

智能可控数字钟

-基于VGA, 声音传感器与温湿度传感器实现

学院: 电子与信息工程学院 专业: 计算机科学与技术

汇报人: xxx 汇报时间: 2020年1月7日

Contents



作品介绍



数字系统划分



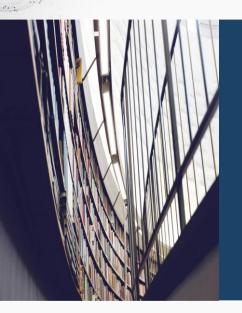
主要原理介绍



结果展示



作品介绍



选题原因:受到电脑开机长时间不操作会进入屏幕保护模式影响,并且会显示时间等信息,就打算做一个拥有屏幕保护多功能控制的数字钟。

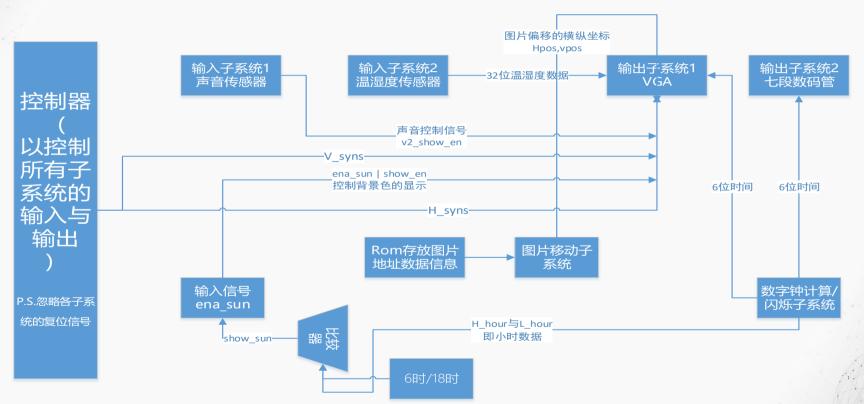
主要功能:在VGA上显示时间/温度与湿度。具体的显示效果受声音传感器、板子上的开关与时间影响。

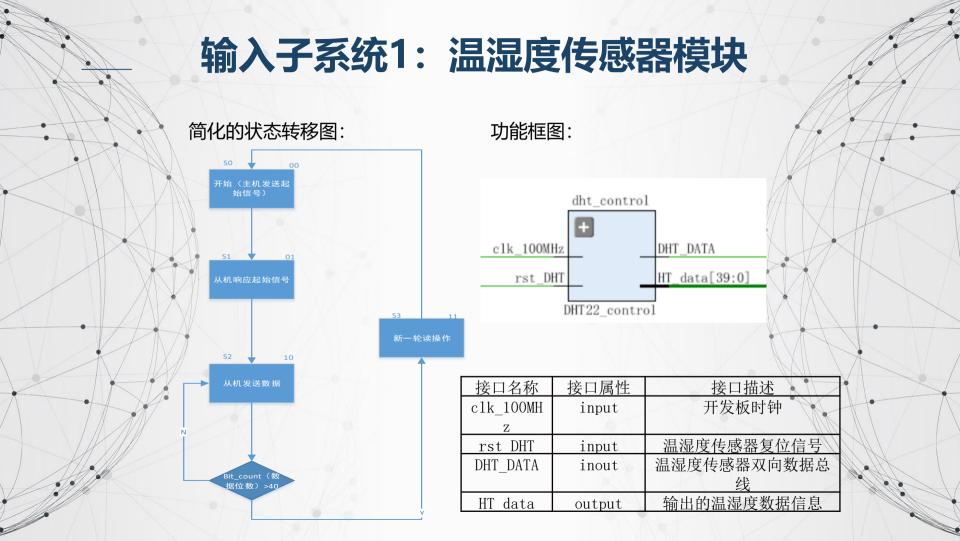
设计思路:采用自顶而下再自下而上的数字系统设计方法。首先划分子模块系统,数字钟模块,温湿度传感器模块,声音传感器控制模块,VGA显示模块,再根据每个模块的需要,向下一级划分。最后将它们同类归并整合,在顶层模块连接。



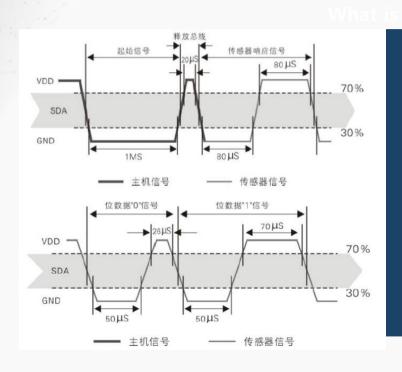
数字系统划分

系统总框图: 主要是两个输入子系统、两个输出子系统以及数字钟实现模块。





单总线通信协议 (1-wire)



主机与传感器间三个步骤即可完成读数据操作

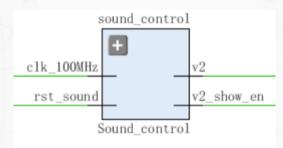
步骤1:上电后,需要等待2s不稳定状态,传感器测环境温度,主机不能发送信号,此后SDA数据总线一直处于高电平,时刻检测外部信号。

步骤2: 即单总线的初始化过程。主机即FPGA开发板发送初始信号,表现为拉低总线1毫秒。再释放总线,使得总线呈现20微秒高电平。随后从机即温湿度传感器发送响应信号,表现为先拉低总线80微秒,再拉高总线80微秒,通知外设准备接受数据。

步骤3:即主机读取数据过程。由数据总线 SDA 连续串行输出 40位数据,高位先出,依次传出湿度高位,湿度低位,温度高位,温度低位,主机根据 I/O 电平的变化接收 40 位数据。此后,从机输出50微秒低电平转为输入状态,等待一次的初始信号。

输入子系统2:声音传感器模块

功能框图:



接口信号定义:

接口名 称	接口属 性	接口描述
c1k_100 MHz	input	开发板时钟
rst_sou nd	input	声音复位信号
v2	inout	声音传感器双向数据 总线
v2_show en	output	输出的控制信号



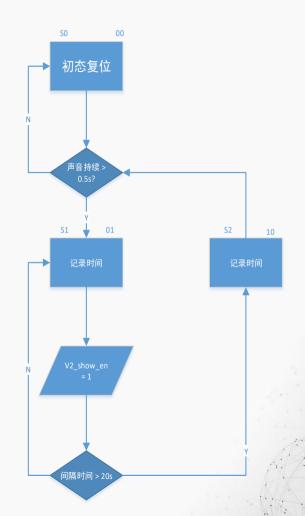
声音传感器操作原理

Sound Senor V2:

内含音频处理芯片LM386,可以音频放大、检测环境声音的有无或判断声音强度的大小(不能测量具体的分贝数)。

由于没有找到相关的Verilog资料,不知如何判断声音强度。但是碰巧发现在板子上插上声音传感器时,只要检测到声音,传感器上的灯就会亮。

所以利用这个特性,并猜测与温湿度传感器类似都只有一根总线,所以利用总线上读取的数据(即有无声音)进行一些延时操作,以达到控制目的。



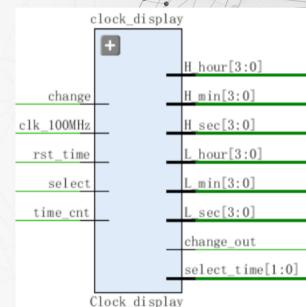
数字钟模块

包括两个子模块: 数字钟计算模块 (Clock_display) 和数字钟数码管显示模块 (display clock state)

数字钟计算整体包括自动计时与手动计时两部分。

计时的逻辑:

将系统100MHz分频为1Hz,即1s为单位。 自动计时将时,分,秒的每一位单独拿出来,从秒 开始依次向高位判断,如果超过了该位能表示数据 的最大值,该位归零,向前进1. 手动计时与自动计时最大的区别就是达到该位能表 达数据最大值时不要向前进1,只需归零。 手动计时的结果被自动计时保留。





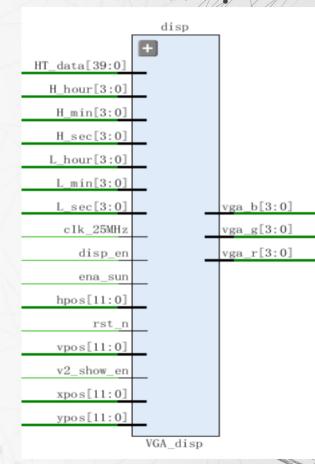
输出子系统1: VGA模块

该模块是仅此于顶层模块级别的模块。

其输入信号来自此前的数字钟模块的6个时间数据,温湿度传感器传来的40位数据,以及让图片移动的偏移量。除了数据信号输入,还有声音传感器传入的控制信号和N4板上的控制信号。

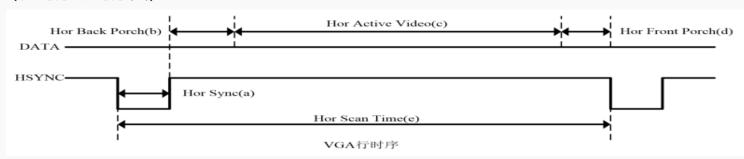
输出VGA的RGB三个通道数据。

功能框图:



VGA驱动原理

VGA的时序主要包括行时序与场时序两个部分。它的扫描过程是从左到右,从上到下 其中行时序主要包括:行同步、行消隐、行有效和行前肩 (行时序的时序图)





只有在行时序与场时序都处于显示阶段(即显示屏的区域内),才能说明显示输出信号有效。

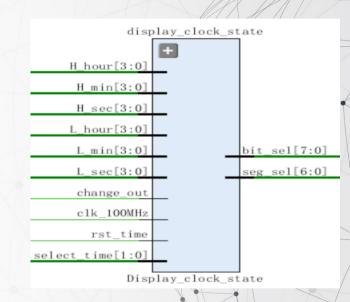
输出子系统2: 七段数码管模块

一、数码管上动态显示时间

具体的数字0~9是通过数码管段选(seg_sel)实现,即选择每个数码管上的灯,低电平有效。 而位选(bit_sel)就是选择哪一个数码管。 问题:如何让不同的数码管显示不同的数值并看

问题:如何让不同的数码管显示不同的数值并看 起来是同时显示。

利用人眼的视觉暂留,当不同数码管间变化时间 少于40毫秒,人眼就无法识别变化过程。对数码 管采用1000Hz的刷新频率,利用计数器型的状态 转移,每一个状态显示一位数码管,如此往复。



二、闪烁效果实现

最直观的想法就是把位选信号取反,位选信号同段选0标识亮, 1标识不亮。

采用一个临时的8位数据8'b1111_1111,每400ms取反一次,再与要闪烁的数码管取或运算。



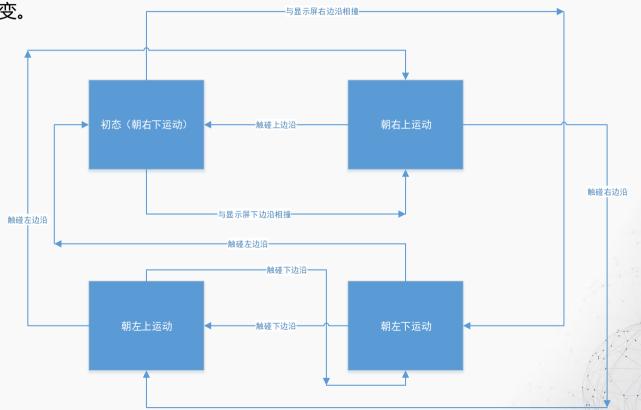
图片运动在VGA上运动

利用状态机实现,每当图片与显示器边沿相撞,则改变原有的运动状态。图片移动的关键在于图片相对

于原点横纵坐标偏移量的改变。

图片偏移量的是否改变取决于VS_negedge,即场时序的下降沿。

由于 VGA 规格 为800*525*60,即每秒刷新60次,所以图片的变化要与VGA刷新频率保持一致,否则图片会出现闪烁,甚至无法呈现出图像。



VGA上呈现字符数据

1. 与VGA上显示图片类似,同样需要调用IP核中的Block Memory Generator,用图片/字符的coe文件初始化一个单口Rom。

但不同的是图片是一个整体,只要按照一行一行的顺序(从左到右,从上至下) 传入Rom地址,就能获取对应像素点的RGB数据。

- 2. 这里获取字符的01数据需要借助字模转化工具,这里字模数据生成与VGA扫描规则同步,即从左到右,从上到下。
- 3. 而所有需要的字符都在同一个coe文件中,所以要显示字符,需要先判断显示的数字在什么区域,就把这个字符从字模中取出写入到这里,是从每个像素点出发。

此处VGA上显示的字符都是16*32,即每一行16个像素点,整个字有32行数据。 所以要定位到每个像素点,必须根据纵坐标找到是哪一行,再由横坐标找到是16 中的哪一个二进制数。



