

## 课题一 七段译码显示

### 实验目的

- 1、学习7段数码管显示译码器的设计。
- 2、学习动态扫描显示的原理及电路的设计。
- 3、学习LPM兆功能模块的调用。

### 实验器材

- 1、SOPC 实验箱
- 2、计算机（装有 Quartus II 9.0 软件）

### 实验预习

- 1、掌握数码管 LED 的显示原理及动态扫描显示的原理。
- 2、提前预习，编写好主模块的 Verilog HDL 程序。

### 实验原理

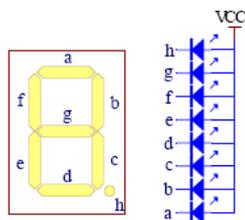


图1.1 共阳数码管及其电路

数码管LED显示是工程项目中使用较广的一种输出显示器件。常见的数码管有共阴和共阳两种。共阴数码管是将8个发光二极管的阴极连接在一起作为公共端，而共阳数码管是将8个发光二极管的阳极连接在一起作为公共端。公共端常被称作位码，而将其他的8位称作段码。如图10.1所示为共阳数码管及其电路，数码管有8个段分别为：h、g、f、e、d、c、b和a（h为小数点），只要公共端为高电平“1”，某个段输出低电平“0”则相应的段就亮。例如数码管的8个段h、g、f、e、d、c、b、a分别接1、0、1、0、0、1、0、0，数码管就显示“2”。

SmartSOPC 实验箱上有2个4位动态共阳数码管LED12和LED13。其中8个位码DIG0-DIG7和8位段码SEG0-SEG7与分别与FPGA相应的引脚相连。8个数码管，其中的每个数码管的8个段：a、b、c、d、e、f、g、h（h是小数点）都分别连到SEG0-SEG7，8个数码管分别由8个选通信号DIG0-DIG7来选择。被选通的数码管显示数据，其余关闭。如在某一时刻DIG2为低电平“0”，其余选通信号为高电平“1”，这时仅DIG2对应的数码管显示来自段码信号端的数据，而其它7个数码管呈现关闭状态。根据这种电路状态，如果希望8个数码管显示希望的数据，就必须使得8个选通信号DIG0-DIG7分别被单独选通，并在此同时，在段信号输入口加上希望在该对应数据管上显示的数据，于是随着选通信号的扫描就能实现扫描显示的目的。虽然每次只有一个LED显示，但只要扫描显示速率够快，由于人的视觉余辉效应，使我们仍会感觉所有的数码管都在同时显示。

动态扫描显示时，刷新率最好大于50Hz，即每显示完一轮的时间不超过20ms，每个数码管显示的时间不能太长也不能太短，时间太长可能会影响刷新率，导致总体显示呈现闪烁的现象；时间太短发光二极管的电流导通时间也就短，会影响总体的显示亮度。一般控制在1ms左右。

## 实验内容

本实验的内容是建立数码管动态扫描显示模块，具体内容如下：在SmartSOPC 实验箱上完成LED数码管的动态显示“1--8”8个数字；并且可通过放慢扫描速度演示动态显示的原理过程。

- 1、启动Quartus II建立一个空白工程，然后命名为dled.qpf。
- 2、新建VERILOG HDL源程序文件scan\_led.v，输入程序代码并保存进行综合编译，若在编译过程中发现错误，则找出并更正错误，直至编译成功为止。
- 3、从设计文件创建模块，由scan\_led.v生成名为scan\_led.bsf 的模块符号文件。
- 4、添加常量兆功能模块。

1)选择Tools>>MegaWizard Plug-In Manager 打开如图1.2所示添加兆功能模块向导。选择Create a new custom megafunction variation 新建一个新的兆功能模块。

2)在图1.2中按Next进入向导第2页。按图1.3所示选择和设置，注意标记部分

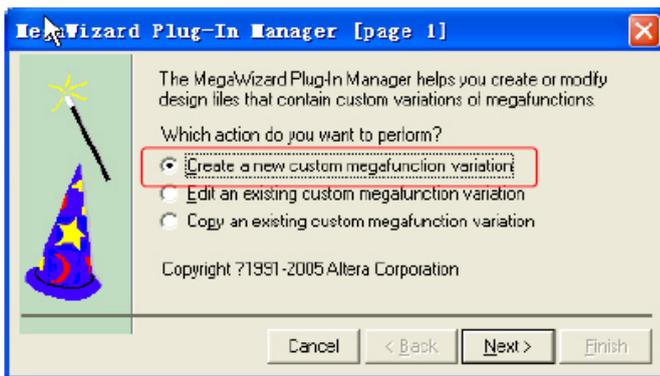


图1.2 添加兆功能模块向导对话框

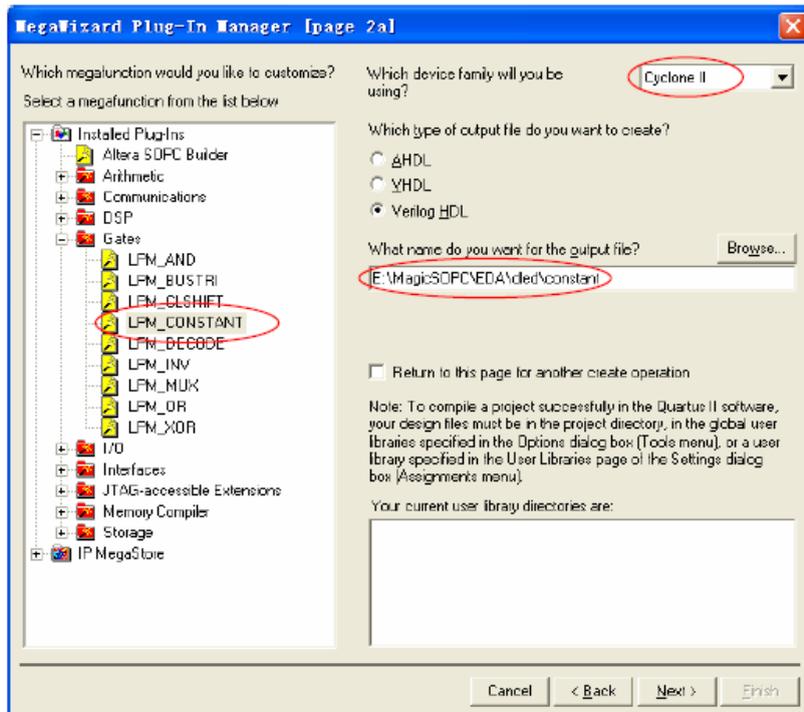


图1.3 添加常量兆功能模块

3)在图1.3中单击Next进入向导第3页。按图1.4所示选择和设置，注意标记部分。

4)在图1.4中单击Next进入向导第4页，单击Finish完成常量兆功能模块的添加。

5)新建图形设计文件命名为dled.bdf 并保存。在空白处双击鼠标左键，分别将symbol对话框中libraries:project 下的constant 和scan\_led 模块放在图形文件dled.bdf 中，加入输入、输出引脚，双击各引脚符号，进行引脚命名。把各模块连接起来如图1.5所示。

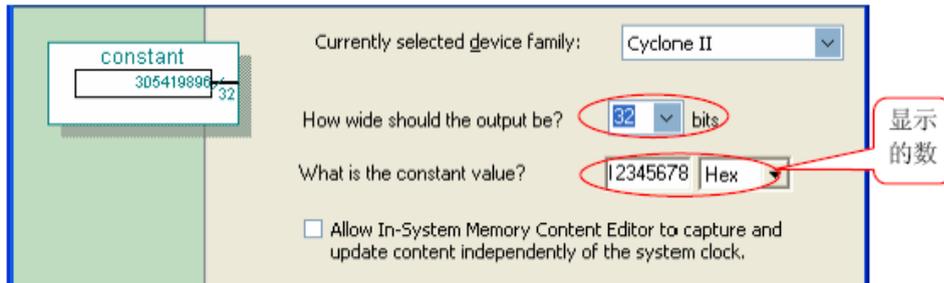


图1.4 设置常量模块的位宽及量值

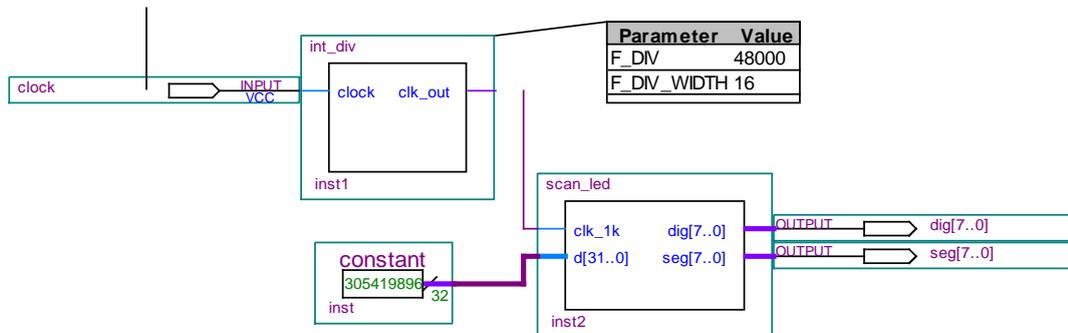


图1.5 动态LED 显示顶层模块

6)选择目标器件并对相应的引脚进行锁定，在这里所选择的器件为Altera 公司Cyclone 系列的EP1C12Q240C8芯片，具体引脚关系，查附录表一。将未使用的引脚设置为三态输入（一定要设置，否则可能会损坏芯片）。

7) 将dled.bdf 设置为顶层实体。对该工程文件进行全程编译处理，若在编译过程中发现错误，则找出并更正错误，直至编译成功为止。

8) 连接硬件，下载程序：用实验箱配置的连线将核心板上 PACK 区的引脚 236, 237, 238 和 239 分别与数码管区的 COM3 (DIG\_COM) 的 DIG4-DIG7 相对应连接；把程序下载到 FPGA 器件中。

9)观察数码管上的数字是否为“12345678”；然后将分频模块的参数改为:F\_DIV 为 24000000, F\_WIDTH 为 25, 再观察数码管上的显示数据，可以看到数据“12345678”动起来了，这正是动态扫描的方法和过程。

## 问答与习题

1、若要显示其他数字，应该做任何处理？

2、若输入时钟改为系统时钟50MHz,为了能够实现动态扫描，该做如何处理？

## 课题二 按键去抖动

### 1、实验目的

学习按键去抖动电路的硬件设计

### 2、实验内容

建立按键消抖模块，通过实验箱上的按键 KEY1（经过消抖）或 KEY2（没经过消抖）控制数码管显示数字，对比有加消抖模块和没加消抖模块电路的区别。

### 3、实验原理

按键去抖动关键在于提取稳定的低电平状态（按键按下时为低电平），滤除前沿、后沿抖动毛刺。对于一个按键信号，可以用一个脉冲对它进行取样，如果连续三次取样为低电平，则可以认为信号已经处于稳定状态，这时输出一个低电平的按键信号。继续取样的过程中，如果不满足连续三次取样为低，则认为按键稳定状态结束，这时输出变为高电平。

抖动时间的长短由按键的机械特性决定，一般为 5ms~10ms。这是一个很重要的时间参数，在很多场合都要用到。

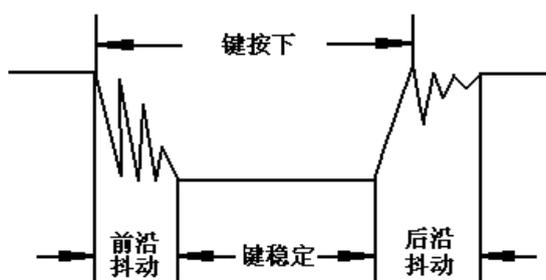


图 2.1 按键电平抖动示意图

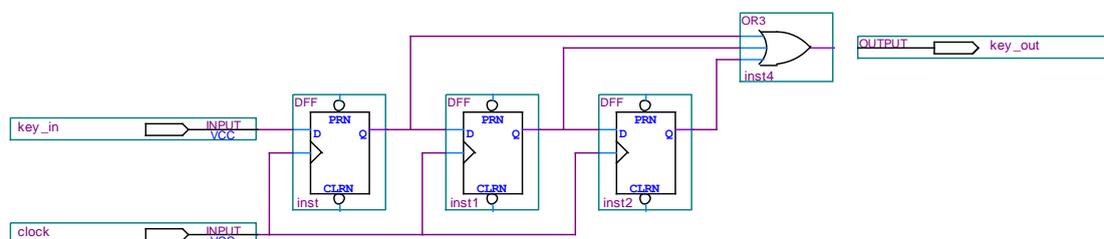


图 2.1 按键消抖硬件原理图

### 实验步骤

- 1、创建一个空白工程，命名为 key\_debounce.qpf。
- 2、将图 2.2 所示电路用 Verilog HDL 语言或原理图方式描述出来，创建 debounce.bsf 模块符号文件。
- 3、设计如图 2.3 所示顶层模块原理图。
- 4、选择目标器件并对相应的引脚进行锁定，引脚锁定参照附录表一所示。将未使用的引脚设置为三态输入（一定要设置，否则会损毁芯片）
- 5、对顶层文件 key\_debounce.bdf 进行编译仿真。
- 6、连接硬件，下载程序：
  - 1) 用实验箱配置的连线将核心板上 PACK 区的引脚 236，237，238 和 239 分别与数码

管区的 COM3 (DIG\_COM) 的 DIG4-DIG7 相对应连接;

2) 将跳线短接帽跳接到 JP6 的 KEY1、KEY2 间, 使 KEY1 与 FPGA 对应的引脚相连;

7、连续按下 KEY1, 观察数码管的显示状态, 看数值是否连续递增; 连续按下 KEY2, 观察数码管的显示状态, 看数值是否连续递增, 比较前后两次操作有何不同。

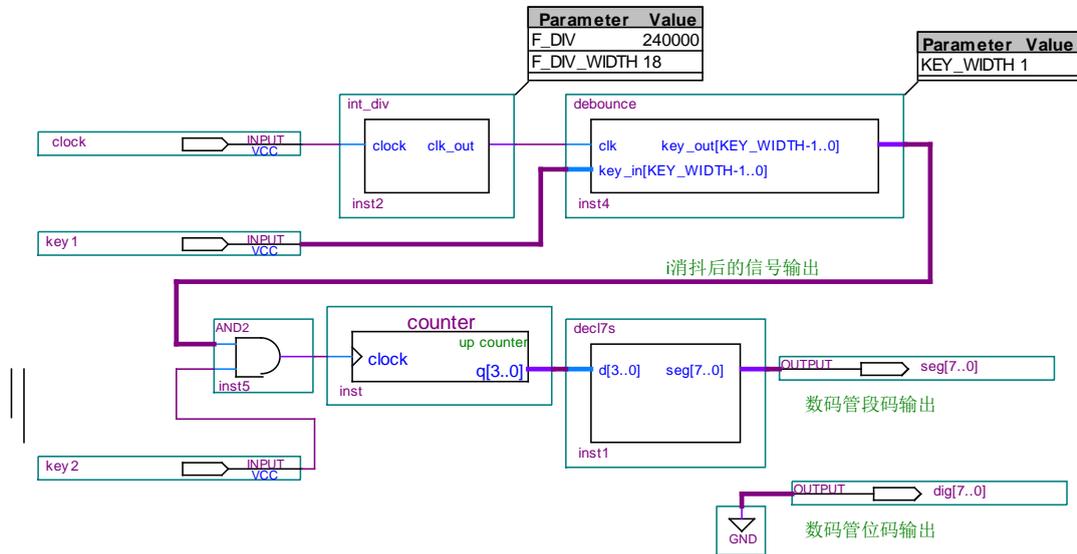


图 2.3 顶层模块

电路图上标号	FPGA 管脚	I/O 方向	外设名称
KEY2	122	I	
KEY1	121	I	
SEG0	169	O	七段数码管 段码
SEG1	170	O	
SEG2	167	O	
SEG3	168	O	
SEG4	165	O	
SEG5	166	O	
SEG6	163	O	
SEG7	164	O	
DIG0	160	O	七段数码管 位码
DIG1	159	O	
DIG2	162	O	
DIG3	161	O	
DIG4	236	O	
DIG5	237	O	
DIG6	238	O	
DIG7	239	O	
Clock	28	O	48M 时钟信号

## 课题三 数字时钟

### 实验目的

设计一个可以计时的数字时钟，其显示时间范围是 00:00:00~23:59:59，且该时钟具有暂停计时、清零等功能。

### 实验器材

- 1、SOPC 实验箱
- 2、计算机（装有 Quartus II 9.0 软件）

### 实验预习

- 1、了解时钟设计原理和各主要模块的设计方法。
- 2、提前预习，编写好主模块的 Verilog 程序。

### 实验原理

数字时钟框图如图 3.1 所示，一个完整的时钟应由 4 部分组成：秒脉冲发生电路、计数部分、译码显示部分和时钟调整部分。

1、秒脉冲发生：一个时钟的准确与否主要取决秒脉冲的精确度。可以设计分频电路对系统时钟 48MHz 进行 48000000 分频从而得到稳定的 1Hz 基准信号。定义一个 48000000 进制的计数器，将系统时钟作为时钟输入引脚 clk，进位输出即为分频后的 1Hz 信号。

2、计数部分：应设计 1 个 60 进制秒计数器、1 个 60 进制分计数器、1 个 24 进制时计数器用于计时。秒计数器应定义 clk（时钟输入）、rst（复位）两个输入引脚，Q3~Q0（秒位）、Q7~Q4（十秒位）、Co（进位位）9 个输出引脚。分、时计数器类似。如需要设置时间可再增加置数控制引脚 Set 和置数输入引脚 d0~d7。

3、译码显示部分：此模块应定义控制时钟输入、时分秒计数数据输入共 25 个输入引脚；8 位显示码输出（XQ7~XQ0）、6 位数码管选通信号（DIG0~DIG5）共 14 个输出引脚。在时钟信号的控制下轮流选择对十时、时、十分、分、十秒、秒输入信号进行译码输出至 XQ7~XQ0，并通过 DIG0~DIG5 输出相应的选通信号选择数码管。每位显示时间控制在 1ms 左右。时钟信号可由分频电路引出。

4、对于校时电路可自由发挥，如定义 3 个键 keystart、keymon、keyadd，分别用于控制时钟的计时开始、调整功能选择和加 1 处理，从而完成对现在时间的调整。

5、各模块连接方式如图 3.1 所示。

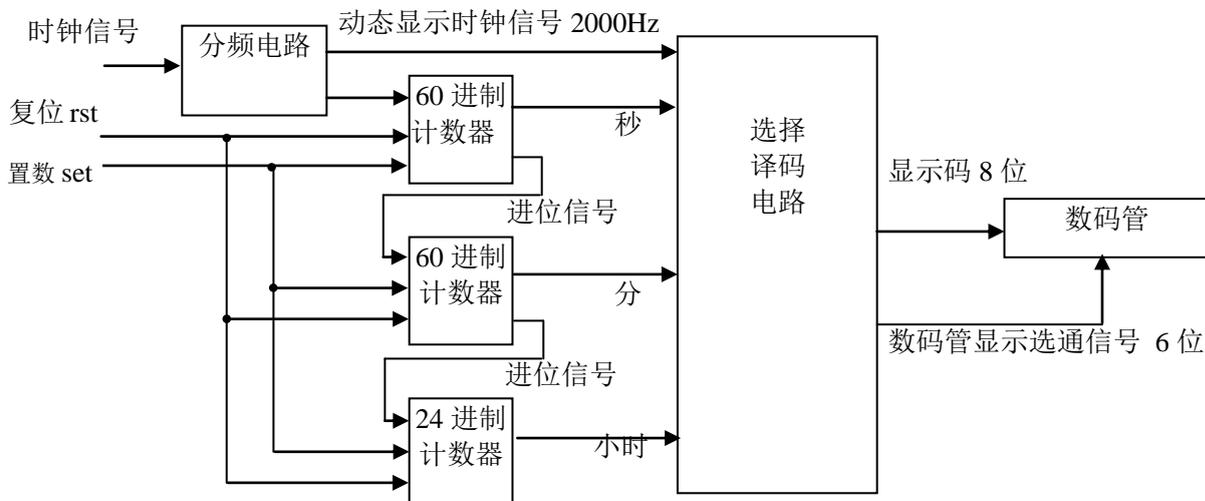


图 3.1 数字时钟框图

## 实验内容

- 1、启动 Quartus II 建立一个空白工程并命名。
- 2、新建 Verilog HDL 源程序文件，输入程序代码并保存，进行综合编译，若在编译过程中发现错误，则找出并更正错误，直至编译成功为止。并生成图形符号文件。
- 3、波形仿真验证。
- 4、选择目标器件 rst、set 接拨挡开关，时钟输入引脚接系统时钟(50M)，显示码输出 XQ0~XQ7 接 seg0-seg7,数码管选通信号 DIG0~DIG5 接实验箱上 dig0~dig5,并对相应的引脚进行锁定，引脚对应关系查实验指导书附录。**将未使用的管脚设置为三态输入**（一定要设置，否则可能会损坏芯片）。
- 5、对该工程文件进行全程编译处理，若在编译过程中发现错误，则找出并更正错误，直至编译成功为止。
- 6、设计下载
  - 1) 使用下载线，连接计算机并口和实验箱JTAG下载口（注意插口方向），打开实验箱电源。
  - 2) 启动下载界面，确认已选中下载线。
  - 3) 完成下载界面的设置，启动下载。
  - 4) a: 用实验箱配置的连线将核心板上 PACK 区的引脚 236, 237, 238 和 239 分别与数码管区的 COM3 (DIG\_COM) 的 DIG4-DIG7 相对应连接;  
b: 将跳线短接帽跳接到 JP6 的 KEY0、KEY1 间; 按动按键开关 KEY1 观察数字钟上的时间变化。。

## 问答与习题

如何实现时间的调整?

## 课题四 数字频率计

### 实验目的

在 SmartSOPC 实验箱上实现 8 位十进制频率计的设计。

### 实验器材

- 1、SOPC 实验箱
- 2、计算机（装有 Quartus II 9.0 软件）

### 实验内容

被测信号从 freq\_input 引脚（13 脚）输入，经过检测后测得的频率值用数码管 1~8 显示。被测信号由测频模块内部产生并输出到外部引脚，其频率对应关系如表 4.1 所示

表 4.1

引脚	频率值/Hz	引脚	频率值/Hz
20	24000000	42	750000
21	12000000	43	375000
23	6000000	44	187500
39	3000000	45	93750
41	1500000	46	46875

### 实验原理

频率即信号 1s 内振动次数，因此测定信号的频率必须有一个脉宽为 1 秒的输入信号作为计数允许的信号；1 秒计数结束后，计数值锁入锁存器，并为下一测频计数周期作准备的计数器清零。

数字频率计框图如图 4.1 所示。由控制、计数、锁存、译码显示四部分组成。工作原理为：控制信号产生电路对系统时钟分频后产生 0.5Hz 的门控信号 gate，锁存允许信号 LE，清零信号 MR。当 gate 为高电平时，计数器对被测信号 cin 进行计数；1s 后 gate 变为低电平，计数器停止计数；当 gate 为低电平、LE 上升沿这两个条件同时满足时，锁存电路将 32 位计数结果锁存送译码显示电路；当 gate 为低电平、MR 上升沿这两个条件同时满足时，计数器清零，为下一次计数做准备。各信号之间的时序关系见图 4.2 所示。

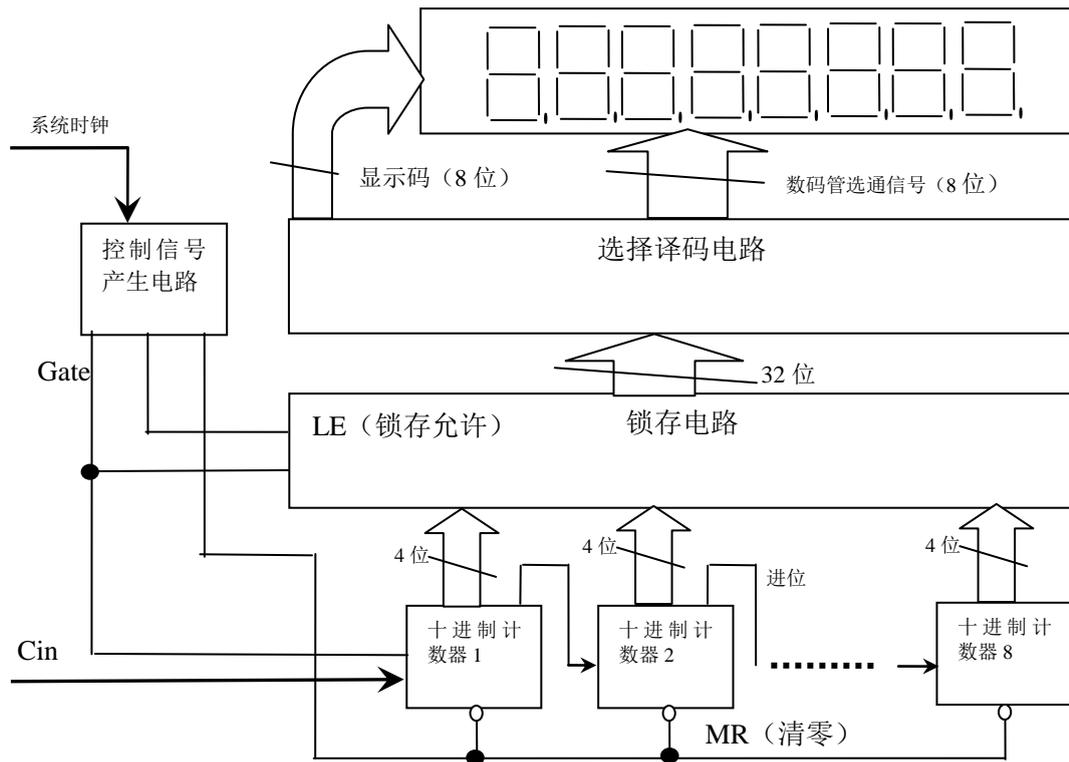


图 4.1 数字频率计框图

1、控制信号产生电路：根据选定的输入时钟信号设定分频系数，要求输出 0.5Hz 门控信号 gate、1Hz 锁存允许信号 LE 和 1Hz 清零信号 MR。这几个信号控制整个系统的工作，非常关键，要求先锁存后清零，否则计数结果可能丢失，参考时序图 4.2 所示。

2、计数模块：定义十进制计数器元件，有 cp（时钟输入）、MR（清零输入，上升沿有效）、gate（门控信号）三个个输入引脚，Q0~Q3、co（进位）5 个输出引脚。功能定义为 gate 为高电平时在 cp 上升沿计数；gate 为低 MR 为高时清零。利用元件调用的方法组成 8 位十进制计数器

3、锁存电路：设计一 32 位锁存器，定义 gate（门控信号）、LE（锁存允许，上升沿有效）d0~d31 共 34 个输入引脚；Q0~Q31 共 32 个输出引脚。功能定义为 gate 为低时在 LE 上升沿锁存。

4、译码显示模块：参考数字钟。

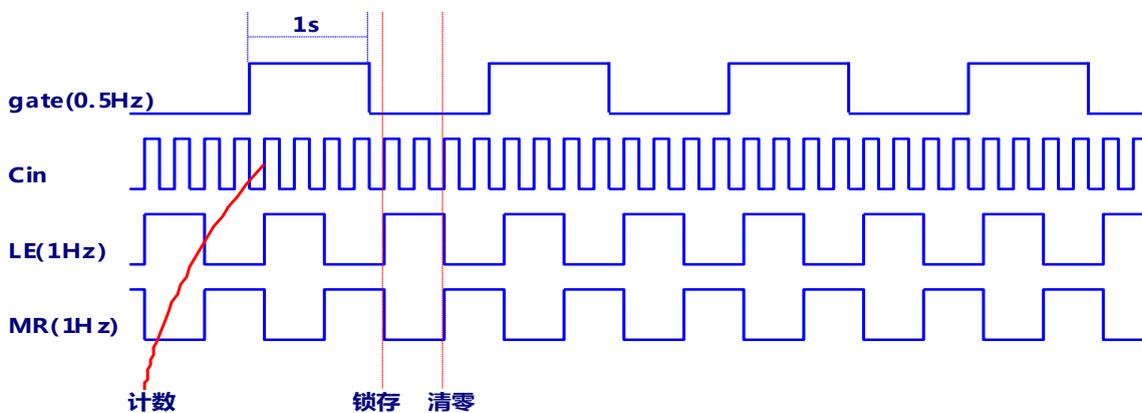


图 4.2 各信号之间的时序关系

## 实验步骤

- 1、启动 Quartus II 建立一个空白工程并命名为 freqtest.qpf。
- 2、新建 Verilog HDL 源程序文件，输入程序代码并保存，进行综合编译，若在编译过程中发现错误，则找出并更正错误，直至编译成功为止。并生成图形符号文件。
- 3、波形仿真验证。
- 4、选择目标器件，时钟输入引脚接系统时钟（48M），并对相应的引脚进行锁定，引脚对应关系查实验指导书附录。**将未使用的管脚设置为三态输入**（一定要设置，否则可能会损坏芯片）。
- 5、对该工程文件进行全程编译处理，若在编译过程中发现错误，则找出并更正错误，直至编译成功为止。
- 6、设计下载
  - 1) 使用下载线，连接计算机并口和实验箱JTAG下载口（注意插口方向），打开实验箱电源。
  - 2) 启动下载界面，确认已选中下载线。
  - 3) 完成下载界面的设置，启动下载。
  - 4) a: 用实验箱配置的连线将核心板上 PACK 区的引脚 236, 237, 238 和 239 分别与数码管区的 COM3 (DIG\_COM) 的 DIG4-DIG7 相对应连接;  
b: 分别用连接线将输出的频率信号 (PIN\_20\21\23\39\41\42\43\44\45\46, 位于核心板左上角) 引接到测频输入端 (PIN\_13), 观察数码管显示的数值。

### 4、练习与思考

- 1、若被测信号频率较低，如 0.5Hz，应如何测量？
- 2、若被测信号频率较高，如 1000000Hz，能否想法提高测量速度？

## 课题五 交通灯实验

### 实验目的

通过此实验让用户进一步了解、熟悉和掌握 CPLD/FPGA 开发软件的使用方法及 Verilog HDL 的编程方法。学习简单状态机的设计和硬件测试。

### 实验内容

本实验的内容是设计一个简易的交通灯控制器，要求能实现红、绿、黄三种灯的控制并通过数码管显示倒计时时间。整个设计在 SmartSOPC 实验箱上实现。

### 实验原理

本实验设计的交通灯控制器要求实现对 A、B 两个方向的红、绿、黄灯三种灯的控制，并能实现时间显示的倒计时。因此每个方向的灯可用一个状态机实现，状态的跳转顺序为红灯-绿灯-黄灯-红灯（另一个的状态应为绿-黄-红-绿）同时设计一个计时器，来记录每种灯的倒计时时间。最后将交通灯的状态信息输出，至数码管显示模块和交通灯模块。注意，一个方向的红灯时间应和另一个方向的绿黄灯时间总和相等。

### 实验步骤

- 1) 启动 Quartus II 建立一个空白工程，然后命名为 traffic\_test.qpf。
- 2) 新建 Verilog HDL 源程序文件 traffic.v、traffic\_test.v、traffic\_top.v，输入程序代码并保存（完整的 Verilog HDL 程序参考程序清单 1）进行综合编译，若在编译过程中发现错误，则找出并更正错误，直至编译成功为止。
- 3) 选择目标器件并对相应的引脚进行锁定，在这里所选择的器件为 Altera 公司 Cyclone 系列的 EP1C6Q240C8 芯片引脚锁定方法如表 1 所示将未使用的管脚设置为三态输入（一定要设置，否则可能会损坏芯片）设置方法见《EDA 实验与实践教程（一）》错误！未找到引用源。节。

表 1 引脚锁定方法

信号	引脚			信号	引脚			信号	引脚		
	1C6	1C12	EDA		1C6	1C12	EDA		1C6	1C12	EDA
seg[0]	169	169	169	dig[0]	160	160	160	lampa[2]	46	46	46
seg[1]	170	170	170	dig[1]	159	159	159	lampa[1]	45	45	45
seg[2]	167	167	167	dig[2]	162	162	162	lampa[0]	43	43	43
seg[3]	168	168	168	dig[3]	161	161	161	lampb[2]	44	44	44
seg[4]	165	165	165	dig[4]	204	139	215	lampb[1]	41	41	41
seg[5]	166	166	166	dig[5]	199	138	216	lampb[0]	42	42	42
seg[6]	163	163	163	dig[6]	198	137	213	rst_n	152	152	152
seg[7]	164	164	164	dig[7]	205	136	214	clock	28	28	28

4) 对该工程文件进行全程编译处理，若在编译过程中发现错误，则找出并更正错误，直至编译成功为止。

#### 5) 硬件连接

(1) 如果核心板是 QuickSOPC-1C12 需执行此步骤，如果不是则跳过此步。拔掉实验箱上 JP6 中 MotorA、MotorB、8563INT、LM750S 上的跳线，拿出实验箱配置的连线将实验箱上 JP6 的 MotorA、MotorB、8563INT 和 LM750S 引脚（注意，连线要插在 JP6 的左边引脚）分别与数码管显示区的 COM3(DIG\_COM)的 DIG4~DIG7 相对应连接。

(2) 将 Traffic PACK 交通灯模块插到核心板上。

(3) 拿出 Altera ByteBlasterII 下载电缆并将此电缆的两端分别接到 PC 机的打印机并口和 QuickSOPC 核芯板上的 JTAG 下载口上，打开实验箱和显示器的电源，执

行下载命令，  
把程序下载到 FPGA 器件中。

6) 按核心板上的复位按键，观察交通灯是否正常运行。

### 问答与习题

如何同时显示两个方向的时间？

## 课题六 A/D转换

### 实验目的

熟悉串行模/数转换芯片TLC549的使用方法，掌握利用有限状态机实现一般时序逻辑分析的方法，了解一般状态机的设计与应用。

### 实验器材

- 1、SOPC 实验箱
- 2、计算机（装有 Quartus II 9.0 软件）

### 实验预习

- 1、查找相关资料，熟悉TLC549的功能，时序和使用方法。
- 2、设计相应的控制流程，书写完整的Verilog程序，写出实验预习报告。

### 实验原理

TLC549是一个8位的串行模数转换器，AD转换时间最大17微秒，I/O时钟可达1.1MHz。如图6.1所示为TLC549的访问时序，从图中可以看出当CS/拉低时，ADC前一次的转换数据（A）的最高位A7立即出现在数据线DATA OUT上，之后的数据在时钟I/O CLOCK的下降沿改变，可在I/O CLOCK的上升沿读取数据。读完8位数据后，ADC开始转换这一次采样的信号（B），以便下一次读取。

转换时片选信号CS/要置高电平。设计操作时序是要注意 $T_{su}(CS)$ 、 $T_{conv}$ 及I/O CLOCK的频率几个参数。 $T_{su}(CS)$ 为CS/拉低到I/O CLOCK 第一个时钟到来时间，至少要1.4 秒； $T_{conv}$ 为ADC的转换时钟，不超过17微秒；I/O CLOCK不能超过1.1MHz。

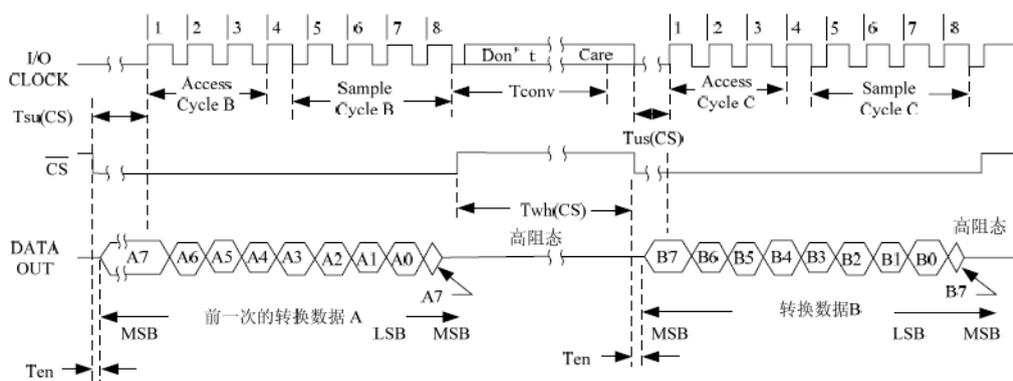


图6.1 TLC549 访问时序

由于ADC是8位的，所以采样的电压值为：

$$V = (D/256) * V_{ref}$$

其中V 为采样的电压值，D为ADC 转换后读取的8 位二进制数， $V_{ref}$  为参考电压值，这里是2.5V。

### 实验内容

本实验的内容是使用状态机实现对TLC549的采样控制，实现一个简易的电压表。模数转

换电路原理图如附录二中所示。利用Quartus II完成设计、仿真等工作，最后在SmartSOPC实验箱上进行硬件测试。实验时通过调节电位器W1改变ADC的模拟输入值，数据采样读取后由数码管显示。最后用万用表测量输入电压，并与读取到的数据做比较。

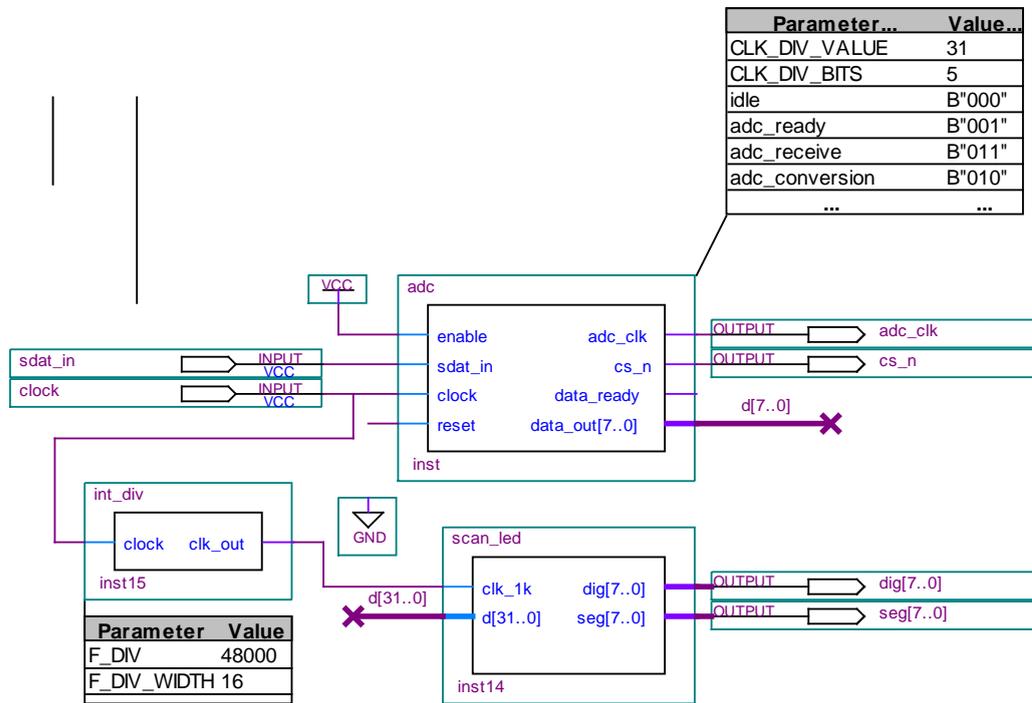


图6.2 TLC549采样控制顶层模块原理图

## 实验步骤

- 1、启动Quartus II建立一个空白工程，然后命名为tlc549adc.qpf。
- 2、新建Verilog HDL 源程序文件adc.v，写出程序代码并保存，进行综合编译，若在编译过程中发现错误，则找出并更正错误，直至编译成功为止。
- 3、建立波形仿真文件并对adc.v进行仿真验证，验证仿真结果是否与设计相符合，如果不符合，需重新设计文件，再进行综合编译、仿真。直到仿真结果与设计相符合为止。
- 4、新建图形设计文件进行硬件测试，命名为tlc549adc.bdf并保存，如图6.2所示。
- 5、选择目标器件并对相应的引脚进行锁定、引脚设置查附录三。**将未使用的引脚设置为三态输入**（一定要设置，否则可能会损坏芯片）。
- 6、将tlc549adc.bdf 设置为顶层实体。对该工程文件进行全程编译处理，若在编译过程中发现错误，则找出并更正错误，直至编译成功为止。
- 7、设计下载
 

使用下载线，连接计算机并口和实验箱JTAG下载口（注意插口方向），打开实验箱电源。启动下载界面，确认以选中下载线。完成下载界面的设置，启动下载。
- 8、用实验箱配置的连线将核心板上PACK区的引脚236，237，238和239分别与数码管区的COM3（DIG\_COM）的DIG4-DIG7相对应连接；
- 9、调节电位器RW1，用万用表测电压档测量模数转换器信号输入端的电压（两表笔分别接ADIN、VSSA两个测试点，测试点位于“串行模/数转换”区域内），观察数码管的数值，将数码管的数据转换计算后与万用表的读数做比较。

## 引脚锁定

信号	引脚			信号	引脚			信号	引脚		
	1C6	1C12	EDA		1C6	1C12	EDA		1C6	1C12	EDA
seg[0]	169	169	169	dig[0]	160	160	160	adc_clk	132	132	132
seg[1]	170	170	170	dig[1]	159	159	159	cs_n	134	134	134
seg[2]	167	167	167	dig[2]	162	162	162	sdat_in	131	131	131
seg[3]	168	168	168	dig[3]	161	161	161	clock	28	28	28
seg[4]	165	165	165	dig[4]	204	139	215				
seg[5]	166	166	166	dig[5]	199	138	216				
seg[6]	163	163	163	dig[6]	198	137	213				
seg[7]	164	164	164	dig[7]	205	136	214				

## 问答与习题

如何提高 TLC549 的转换精度和转换速率？

## 课题七 简易正弦信号发生器

### 实验目的

进一步熟悉 QuartusII 及 LPM-RAM 宏模块与 FPGA 硬件资源的使用方法。学习利用 LPM\_ROM 设计一个简易的正弦信号发生器。

### 实验设备

计算机、和软件QuartusII和EDA/SOPC试验箱

### 实验原理

如图所示的简易的正弦信号发生器的结构由以下四个部分组成：

- 地址信号发生器，由计数器构成。这里根据ROM的参数，选择6位输出。
- 正弦信号数据存储ROM，含有64个8位波形数据（一个正弦波周期）。
- Verilog HDL顶层程序设计。
- 8位D/A转换器。

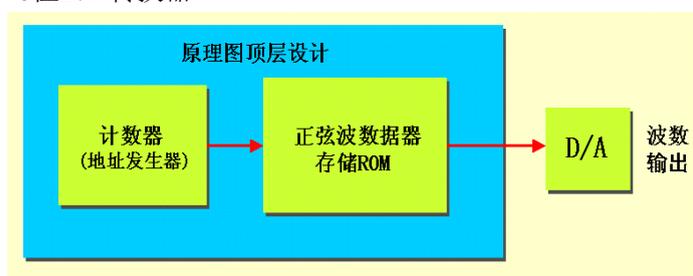


图 7.1 简易正弦信号发生器结构框图

打开 QuartusII 软件，在连接试验电路之前调入 LPM-RAM-DQ 宏模块，PLM-COUNER 模块，再连接电路图，实验原理设计图如下：

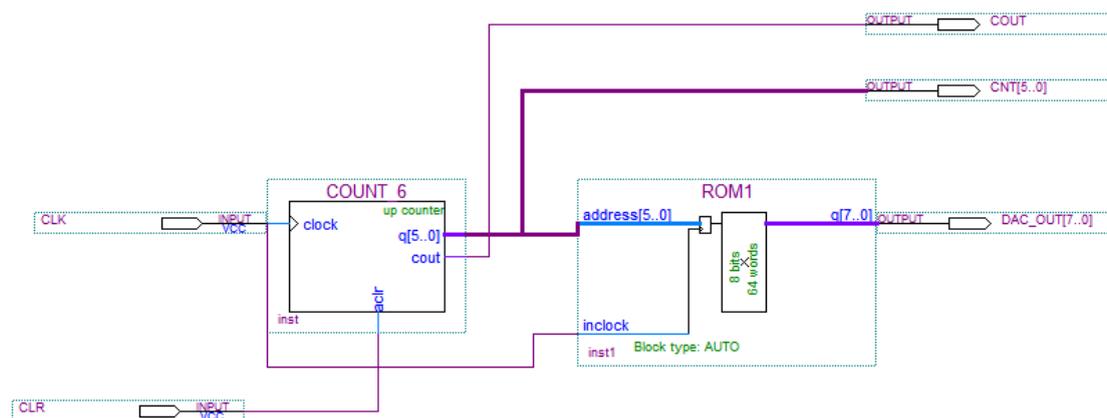


图 7.2 简易正弦信号发生器顶层电路设计

### 实验内容

- 启动Quartus II建立一个空白工程，然后命名为sine\_generator.ppf。
- 定制ROM初始化数据文件。
- 调入LPM-ROM宏模块，LPM-COUNER模块构建如图7.2所示顶层文件原理图，并设置为顶层文件，进行综合编译，若在编译过程中发现错误，则找出并更正错误，直至编译

成功为止。

- 4、建立波形仿真文件并对sine\_generator.bdf进行仿真验证，验证仿真结果是否与设计相符合，如果不符合，需重新设计文件，再进行综合编译、仿真。直到仿真结果与设计相符合为止。
- 5、选择目标器件并对相应的引脚进行锁定、引脚设置查附录表一。将未使用的引脚设置为三态输入（一定要设置，否则可能会损坏芯片）。
- 6、使用下载线，连接计算机并口和实验箱JTAG下载口（注意插口方向），打开实验箱电源。启动下载界面，确认以选中下载线。完成下载界面的设置，启动下载。
- 7、调用 SignalTap II 工具以及 ROM 的在系统数据读写测试工具进行测试

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	255	254	252	249	245	239	233	225
8	217	207	197	186	174	162	150	137
16	124	112	99	87	75	64	53	43
24	34	26	19	13	8	4	1	0
32	0	1	4	8	13	19	26	34
40	43	53	64	75	87	99	112	124
48	137	150	162	174	186	197	207	217
56	225	233	239	245	249	252	254	255

图 7.3 键入 64 个正弦信号数据

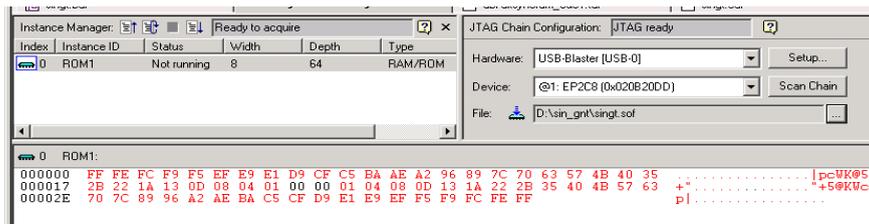


图 7.4 利用 In-System Memory Content Editor 读取 LPM\_ROM 中数据

### 1. 打开 SignalTap II 编辑窗

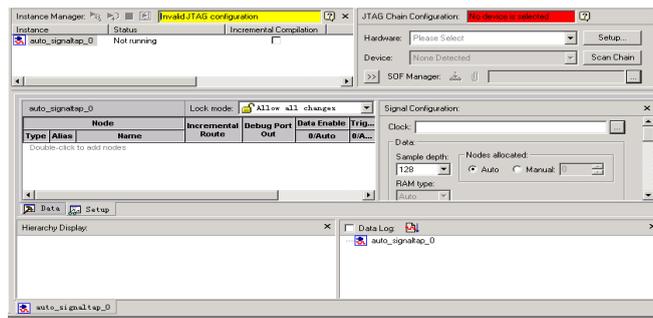


图 7.5

### 2. 调入待测信号窗

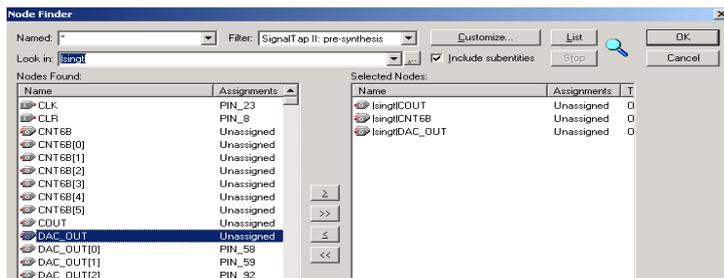


图 7.6

### 3、设置 SignalTap II 工作参数

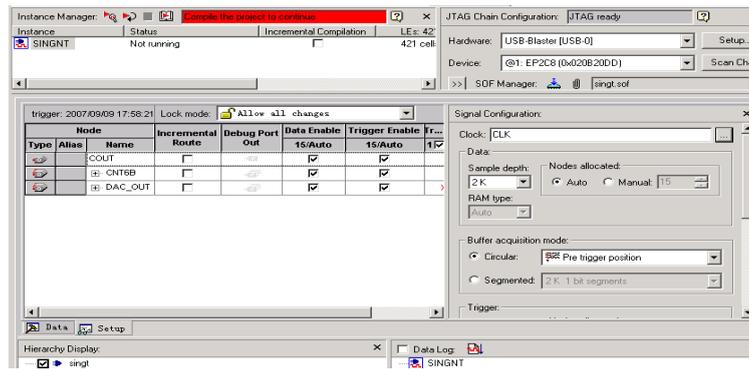


图 7.7

### 4、设置 SignalTap II 的触发信号和触发方式

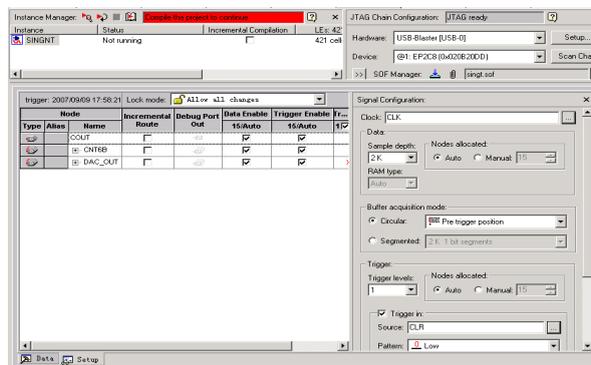


图 7.8

### 5、文件存盘编译下载

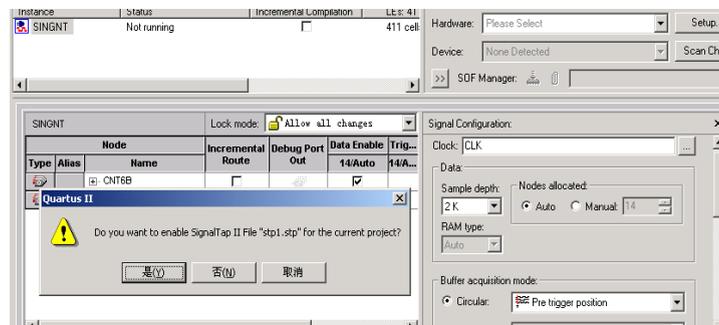


图 7.9

### 6. 启动 SignalTap II 进行采样与分析

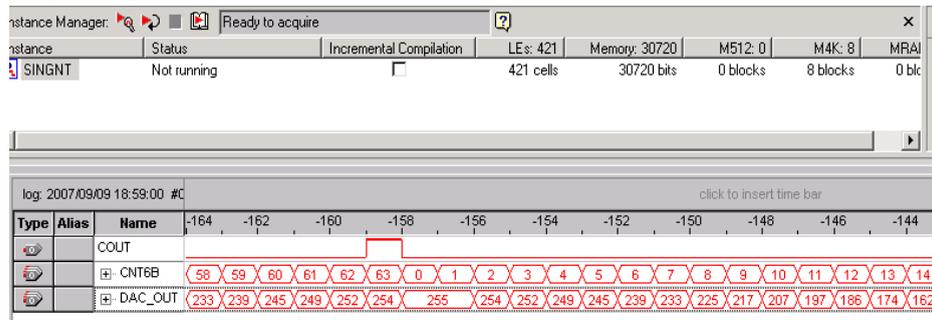


图 7.10

## 7、设置 SignalTap II 窗口中波形数据显示方式

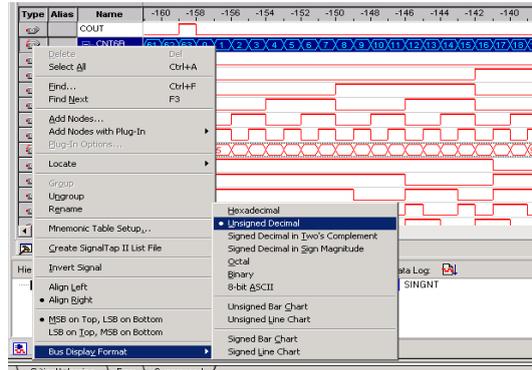


图 7.11

## 8、显示输出的正弦波信号

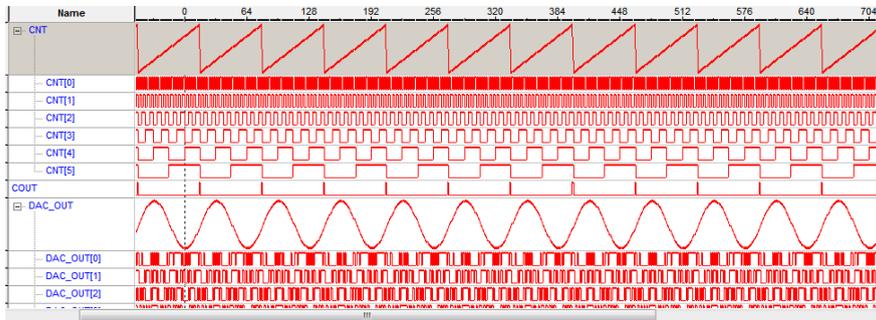


图 7.12

## 附录一:实验操作的基础知识

### 一、实验SOPC平台注意事项

1) 在设计中一定要将未定义的引脚定义为三态输入,注意一定不能将未定义(不使用)引脚模式设为输出,并连接到地(As outputs,driving ground)。否则会造成未使用的芯片冲突而损坏芯片。

#### 2) 数字信号源

数字信号源能产生0.5Hz~100MHz多组数字信号源,分CLOCK0~CLOCK3四组信号输出,通过短路帽跳线分别可选择不同的时钟信号输入到FPGA, CLOCK0~CLOCK3 对应的排针左边为不同频率的数字信号,排针右边是连到一起的,所以每组信号源只能插一个短路帽,否则可能会损坏芯片。使用时要将数字信号源的电源开关JP11的短路帽跳到ON 位置。

#### 3) DDS 函数信号发生器

DDS 函数信号发生器的时钟源由数字信号源提供,所以使用时必须确保数字信号源的电源开关JP11 的短路帽跳到ON位置,DDS时钟输入端JP10的短路帽处于24M/OUT 位置,同时要打开DDS函数信号发生器的电源开关DDS\_Power(SW10)。

输出信号由正弦波或三角波切换为方波时,输出幅值可能会变得很大,所以在切换之前应将输出幅值调小(调节W2)。

#### 4) TFT 真彩液晶显示器

不使用TFT LCD 液晶显示器时应将LCD 的背光电源关掉(SW11)。

#### 5) FPGA 编程

通过JTAG 接口下载FPGA 程序时,最要将下载器插在主板上的JTAG 接口,不要插在核心板的JTAG 接口上,以防插反,最好是断电后再插拔;如果使用AS接口下载程序时,一定要断电后才能把下载器插到AS 接口,并且确保不插反才能上电。

#### 6) 扩展I/O

J38 的扩展I/O中的AA7、E5 是和扩展模块中J32、J33 扩展I/O 的AA7、E5 共用FPGA 引脚的;J38 的扩展I/O 中的AC11、AF13、AA11 是和扩展总线的控制I/O PIO1、PIO2、PIO3 共用FPGA引脚;使用时要注意I/O 分配。

7) 未使用或不清楚的电路部分,不要随意短接或使用,一定要详细查询资料。

### 二、实验前的准备及实验要求

1) 首次实验时,应详细阅读实验箱的相关介绍知识,熟悉软件使用方法。

2) 阅读实验注意事项及实验室相关规定,听从实验指导教师的安排,按时完成实验任务。

3) 实验前必须预习,写出详细的实验报告,明确实验原理,熟悉实验操作内容。

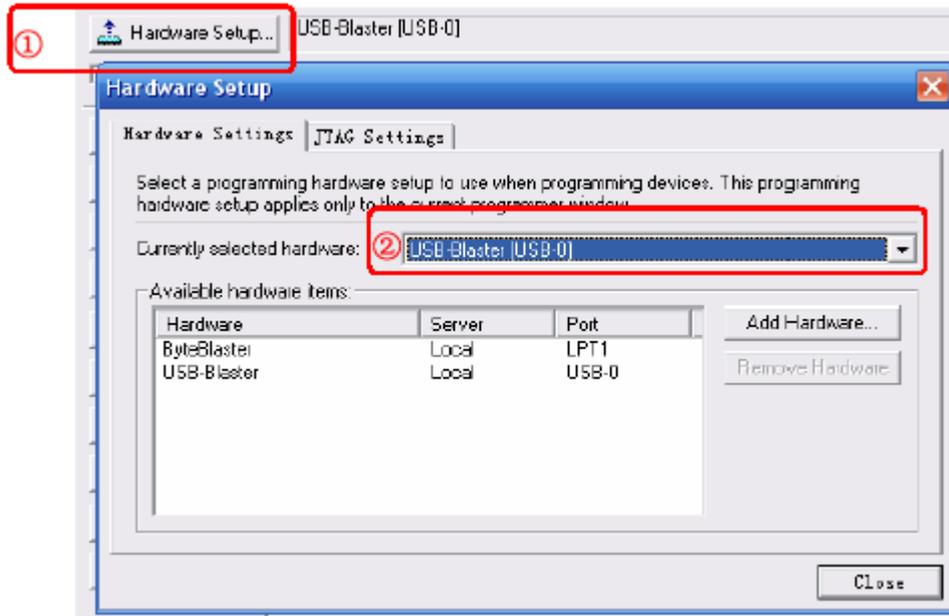
### 三、安装Z-Blaster下载电缆

从“开始”>>“程序”>>“Altera”>>“Quarts II 7.0”打开Quarts II 9.0 软件。在Quartus II软件中选择“Tools”>>“Programmer”。打开编程器窗口,如图1.1 所示。察看编程器窗口左上角的Hardware Setup栏中硬件是否已经安装,如果是No Hardware,表明没有安装下载电缆。

Z-Blaster下载电缆安装过程如下:

1. 把Z-ByteBlaster 插上电脑的并口,电脑会提示发现新硬件,手动安装Z-ByteBlaster 驱动,驱动文件位于[Quartus 安装路径]\altera\90\quartus\drivers\usb-blaster\x32
2. 安装驱动完成后,在Quartus 中点击 , 进入图F1.1所示的Hardware Setup对话框;
3. 在图示的Currently selected hardware: 下拉框中选择Byte-Blaster,最后点击“Close”关闭窗

口。



图F1.1 ByteBlaster II下载电缆安装

4. 亦可在Windows操作界面时，直接安装Z-Blaster下载电缆的驱动，方法类似普通硬件的安装，安装驱动在Quartus安装路径\altera\70\quartus\drivers\usb-blaster\x32。安装好后，启动Quartus软件下载时，需用上面同样方法选中下载线。

表一

序号	电路图上标号	FPGA管脚	I/O方向	外设名称	跳线及复用
1	LED1	50	O	8 个独立 LED 灯， 主板上的 LED1-8 与核心板上 LED1- 8 共用管脚	需要短接主板上 JP6 的 对应跳线
2	LED2	53	O		
3	LED3	54	O		
4	LED4	55	O		
5	LED5	176	O		
6	LED6	47	O		
7	LED7	48	O		
8	LED8	49	O		
9	KEY8	156	I	独立按键· 核心 板上 KEY1-4 与 主板上 KEY1-4 共用	
10	KEY7	158	I		
11	KEY6	141	I		
12	KEY5	143	I		
13	KEY4	124	I		
14	KEY3	123	I		
15	KEY2	122	I		
16	KEY1	121	I		
17	BEEP	175	O	蜂鸣器	
18	LCD_LIGHT	174	O	液晶显示控制	
19	LCD_EN	173	O		
20	DcMotorSpeed	140	O	直流电机 注意短 接 JP1 电源 跳线	
21	DcMotorA	139	O		
22	DcMotorB	138	I		
23	8563_INT	137	I	8563	
24	LM75_OS	136	I	LM75	

25	SEG0	169	O	七段数码管 段码	—
26	SEG1	170	O		
27	SEG2	167	O		
28	SEG3	168	O		
29	SEG4	165	O		
30	SEG5	166	O		
31	SEG6	163	O		
32	SEG7	164	O	七段数码管 位码	1C12 时要通过连接线 从主板上的 JP6 或核 心板上的 PACK 复用
33	DIG0	160	O		
34	DIG1	159	O		
35	DIG2	162	O		
36	DIG3	161	O		
37	DIG4	204	O		
38	DIG5	199	O		
39	DIG6	198	O	232 串口	—
40	DIG7	205	O		
41	UART_RXD	135	I	ADC	—
42	UART_TXD	133	O		
43	AD_nCS	134	O		
44	AD_DAT	131	I	DAC	—
45	AD_CLK	132	O		
46	DAC_LDAC	125	O		
47	DAC_LOAD	126	O		
48	DAC_DATA	127	O	步进电机 注 意短接 JP4 电源跳	1C12 时要通过连接线 从主板上的 JP6 或核 心板上的 PACK 复用
49	DAC_CLK	128	O		
50	ULN2003A	103	O		
51	ULN2003B	102	O		
52	ULN2003C	97	O		
53	ULN2003D	96	O	红外收发	1C12 时要通过连接线
54	IR_CLK	221	O		

55	IR_TX	81	O	通过 JP2 跳线来选	从主板上的 JP6 或核	
56	IR_RX	80	I			
57	RS485_DI	待定	O	485 接口	要通过连接线从主板上的 JP6 或核心板上	
58	RS485_RE_DE		O			
59	485_RO		I			
60	SD_WP	待定	I	SD/MMC 卡接口	要通过连接线从主板上的 JP6 或核心板上的 PACK 复用	
61	SD_INSERT		I			
62	SPI_MISO		I			
63	SPI_CLK		O			
64	SPI_MOSI		O			
65	SD_nCS		O			
66	SD_POWER		O			
67	VGA_VSYNC	待定	O	VGA 接口	要通过连接线从主板上的 JP6 或核心板上的 PACK 复用	
68	VGA_HSYNC		O			
69	VGA_R0		O			
70	VGA_R1		O			
71	VGA_R2		O			
72	VGA_G0		O			
73	VGA_G1		O			
74	VGA_G2		O			
75	VGA_B0		O			
76	VGA_B1		O			
77	LATTICE_STR	待定	O	LED 点阵接口	要通过连接线从主板上的 JP6 或核心板上的	
78	LATTICE_SI		O			
79	LATTICE_SCK		O			
80	MS_DATA	待定	IO	鼠标接口	要通过连接线从主板上的 JP6 或核心板上的	
81	MS_CLK		IO			
82	KB_DATA		IO	IO		键盘接口
83	KB_CLK					
84	RTL8019_INT	待定	I	以太网接口	要通过连接线从主板上的 JP6 或核心板上的	
85	RTL8019_nCS		O			
86	RTL8019_RST		O			
87	USBVIN	待定	I	USB 接口	要通过连接线从主板上的 JP6 或核心板上的 PACK 复用	
88	USB_SUSPD		IO			
89	USB_nRST		O			
90	USB_nINT		I			
91	USB_nCS		O			
92	P_IO1	待定	根据使用来定	主板上 PACK1 用户 IO 口	要通过连接线从主板上的 JP6 或核心板上的	
93	P_IO2					
94	P_IO3					
95	P_IO4					
96	P_IO5					
97	P_IO6					
98	EXT_nCS		O	PACK1 片选	PACK 复用	

### 核心板引脚分配

序号	引脚标号	FPGA 管脚	IO 口方向	功能	备注
1	SDRAM_A0	184	O	SDRAM 地址线	
2	SDRAM_A1	183			
3	SDRAM_A2	182			
4	SDRAM_A3	181			
5	SDRAM_A4	179			
6	SDRAM_A5	180			
7	SDRAM_A6	187			
8	SDRAM_A7	188			
9	SDRAM_A8	193			

10	SDRAM_A9	194				
11	SDRAM_A10	185				
12	SDRAM_A11	195				
13	SDRAM_A12	196				
14	SDRAM_D0	226	IO	SDRAM 数据线		
15	SDRAM_D1	225				
16	SDRAM_D2	224				
17	SDRAM_D3	223				
18	SDRAM_D4	222				
19	SDRAM_D5	215				
20	SDRAM_D6	214				
21	SDRAM_D7	213				
22	SDRAM_D8	216				
23	SDRAM_D9	217				
24	SDRAM_D10	218				
25	SDRAM_D11	219				
26	SDRAM_D12	227				
27	SDRAM_D13	228				
28	SDRAM_D14	233				
29	SDRAM_D15	234				
30	SDRAM_BA0	197	O	BA0、1 决定哪个		
31	SDRAM_BA1	186	O			
32	SDRAM_DQML(0)	207	O	输入/输出屏蔽 Masked		
33	SDRAM_DQMH(1)	208	O			
34	SDRAM_nRAS	202	O	命令的读入		
35	SDRAM_nCAS	203	O			
36	SDRAM_nWE	206	O			
37	SDRAM_CLK	144	O	SDRAM 时钟		
38	SDRAM_CKE	201	O	时钟使能		
39	SDRAM_nCS1	200	O	SDRAM1 片选		
40	SDRAM_nCS2	235	O	SDRAM2 片选		
41	D0	93	IO	Nor Flash 以及外扩的 数据总线		
42	D1	94				
43	D2	95				
44	D3	98				
45	D4	99				
46	D5	100				
47	D6	101				
48	D7	104				
49	D8	74				
50	D9	73				
51	D10	68				
52	D11	67				
53	D12	66				
54	D13	65				

55	D14	61			
56	D15	62			
57	A0	115	O	Nor Flash 以及外扩的地址总线	
58	A1	83			
59	A2	84			
60	A3	85			
61	A4	86			
62	A5	87			
63	A6	106			
64	A7	107			
65	A8	108			
66	A9	113			
67	A10	114			
68	A11	82			
69	A12	79			
70	A13	78			
71	A14	77			
72	A15	76			
73	A16	60			
74	A17	59			
75	A18	58			
76	A19	75			
77	A20	117			
78	A21	118			
79	A22	119			
80	nWE	105	O	写信号	
81	nOE	56	O	读信号	
82	FLASH_nCS1	120	O	Flash1 片选	
83	FLASH_nCS2	116	O	Flash2 片选	
84	FLASH_R/nB	29	I	Flash 读写忙信号	
85	SRAM_nCS1	88	O	SRAM1 片选	
86	SRAM_nCS2	57	O	SRAM2 片选	
87	UPPER_BYTE	64	O	高字节选通	

88	LOWER_BYTE	63	O	低字节选通	
89	I2C_SCL	177	IO	I2C 时钟	
90	I2C_SDA	178	IO	I2C 数据	
91	SYS_CLK0	28	I	时钟输入	
92	SYS_CLK2	153	I	时钟输入	
93	SYS_nRST	152	I	FPGA 复位输入	