

# 基于 FPGA 的电磁超声脉冲压缩检测系统

赵超凡 钱孜洋 杨冉

## 第一部分 设计概述

### 1.1 设计目的

电磁超声技术相对于传统的压电超声技术具有无需耦合剂,可实现非接触式测量等诸多优点。但是由于电磁超声换能效率低,匹配良好的换能器产生的回波信号一般也仅有数十微伏,这严重限制了电磁超声技术的发展与应用。脉冲压缩技术广泛应用于雷达信号处理中,能够有效提高雷达信号信噪比,提升雷达距离分辨力和抗干扰性能。因此本设计基于 ZYNQ-7020,搭建电磁超声测厚系统,并应用脉冲压缩技术,有效提高回波信号的信噪比,实现对金属试件的精准测厚。

### 1.2 应用领域

电磁超声测厚系统可对金属试件进行精准测厚,而厚度对于管道、承压设备等金属制材而言有着重要的意义:厚度的减薄容易导致裂纹、断裂等缺陷,进而导致爆炸等重大安全事故。因此,对金属制材的厚度进行检测,对于保障工业设备的健康稳定运行,推动工业高质量发展具有重要意义。电磁超声技术具有无需超声波耦合剂的技术优势,可以实现非接触式测量,因而还适用于高速移动的自动化检测场景。

### 1.3 主要技术特点与创新

- (1)电磁超声测厚系统无需超声波耦合剂,检测时无需对试件进行表面处理,相对压电超声,简化了测量流程,且具有更广的应用范围。
- (2)采用脉冲压缩技术,编码方式选择互补格雷码,理论上码长可无限长,通过合理的构造方法可以得到足够长的编码序列,能显著提高回波信号的信噪比。
- (3)基于 ZYNQ-7020 搭建电磁超声测厚系统,充分发挥了 FPGA 和嵌入式的优点,扩展性强,可进一步开发为便携式仪器,具有较高的商业价值。

### 1.4 关键性能指标

本文设计的基于 ZYNQ-7020 的电磁超声脉冲压缩检测系统具有如下技术指

标:

- (1) 检测对象: 5-100mm 厚金属试件
- (2) 测厚精度:  $\leq 0.2\text{mm}$
- (3) 检测速度:  $\leq 10\text{s}$
- (4) 信噪比:  $\geq 6\text{dB}$

## 第二部分 系统组成及功能说明

### 2.1 系统整体介绍

电磁超声测厚系统结构框图如图 2-1 所示。系统以 ZYNQ 7020 为核心，配备高压模块、发射电路、接收电路与 ADC 模块、电磁超声换能器。

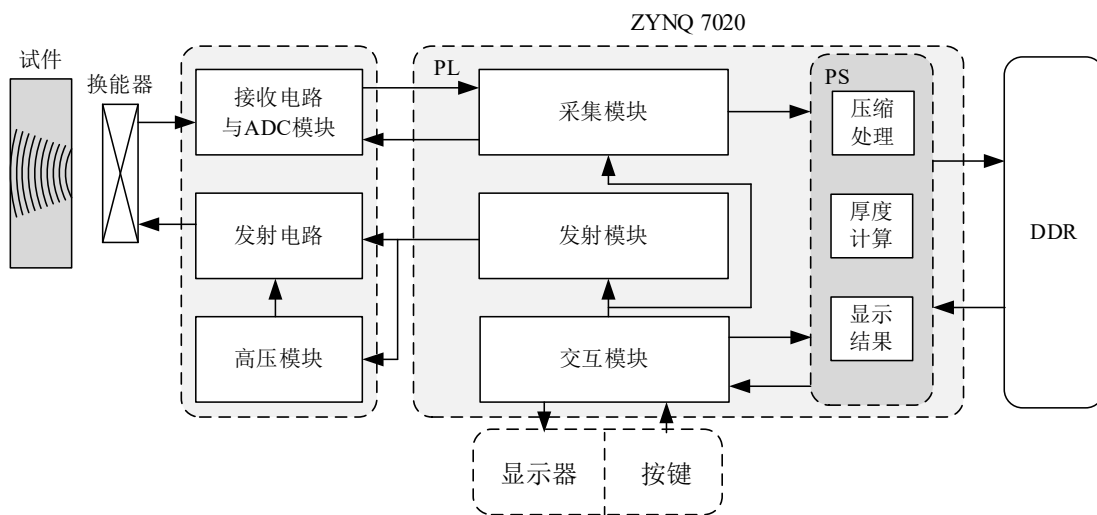


图 2-1 系统整体结构框图

工作过程中，ZYNQ 控制高压模块产生 100V 直流信号，再由发射电路逆变为  $\pm 100\text{V}$  高频交变信号，通过电磁超声换能器激励超声波；换能器接收到的回波信号由接收电路进行滤波与放大，并由 ADC 模块转换为数字信号，在 ZYNQ 中进行脉冲压缩处理和厚度计算。最后，将结果显示在液晶屏上。DDR 主要用于数据的缓存，用户可通过按键和显示器进行人机交互。

在本系统中，ZYNQ 的主要作用包括：1)对发射接收电路的时序控制；2)回波信号采集；3)脉冲压缩算法处理；4)被测试件的厚度计算；5)人机交互。其中，在 PL 端设计了发射模块、采集模块用于对发射接收电路的时序进行控制，交互模块用于读取用户的输入信息的读取和显示器的控制。在 PS 端设计了脉冲压缩处理算法和厚度计算算法，通过算法提高回波信号的信噪比并自动计算厚度。

### 2.2 脉冲压缩原理

脉冲压缩技术的基本原理如图 2-2 所示，系统通过发射频率调制或相位调制的一系列脉冲信号，而在接收回波信号时采用与发射信号相匹配的滤波器对其进行压缩处理，使得大时宽的信号转换成为窄脉冲信号，从而实现发射接收信号的脉冲压缩。

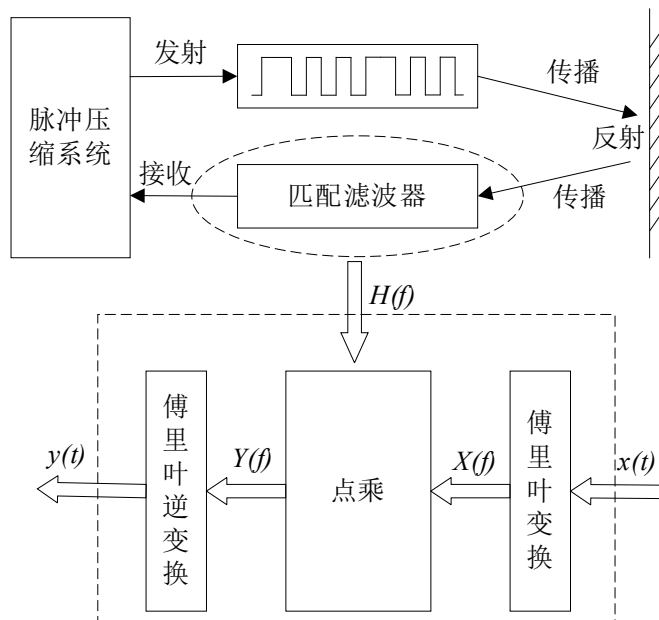


图 2-2 脉冲压缩技术的原理框图

由于发射脉冲为一系列脉冲信号，与单脉冲发射信号相比，发射信号的总功率远大于单脉冲发射信号，这一系列脉冲信号反射回来后经过一特定的匹配滤波器，能够使其还原成窄脉冲信号，故不会失去距离分辨力。因此脉冲压缩是一种不需要高能量窄脉冲所需要的高峰值功率，就可同时实现宽脉冲的能量强度和窄脉冲的高分辨力的技术，对于解决电磁超声换能效率低从而导致信号信噪比低问题有着重要意义。

脉冲压缩信号处理过程如图 2-2 中所示，输入信号  $x(t)=s(t)+n(t)$ ，其中  $s(t)$  为有效信号， $n(t)$  为噪声信号； $s(t)$  的频谱密度函数为  $S(f)$ ； $y(t)$  是输出的有效信号和噪声信号之和， $y(t)=s_0(t)+n_0(t)$ ， $s_0(t)$  是输入有效信号经过压缩处理后的信号，处理过程可以表达为：

$$s_0(t) = \int_{-\infty}^{+\infty} H(f)S(f)e^{j2\pi ft} df \quad (2-1)$$

经处理后信号中的噪声功率谱密度等于输入信号中的噪声功率谱密度乘以匹配滤波器系统传递函数模块的平方，由此得到输出信号中噪声的表示式为：

$$N_0 = \frac{n_0}{2} \int_{-\infty}^{+\infty} |H(f)|^2 df \quad (2-2)$$

因此，在采样时刻  $t_0$  经处理后有效信号的瞬时功率与噪声的平均功率之比为：

$$r = \frac{|s_0(t_0)|}{N_0} = \frac{\left| \int_{-\infty}^{+\infty} H(f)S(f)e^{j2\pi ft} df \right|^2}{\frac{n_0}{2} \int_{-\infty}^{+\infty} |H(f)|^2 df} \quad (2-3)$$

由施瓦尔兹不等式可知：

$$\left| \int_{-\infty}^{+\infty} f_1(x)f_2(x)dx \right|^2 \leq \int_{-\infty}^{+\infty} |f_1(x)|^2 dx \cdot \int_{-\infty}^{+\infty} |f_2(x)|^2 dx \quad (2-4)$$

该不等式当且仅当  $f_1(x)=kf_2^*(x)$  时取等号，其中  $k$  为任意常数。

因此，令  $f_1(x)=H(f)$ ,  $f_2(x)=S(f)e^{j2\pi ft}$ ，则有以下公式成立：

$$r \leq \frac{\int_{-\infty}^{+\infty} |H(f)|^2 df \cdot \int_{-\infty}^{+\infty} |S(f)|^2 df}{\frac{n_0}{2} \int_{-\infty}^{+\infty} |H(f)|^2 df} = \frac{2E}{n_0} \quad (2-5)$$

其中  $E$  为有效信号的能量，上式取等号时当且仅当  $H(f)$  满足以下条件：

$$H(f) = kS^*(f)e^{-j2\pi ft} \quad (2-6)$$

通过以上分析说明匹配滤波器输出的最大信噪比仅与有效信号的能量和输入信号噪声的功率谱密度有关，而与输入信号的形式无关。而脉冲压缩所需要的匹配滤波器的频率响应与输入信号频谱的复共轭相同。从本质上来说，匹配滤波器的效果与互相关器相同，其最终输出为有效信号的自相关函数和其与输入信号噪声的互相关函数之和。基于卷积定理，信号自相关函数与其频谱之间的关系可表示为：

$$y(u') = C_{ss}(u) = \int_{-\infty}^{+\infty} s(t)s(t-u)dt = \int |S(f)|^2 e^{j2\pi ft} df \quad (2-7)$$

因此，从上面的分析可知，匹配滤波器的输出也是输入信号的功率谱密度的傅立叶逆变换。

为了达到脉冲压缩的目的，通过选择合适的滤波器对输入信号进行相应的处理，可以使得输入信号中的噪声得到有效滤除，使得输出信号信噪比极大，通过分析可知，当滤波器传递函数为有效信号的傅氏变换的共轭时，此时输出信号信噪比最大。典型的相位调制的脉冲信号，发射波形以及经过脉冲压缩后的回波信号如图 2-3 所示。

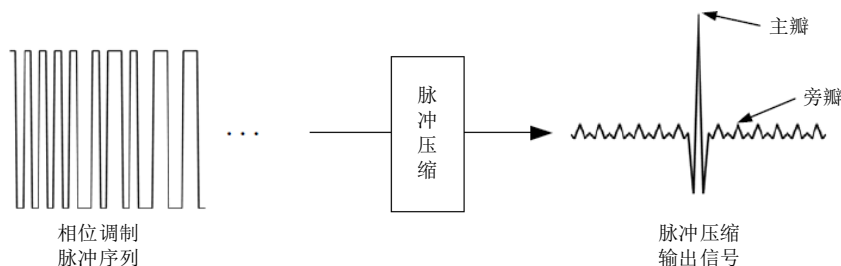


图 2-3 相位调制脉冲发射信号的脉冲压缩示意图

图 2-3 中的相位调制脉冲序列是将一系列的不同相位的子脉冲串起来，形成特定的编码。通过之前的分析，脉冲压缩输出信号是输入的有效信号的自相关函数加上有效信号与噪声的互相关函数。故可以得到脉冲压缩发射信号的编码方式会很大程度地影响最终的脉冲压缩性能。常用的编码为巴克码、互补格雷序列、勒让德序列、伪随机序列等。

其中，巴克码是一种较为理想的编码形式。由于它是典型的二相编码序列，所以仅用正负号代表相位，“+”和“-”表示相位相差  $180^\circ$ 。

它具有近似于图钉型的模糊函数，其非周期自相关函数为：

$$\phi(k) = \sum_{i=1}^N a_i \cdot a_{i+k} \quad (-(N-1) \leq k \leq (N-1)) \quad (2-8)$$

其中， $N$  为码元中的子脉冲数， $a_i$  为二元编码序列。它比起其他编码序列的自相关函数来有很大的优越性，主要是编码长度即为自相关函数的主旁瓣比，即为压缩比。如图 2-4 所示为 5、7、11 和 13 位巴克码的自相关函数。

从图 2-4 中可以看出，对于单纯的巴克码而言，其自相关函数的旁瓣幅值是相对固定的，主瓣幅值随着编码的长度增加而增加，编码长度即为脉冲压缩比，同时旁瓣数量随着编码长度增加而增加。

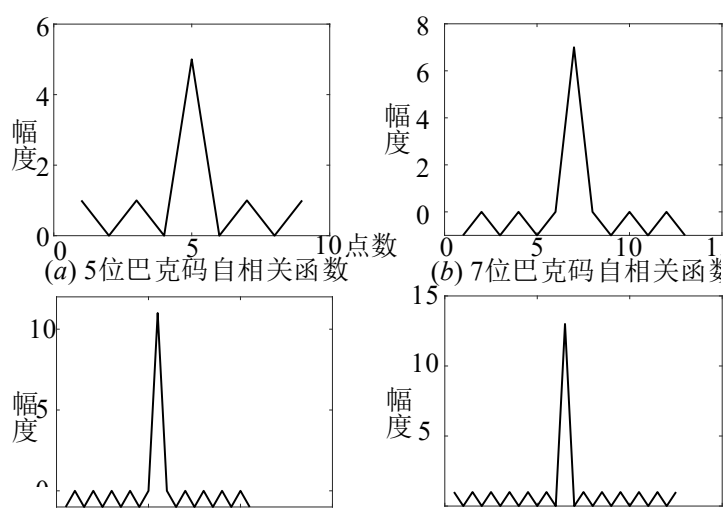


图 2-4 不同码长的巴克码自相关函数

尽管巴克码是一种性能较为优良的二相编码信号，但是仍存在一定缺点。由于目前已找到的巴克码只有九种，而且最长只有十三位，即  $N=13$ ，因此巴克码本身能给出的最大压缩比为 13，很难达到电磁超声系统的要求，这限制了巴克码在本系统中的使用。

互补格雷码的定义如下：码长为  $N$  的一对序列  $A$  和  $B$ ，若它们的自相关函数之和仅在零位移处为一非零值，其余位置均为零，那么这两个序列被称为互补

格雷码。理想的 16 位互补格雷码如图 2-5 所示。

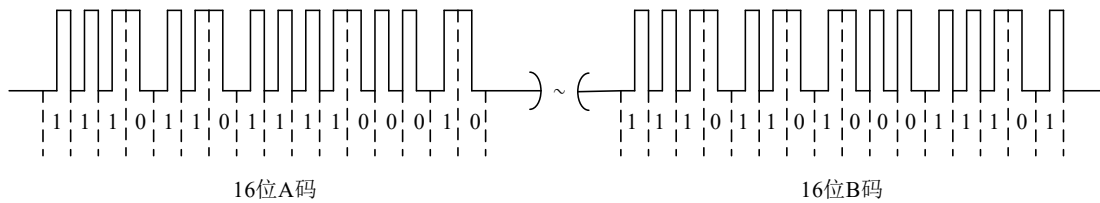


图 2-5 理想 16 位互补格雷码信号

实际上，互补格雷码中的任意一组序列的自相关函数都存在着一旁瓣，但是相加后可被消除。A 和 B 的自相关函数以及它们的和分别为：

$$A_M(k) = \sum_{i=1}^N a_i \cdot a_{i+k} \quad (-(N-1) \leq k \leq (N-1)) \quad (2-9)$$

$$B_M(k) = \sum_{i=1}^N b_i \cdot b_{i+k} \quad (-(N-1) \leq k \leq (N-1)) \quad (2-10)$$

$$G_M(k) = A_M(k) + B_M(k) \quad (-(N-1) \leq k \leq (N-1)) \quad (2-11)$$

对上述 16 位的互补格雷码进行自相关分析，得到结果如图 2-6 所示。从图中可以看出其中序列 A 和序列 B 的自相关函数的峰值等于编码长度，而其旁瓣幅值为自相关结果的峰值的 18.75%，而当两个序列的自相关结果相加后，信号峰值为原来的两倍，而旁瓣幅值均为零。

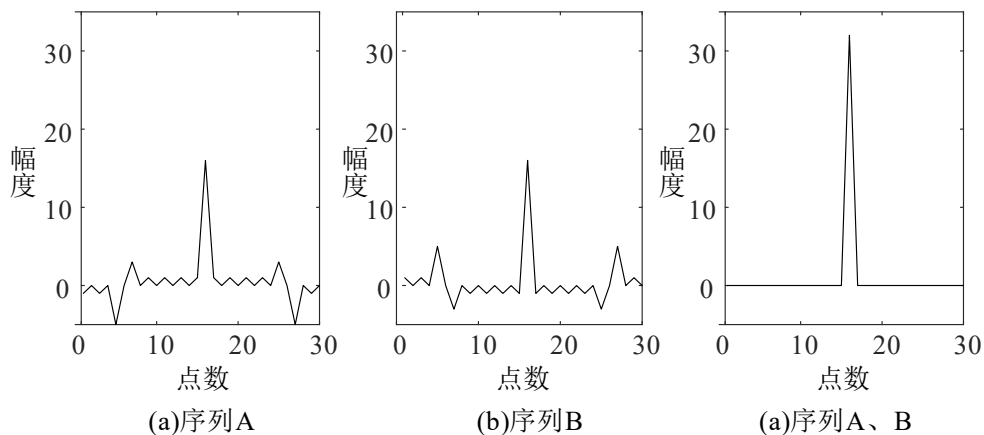


图 2-6 互补格雷码的自相关函数示意图

互补格雷码可以通过两组序列求和来达到消除旁瓣信号，进而提高信号信噪比的目的，除此之外，互补格雷码还有一个较为优越的特性，那就是其长度理论上可以是无限的。通过合理的构造方法可以得到足够长的编码序列，这一点是巴克码无法实现的。

因此，本设计选用互补格雷码作为脉冲压缩信号的编码方式，利用 PL 端在时序控制方面的优势生成互补格雷码作为电磁超声检测系统的激励信号。



## 2.3 硬件电路设计

### 2.3.1 接收电路与 ADC 采集模块

接收电路主要包括前级放大电路、窄带滤波电路、可控增益放大电路。回波信号先由接收电路进行滤波、放大处理后,再由 ADC 采集模块转换为离散波形,输入至 ZYNQ。接收电路与 ADC 采集模块结构框图如图 2-7 所示。

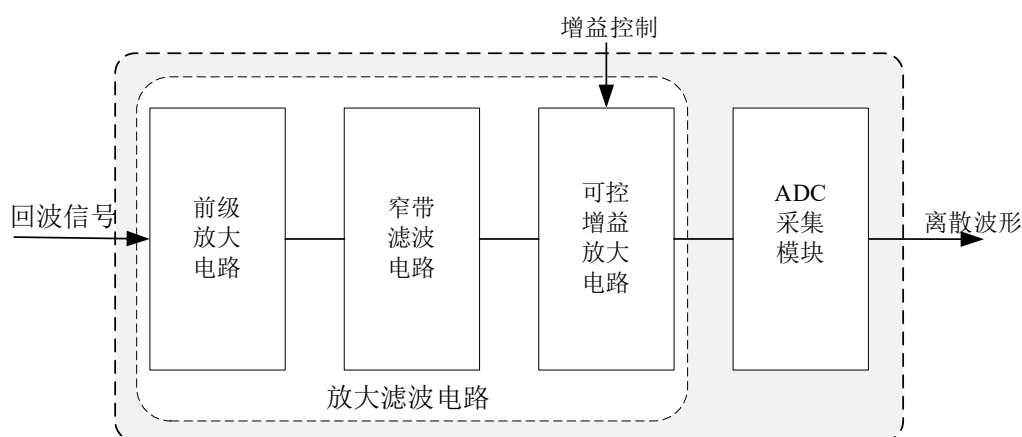


图 2-7 接收电路与 ADC 采集模块结构框图

电磁超声换能器的回波信号电压等级通常为微伏级别,因此需要对回波信号进行放大,本系统设计前级放大电路的增益为 20dB,可控增益放大电路的增益范围为 0-80dB。同时,换能器中线圈为敏感元件,回波信号中耦合有大量噪声与干扰,采用中心频率为 4MHz,带宽 4MHz 的窄带滤波器进行滤波处理,提高信噪比。

ADC 采集模块采用了高速 AD 模块 AN9238。该模块采用了 ADI 公司的模数转换芯片 AD9238,该芯片有两路转换通道,采样速率最高为 65MSPS,采样位数为 12 位,可以满足本系统的要求。ADC 采集模块如图 2-8 所示。

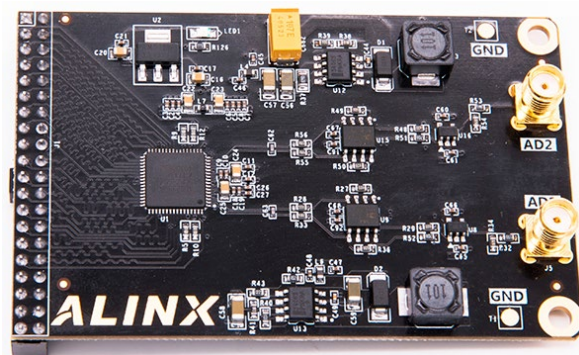


图 2-8 ADC 采集模块 AN9238

### 2.3.2 发射电路与高压模块

发射电路主要包括隔离驱动电路和 H 桥功率放大电路。工作过程中，ZYNQ 控制高压模块产生 100V 直流信号，再由发射电路逆变为  $\pm 100\text{V}$  高频交变信号，通过电磁超声换能器激励超声波。发射电路与高压模块结构框图如图 2-9 所示。

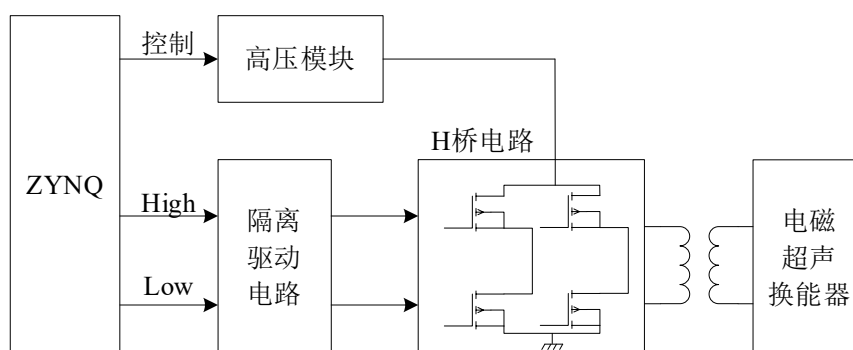


图 2-9 发射电路与高压模块结构框图

### 2.3.3 电磁超声换能器

电磁超声换能器主要由激励线圈、接收线圈和永磁铁构成。其结构如图 2-10 所示。

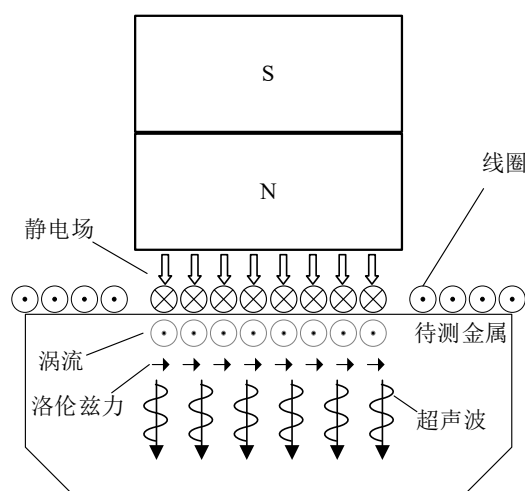


图 2-10 电磁超声换能器

发射过程中，先在换能器线圈中通以频率为兆赫兹级别的大功率电流。根据电磁场理论，该电流会在被测金属试件表面集肤深度内感生出与方向相反、频率相同的涡流。交变的涡流在磁铁施加的静电场下会受到洛伦兹力作用，该力的方向可由左手定则判断。金属表面在洛伦兹力的作用下就会产生周期性的振动和弹性形变。当这种振动以波的形式在金属中传播时，就形成了超声波。接收过程是发射过程的逆过程。

电磁超声以电磁感应为换能机理，因此可以实现非接触式测量，检测时无需对试件进行表面处理，无需涂抹耦合剂，相对压电超声，大大简化了测量流程。



## 2.4 ZYNQ 程序设计

### 2.4.1 PL 端发射模块设计

发射模块主要由高压控制器与触发信号生成器构成，其结构如图 2-11 所示。高压控制器控制高压模块向电容充电，并在接收电路工作时将其关闭，避免高压模块工作时对接收电路产生干扰。充电完成后，由高压充电控制器向触发信号生成器发送互补格雷码的 A 码使能信号，由采集模块的 BRAM 半满信号作为互补格雷码的 B 码使能信号，产生互补格雷码触发信号，发射模块的 Block Design 如图 2-12 所示。

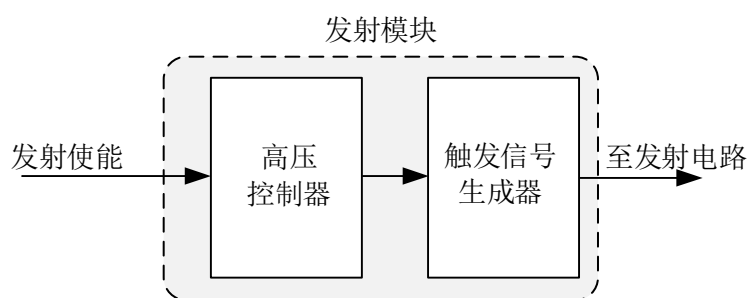


图 2-11 发射模块结构框图

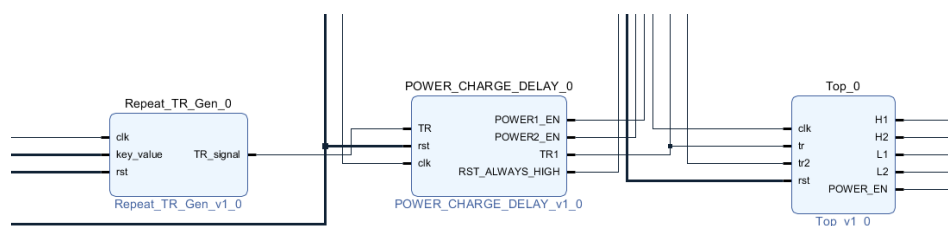


图 2-12 发射模块 Block Design

### 2.4.2 PL 端采集模块设计

采集模块主要负责 ADC 输出数据的读取，并将数据通过 HP AXI 端口存储至 DDR；为了充分利用 ADC 的输入范围，通过 IIC 控制接收电路中可控增益放大电路的增益。采集模块设计框图如图 2-13 所示。

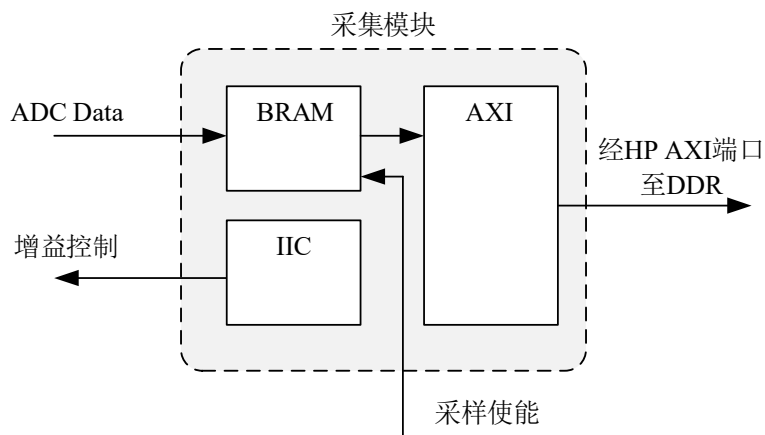


图 2-13 采集模块设计框图

ADC 输出数据的读取通过控制 BRAM 的读写实现, BRAM 的写时钟与 ADC 的采样时钟相连, 读时钟与 HP AXI 从端口时钟相连。由于需要分别发射互补格雷码, 因此以 BRAM 半满信号为界, 分别存储序列 A 和序列 B 对应的超声回波数据。当达到采样深度, 即 BRAM 写满时, 停止采样, 并向 AXI 发送读请求; AXI 接收到 BRAM 的读请求后向 BRAM 发送读使能信号, 将采样数据从 BRAM 中取出, 搬移至连接在 PS 端的 DDR3 中暂存。采集模块的 Block Design 如图 2-14 所示。

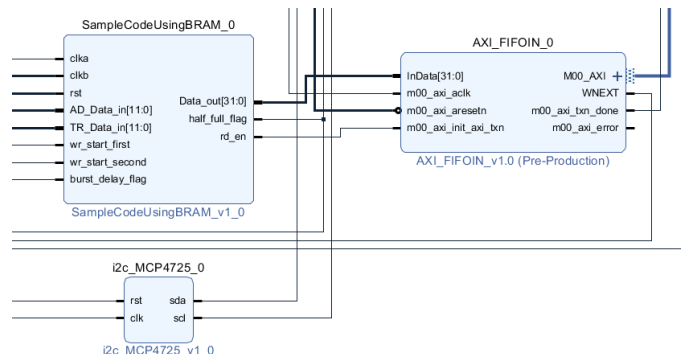


图 2-14 采集模块 Block Design

### 2.4.3 PL 端交互模块设计

交互模块的结构如图 2-15 所示。主要包括 HDMI 模块的例化, 按键中断的分配。HDMI 模块读取压缩处理后的回波数据, 并将其显示在显示器上。按键控制发射采样和光标移动。

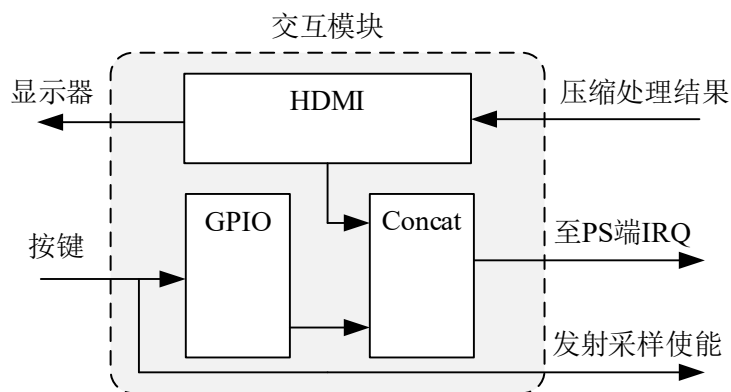


图 2-15 交互模块结构框图

其中，HDMI 模块的具体结构如图 2-16 所示。主要包括了时序控制模块、数据输出模块和格式转化模块。其主要功能为不断读取压缩后的回波数据，并显示在显示器上。

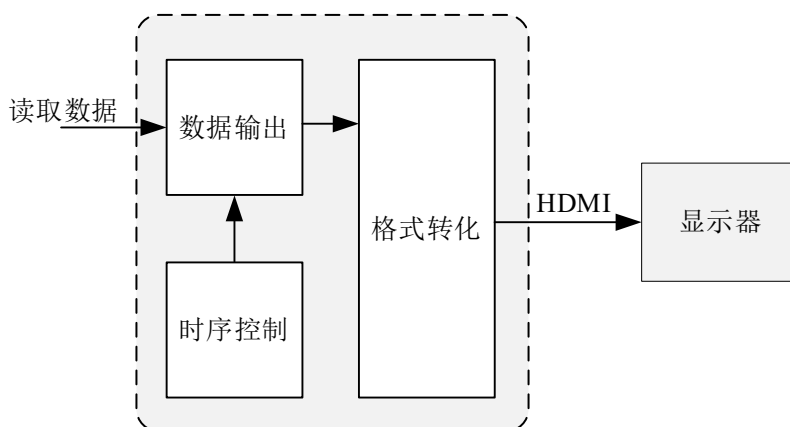


图 2-16 HDMI IP 结构框图

### (1) 数据输出模块

数据输出模块使用 Xilinx 的 VDMA IP 核，该模块主要实现读取 DDR 中的数据并以数据流的形式输出。配置该 IP 时，使能 MM2S 接口，通过 AXI4 总线，与 ZYNQ HP 口进行数据交互，读取 PS 端 DDR 中经过脉冲压缩的的波形数据。

### (2) 时序控制模块

时序控制模块主要包括两个部分，分别是 Xilinx 的 VTC IP 核以及自定义的动态时钟产生模块。其中，VTC IP 负责产生一帧画面显示的行场同步时序。为了适配不同分辨率的显示器，需要设置不同的分辨率及对应的像素时钟。自定义的动态时钟模块可以根据不同的分辨率配置锁相环的时钟输出，产生相应的像素时钟。

### (3) 格式转化模块

格式转化模块使用 Xilinx 的 RGB2Dvi IP 核，将图像的 RGB 的数据转换用



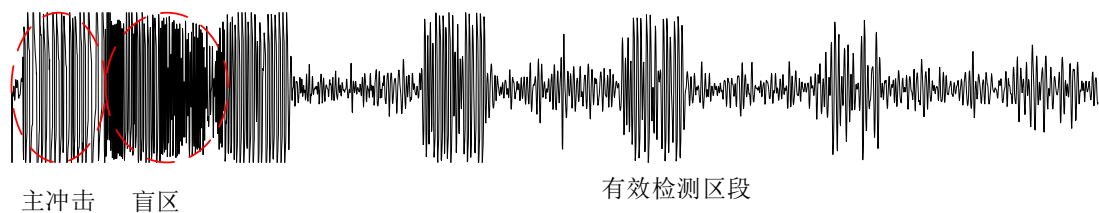


图 2-19 回波各段组成部分

将匹配滤波器与回波信号卷积处理，大时宽的回波信号会被压缩为窄的脉冲信号，信噪比将大大提高，由于两组互补的激励信号自相关函数加和后，旁瓣完全消除，所以将两组互补的回波信号分别处理，结果相加后信噪比将得到进一步提高。具体操作流程如图 2-20 所示。

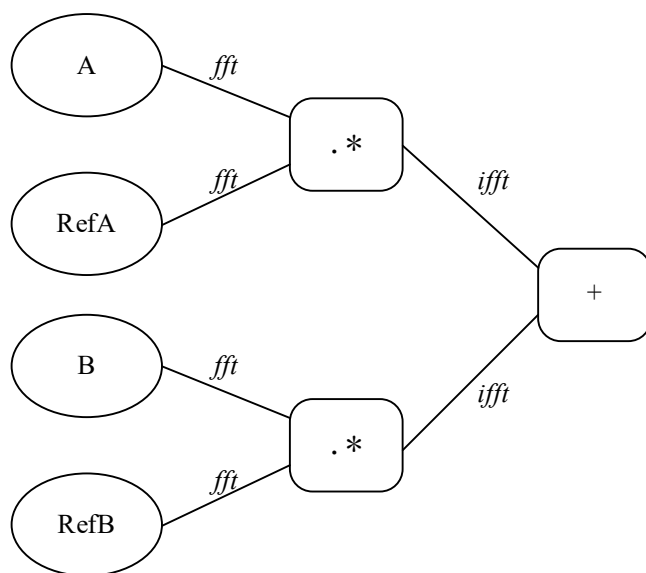


图 2-20 压缩处理流程

#### 2.4.5 厚度计算模块

厚度计算模块根据压缩后的回波数据自动精确测量试件厚度。计算流程如图 2-21 所示。首先，将处理后的浮点型回波数据转化为整型，提取其包络后对比分析不同的信号极值，选择两个有效值，作为两个回波信号峰值。通过计算采样点数结合采样率和预置声速即可计算出试件厚度。

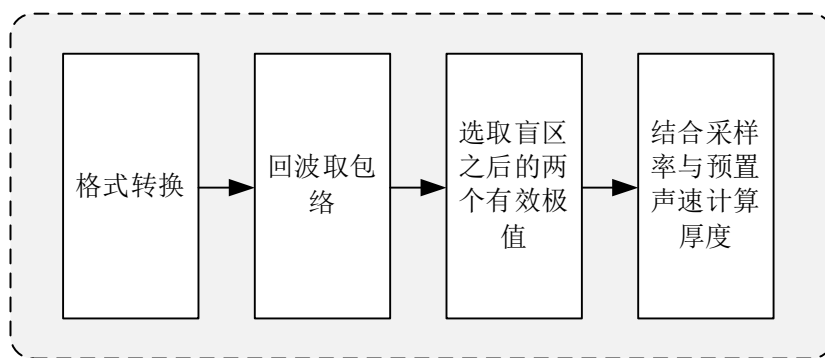


图 2-21 厚度计算模块结构框图

超声测厚利用两个回波之间的声时差，声时差的测量精度决定了测厚的精度，为了保证声时差的测量准确，需要首先精确解析回波信号的包络线，利用包络线上的关键点测量声时差。本设计采用 3 点运动均值滤波器提取包络线，先前向再后向处理找到局部最大值及其对应的位置，计算第  $n$  个波谷与波峰的差值  $VPD(n)$ ，用来去掉那些假的波峰点，如果满足

$$VPD(k) < 0.7 \times \{VPD(k-1) + VPD(k) + VPD(k+1)\} / 3 \quad (2-12)$$

则此波峰为假，需要移除。如此逐峰值点进行比较，提取出回波包络线。提取包络信号后，要通过对比分析找到回波信号。因为电磁超声存在着盲区，因此在舍弃主冲击和拖尾信号的无效数据后。通过比较包络信号的值，找到两个最大的两个极值点作为回波信号峰值，由此得到两个峰值点之间的采样点数  $N$ 。

厚度的计算需要结合采样率和预置的试件材料声速值。计算公式如下。

$$H = \frac{NV}{2S_r} \quad (2-13)$$

其中， $H$  试件厚度； $V$  预置声速； $N$  采样点数； $S_r$  采样率

#### 2.4.6 结果显示模块

采集的数据经过处理进行显示时。首先固定在一帧缓存的相应位置上显示菜单栏。然后在各个波形缓存中绘制网格数据并根据波形数据在波形缓存中叠加相应的波形数据，最后将不同的波形缓存刷新到屏幕的一帧缓存中，显示显示相应的波形和菜单栏。观察波形时，可以通过按键调整两个光标的位置，并菜单栏上实时显示各个光标的位置和以及自动计算的试件厚度值。整个流程如图 2-22 所示。



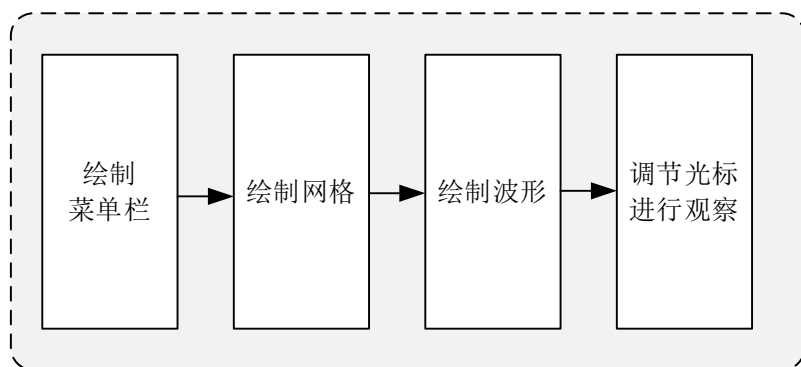


图 2-22 结果显示模块结构框图

菜单栏显示试件厚度的实时测量值、光标的像素点位置和试件中声速的相关信息。通过在画面上叠加网格，使波形数据更加易于观察。绘制波形结束后，通过按键可以移动两个光标，并根据光标的位置手动计算试件的厚度，可以对自动计算结果进行校正。

### 第三部分 完成情况及性能参数

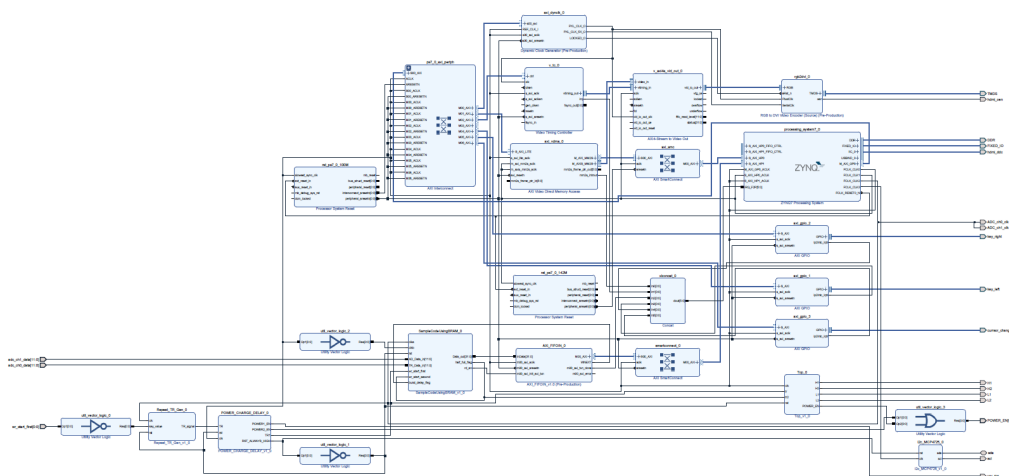


图 3-1 PL 端 Block Design

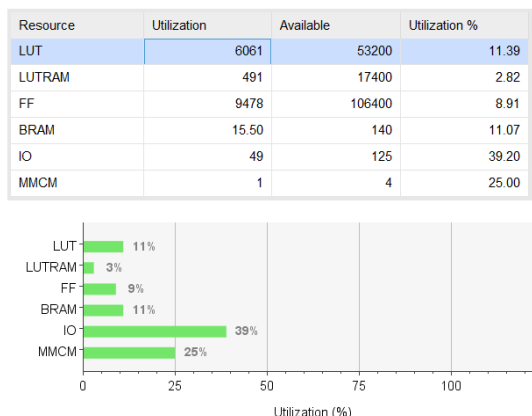


图 3-2 ZYNQ 资源消耗

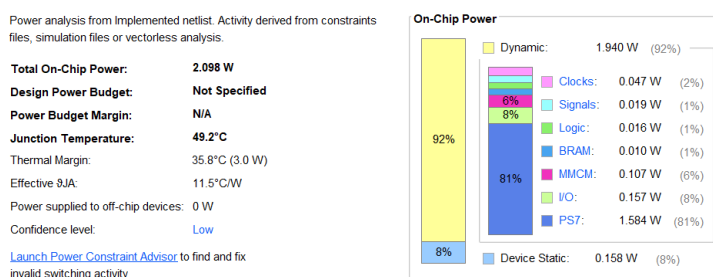


图 3-3 片上功耗

PL 端 Block Design 如图 3-1 所示，ZYNQ 资源消耗如图 3-2 所示，片上功耗如图 3-3 所示。由于 PL 端主要进行时序控制，不涉及大规模计算，因此 PL 端资源占用较少，脉冲压缩算法与自动测厚算法均在 PS 端完成，片上功耗图可以看出 PS 端功耗较高，占比为 81%。后续考虑采用硬件加速的思想，通过 PL 端实现脉冲压缩算法，充分发挥 ZYNQ 灵活、高速的优势，提升对超声波信号的处理速度。

本设计以 ZYNQ 7020 为核心，实现了电磁超声脉冲压缩测厚系统，系统实物如图 3-4 所示。利用此系统完成了 20mm 厚碳钢试件的自动精确测厚，可以将处理前的波形与处理后的波形显示在屏幕上。千分尺测量结果如图 3-5 所示，为 19.869mm，自动计算结果如图 3-6 所示，为 19.849mm，测量误差为 0.02mm，小于预期性能指标的 0.2mm。

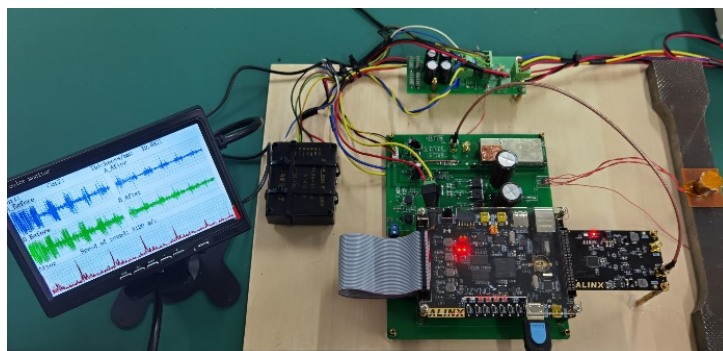
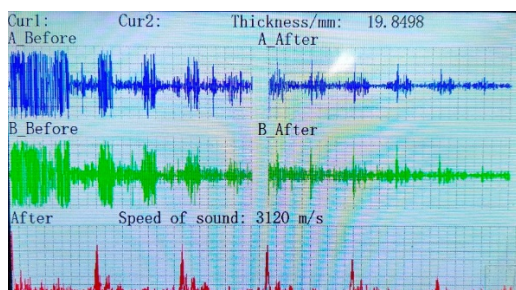


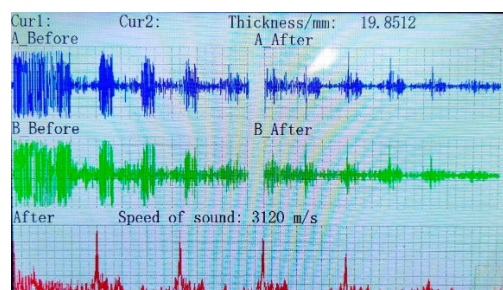
图 3-4 系统实物图



图 3-5 千分尺测量结果



(1)第一次检测结果



(2)第二次检测结果

图 3-6 自动测量结果

以第一次测厚结果为例，对其处理前后的回波信号进行分析，得到信噪比提升结果如图 3-7 所示，本系统可以将回波信号信噪比提高至 8dB，满足关键性能指标的要求，提升效果明显。

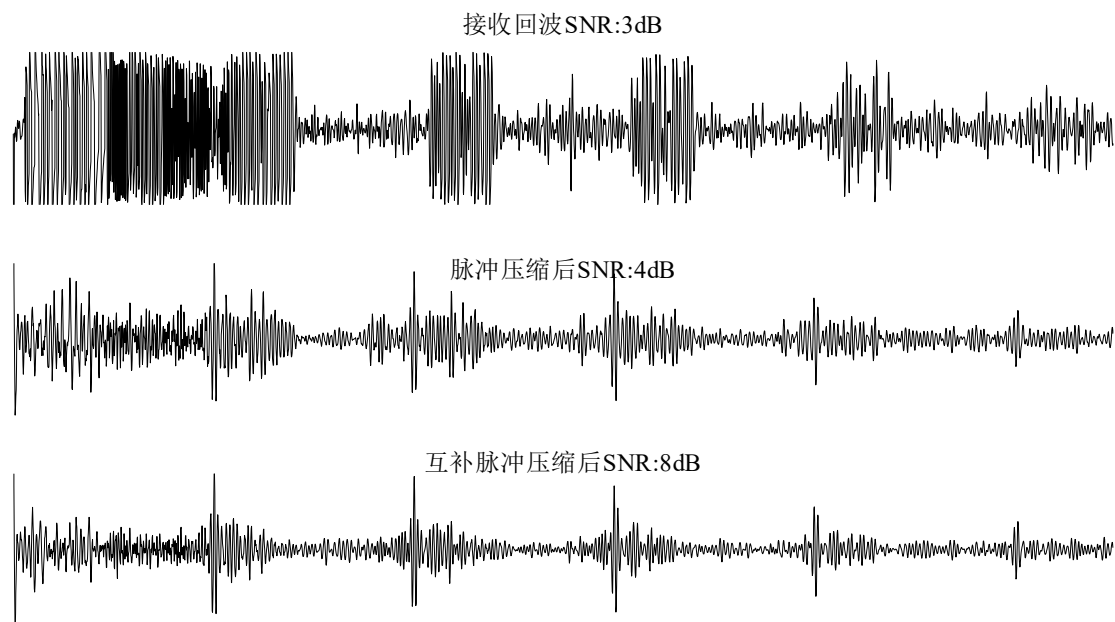


图 3-7 SNR 提升结果

## 第四部分 总结

### 2.5 可扩展之处

(1) 采用硬件加速的思想,通过 PL 端实现脉冲压缩算法,充分发挥 ZYNQ 灵活、高速的优势,提升对超声波信号的处理速度。

(2) 对于较薄的试件,回波信号将淹没在较宽的发射脉冲中,可采用发射-接收编码式脉冲压缩技术,实现系统发射和接收工作状态的分离,实现对薄试件的测量。

(3) 超声波回波信号不仅可用于厚度测量,也可用于裂纹、缺陷的测量,因此可对回波信号进一步分析,拓展系统的检测内容。

### 2.6 心得体会

比赛的 2 个月期间,小组成员进行了 FPGA 选型,硬件电路设计与调试,软件编写,实验测试,完整完成了电磁超声脉冲压缩检测系统这一设计。在这过程中,每位成员都积极完成各自的任务,收获满满,同时也发现了诸多问题,例如在调试 PL 端程序时,小组成员对问题分析缺乏条理,做了很多重复性的工作,而 PL 端编译、综合时间又较长,耽误了较多时间;在调试电磁超声电路时,分析还不够严谨,忽略了很多隐藏在正常现象下的问题,最终导致调试过程中 MOS 管的烧毁。不过,发现问题才能改进,小组成员对本次设计做了深刻的反思,通过此次竞赛,小组成员在硬件调试、程序设计与调试、团队合作等能力都有了显著的提升。

最后，需要感谢实验室的各位老师与师兄对我们不厌其烦的指导，在很多关键时期给予我们建设性的意见，也要感谢组委会提供比赛这样一个平台，推动我们对 FPGA 的学习、理解，同时很温馨地通过短信提醒我们比赛的各个节点，通过 QQ 群及时解答我们的疑惑，今年是大赛成立以来的第四届，还是很年轻的比赛，愿我们一同成长！

## 第五部分 参考文献

- [1] Hirao M., Ogi H. Electromagnetic Acoustic Transducers:Non-contacting Ultrasonic Measurements using EMATs[M], Springer, 2016.
- [2] Abouelfadl A., Samir A. Mohamed, Ahmed Fathy M., et al. Performance analysis of LFM pulse compression radar under effect of convolution noise jamming, 2/2016, 2016 [C]. IEEE.
- [3] Anastasi R.F., Madaras E.I. Pulse compression techniques for laser generated ultrasound, 1999, 1999 [C]. IEEE.
- [4] 周正干, 马保全, 孙志明, 姜靖涛. 相位编码脉冲压缩方法在空气耦合超声检测信号处理中的应用[J], 机械工程学报, 2014, 50(02):48-54.
- [5] 何宾. Xilinx Zynq-7000 嵌入式系统设计与实现：基于 ARM Cortex-A9 双核处理器和 Vivado 的设计方法[M], 电子工业出版社, 2019.

## 第六部分 附录

### PL 端主要代码

```
//发射模块状态机设计
//发射 4MHz 二相编码序列

reg [CODE_LENGTH-1:0] BI_PHASE_CODE_REG;//二相码值寄存器
reg [12:0] CODE_LENGTH_COUNTER = 'b0;//码长计数器
assign CODE = BI_PHASE_CODE_REG[0:0];//由低位到高位依次发射

reg [2:0] TR_STATUS = 1'b0;
parameter TR_WAIT = 0;
parameter TR_ING = 1;
parameter TR_DONE = 2;

always @(posedge clk_4MHz or posedge rst) begin
    if (rst == 1) begin
        CODE_LENGTH_COUNTER<=0;
        BI_PHASE_CODE_REG<=0;
    end
    else
    begin
        case (TR_STATUS)
            TR_WAIT://等待状态, 检测两个触发脉冲
            begin
                if (init_tr_pulse==1) begin
                    TR_STATUS<=TR_ING;
                    BI_PHASE_CODE_REG<=BI_PHASE_CODE;
                end
                else if(init_tr2_pulse==1)begin
                    TR_STATUS<=TR_ING;
                    BI_PHASE_CODE_REG<=BI_PHASE_CODE_COMP;
                end
                else begin
                    TR_STATUS<=TR_STATUS;
                    CODE_LENGTH_COUNTER<=0;
                    BI_PHASE_CODE_REG<=BI_PHASE_CODE_REG;
                end
            end
        end

        TR_ING://发送状态, 检测已发送编码数量
        begin
            if (CODE_LENGTH_COUNTER==CODE_LENGTH-1) begin
                TR_STATUS<=TR_DONE;
                CODE_LENGTH_COUNTER<=0;
            end
        end
    end
end
```



```
        BI_PHASE_CODE_REG<=0;
    end
    else
    begin
        TR_STATUS<=TR_STATUS;
        CODE_LENGTH_COUNTER<=CODE_LENGTH_COUNTER+1;
        BI_PHASE_CODE_REG<=BI_PHASE_CODE_REG>>1;
    end
end

TR_DONE://结束状态，自动转向等待状态
begin
    TR_STATUS<=TR_WAIT;
    CODE_LENGTH_COUNTER<=0;
    BI_PHASE_CODE_REG<=0;
end
default;;
endcase
end
end

assign H1 =
(TR_STATUS[0:0]&&clk_4MHz&&CODE) || (TR_STATUS[0:0]&&~clk_4MHz&&~CODE);
assign L1 = ~H1;
assign H2 = (TR_STATUS[0:0]&&~H1) || (~TR_STATUS[0:0]&&H1);
assign L2 = ~H2;
assign POWER_EN = TR_STATUS[0:0]; //发射时打开
```

```
//采集模块状态机设计
reg [2:0] status='b0;
parameter IDLE = 0;
parameter FIRST_WR = 1;
parameter WR_WAITING = 2;
parameter SECOND_WR = 3;
parameter WR_DONE = 4;

always@(posedge clka or posedge rst)
begin
    if(rst==1)begin
        status<=IDLE;
        PORTA_EN<=0;
        full_flag<=0;
        half_full_flag<=0;
    end
```

```
else begin
    case(status)
        IDLE://初始状态，检测写使能脉冲
        begin
            if(init_wr_pulse==1)begin
                status<=FIRST_WR;
                PORTA_EN<=1;
            end
            else begin
                status<=status;
                PORTA_EN<=0;
                half_full_flag<=0;
            end
        end
    end

    FIRST_WR://采集第一组回波信号，检测写地址
    begin
        if(addr_a==13'b0_1111_1111_1111)begin
            status<=WR_WAITING;
            PORTA_EN<=0;
            full_flag<=0;
            half_full_flag<=1;
        end
        else begin
            status<=status;
        end
    end

    WR_WAITING://等待状态，检测第二个写使能脉冲
    begin
        if(init_wr2_pulse==1)begin
            status<=SECOND_WR;
            PORTA_EN<=1;
        end
        else begin
            status<=status;
        end
    end

    SECOND_WR://采集第二组回波信号，检测写地址
    begin
        if(addr_a==13'b1_1111_1111_1111)begin
            status<=WR_DONE;
            PORTA_EN<=0;
        end
    end
end
```

```

        full_flag<=1;
        half_full_flag<=0;
    end
    else begin
        status<=status;
    end
end

WR_DONE://写完成，转向初始状态
begin
    status<=IDLE;
end

default;;
endcase
end
end

```

## PS 端主要代码

```

/*
void FFT(float dataR[],float dataI[],float dataOR[],float
dataOI[],int N,int M)
    FFT 函数体接口说明
    dataR[]:输入信号实部
    dataI[]:输入信号虚部，此处为零
    dataOR[]:频谱实部
    dataOI[]:频谱虚部
    N:变换点数
    M:2^M=N
*/
/*
void IFFT(float dataR[],float dataI[],float dataA[],int N,int M)
    IFFT 函数体接口说明
    dataR[]:输入频谱实部
    dataI[]:输入频谱虚部
    dataA[]:输出时域信号
    N:变换点数
    M:2^M=N
*/
/*
double MeasureEchoPosition_SimilarEhco(const RawData* Data_D,
int32_t GainDB, IndexType nSize, IndexType *PeakIndex1, IndexType*
PeakIndex2, IndexType* PeakNum)
    MeasureEchoPosition_SimilarEhco 函数体接口说明

```

```
IndexType:int32_t
Data_D:输入波形
GainDB:增益,此处为零
nSize:检测长度
PeakIndex1:首个极值位置
PeakIndex2:第二个极值位置
PeakNum:极值数量
返回值:两个极值之间的采样点距离
*/

//核心算法
void MY_FFTR1(void){
    int temp;
    int i;

    //取值,格式转换
    for(i=4;i<16384;i=i+4){
        temp = (int)Xil_In16(0x10000000+i+2);
        temp = temp&0xFFFF;
        Echo_1[i/4] = temp*Volt_Per_Bit-4.8;
        Echo_1_COPY[i/4] = Echo_1[i/4];
    }
    Echo_1[0] = Echo_1[4095];
    Echo_1_COPY[0] = Echo_1[0];
    for(i=16388;i<32768;i=i+4){
        temp = (int)Xil_In16(0x10000000+i+2);
        temp = temp&0xFFFF;
        Echo_2[(i-16384)/4] = temp*Volt_Per_Bit-4.8;
        Echo_2_COPY[(i-16384)/4] = Echo_2[(i-16384)/4];
    }
    Echo_2[0] = Echo_2[4095];
    Echo_2_COPY[0] = Echo_2[0];

    //取主冲击
    for(i=0;i<4096;i++){
        if(i<200){
            Trig_1[i] = Echo_1[i];
            Trig_2[i] = Echo_2[i];
        }else{
            Trig_1[i]=0;
            Trig_2[i]=0;
        }
    }
}
```

```
//A 相回波 FFT
FFT(Echo_1_COPY,yI,Echo_FFT_REAL,Echo_FFT_IMG,Sample_Point,M);
for(i=0;i<4096;i++){
    yI[i]=0;
}
//A 相滤波器构造
FFT(Trig_1,yI,Trig1_FFT_REAL,Trig1_FFT_IMG,Sample_Point,M);
for(i=0;i<4096;i++){
    yI[i]=0;
}
//A 相脉冲压缩
for(i=0;i<Sample_Point;i++){
    Trig1_FFT_IMG[i]=-Trig1_FFT_IMG[i];
    Data_Real[i]=(Echo_FFT_REAL[i]*Trig1_FFT_REAL[i]-
Trig1_FFT_IMG[i]*Echo_FFT_IMG[i]);
    Data_Img[i]
=(Echo_FFT_REAL[i]*Trig1_FFT_IMG[i]+Trig1_FFT_REAL[i]*Echo_FFT_IMG[i]
);
}
IFFT(Data_Real,Data_Img,Echo_Proed1,Sample_Point,M);

//B 相回波 FFT
FFT(Echo_2_COPY,yI,Echo_FFT_REAL,Echo_FFT_IMG,Sample_Point,M);
for(i=0;i<4096;i++){
    yI[i]=0;
}
//B 相滤波器构造
FFT(Trig_2,yI,Trig2_FFT_REAL,Trig2_FFT_IMG,Sample_Point,M);
for(i=0;i<4096;i++){
    yI[i]=0;
}
//A 相脉冲压缩
for(i=0;i<Sample_Point;i++){
    Trig2_FFT_IMG[i]=-Trig2_FFT_IMG[i];
    Data_Real[i]=(Echo_FFT_REAL[i]*Trig2_FFT_REAL[i]-
Trig2_FFT_IMG[i]*Echo_FFT_IMG[i]);
    Data_Img[i]
=(Echo_FFT_REAL[i]*Trig2_FFT_IMG[i]+Trig2_FFT_REAL[i]*Echo_FFT_IMG[i]
);
}
IFFT(Data_Real,Data_Img,Echo_Proed2,Sample_Point,M);

//加和, 格式转换
for(i=0;i<Sample_Point;i++){
```

```
Echo_Proed[i]=Echo_Proed1[i]+Echo_Proed2[i];
if(Echo_Proed[i]>100){
    Echo_Proed_2_INT[i]=2000;
}else if(Echo_Proed[i]<-100){
    Echo_Proed_2_INT[i]=0;
}else{
    Echo_Proed_2_INT[i] = (int)Echo_Proed[i]*10+1000;
}
}

//自动计算采样点距离
distance =
MeasureEchoPosition_SimilarEhco(Echo_Proed_2_INT,0,2048,PeakIndex1,Pe
akIndex2,PeakNum);
}
```