

EDA 课程设计洗衣机

目录

1 绪论	1
1.1 知识背景	1
1.2 任务要求	1
1.3 分析理解	2
2 方案的论证及选择	2
2.1 设计原理及总体框图	2
3 单元电路的设计	3
3.1 预置时间和编码电路	3
3.2 数码管显示模块	4
3.3 时序控制电路	6
3.4 译码器模块	7
3.5 减法计数器模块	8
4 电路图的绘制	10
5 电路的仿真及调试	12
6 心得体会	13
参考文献	15

1 绪论

1.1 知识背景

EDA 技术是在电子 CAD 技术基础上发展起来的计算机软件系统，是指以计算机为工作平台，融合了应用电子技术、计算机技术、信息处理及智能化技术的最新成果，进行电子产品的自动设计。利用 EDA 工具，电子设计师可以从概念、算法、协议等开始设计电子系统，大量工作可以通过计算机完成，并可以将电子产品从电路设计、性能分析到设计出 IC 版图或 PCB 版图的整个过程的计算机上自动处理完成。现在对 EDA 的概念或范畴用得很宽。包括在机械、电子、通信、航空航天、化工、矿产、生物、医学、军事等各个领域，都有 EDA 的应用。目前 EDA 技术已在各大公司、企事业单位和科研教学部门广泛使用。例如在飞机制造过程中，从设计、性能测试及特性分析直到飞行模拟，都可能涉及到 EDA 技术。本文所指的 EDA 技术，主要针对电子电路设计、PCB 设计和 IC 设计。

利用可编程逻辑器件丰富的内部资源，借助 EDA（电子设计自动化）工具（如 MAX+PLUS II）把家电控制器电路集成在一片 FPGA（现场可编程门阵列）芯片内，这样就无需专门的单片机和外部逻辑电路。从而减小了电路的体积、提高了系统的稳定性。一个简易的全自动洗衣机就可以利用这个方法来实现对洗涤过程的控制。

1.2 任务要求

初始条件：

设计一个洗衣机洗涤程序控制器，具有根据衣量设定不同洗涤时间、洗涤完毕有声音提醒等功能，在洗涤过程中通过不同颜色的指示灯反映洗涤三种状态：正转、反转和暂停。

要求完成的主要任务：

- 1) 洗衣机工作时间可以在 1~15 分钟任意设定。
- 2) 电机运行规律为正转 20S，停 10S，反转 20S，停 10S，再正转 20S，…… 依此规律反复运行。
- 3) 以分为单位显示洗衣机剩余工作时间，每当电机运行 1 分钟，显示器自动减 1，直到显示器为“0”，电机停止运转，并发出 3 声蜂鸣声提醒洗涤完毕。
- 4) 电机正转、反转和暂停通过 LED 灯区分。

原创力文档

maxbook118.com
预览与源文档一致，下载高清无水印

- 5) 广泛调研, 提出几种可行的方案, 多方论证, 确定设计方案。在 EDA 仿真软件上完成设计和仿真。

1.3 分析理解

针对任务具体要求, 可以设想这样一个具有定时、显示、提醒、控制等功能的洗衣机应当包含有多个模块, 这些模块相互配合才能完成全部的要求。定时需要减计数器, 显示需要数码管, 正转反转暂时的实际也需要计数器的控制, 同时设定不同状态亮起不同颜色的 LED 灯, 这需要译码器的使用, 最后的报警则应设在时间到零时刻的时候对蜂鸣器进行触发。

2 方案的论证及选择

2.1 设计原理及总体框图

洗衣机控制器的设计主要是定时器的设计, 由一片 FPGA 和外围电路构成了电器控制部分。FPGA 接收键盘的控制命令, 控制洗衣机的进水、排水、水位和洗衣机的工作状态、并控制显示工作状态以及设定直流电机速度、正反转控制、制动控制、起停控制和运动状态控制。对 FPGA 芯片的编程采用模块化的 VHDL (硬件描述语言) 进行设计, 设计分为三层实现, 顶层实现整个芯片的功能。顶层和中间层多数是由 VHDL 的元件例化语句实现。中间层由无刷直流电机控制、运行模式选择、洗涤模式选择、定时器、显示控制、键盘扫描、水位控制以及对直流电机控制板进行速度设定、正反转控制、启停控制等模块组成, 它们分别调用底层模块。总体框图如图 1 所示:

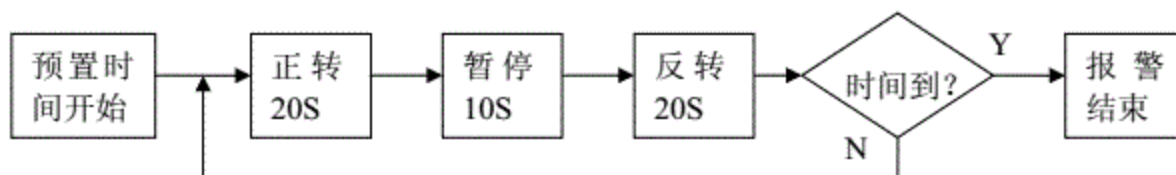


图 1 总体框图

从框图看来, 整个设计可以分为五个模块: 数码管显示模块、时序控制电路、预置时间和编码电路、译码器模块、减法计数器模块。从预置时间开始, 给开始信号让电机开始运转, 由数码管显示模块控制时间的显示, 同时减法计数器进行计时操作, 时序控制电路

控制洗衣机的运转情况，通过译码模块来决定电机运转状态，即正转反转还是暂停，各个运转状态的时间也需减法计数器来控制。下面进行分模块讨论。

3 单元电路的设计

3.1 预置时间和编码电路

定义实体 DINGSHI，具有 LOAD、FEN 两个输入端口，O 一个输出端口。LOAD 是一个时钟信号，当其出现上升沿时，将输入的时间进行预置，而 FEN 的代表分钟数，由输入的信号来进行时间预置分钟的判断，分别列举 1-15 的 BCD 码，根据不同情况进行选择输出分钟数 0。

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY DINGSHI IS PORT(START:IN STD_LOGIC;
    FEN:IN STD_LOGIC_VECTOR(3 DOWNTO 0);
    O:OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
END DINGSHI;
ARCHITECTURE MK1 OF DINGSHI IS
SIGNAL P1:STD_LOGIC_VECTOR(7 DOWNTO 0);
BEGIN
    PROCESS(START)
    BEGIN
        IF(START'EVENT AND START='1')
        THEN
            CASE FEN IS
                WHEN "0000"=>P1<="00000000";
                WHEN "0001"=>P1<="00000001";
                WHEN "0010"=>P1<="00000010";
```

```

    WHEN "0011"=>P1<="00000011";
    WHEN "0100"=>P1<="00000100";
    WHEN "0101"=>P1<="00000101";
    WHEN "0110"=>P1<="00000110";
    WHEN "0111"=>P1<="00000111";
    WHEN "1000"=>P1<="00001000";
    WHEN "1001"=>P1<="00001001";
    WHEN "1010"=>P1<="00010000";
    WHEN "1011"=>P1<="00010001";
    WHEN "1100"=>P1<="00010010";
    WHEN "1101"=>P1<="00010011";
    WHEN "1110"=>P1<="00010100";
    WHEN "1111"=>P1<="00010101";
    WHEN OTHERS=>P1<="00000000";
    END CASE;
    END IF;
END PROCESS;
O<=P1;
END MK1;

```

3.2 数码管显示模块

采用共阴极的数码管，当有高电平时相应管脚亮，对应数字 0-9，七段管脚会有相应的亮暗，将规则列出，可以将输入的数字 NUM 分成高四位和低四位，分别用两个数码管来显示。

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.All;
USE IEEE.STD_LOGIC_UNSIGNED.All;
ENTITY SHOWTIME IS
    PORT(NUM:IN STD_LOGIC_VECTOR(7 DOWNTO 0);

```

```
    CIK:IN STD_LOGIC;
    Q1,Q2:OUT STD_LOGIC_VECTOR(6 DOWNT0 0)
);
END SHOWTIME;
ARCHITECTURE ONE OF SHOWTIME IS
SIGNAL TEMP1,TEMP2:STD_LOGIC_VECTOR(6 DOWNT0 0);
SIGNAL BCD1,BCD2:STD_LOGIC_VECTOR(3 DOWNT0 0);
BEGIN
PROCESS(CIK)
BEGIN
    IF(CIK'EVENT AND CIK='1')
    THEN  BCD1<=NUM(7 DOWNT0 4); BCD2<=NUM(3 DOWNT0 0);
    END IF;
END PROCESS;
PROCESS(BCD1,BCD2)
BEGIN
    CASE BCD1 IS
WHEN  "0000"=> TEMP1<= "1111110" ;
WHEN  "0001"=> TEMP1<= "0110000" ;
WHEN  "0010"=> TEMP1<= "1101101" ;
WHEN  "0011"=> TEMP1<= "1111001" ;
WHEN  "0100"=> TEMP1<= "0110011" ;
WHEN  "0101"=> TEMP1<= "1011011" ;
WHEN  "0110"=> TEMP1<= "1011111" ;
WHEN  "0111"=> TEMP1<="1110000" ;
WHEN  "1000"=> TEMP1<="1111111" ;
WHEN  "1001"=> TEMP1<= "1111011" ;
WHEN OTHERS=>TEMP1<="0000000";
    END CASE;
END PROCESS;
END ONE;
```

```

CASE BCD2 IS
WHEN  "0000"=> TEMP2<= "1111110";
WHEN  "0001"=> TEMP2<= "0110000";
WHEN  "0010"=> TEMP2<= "1101101";
WHEN  "0011"=> TEMP2<= "1111001";
WHEN  "0100"=> TEMP2<= "0110011";
WHEN  "0101"=> TEMP2<= "1011011";
WHEN  "0110"=> TEMP2<= "1011111";
WHEN  "0111"=> TEMP2<="1110000";
WHEN  "1000"=> TEMP2<="1111111";
WHEN  "1001"=> TEMP2<= "1111011";
WHEN  OTHERS=>TEMP2<="0000000";
END CASE;
END PROCESS;
Q1<=TEMP1; Q2<=TEMP2;
END ONE;

```

3.3 时序控制电路

该模块中，输入端口有时钟信号 CLK，清零信号 RD，高电平有效，开始信号 START，高电平触发，当使能有效且 START 为高电平时，每到来一个时钟上升沿，中间信号 Q 就减 1，Q 从 59 减至 40，输出洗涤状态信号 XIDI 为 100，代表正转 20S，Q 从 39 减至 30，对应 XIDI 为 001，代表暂停 10S，Q 从 29 减至 10，对应 XIDI 为 010，代表反转 20S，Q 从 10 减至 0，对应 XIDI 为 001，代表暂停 10S，当 Q=0 时，又重新装载数值 59，进行新一轮计数，如此循环。

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY SHIXU IS

```



```

PORT(CLK:IN STD_LOGIC;
RD:IN STD_LOGIC;
START:IN STD_LOGIC;
XIDI:BUFFER STD_LOGIC_VECTOR(2 DOWNTO 0)
);
END SHIXU;
ARCHITECTURE MK3 OF SHIXU IS
SIGNAL Q:STD_LOGIC_VECTOR(5 DOWNTO 0);
BEGIN
PROCESS(CLK,EN,Q)
BEGIN
IF(RD='1')OR(START='0') THEN XIDI<="001";Q<="111011";
ELSE IF(CLK'EVENT AND CLK='1')THEN
IF Q="000000" THEN Q<="111011" ;SHUCHU<="100" ;
ELSIF Q="111010"THEN XIDI<="100" ;
ELSIF Q="100111"THEN XIDI<="001";
ELSIF Q="011101"THEN XIDI<="010";
ELSIF Q="001001"THEN XIDI<="001";
END IF;
Q<=Q-1;
END IF;
END IF;
END PROCESS;
END MK3;

```

3.4 译码器模块

输入信号 SHURU 代表电机运行状态，100 正转，001 暂停，010 反转，当 SHURU=100, 输出信号 RUN 输出高电平, 其它为低电平, 当 SHURU=001, 输出信号 PAUSE 输出高电平, 其它为低电平, 当 SHURU=010, 输出信号 REV 输出高电平, 其它为低电平。

通过此程序段，将代码变成控制信号，对电机运转真正实现控制。

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY YIMA IS
PORT(SHURU:IN STD_LOGIC_VECTOR(2 DOWNTO 0);
RUN:OUT STD_LOGIC;
REV:OUT STD_LOGIC;
PAUSE:OUT STD_LOGIC;
);
END YIMA;
ARCHITECTURE MK4 OF YIMA IS
BEGIN
PROCESS(SHURU)
BEGIN
CASE SHURU IS
WHEN "001"=>REV<='0';RUN<='0';PAUSE<='1';
WHEN "010"=>REV<='1';RUN<='0';PAUSE<='0';
WHEN "100"=>REV<='0';RUN<='1';PAUSE<='0';
WHEN OTHERS=>REV<='0';RUN<='0';PAUSE<='1';
END CASE;
END PROCESS;
END MK4;
```

3.5 减法计数器模块

由于数码管显示的是分钟数递减，而时钟信号的周期为 1S，所以要设计一个 60 进制的减计数器。CHUSHITIME 是设定的预置时间，XIANSHI 是数码管应显示的剩余分钟数，通过变量 TIME_SECOND 来实现。TIME_SECOND 初值为 59，当时钟上升沿到来，

TIME_SECOND 减 1, 直到为 0 时, 又进行新一轮数据的装载为 59。当 CHUSHITIME 高四位不为 0 而低四位为 0 时, 让高四位为 0, 低四位变成 9; 若高四位低四位均不为 0, 则高四位不变, 低四位做减计数直到 0; 若高四位为 0 低四位不为 0 则低四位做减计数直到 0。

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY COUNTER IS
PORT(CLK,START:IN STD_LOGIC;
      CHUSHITIME:IN STD_LOGIC_VECTOR(7 DOWNTO 0);
      XIANSHI:BUFFER STD_LOGIC_VECTOR(7 DOWNTO 0 );
      CF:BUFFER STD_LOGIC);
END COUNTER;
ARCHITECTURE MK5 OF COUNTER IS
BEGIN
  PROCESS(CLK)
    VARIABLE TIME_SECOND:INTEGER RANGE 0 TO 59:=59;
  BEGIN
    IF(CLK'EVENT AND CLK='1')
      THEN
        IF(START='0')
          THEN
            XIANSHI<=CHUSHITIME;
            CF<='1';
          ELSE
            IF(CF='1')
              THEN
                IF(TIME_SECOND=0 AND XIANSHI(7 DOWNTO 0)=0)
                  THEN CF<='0';
                
```

```

ELSE
    IF(TIME_SECOND=0)
    THEN IF(XIANSHI(3 DOWNT0 0)=0)
        THEN
            XIANSHI(7 DOWNT0 4)<=XIANSHI(7 DOWNT0 4)-1;
            XIANSHI(3 DOWNT0 0)<="1001";
            TIME_SECOND:=59;
        ELSE
            XIANSHI(7 DOWNT0 4)<=XIANSHI(7 DOWNT0 4);
            XIANSHI(3 DOWNT0 0)<=XIANSHI(3 DOWNT0 0)-1;
            TIME_SECOND:=59;
        END IF;
    ELSE
        TIME_SECOND:=TIME_SECOND-1;
    END IF;
END IF;
END IF;
END IF;
END IF;
END PROCESS;
END MK5;

```

原创力文档

max.book118.com

PDF阅读器请下载高清无水印

4 电路图的绘制

将上述五个模块分别生成相应芯片，在图纸上绘制总电路图，总体的输入有四个：时钟信号 CLK 和 LOAD，预置时间信号 FEN，开始信号 START。定时模块的输出作为减计数器的输入，由此进行计数，减计数器的输出接数码管的输入，将当前数字通过数码管进行显示，进位信号作为时序电路的清零端，时钟信号和开始信号触发时序电路进行对电机状态的控制描述，时序电路的输出则接入译码器，将电机状态描述进行具体执行，即为正

转或反转或暂停。当 CF 由高变低时，可以利用这个一个下降沿触发蜂鸣器的报警。
 连接总电路图如图 2:

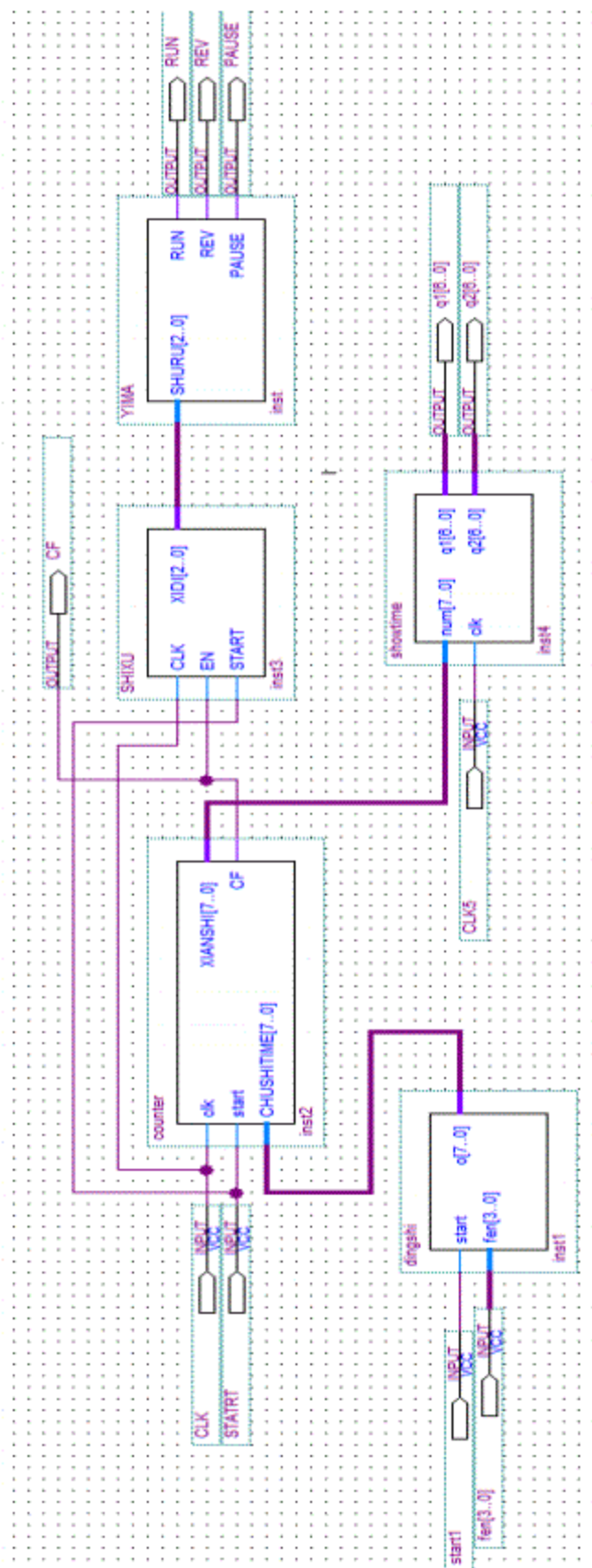


图 2 总电路图

5 电路的仿真及调试

各个单元电路仿真结果图：

1) 预置时间和编译电路仿真。对 fen 信号进行“1110”的设定，表示 14 分钟的预置时间，即十位数为 1，个位数为 4，对应 BCD 码及为 0001,0100，当 start 高电平到来，则把组成八位二进制数 00010100 赋给 o 输出。仿真结果如图 3：

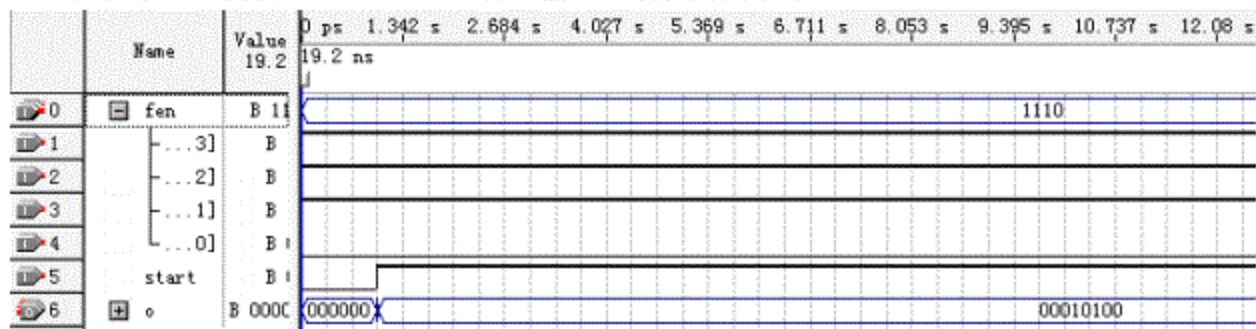


图 3 预置时间和编码电路仿真结果

2) 数码管显示模块仿真。当有时钟到来，便把 num 所表示的 BCD 码高四位即十位让 q1 数码管显示，低四位即个位让 q2 数码管显示。仿真结果如图 4：

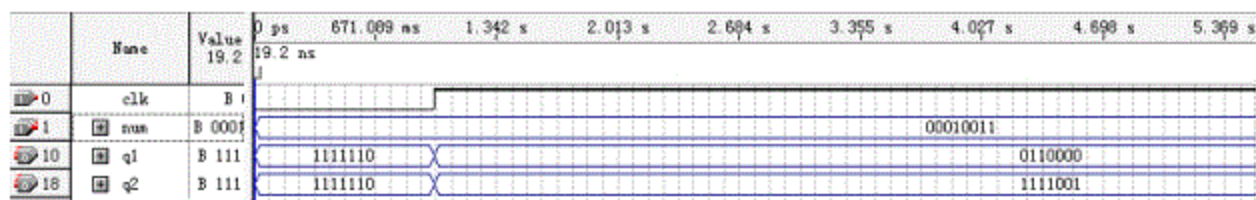


图 4 数码管显示模块仿真结果

3) 时序控制电路仿真。在清零信号 EN 低电平无效和 START 高电平有效时，实现 XIDI 信号描述电机每分钟：正转 20s→暂停 10s→反转 20s→暂停 10s 的循环。仿真结果如图 5：

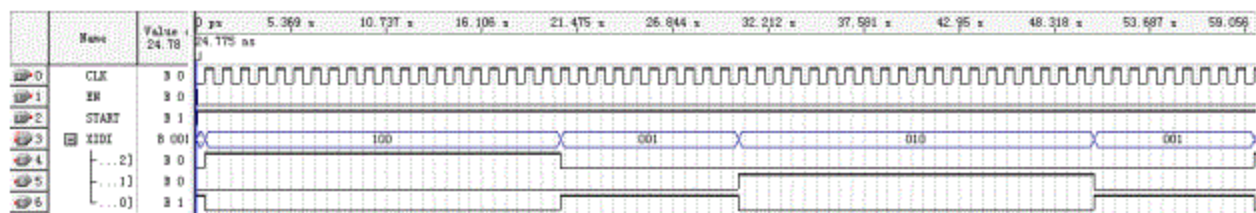


图 5 时序控制电路仿真结果

4) 译码器模块仿真。可以看到，SHURU=100，输出信号 RUN 输出高电平，其他为低

电平，SHURU=001，输出信号 PAUSE 输出高电平，其它为低电平，当 SHURU=010，输出信号 REV 输出高电平，其它为低电平。仿真结果如图 6:

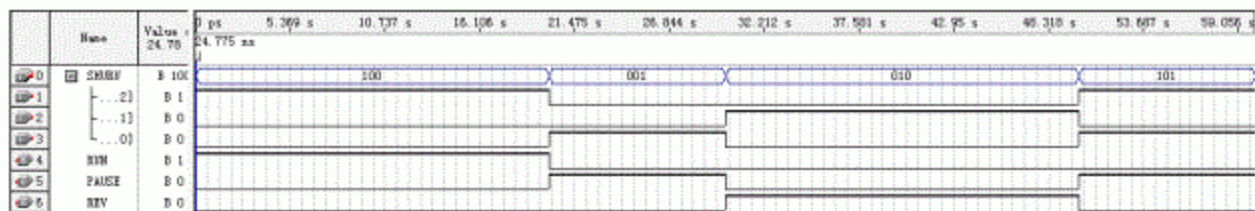


图 6 译码器模块仿真结果

5) 减计数器仿真。start 信号有效，设定 CHUSHITIME 为 00010010，可以看到，输出信号 XIANSHI 每过 60s 便减 1，直至为 0。仿真结果如图 7:

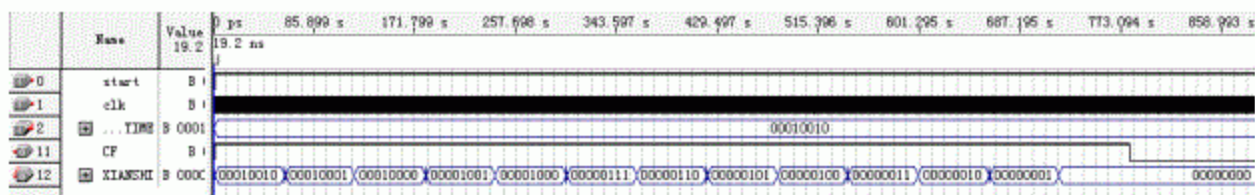


图 7 减计数器仿真结果

总电路图仿真。当 start 信号高电平有效时，预置分钟数 fen 为 1011，电机加入时钟信号，电机每分钟进行正转暂停反转暂停的循环，q1 显示剩余分钟数的十位，2 分钟后，由 1 减为 0，q2 显示剩余分钟数的个位，前 2 分钟，由 1 减为 0，后 9 分钟则由 9 每分钟减 1 直至为 0。仿真结果如图 8:

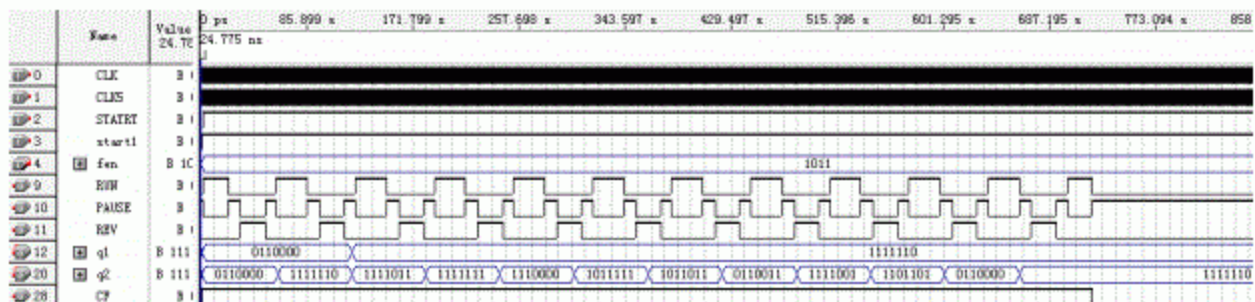


图 8 总电路图仿真结果

6 心得体会

这次课设时间紧迫，中间遇到很多困难，但最终的总电路图的仿真结果终于出来了，并且正确。在每个分模块中，所要表达的意思想要达到的目的都还是做到了，仿真结果也都正确。

通过本次课设，我加强了对 VHDL 语言的认识，对其的运用也变得更为熟练，另一方面，通过实际的软件操作，使我更熟悉了 QUARTUS 软件的操作和使用，应用起来也更为得心应手。本次课题是设计全自动化的洗衣机控制装置，通过分析要求而对所要做的事情进行模块分类，这样的思想很妙，很具有逻辑性，进行分模块后目标性也变强了，做起来不会茫然不知所措，最终将模块组合调整便能得到想要的结果。

最后，谢谢老师的指导和给予的意见。

参考文献

- [1] 潘松著.EDA 技术实用教程(第三版). 北京:科学出版社,2009
- [2] 康华光主编.电子技术基础(数字部分). 北京:高教出版社,2010
- [3] 阎石主.数字电子技术基础. 北京:高教出版社,2008
- [4] 赵岩岭 刘春等.在 MAX+PLUS II 平台下用 VHDL 进行数字电路设计. 西安:西电出版社,2008
- [5]徐志军.EDA 计数与 VHDL 设计. 北京:电子工业出版社,2009.1