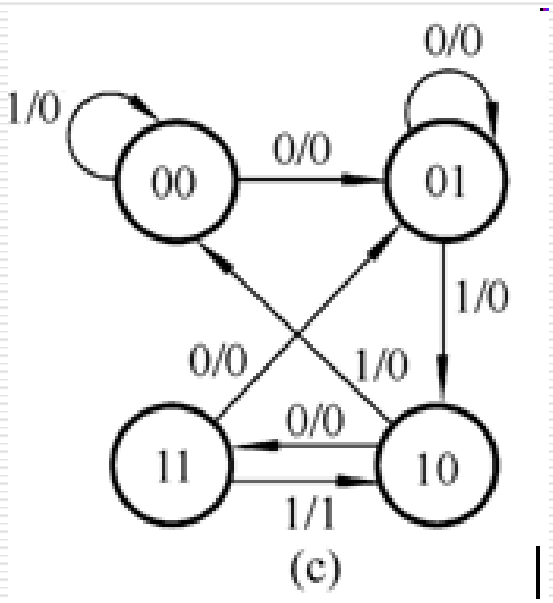
其次，确定状态机的状态图。根据设计要求，该电路至少应有四个状态，分别用S_1、S_2、S_3、S_4表示。若假设电路的初始状态用S_0表示，则可用五个状态来描述该电路。根据分析，可以画出图(a)所示的原始状态图。

（一）有限状态机建模（P239）



观察该图可以看出，S_2、S_4为等价状态，可用S_2代替S_4，于是得到简化状态图，如图(b)所示。


0/0
1/0
0/0
S_0
S_1
1/0
0/0
1/0
0/0
S_3
S_2
1/1
(b)


0/0
1/0
0/0
00
01
1/0
0/0
1/0
0/0
11
10
1/1
(c)

状态编码，如图(c)所示。

状态图描述方法

主要包含四部分内容:

1. 利用parameter 描述状态机中各个状态的名称，并指定状态编码。例如，对序列检测器的状态分配使用最简单的自然二进制码，其描述如下:

```
parameter  S0=2'b00, S1=2'b01, S2 = 2'b10, S3 = 2'b11;
```

2. 用always 块描述状态触发器实现状态存储。

3. 使用敏感表和case语句(也可以采用if-else等价语句)描述的状态转换逻辑。

4. 描述状态机的输出逻辑。

状态图描述方法

用两个always块对该例的状态机进行描述，其代码如下：

```
module  Detector ( Sin, CP, nCR, Out) ;
    input Sin, CP, nCR;   //声明输入
    output reg Out ;        //声明输出
    reg [ 1 : 0 ] Current_state, Next_state;
parameter  S0=2'b00, S1=2'b01, S2 = 2'b10, S3 = 2'b11;

always @(posedge CP or negedge nCR )  //状态转换
   begin
         if (~nCR)
               Current_state <= S0;              //异步清零
         else
               Current_state <= next_state;   //触发器翻转
    end
```

```
always @( Current_state or Sin)
   begin
    case(Current_state )
     S0:   begin Out =1'b0;
                  Next_state = (Sin==1)? S0 : S1;   end
     S1:   begin Out =1'b0;
                  Next_state = (Sin==1)? S2 : S1;   end
     S2:   begin Out =1'b0;
                  Next_state = (Sin==1)? S0 : S3;   end
     S3:   if (Sin==1)
                  begin Out =1'b1; Next_state = S2; end
           else
                  begin Out =1'b0; Next_state = S1; end
     endcase
   end
endmodule
```

有限状态机实验任务

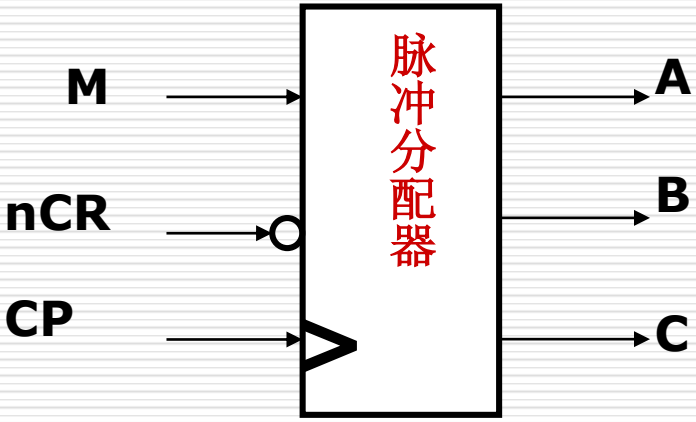
步进电机脉冲分配器设计

试用 Verilog HDL设计一个能够自启动、具有正反转功能的三相六拍步进电机脉冲分配器电路。

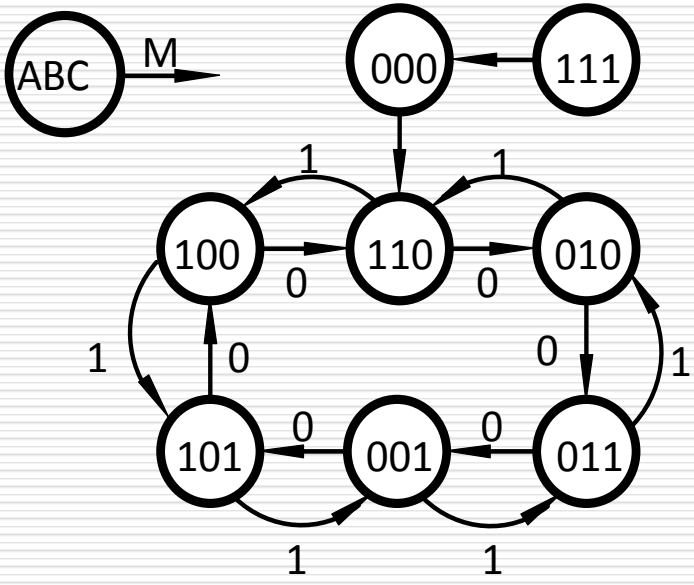
此电路的状态转换图如图7.3.3所示。

当M=0时，按顺时针方向转；

当M=1时，则按反时针方向转。



设计框图



有限状态机实验任务

步进电机脉冲分配器设计

设计步骤与要求:

①创建子目录E:\EDA_Lab\Lab5，并新建一个工程项目。

② 使用Verilog HDL设计电路，并进行仿真分析。

③ 用FPGA开发板实现步进电机脉冲分配器，并实际测试逻辑功能。（A、B、C用发光二极管代替）

④ 根据实验流程和实验结果，写出实验总结报告，并对波形图和实验现象进行说明。

（二）多功能数字钟设计

实验内容要求（P250）

□ 基本功能

- 准确计时，以数字形式（十二进制）显示时、分、秒的时间(时、分用数码管、秒用LED显示，用LED区分上午和下午）
- 校正时间：可对时、分进行调整

（二）多功能数字钟设计

实验内容要求（P250）

提高：

- （1）任意闹钟；
- （2）小时为12/24进制可切换
- （3）报正点数（几点钟LED闪烁几下）

（二）多功能数字钟设计

为什么要采用层次化、模块化设计方法？

- “分而治之”
- 对于一个复杂的数字系统，运用层次化设计方法，使设计课题进一步细化，分块设计，条理清晰。另外，在调试时可采用逆向调试方式，即从模块调试向总体调试方向开展调试工作，使设计中出现的问题在模块级就能发现，及时处理，这样就会使一个复杂的设计变得容易调试，缩短了设计时间。

（二）多功能数字钟设计

- 如何采用层次化设计方法？

自顶向下　　自底向上

自顶向下---先设计顶层总框图，该框图由若干个具有特定功能的源模块组成。下一步针对这些具有不同功能的模块进行设计，对于有些功能复杂的模块，还可以将该模块继续化分为若干个功能子模块，这样就形成模块套模块的层次化设计方法。

Verilog HDL结构描述

基本门的调用

门名 调用名（端口名表项）

门名：**14**种基本门级元件

模块的调用

结构化描述方式

模块的调用

基本方式： 模块名　调用名（端口名表项）

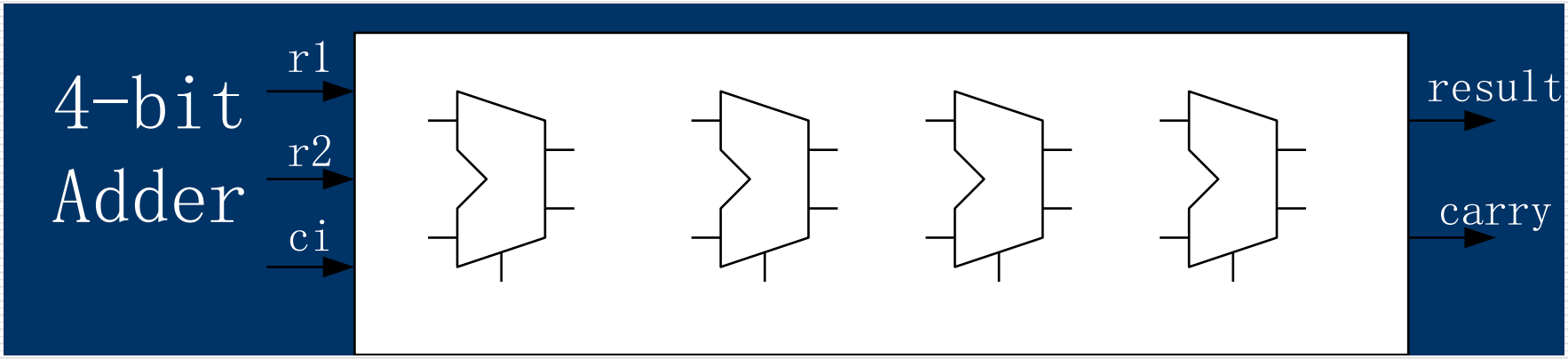
调用方式一：位置对应调用方式

调用方式二：端口名对应调用方式

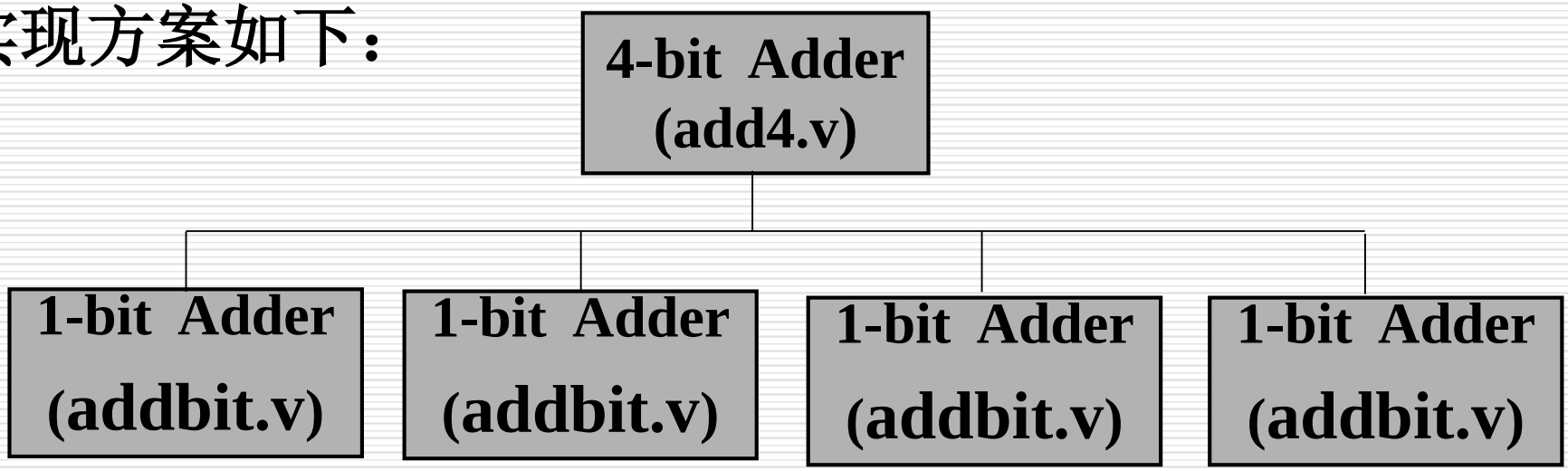
调用方式三：存在不连接端口的调用方式

层次化设计方法举例

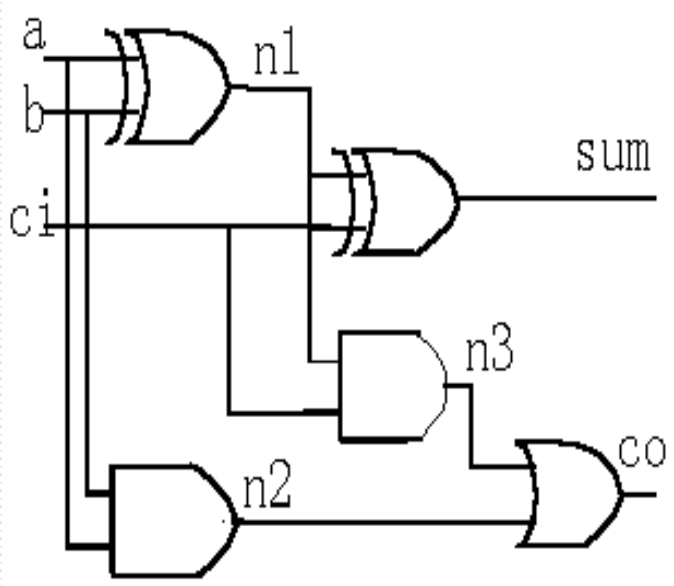
例 请用层次化的方法设计一个4位全加器，框图如下：



实现方案如下：



1. 底层模块——1位全加器实例：




```
module addbit (a, b, ci, sum, co);
input   a,  b,  ci;
output  sum,  co;
wire    a, b, ci, sum, co, n1, n2, n3;

    xor    (n1, a, b,);
    xor    (sum, n1, ci);
    and    (n2, a, b);
    and    (n3, n1, ci);
    or     (co, n2, n3);

endmodule
```

一些Verilog原型(Primitive)

列出结构化的元件

并按网表连接

2. 顶层模块调用底层模块实例 – 通过位置关联

Order must match exactly

```
models  add4  (result, carry, r1, r2, ci);
   output  [3:0]  result;
   output         carry;
   input   [3:0]  r1, r2;
   input          ci;
   wire    [3:0]  r1, r2, result;
   wire     ci, carry, c1, c2, c3;
    addbit  u1 (r1[0], r2[0], ci, result[0], c1);
    addbit  u2 (r1[1], r2[1], c1, result[1], c2);
    addbit  u3 (r1[2], r2[2], c2, result[2], c3);
    addbit  u4 (r1[3], r2[3], c3, result[3], carry);
endmodule
```

```
module addbit  (a, b, ci, sum,co);
      input   a, b, ci;
      output   sum, co;

      Structural or behavioral
      model

endmodule
```

3. 顶层模块调用底层模块实例 – 通过名字关联

```
module add4 (result, carry, r1, r2, ci);
   output  [3:0]    result;
   output  carry;
   input   [3:0] r1, r2;
   input   ci;
   wire    [3:0] r1, r2 ,   result;
   wire    ci, carry, c1, c2 c3;
   addbit u0 (.co(c1)    , .sum(result[0]), .ci(ci),.b(r2[0]),.a(r1[0]));
   addbit u1 (.co(c2)    , .sum(result[1]), .ci(c1),.b(r2[1]),.a(r1[1]));
   addbit u2 (.co(c3)    , .sum(result[2]), .ci(c2),.b(r2[2]),.a(r1[2]));
   addbit u3 (.co(carry), .sum(result[3]), .ci(c3),.b(r2[3]),.a(r1[3]));
endmodule
```

here names must match exactly

注意：该描述应严格保持名字的一致！

层次化设计举例 —100进制计数器

由10进制计数器构成100进制计数器

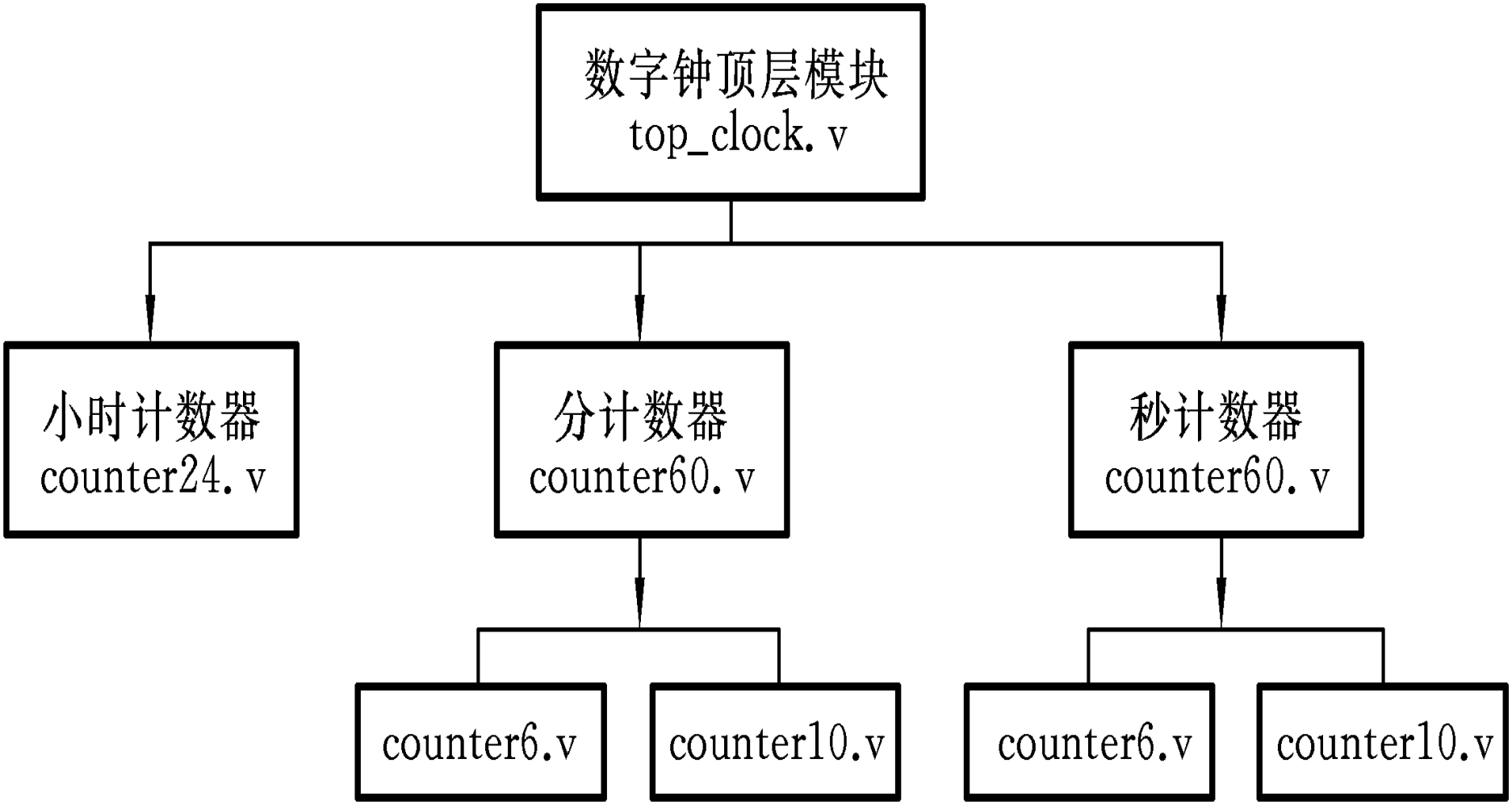
```
//********************* counter100.v (BCD: 00~59)**************
//100进制计数器：调用10进制模块两次
module counter100(Cnt, nCR, EN, CP);
   input CP, nCR, EN;
   output [7:0] Cnt;       //模60计数器的输出信号
  wire  [7:0] Cnt;         //输出为8421 BCD码
   wire  ENP;              //计数器十位的使能信号（中间变量）

 counter10 UC0 (Cnt[3:0], nCR, EN, CP);  //计数器的个位
 counter10 UC1 (Cnt[7:4], nCR, ENP, CP); //计数器的十位
 assign  ENP = (Cnt[3:0]==4'h9);  //产生计数器十位的使能信号

endmodule
```

层次化设计举例（续） —100进制计数器

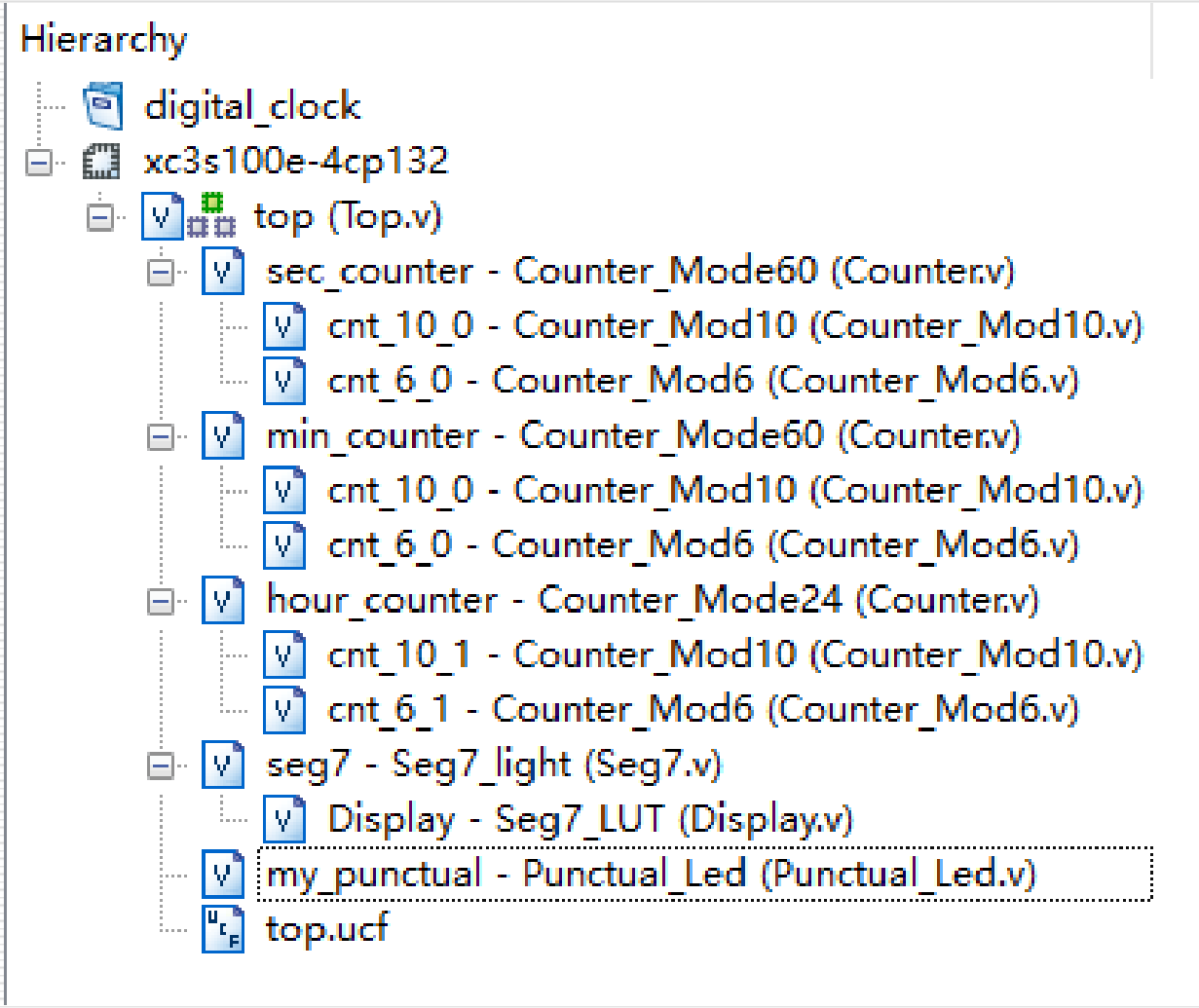
由10进制计数器构成100进制计数器

```
//******************* counter10.v ( BCD: 0~9 ) ****************
module counter10(Q, nCR, EN, CP);
   input CP, nCR, EN;
   output [3:0]       Q;
   reg    [3:0]       Q;
   always @(posedge CP or negedge nCR)
   begin
     if(~nCR)    Q <= 4'b0000;        // nCR＝0，计数器被异步清零
     else if(~EN)  Q <= Q;             //EN=0,暂停计数
     else if(Q == 4'b1001)  Q <= 4'b0000;
     else        Q <= Q + 1'b1;      //计数器增1计数
   end
endmodule
```

数字钟主体计时模块框图举例



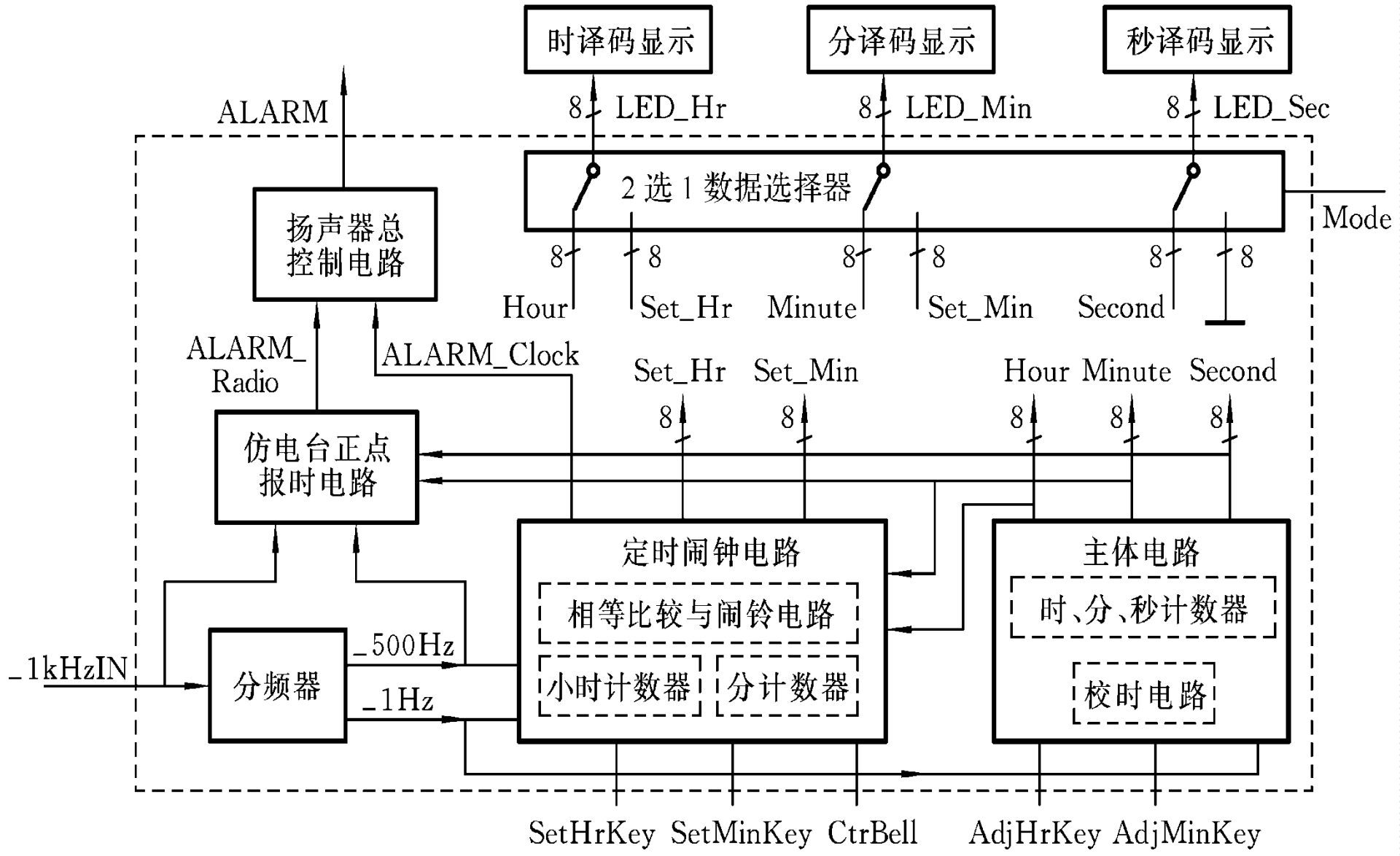
数字钟主体计时模块框图举例



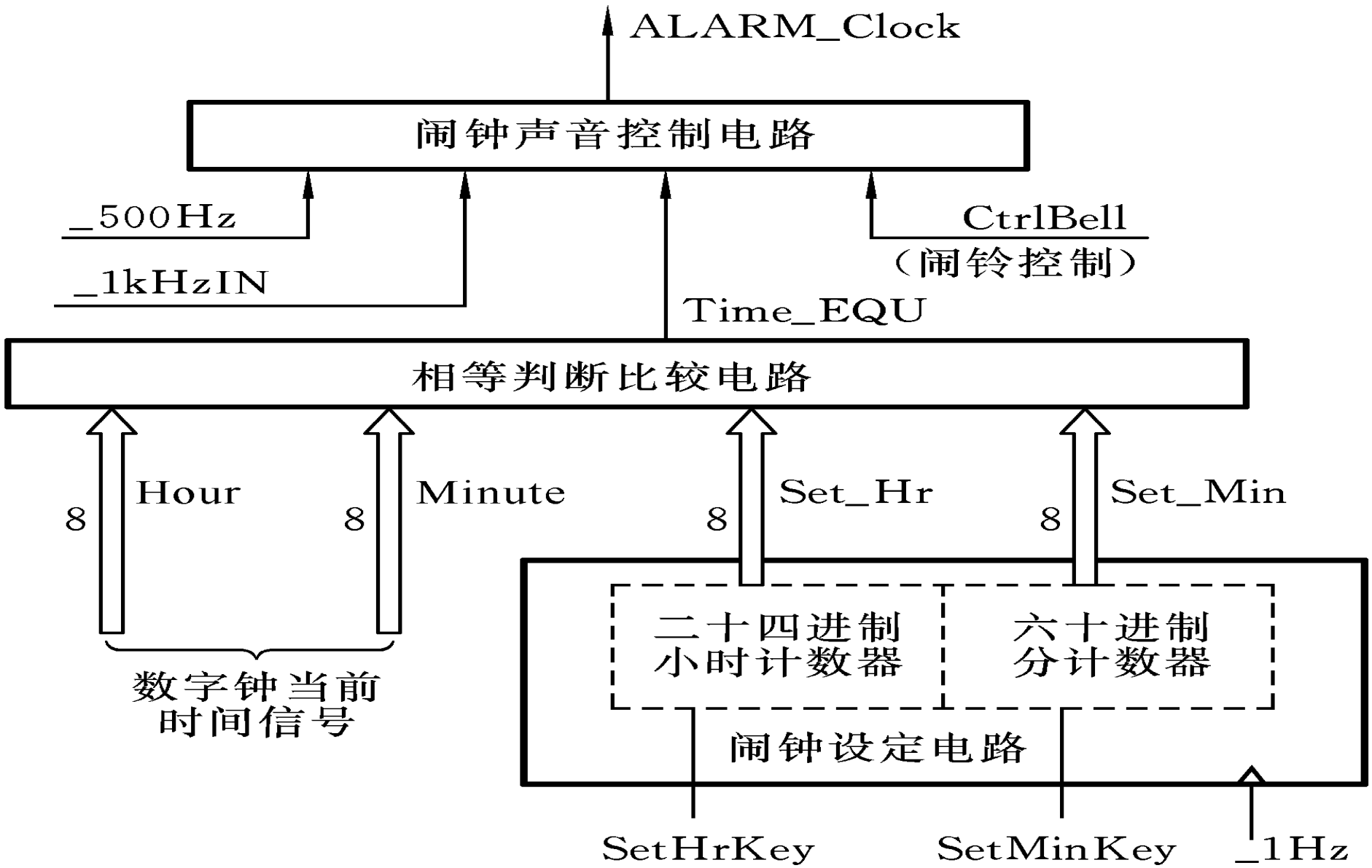
时、分、秒计数器的设计

- 分和秒计数器都是模M=60的计数器
  - 其计数规律为00—01—…—58—59—00…
- 时计数器：
  - 若采用24小时制：计数器为24进制，其计数规律为00—01……—02—23—00….
  - 若采用12小时制：计数器为12进制，其计数规律为01—02……—12—01….
- 24小时制：当数字钟运行到23时59分59秒时，秒的个位计数器再输入一个秒脉冲时，数字钟应自动显示为00时00分00秒。
- 12小时制：当数字钟运行到12时59分59秒时，秒的个位计数器再输入一个秒脉冲时，数字钟应自动显示为01时00分00秒。

多功能数字钟总体框图举例



任意闹钟设定作用模块举例



Xilinx开发板Nexys4DDR

The Nexys4 DDR

Nexys4DDR

XC7A100T-1CSG324C

动态扫描 **P395**

静态扫描显示方式

动态扫描显示方式

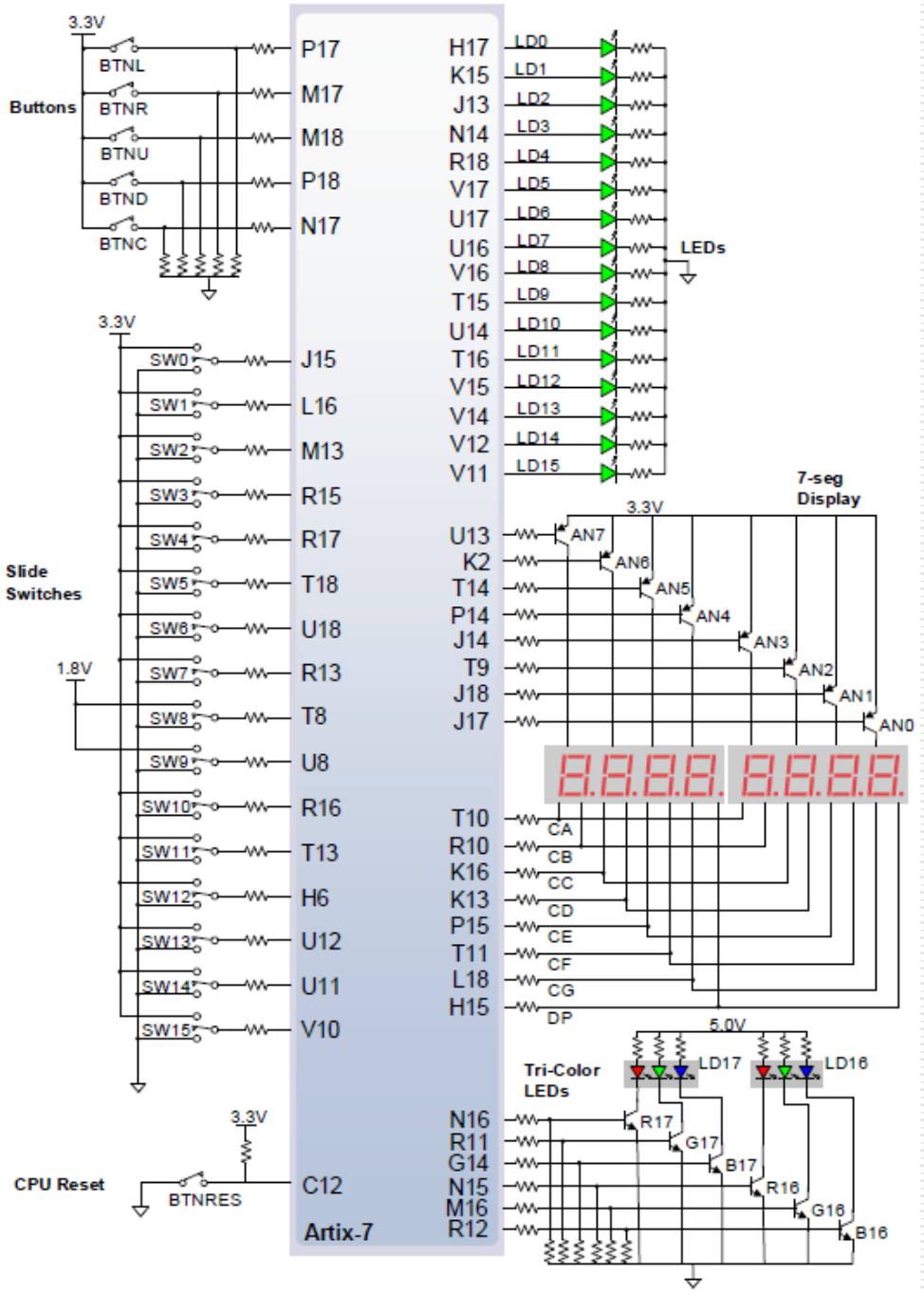
人眼的视觉暂留效应

数码管扫描频率足够大

管脚约束：

nexys4ddr_master_ucf

根据实际使用管脚进行修改



实验报告要求

- 实验名称
- 实验任务及要求
- 实验条件（实验仪器、软件、实验板等）
- 系统设计说明：
  - 组成框图、工作原理
  - 模块设计（源代码及注释）
  - 仿真文件及仿真波形（波形分析）
- 调试过程：
  - 调试中碰到的问题及解决方法
  - 最后观察到的实验结果
- 实验总结
  - 功能扩展
  - 实验的收获、体会及待改进的问题

下阶段实验

篮球竞赛24s定时器设计（P195）---硬件插板

P195设计课题1 24s定时器（含555振荡器1kHz）

检查《电子线路设计、测试与实验（二）》单元七测验结果