

西安交通大学

数字电子技术综合实验报告

题 目：基于 EGO-1 口袋实验室的波形测量仪设计

电信 学院 自动化 系

学生姓名 吴思源

班级 自动化钱 71

学号 2171310846

指导教师 孙敏

2019 年 12 月

实验题目：基于 EGO-1 口袋实验室的波形测量仪设计

学生姓名：吴思源

指导教师：孙敏

摘 要

电路实验中经常需要用到测量波形的峰峰值与频率数据。本次实验中采用 EGO-1 口袋实验室板卡，使用 Verilog HDL 硬件描述语言对 FPGA 进行硬件编程，设计了一个简易的波形测量仪。波形测量仪可以借助 AD7476 模块对波形进行采样和模数转换，然后计算波形的峰峰值与频率，然后在数码管上显示测量数据或通过 UART 串口将测量数据以 ASCII 码发送到上位机显示。

硬件电路设计中采用了对待测波形上升沿计数的方法计算频率，采用了最大值最小值采样保持的方法进行峰峰值的测量与保持。对于二-十进制转换，使用了一种资源占用较少的方法进行实现。

波形测量仪可以对方波、三角波、正弦波等波形进行较为准确的测量。波形的峰峰值测量范围为 0~3.4V。频率的测量范围为 100Hz~50kHz，在高频部分的测量误差可以控制在 5% 以内。

关 键 词：波形测量 FPGA 数模转换 串口通信 进制转换

目 录

1 前 言	1
2 实验目的与实验内容	2
2.1 实验目的	2
2.2 实验内容	2
2.2.1 ADC 数据采集	2
2.2.2 峰峰值计算	2
2.2.3 频率计算	2
2.2.4 数码管数据显示	3
2.2.5 按键切换	3
2.2.6 UART 串口通讯	3
3 实验原理	4
3.1 频率计算原理	4
3.2 ADC 数据采集时序逻辑	4
3.3 串行通讯时序逻辑	5
3.4 二-十进制转换算法	5
4 系统设计逻辑	7
4.1 读取模块	7
4.2 计算模块	8
4.2.1 峰峰值计算	8
4.2.2 频率计算	8
4.3 显示模块	8
4.4 串口通信模块	9
4.4.1 发送字符	9
5 A/D 转换仿真时序图	10
6 资源占用分析	12
6.1 各类元件简介	12
6.1.1 LUT	12
6.1.2 FF	12
6.1.3 BUFG	12
6.1.4 IO	12
6.2 资源占用情况	13
6.3 资源占用对比分析	14

6.3.1 与杨子轩同学进行对比分析	14
6.3.2 与蒋施达同学对比分析	15
7 设计与实验过程中存在的问题与改进	16
7.1 二-十进制转换的问题	16
7.1.1 使用直接除法带来的问题	16
7.1.2 按位转换	16
7.2 精确度的问题	16
7.2.1 测峰峰值的精确度	16
7.2.2 测频率的精确度	16
8 实验总结	17
9 心得与建议	18

1 前言

在电路实验中经常需要对未知波形进行测量，并且得到其特征数据，方便后续实验的进行。峰峰值描述了波形中最大值与最小值之间的电压差，频率描述了每秒波形的周期个数，二者对于描述波形特征意义重大。因此，只需要测量出波形的峰峰值和频率，就可以近似的了解未知信号的特征，进行后续分析。

可编程逻辑门阵列（FPGA）可以通过硬件编程设计特定的电路，快速高效的实现某种特定的功能。FPGA 的编程主要是使用 Verilog HDL 硬件描述语言来描述的。Verilog HDL 可以通过编程实现译码器、加法器、比较器、ALU 等各种组合逻辑电路以及触发器、锁存器等各种时序逻辑电路，并且将各个模块组合成需要功能的电路。

本实验中主要使用 Verilog 硬件编程语言，在 EGO-1 口袋实验室中进行 FPGA 编程，实现了一个简易的波形测量装置。波形测量装置可以实现方波、三角波、正弦波等波形的测量，并且读取其峰峰值、频率数值，在 EGO-1 板卡上的数码管上显示测量出的结果，并且使用按键进行重置和显示切换。此外，此程序还可以实时将数据通过 UART 协议串行传送到上位机的串口通信软件中进行显示。

2 实验目的与实验内容

2.1 实验目的

本实验主要对 FPGA 硬件编程，实现简易波形测量装置。将待测波形通过 EGO-1 板卡上的 ADC 模块（模数转换模块）进行数据采集，然后通过 FPGA 硬件电路设计对于采集到的数字数据进行处理，使用设计出的电路计算出待测波形的峰峰值、频率，之后在数码管上进行十进制显示。显示时需要能够使用按键切换峰峰值和频率的数据显示。在完成显示后，还可以将频率和峰峰值数据通过板卡上的串口使用 UART 协议发送到上位机，在上位机上显示频率和峰峰值数据。

2.2 实验内容

本实验中包含了 ADC 数据采集、峰峰值计算、频率计算、数码管数据显示、按键切换、UART 串口传输共六部分内容。

2.2.1 ADC 数据采集

实验中使用了 AD7476A 的 Pmod 外接采集模块板进行数据采集操作。实验中需要阅读 AD7476A 的数据手册，了解到 AD7476A 板卡的最高采样频率是 20MHz，串行时钟的时钟周期应为数十纳秒量级。

2.2.2 峰峰值计算

实验中对于波形需要测量其峰峰值。测量方法是巧妙利用锁存器和触发器，将采集到的最大值与最小值锁存器起来。由于输入波形的峰峰值有可能随时改变，需要设计合理的时序将其不断更新，即每隔一定周期后清零后再重新测量。

2.2.3 频率计算

实验中需要对采集到的数据计算频率。由于待测波形的形状不定，频率未知，需要对其先使用施密特触发器转换为方波，之后统计其某一段时间内的上升沿个数。计算出上升沿个数后与系统时钟在同样时间段内的上升沿个数做商，然后乘以系统的时钟频率，就可以得到待测波形的频率数据

2.2.4 数码管数据显示

实验中需要将波形显示到数码管上，这一部分需要合理操纵八位七段数码管，使其分时点亮，然后动态显示测得的峰峰值和频率。峰峰值显示的单位为毫伏特，频率显示的单位为赫兹。由于板卡的工作电压 VCC 为 3.4 伏特，因此峰峰值显示的最大值为 3400 毫伏特，最小值为 0 毫伏特。频率的显示范围由采样周期和频率计算周期决定，这一部分会在实验原理与系统逻辑部分中详述。

2.2.5 按键切换

实验中需要将峰峰值和频率显示到数码管上，并且可以通过按键进行切换。实验中通过复杂的时序触发关系，实现了点按特定的某个键后进行的数据显示切换。

2.2.6 UART 串口通讯

实验中需要将测量到待测波形的峰峰值、频率等数据发送到上位机，通过上位机的串口接收工具进行显示。串口的发送端为板卡上的 UART 端口，波特率为 9600，采用 ASCII 码规则发送六比特频率数据、四比特峰峰值数据和六比特其它显示数据。

3 实验原理

3.1 频率计算原理

将采集到的离散待测波形通过施密特触发器转换为方波波形，高门限设置为最大值的 $\frac{2}{3}$ ，低门限设置为最大值的 $\frac{1}{3}$ ，通过施密特触发器后的波形如图所示。

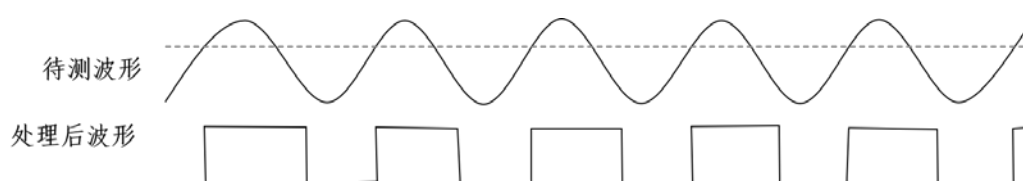


图 1-1 施密特触发器处理前后的波形示意

这样可以将未知的待测波形转化为已知的方波波形，然后可以方便地提取波形的上升沿和下降沿，进行频率的计算。

对系统时钟进行分频，得到参考时钟周期 T_1 ，当参考时钟为高电平时，对这一段时间内经过施密特触发器处理过的未知波形的上升沿进行计数，得到上升沿个数 C_1 ，然后记录同样时间内的系统时钟的上升沿个数 C_2 ，将二者做商得到待测波形的频率为

$$f = \frac{C_1}{C_2} f_{sys} \quad (3.1)$$

其中系统的时钟频率为 100MHz，系统时钟频率为 10ms，通过这个方法就可以准确计算出待测波形的频率。频率测定的下限由参考时钟的频率决定，上限由 AD 采样的时钟频率决定。

3.2 ADC 数据采集时序逻辑

如图所示，片选信号（CS）低有效，时钟信号（SCLK）不大于 20MHz，数字数据信号（SDATA）为串行数据传输信号。

当片选信号（CS）拉低时，AD7476 模块开始向板卡传送数据，以时钟周期 SCLK 为参考，前四个周期向板卡传送四个零，后 12 个周期向板卡传送 12 位的数字数据，数字量是 AD7476 模块接收到的模拟信号进行 AD 转换后得到的。当 AD7476 向板卡传输完 12 位数字信号后，SDATA 自动终止发送，此时应该将片选信号拉高。

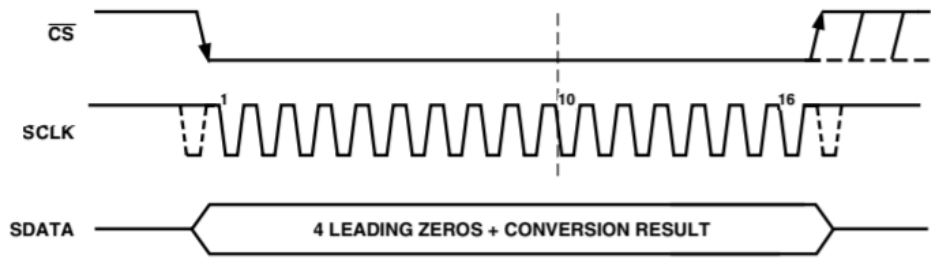


图 3-2 AD7476 时序图

3.3 串行通讯时序逻辑

UART 是实现设备之间低速数据通信的标准协议之一。采用 UART 的两设备之间不需要其它时钟信号进行通讯，只需要两条数据线信号线接收线(RXD)和发射线(TXD)就可以完成数据的相互通信，并且接收和发送可以同时进行，也就是全双工。在发送器空闲时间，数据线处于逻辑 1 状态，当提示有数据要传输时，首先使数据线的逻辑状态为低，之后发送 8 个数据位、一位校验位以及一位停止位，停止位用于标示一帧的结束。接收过程亦类似，当检测到数据线变低时，开始对数据线以约定的频率抽样，完成接收过程。



图 3-3 UART 串口通讯数据帧格式

3.4 二-十进制转换算法

二进制到十进制的转换采用了从左向右移位的方式进行。
实现的算法如下

1. 将二进制的最高位放到寄存器的最低位中
2. 将寄存器左移一位，并将二进制码的下一位放到寄存器的第一位中
3. 找到左移后的对应十进制个位、十位、百位、千位的四位 BCD 码
4. 判断各位是否大于 4，若是则将其加三
5. 若二进制数倒数第二位放入到了寄存器中，则跳转到第六步；若未放入，则跳转到第二步
6. 将二进制左移一次，然后输出转换后到数据

算法的原理，即将二进制码进行按权展开，然后从低到高不断乘 2 装载到输出寄存器中。如果对应的十进制 BCD 码乘 2 后大于 10，则加 6 使其进位。若不大于 10，则保持原来的数据。算法流程的一个示例图如下：

Operation	Hundreds	Tens	Units	Binary	
HEX				F	F
Start				1 1 1 1	1 1 1 1
Shift 1			1	1 1 1 1	1 1 1
Shift 2			1 1	1 1 1 1	1 1
Shift 3			1 1 1	1 1 1 1	1
Add 3			1 0 1 0	1 1 1 1	1
Shift 4		1	0 1 0 1	1 1 1 1	
Add 3		1	1 0 0 0	1 1 1 1	
Shift 5		1 1	0 0 0 1	1 1 1	
Shift 6		1 1 0	0 0 1 1	1 1	
Add 3		1 0 0 1	0 0 1 1	1 1	
Shift 7	1	0 0 1 0	0 1 1 1	1	
Add 3	1	0 0 1 0	1 0 1 0	1	
Shift 8	1 0	0 1 0 1	0 1 0 1		
BCD	2	5	5	http://blog.csdn.net/T200503028	

图 3-4 使用上述算法将 $(11111111)_2$ 转换为十进制

4 系统设计逻辑

系统分为读取模块、计算模块、显示模块、串口通讯模块、进制转换模块五部分实现，模块示意图如下图所示。

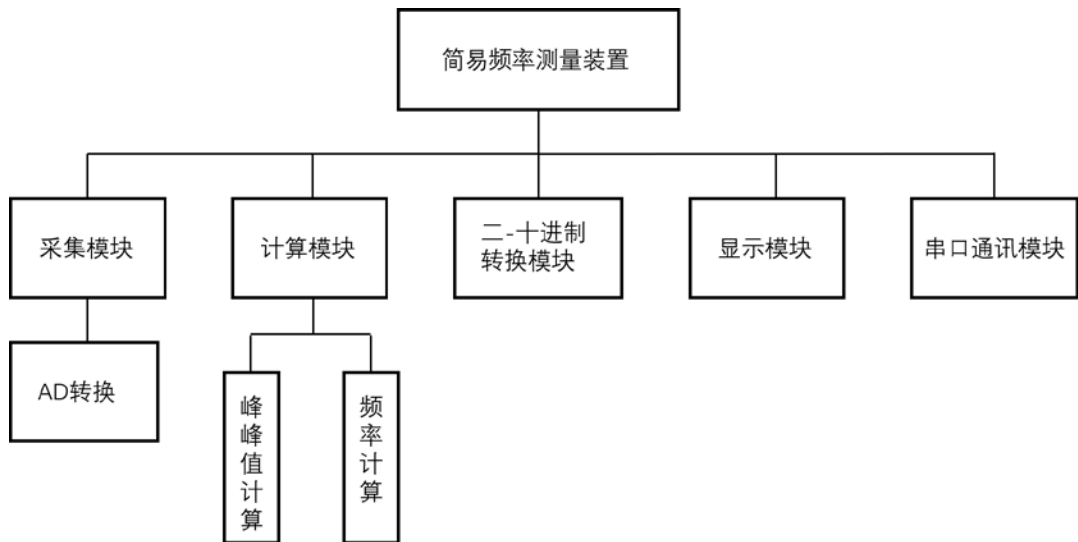


图 2-1 系统模块示意图

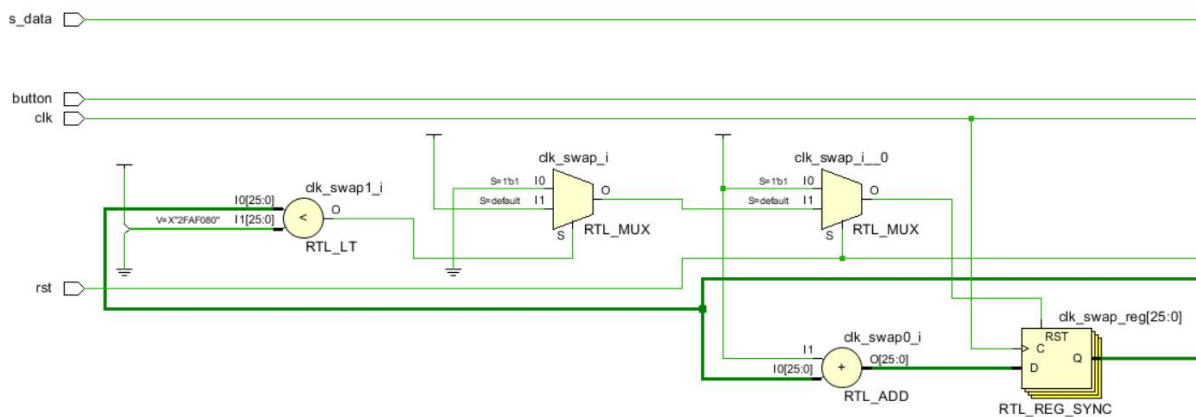


图 3-2 时钟分频部分 RTL 设计图

系统 RTL 设计图

4.1 读取模块

本模块作用为从 AD 转换芯片中按照特定周期的采样时钟信号读取数模转换后的

数字信号并且存入寄存器中，便于下一步的分析处理。这一模块在 `ad_reader.v` 文件中定义的模块实现。

4.2 计算模块

本模块将读取到的数字信号分别进行计算，得到峰峰值与频率数据。

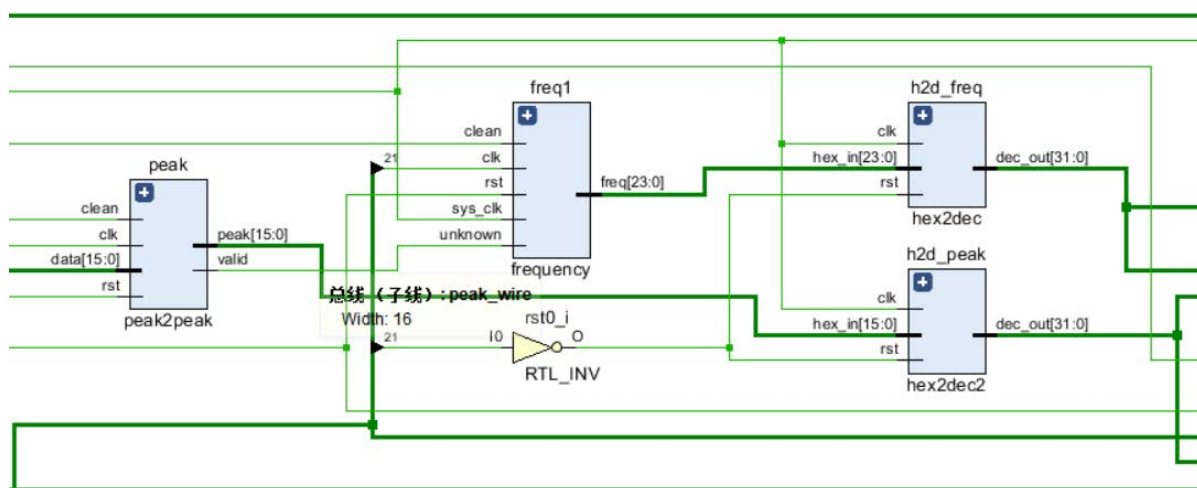


图 3-3 计算部分 RTL 分析

4.2.1 峰峰值计算

这一模块统计某一段时间内读取到的数字信号的最大值与最小值，然后作差计算峰峰值。由于需要动态更新，因此需要每隔一段时间将存储最大值、最小值、峰峰值的寄存器清零重新读取。在这一部分模块中，还计算了施密特触发器的高低门限，并采用施密特触发器将读取到的数字信号转成了方波信号，便于下面频率的计算。这一模块在 `peak2peak.v` 中实现

4.2.2 频率计算

这一部分的算法即计算参考信号为高时的待测信号的上升沿个数，与 100MHZ 系统时钟在同样时间内上升沿个数做商，然后乘以系统时钟的频率得到待测信号的频率。这一模块在 `frequency.v` 中实现。

4.3 显示模块

这一部分对数码管进行操作，使其以毫秒数量级按位刷新。对于频率数据，显示六位，单位为 Hz；对于峰峰值数据，显示四位，单位为 mV。显示频率数据时，将高两位使能无效，对低 6 位不断刷新，分时显示每一位频率数据；显示峰峰值数据时，将高

4 位使能无效，对低 4 位不断刷新，分时显示每一位峰峰值数据；

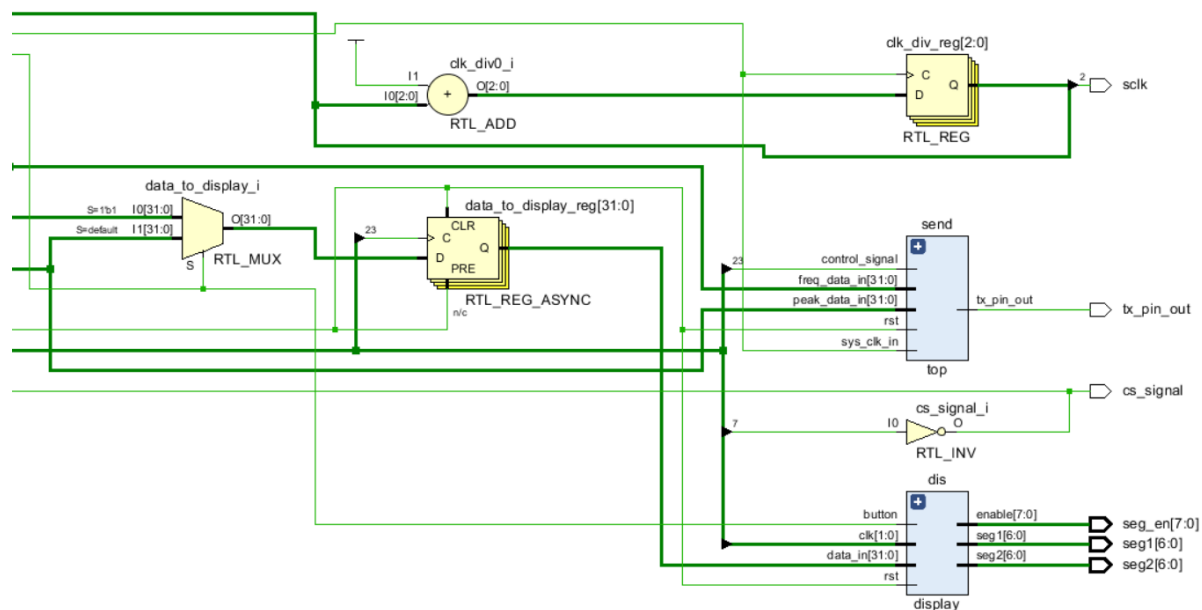


图 3-4 发送及显示部分 RTL 分析图

4.4 串口通信模块

4.4.1 发送字符

串口通信模块使用 ASCII 码发送显示数据，接收端需要为解析 ASCII 码。根据输入的时序，每隔 100ms 发送一次数据，时钟采用系统时钟的分频实现。每次发送数据时，发送 16 次数据，每次 1 个字节。前 2 次发送字母 f，代表频率；之后发送 6 次频率数据，每次发送从高到低的一位频率数值；发送两位 p，代表开始发送峰值数据；然后发送 4 次峰峰值数据，每次从高到低发送一位峰峰值数据；最后发送两次换行符，代表字符已经发送完毕。

发送到上位机的字符串与显示到数码管上的数据应当内容一致。

5 A/D 转换仿真时序图

仿真程序中，通过模仿 AD7476 的工作时序，向波形测量程序输入了一个方波信号。方波信号的低电平向 s_data 引脚发送十二位二进制数 000000110101，高电平向 s_data 引脚发送 111111111111，仿真方波信号的频率为 10kHz。

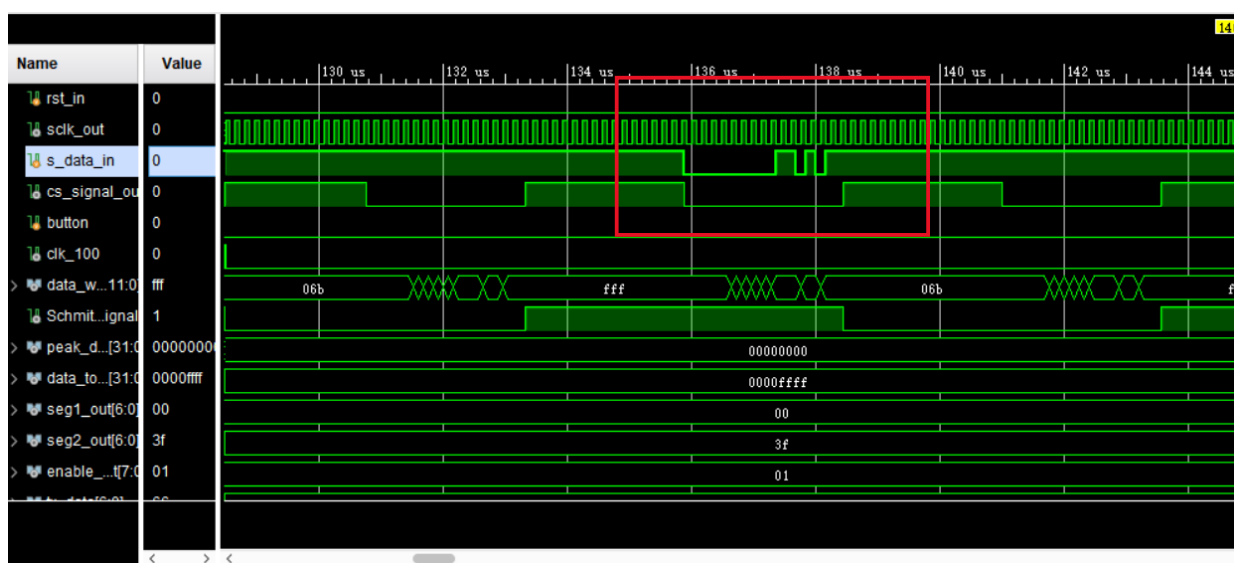


图 5-1 AD 转换仿真时序图

A/D 转换仿真时序图如上图所示。红框中的内容是 AD 转换读取信号的部分。相关信号的含义如下：

- sclk_out 信号为板卡输出到 AD7476 上的时钟信号
- s_data_in 是 AD7476 输入板卡的数据信号
- cs_signal_out 是板卡输出到 AD7476 的片选信号
- rst_in 是板卡输出给 AD7476 的重置信号

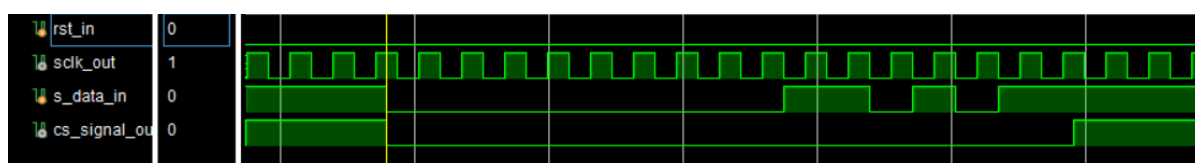


图 5-2 AD 转换中关键信号时序图

读取 AD7476 数据的操作时序如下：

当片选信号（CS）拉低时，AD7476 模块开始向板卡传送数据，以时钟周期 SCLK 为参考，前四个周期向板卡传送四个零，后 12 个周期向板卡传送 12 位的数字数据，

数字量是 AD7476 模块接收到的模拟信号进行 AD 转换后得到的。当 AD7476 向板卡传输完 12 位数字信号后，SDATA 自动终止发送，此时应该将片选信号拉高。

6 资源占用分析

FPGA 的结构类似于数电课程中学习过的 CPLD，主要由可组态逻辑块（CLB）、输入输出块（IOB）以及可编程内部连线器（PI）组成。由于不同的 Verilog HDL 程序的编写会产生不同的硬件设计，而不同硬件设计实现占用的资源不同，合理分析调配资源显得尤为关键。

6.1 各类元件简介

6.1.1 LUT

LUT 即查找表（Look Up Table），属于组合逻辑结构。LUT 的本质类似 RAM。使用 Verilog HDL 语言描述了逻辑电路之后，Vivado 会自动计算此逻辑电路所有可能出现的情况，并自动将真值表输入 RAM。这样当硬件电路的输入中出现某一特定信号时，FPGA 会访问 RAM 并从 RAM 中寻找该输入对应的输出。

6.1.2 FF

FF 即触发器。触发器是 FPGA 中最基本的时序逻辑器件。如果在 Verilog 中用到了大量的触发条件和 `always` 块，则 FF 的资源占用率会相对较高。

6.1.3 BUFG

BUFG 是 BUFG 是 IO 时钟网络，它只能驱动 IO Block 里面的逻辑，不能驱动 CLB 里面的 LUT，REG 等逻辑。

6.1.4 IO

IO 是调用的 IO 接口的数量。这一部分的资源占用与需求强相关。本实验中占用的 IO 端口数目如下表所示

表 5-1 IO 资源占用及引脚详单

引脚功能	寄存器名称	FPGA IO PIN
低四位数码管显示	seg1	D4 E3 D3 F4 F3 E2 D2
高四位数码管显示	seg2	B4 A4 A3 B1 A1 B3 B2
八位数码管使能	seg_en	G6 E1 F1 G1 H1 C1 C2 G2
时钟信号输入	clk	P17
重置信号输入	rst	R15
片选信号输出	cs_signal	H17
ADC 信号输入	s_data	G17
ADC 时钟输出	sclk	J13

串口发送模块	tx_pin_out	T4
显示切换输出	button	U4

6.2 资源占用情况

下面是本人编写的 FPGA 中程序的资源占用情况。由于程序中采用了较规范的书写格式，因此资源占用量相对较少。

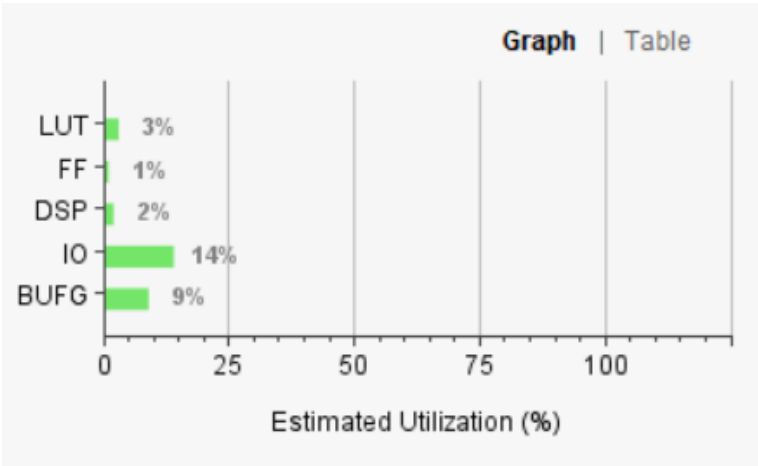


图 5-3 综合后预计资源占用情况图

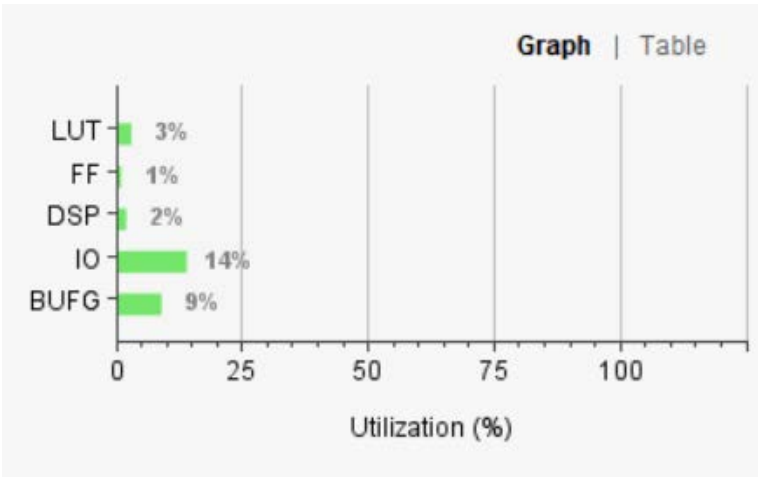


图 5-4 硬件实现后资源占用情况

表 5-2 资源占用情况表

资源	占用量	占用率%
LUT	716	3.44
FF	267	0.64

DSP	2	2.22
IO	29	13.81
BUFG	3	9.38

Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	DSP s (90)	Bonded IOB (210)	BUFGCTRL (32)
▼ N adconvertor	716	267	8	2	29	3
> [I] freq1 (frequency)	10	48	3	2	0	0
[I] h2d_freq (hex2dec)	46	37	1	0	0	0
[I] h2d_peak (hex2dec2)	27	38	0	0	0	0
[I] peak (peak2peak)	543	26	4	0	0	0
[I] reader (ad_reader)	42	17	0	0	0	0
> [I] send (top)	32	40	0	0	0	0

图 5-5 各模块资源占用情况

从上表中可以看到，我的 LUT 占用主要集中在峰峰值计算模块 peak 中，而二-十进制转换、读写模块、峰峰值计算模块的 LUT 计算较少。

6.3 资源占用对比分析

6.3.1 与杨子轩同学进行对比分析

杨子轩同学的电路和程序设计中使用了大量的组合逻辑元件，因此其 LUT 数目惊人的高。他的程序中使用的 LUT 元件的个数是本人设计的 2.7 倍。时序逻辑器件的个数也比本人的设计多 23%。参考他的代码，他在 2-10 进制转换部分使用了大量的除十取余的运算来进行运算，自然他程序中的 LUT 数目会惊人的高。

Resource	Utilization	Available	Utilization %
LUT	1983	20800	9.53
FF	329	41600	0.79
DSP	1	90	1.11
IO	18	210	8.57

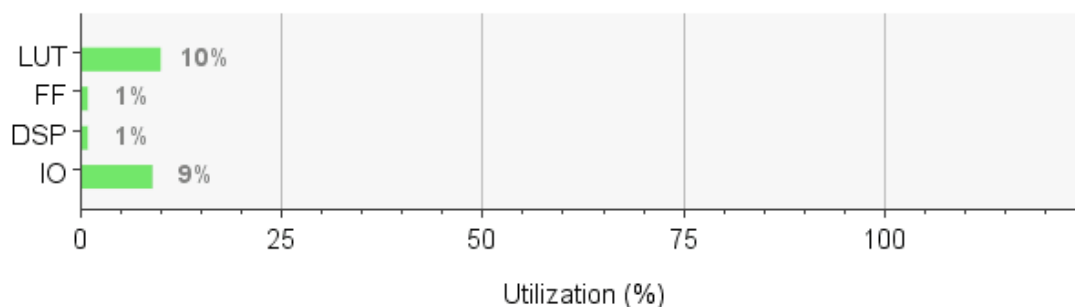


图 5-3 杨子轩同学资源占用情况表

6.3.2 与蒋施达同学对比分析

蒋施达同学的程序实现非常完美，他使用了至少 6 个不同分频的时钟，实现了较高精度的频率和峰峰值测量。他的设计中，每 5.12s 记录一次峰峰值，然后刷新峰峰值数据。如果当前峰峰值大于显示的峰峰值就立即刷新。（相当于存了两个峰峰值，一个是上个 5.12s 区间的峰峰值，一个是这个 5.12s 区间的峰峰值）在他的设计中，还增加了校准键与采样保持键。按下校准键后，显示周期的输出变成 1.0000，实现周期测量的校准；按下采样保持键，峰峰值只会变大不会变小。由于他实现了校准等复杂的组合逻辑功能，因此他的 LUT 使用量非常高，大约是我使用的 LUT 的 3.2 倍。不过，由于他的设计实现了待测信号 2.5000-0.0001s 周期的测量，而且精度很高误差非常小。因此他的方案不失为一个优秀的实现方案。

Resource	Utilization	Available	Utilization %
LUT	2286	20800	10.99
FF	353	41600	0.85
DSP	3	90	3.33
IO	33	210	15.71
BUFG	3	32	9.38

图 5-4 蒋施达同学资源占用情况表

7 设计与实验过程中存在的问题与改进

7.1 二-十进制转换的问题

7.1.1 使用直接除法带来的问题

最开始使用按位除法进行进制转换，即将所得数据除十取余、除百取余，按照这种方法进行进制转换会耗费大量资源，不仅仅需要预先分配多个高位的寄存器，还需要预先考虑到频率数据为不同位数时显示出现的问题。

7.1.2 按位转换

采用上述的按位转换方法可以减少资源占用，减少寄存器变量的使用。但是这种方法对于时序有较高要求。在实验中我使用了系统时钟来驱动二-十进制转换模块，尽量减少二-十进制转换对于时序的占用。

7.2 精确度的问题

7.2.1 测峰峰值的精确度

峰峰值的测定使用的是此时已知的最大采样值与最小采样值相减的方式进行计算。但是由于产生的波形可能会在某一个周期出现毛刺现象，对于峰峰值的精确度有较大影响。虽然在程序中采用了每隔 100ms 清空缓存重新测量峰峰值的办法，但是仍然会出现 10%~20% 的测量误差。

7.2.2 测频率的精确度

频率的精确度在较低频时（100Hz 以下），有较大误差，原因是采用了对待测信号和系统时钟的脉冲上升沿进行计数后做商的方法进行计算，而系统时钟的频率非常高，使用电路进行除法运算时会有一定的舍入误差。而且由于使用的参考时钟信号频率较大，为 100Hz，频率小于 100Hz 的信号均不可进行计数。

在较高频的时候误差很小，大约为 5% 以内，主要是待测信号在 1kHz 到 50kHz 之间时，由于高频时脉冲上升沿计数位置的误差可以忽略不计，因此这种方法对于高频信号的优势很大。

8 实验总结

本次数电综合实验实现了一个简易的波形测量装置，可以测定 100Hz~50kHz 的频率数据与 0V~3.3V 的峰峰值数据，并将数据同时显示在数码管上和上位机上，并且可以实现按键切换数码管的数据显示。

实验中采用了对待测波形上升沿计数的方法计算频率，采用了最大值最小值采样保持的方法进行峰峰值的测量与保持。对于二-十进制转换，使用了一种资源占用较少的方法进行处理。实验中选取了合适频率测量的算法，在较高频率测量中控制误差在 5% 以内，而且资源占用远小于其它同学。

这次实验中，我对 Verilog HDL 语言的编写风格和编程思路有了更加深刻的理解。对于 **reg** 变量和 **wire** 变量理解与使用更加清晰，关于 **assign** 语句下的同步清零和异步清零运用的更加娴熟，对于数码管显示、串口通信、ADC 转换等模块进行了综合功能实现与调用。我感觉收获主要有几个方面。一个是在做最后综合大作业的时候，是模仿课本（旧版）P310 和 P328 页 A/D 转换和 D/A 转换的程序书写的，通过模仿例程的书写方式，思考例程的逻辑，大大减少了我代码中的错误。一个是我是先写仿真，仿真通过之后再进行实验室的上板调试的。在仿真代码中，我定义了一个方波信号，然后测量这个仿真的方波信号的各项参数，再进行调试的。这种方法大大提高了我调试的效率，硬件调试中很快就调出了理想的结果。

9 心得与建议

对实验的心得与建议：实验的难度梯度设置的不合理，初期的实验（指 TTL 与非门、CMOS 与非门部分）难度适中，只需要仔细预习便可以做出。中期的部分实验，（如 Verilog 的基础实验部分，如 3-8 译码器等）较为简单，也便于同学掌握。但是后期的 UART 串口通讯、AD/DA 转换等模块，由于相关资料太少、或资料鱼龙混杂（例如书上的 ADC 程序并不一定可以调通），而且调试复杂（经常是仿真通过但是上板之后程序不正常工作），会耗费大量时间。希望后期的程序可以提供更多资料参考与调试帮助，以免浪费太多不必要的时间。

另外，对于 **Verilog** 语言的学习，也希望教研室可以提供更多更加详细的参考资料。对于某些错误的编程语法与编程逻辑（这里是指与其它高级程序设计语言思路不同的地方），希望教研室不要泛泛而谈，而是详细指出，及时纠正、反馈同学们对于 Verilog 语言的编写错误，以防同学们未意识到自己的程序编写错误和语法错误盲目上板调试、浪费大量时间。对于多模块复杂时序的语法和程序编写，希望可以提供更多更加详尽的范例。例如，可以先学习阅读实验前提供的部分 UART 接收代码，然后基于这份代码写出 UART 发送程序并调试通过。