

基于 FPGA 的多通道高速实时信号处理系统设计

洗友伦, 卢护林, 苏涛

(西安电子科技大学雷达信号处理国家重点实验室 陕西 西安 710071)

摘要: 在高速实时信号处理系统中常采用 FPGA+DSP 的架构, 根据各自的优点, FPGA 主要做前端信号处理和系统控制。本文结合具体工程项目, 分析设计如何将中频正交采样 (DDC) 在 FPGA 中实现, FPGA 如何与 TS201 实现高速链路口通信, 以及多路信号并行实时处理在 FPGA 中的实现。

关键字: 多通道; DDC; 实时信号处理; 链路口通信; FPGA

Real-time multi-channel high-speed signal processing system design based on FPGA

XIAN Youlun, LU Hulin, SU Tao

(Key Lab for Radar Signal Processing, Xidian University, Xi' an, 710071, China)

Abstract: In high-speed real-time signal processing systems are often used FPGA + DSP architecture .In accordance with their respective advantages, FPGA is often used as front-end signal processing and system control. This paper for specific projects, analysis of how to design IF orthogonal sampling (DDC) in FPGA implementation, how to achieve FPGA communication with the TS201 use link-port, and multiple real-time signal parallel processing in the FPGA to achieve.

Key words: Multi-channel; DDC; Real-time signal processing; link-port Communication; FPGA

1. 引言

在很多高速信号处理中常采用 FPGA+DSP 的结构, FPGA 主要做时序控制和前端处理, DSP 主要做后端信号处理。对于 FPGA 而言他的控制能力强, 有很好的并行处理优势这些都是 DSP 所不及的, 但是不像 DSP, FPGA 没有专门的寄存器和控制器, 而且存储空间小。因此, 对于 FPGA 接口电路的设计和数据通信是显得非常关键。同时, FPGA 在处理整数乘累加时很有优势, 但在处理浮点数和除法运算时就非常耗资源, 要把一个信号处理算成功地在 FPGA 上实现, 必须对算法有深刻的理解和详细的分析优化。

本文结合具体信号处理系统, 主要讨论 FPGA 在信号处理和接口通信中的具体实现。该系统 FPGA 主要并行实时处理 4 路高速 AD 采样的中频信号, 首先进行数字下变频 (DDC), 再把处理结果通过链路口传给 TS201, 同时用链路口接收 DSP 的处理结果, 并传给后板, 同时 FPGA 对整个系统做控制。经过系统实测验证, 该系统设计已成功运用在具体工程中。

2. 系统结构

本系统主要由1片FPGA：Altera公司的CycloneII系列EP2C70F672I8和4片ADI公司的TS201，4片Linear公司的高速ADC—Ltc2248；其系统结构如图2.1所示。

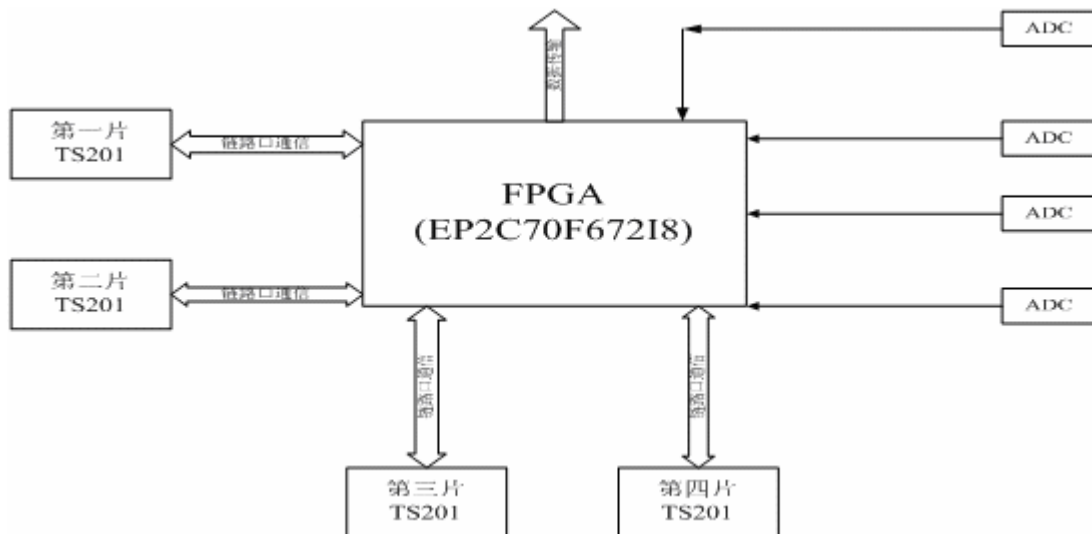


图 2.1 系统结构

该系统主要工作流程是 FPGA 接收 4 路高速 AD 采集的中频信号，通过前端信号处理（DDC）再通过链路口分别传给 4 片 DSP，DSP 把处理好的结果回传给 FPGA，最后 FPGA 再将结果传给后端信号处理板。在这个系统中 FPGA 和 DSP 的时钟由 50Mhz 晶振提供，DSP 工作在 600Mhz 的核时钟，AD 以 40Mhz 的采样速率工作。本文主要讨论 FPGA 的程序设计主要包括信号处理和数据传输部分。FPGA 内部要完成的功能如图 2.2 所示。

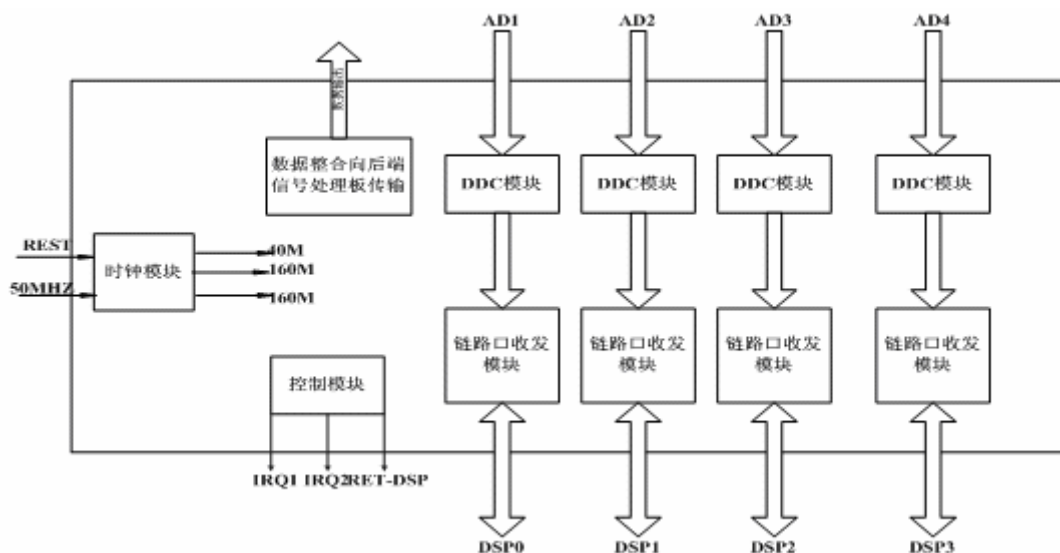


图 2.2 FPGA 主要处理任务框图

3. 基于 FPGA 的中频正交采样

3.1 算法分析

在信号处理领域，对接收到的中频信号进行正交化是十分重要的。它保留了信号复包络的所有信息，同时降低数据率和数据量，因而在高速信号处理中得到了广泛的应用。

一个系统所接收到的中频信号可表示为：

$$X(t) = R_e \left(a(t) \exp[j(2\pi f_0 t + \phi(t))] \right) = x(t) \exp(2\pi f_0 t) \quad (1)$$

式中， f_0 为中频频率， $a(t)$ 和 $x(t)$ 分别表示信号的幅度和相位， $x(t) = a(t) \exp[j\phi(t)] = I + jQ$ ，称为 $X(t)$ 的复包络，它包含了带通信号 $X(t)$ 的所有信息。

令

$$I = X_I = a(t) \cos \phi(t)$$

$$Q = X_Q = a(t) \sin \phi(t)$$

则
$$a(t) = \sqrt{X_I^2(t) + X_Q^2(t)} = \sqrt{I^2 + Q^2} \quad (2)$$

$$\phi(t) = \tan^{-1} \left(\frac{X_I(t)}{X_Q(t)} \right) = \tan^{-1} \left(\frac{I}{Q} \right) \quad (3)$$

$X_I(t)$ 和 $X_Q(t)$ 分别为 $X(t)$ 的同相分量和正交分量，可分别用 I 和 Q 表示。现在要对一个信号实现数字中频直接采样主要有以下几种方法：

- 低通滤波法
- 数字乘积检波法
- 希尔伯特变换法
- 贝塞尔插值法

这些方法的实现过程各有不同，但基本思想是一致的。下面以常见的低通滤波法为例来说明正交采样的实现过程及原理。其实现过程如图 3.1 所示。

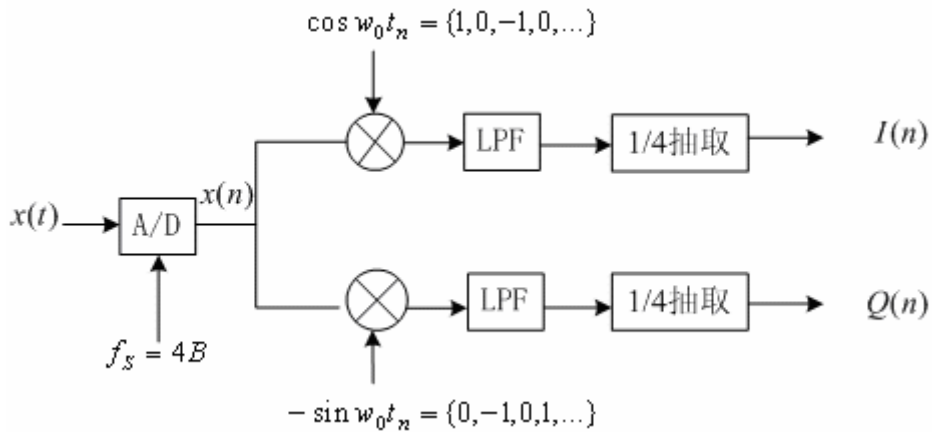


图 3.1 数字中频正交采样

在图 3.1 中，根据带通信号采样定理选取 $f_0 = B$ ， $f_s = 4B$ 。则 A/D 变换后频谱示意如，之后将

信号分别乘以 $\cos w_0 t_n$ 和 $-\sin w_0 t_n$ ，由于 $t_n = \frac{n}{f_s}$ ，而 $f_0 = \frac{1}{4} f_s$ ，所以 $\cos w_0 t_n$ 实际上可以化简为

$\cos \frac{\pi}{2} n$ ，即如图 3.1 中所示其取值为 1, 0, -1, 0 交替变换， $-\sin \omega_0 t_n$ 也是一样。这样相乘以后相

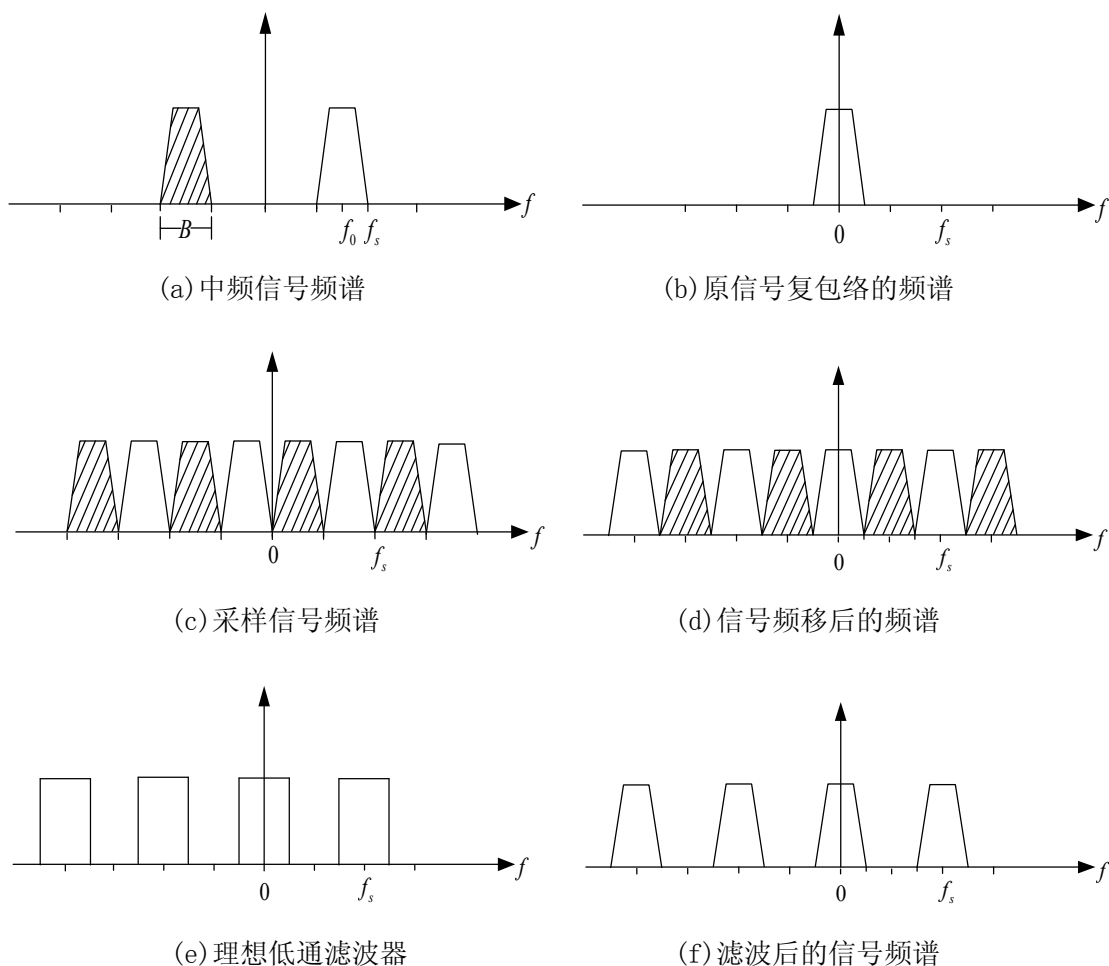
当于频谱中心左移 $\frac{\pi}{2}$ ，即将正频谱的中心移到了零频，时域信号也相应分成了实部和虚部。然后通过

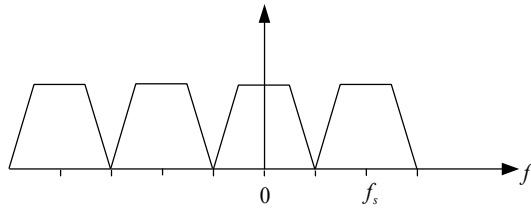
低通滤波器，频域上就是滤除掉高频的镜频分量，而在时域上是将分离出的实虚部通过滤波插值得到 I、Q 两路同一时刻的采样值。滤波后数据率仍为 $4B$ ，进行 1/4 抽取是为了降低数据率，在频域上等效为降低频谱间的间隔，最后得到的就是所需信号的复包络。可见，通过数字正交采样，不仅得到了信号的复包络，可以方便的获取信号的幅度及相位信息，同时还降低了数据率，使后续的信号处理大大减轻了运算负荷。

从上面的分析中我们可以看出要在 FPGA 内部实现数字正交采样可以分为以下几步来实现：

1. 将高速 AD 采样的时域信号正交化分成实部和虚部，通过乘 \cos 和 $-\sin$ 实现。
2. 然后通过低通滤波器，滤除高频部分的镜频分量。
3. 进行抽取是为了降低数据量。

其频谱变化过程如图 3.2 所示。





(g) 抽取后的信号频谱

图 3.2 数字正交采样的频域表示

3.2 数据正交化的 FPGA 实现

在该系统中 AD 的采样率为 40Mhz，信号带宽为 10Mhz，I 路乘 $\cos w_0 t_n$: 1, 0, -1, 0; Q 路乘 $-\sin w_0 t_n$: 0, -1, 0, 1。

I 路: $x(0), -x(2), x(4), \dots$

Q 路: $-x(1), x(3), -x(5), \dots$

Q 路设计与 I 路设计类似。

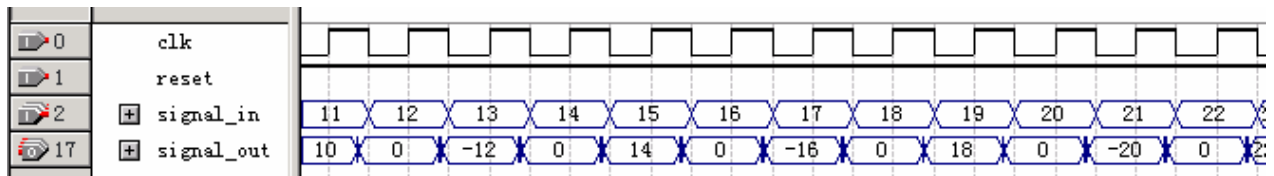


图 3.3 数字正交模块时序仿真

3.3 FIR 滤波的 FPGA 实现

一般来说，FIR 滤波器的系数是固定的，当数据率很高时，若数据/系数位数不高、阶数又不是特别多，用 FPGA 比用 DSP 得到的 FIR 滤波器效率和性能高得多。

为了采用乘加器阵列，硬件上大都基于转置型 FIR 结构，如图 3.4，因为这种结构中，每个乘加器结构都是一样的，且多个 FIR 滤波器可以直接级联。

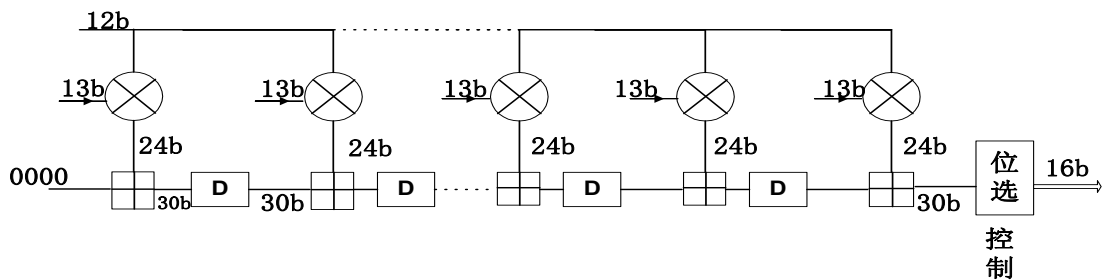


图 3.4 FIR 滤波器直接型的转置结构

本文设计的 FIR 数字滤波器为 32 阶低通滤波器，采用等纹波，滤波器带宽为 5MHz，截止频率为 6MHz，输入数据为 16 位有符号数，滤波器系数同样为 16 位有符号数，采用线性相位型的网络结构。主要分为以下几步来实现：

3.3.1 生成滤波器系数

根据设计要求用 Matlab 产生 FIR 的系数，再将生成的系数转化为整数形式。

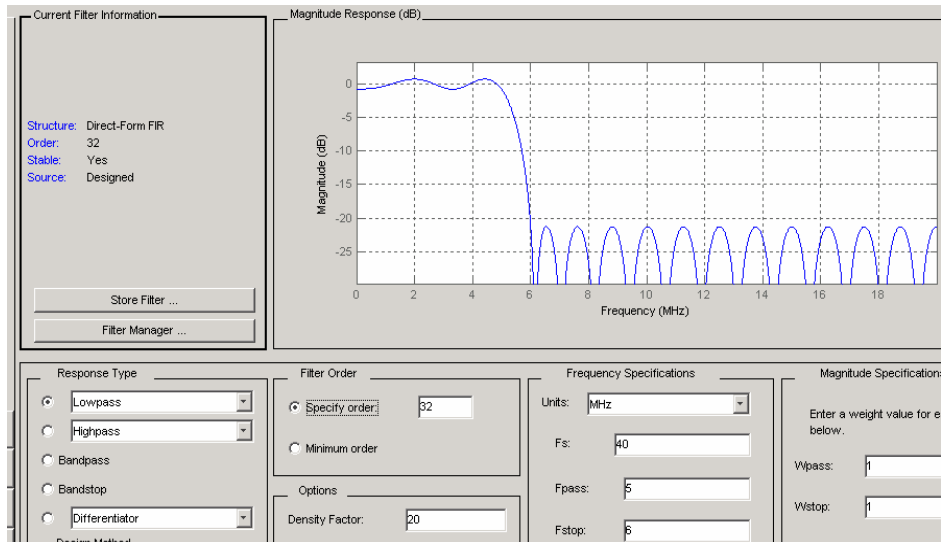


图 3.5 MATLAB 的 Filter Design Tool

3.3.2 移位输入模块

输入引脚有时钟信号 `clk`，为了跟采样率匹配，采用 40MHz 全局时钟；置位清零信号 `reset`，低位有效；16 位输入数据 `signal_in`，依次串行输入，通过 31 级触发器延迟，依次寄存。由于采用线性相位型网络结构，该模块同时完成系数对称位的输入数据的有符号加法，并将 16 个计算结果同时并行输出。每个加法器的输出为 17 位有符号数。

3.3.3 乘法器模块

由于 Cyclone II 新增的嵌入式乘法器在速度和性能上都大大优于用 LE 搭建的乘法器，故该模块直接调用该乘法器进行运算。共使用 16 个 16×17 位的乘法器，其输入为 17 位的移位输入模块输出的数据和 16 位的滤波器系数。输出结果为 33 位有符号数。

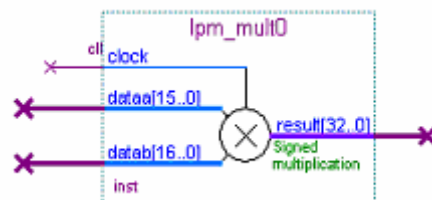


图 3.6 乘法器模块

3.3.4 结果累加模块

乘法器的 16 个 33 位输出数据作为输入数据，输出结果为该 16 个数据的简单累加和。其实现方式可以调用 Quartus II 自带的 `parallel_add` 加法模块，直接生成一个 16 输入的加法器，输出结果为 37 位的有符号整数。从逻辑功能上分析，该加法模块简单明了，输出结果在功能仿真中也是准确无误。然而该滤波器设计是在高速状态下完成，因此在器件实际工作时的时序延迟不得不考虑在内。16 输入加法器的内部组合逻辑相对较复杂，其相应的逻辑延迟 t_{logic} 就会相对较大，这会严重影响电路工作的最小时钟周期，而无法满足高速数字信号处理的需求。于是在设计该模块时，实际采用的是 4 个 4 输入和 3 个 2 输入的加法器（采用 3 级流水结构，第一级为 4 个 4 输入加法器，第二级为 2 个 2 输入加法器，第三级为 1 个 2 输入加法器）来实现，这样可以减小逻辑延迟的时间，从而提高电路工作的最

小时钟周期。

3.3.5 FIR 的结果分析

对设计进行时序仿真，其波形仿真的结果如图 3.7 所示。fir_output 为滤波器的输出结果，其中前 31 个有效数据为暂态响应结果，从第 32 位开始为我们所需要的滤波器稳定输出的结果。

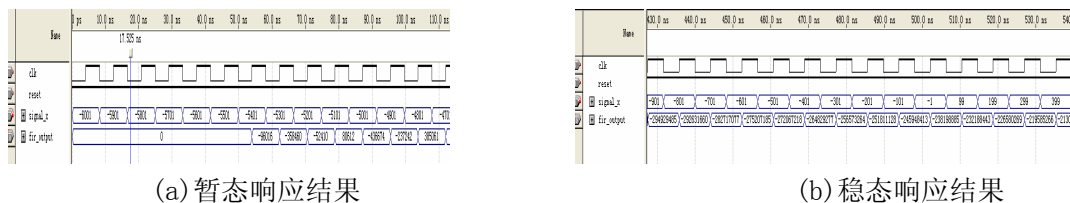


图 3.7 滤波器功能仿真波形图

将波形文件保存成 .tbl 格式的，然后从 MATLAB 中调用该文件经过处理可得到滤波器输出结果的数据。图 3.8 为 quartus II 中得到的滤波器输出结果，图 3.9 为 MATLAB 中运算得到的滤波器输出结果。图 3.10 为两者的相对误差。经比较可以发现相对误差在 10^{-5} 数量级，处理结果理想。

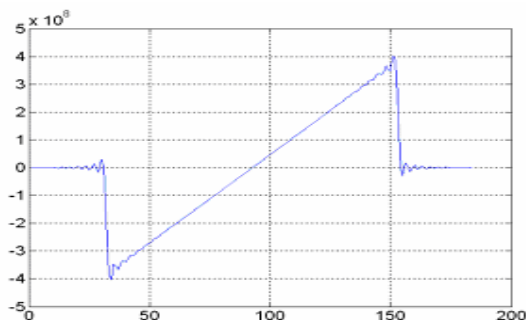


图 3.8 quartus II 中的滤波器输出结果

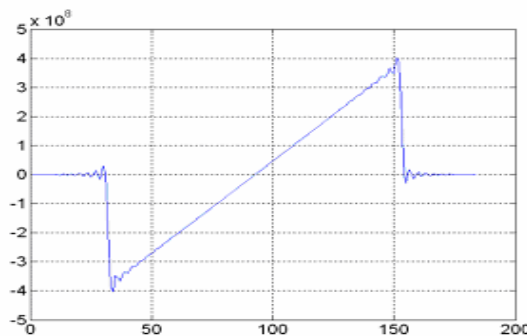


图 3.9 MATLAB 中的滤波器输出结果

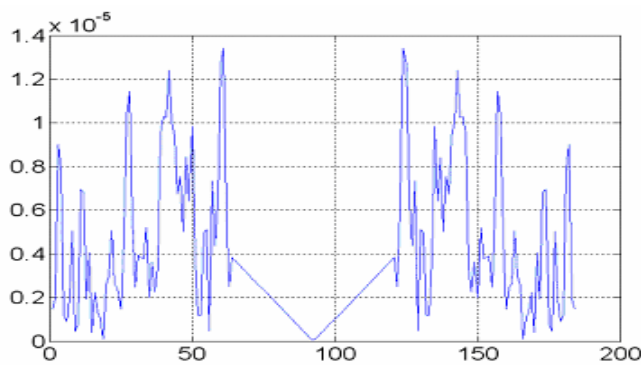


图 3.10 仿真结果与运算结果相对误差

3.4 抽取

为了降低数据处理量需要对滤波结果进行抽取，该系统根据输入信号的不同做不同倍数的抽取，主要有 2, 4, 8 倍抽取。低通滤波的数据率是 40MHz，我们只需对 40MHz 时钟进行相应分频，然后用分频的时钟对数据采样就可以了。在这里为了便于调整时钟相位用 160MHz 时钟进行分频，产生 20MHz, 10MHz, 5MHz 的抽取时钟。

3.5 整体结构与结果分析

根据前面的设计把正交化模块、FIR 模块、抽取模块组合在一起就构成 DDC 模块最后的结构如图 3.11 所示。用 Matlab 模拟产生一批数据放到 ROM 中，让 FPGA 处理,用 signal TapII 观察器处理结果，通过 Matlab 分析可得该 DDC 满足要求，其结果如图 3.12 所示。

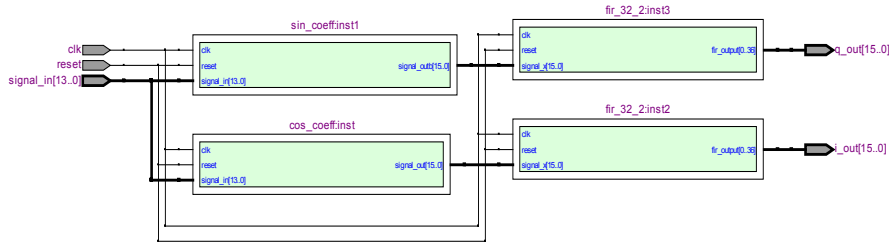


图 3.11 DDC RTL 结构图

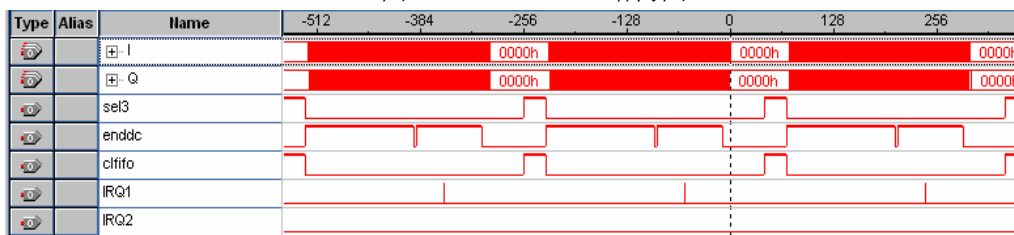


图 3.12 FPGA 处理 DDC 的结果

4. FPGA 与 TS201 之间的高速链路口通信设计

4.1 TS201 链路口结构

在该系统中由于 FPGA 与多片 DSP 之间通信,如果采用总线的方式存在两个问题,第一是 DSP 与 FPGA 之间容易产生总线冲突;二是 DSP 的数据线与地址线较多,占用 FPGA 的 IO 口资源,布线复杂。而采用链路口通信可以避免以上两个问题,而且传输速度快,因此采用链路口的方式更适合 DSP 与 FPGA 的通信。

表4.1 链路口引脚定义

引脚	定义
LxDAT03-0P	链路口发送通道数据线, LVDS P 端
LxDAT03-0N	链路口发送通道数据线, LVDS N 端
LxCLKOUTP	链路口发送通道时钟, LVDS P 端
LxCLKOUTN	链路口发送通道时钟, LVDS N 端
LxACKI	链路口发送通道确认输入信号, 通过该信号, 接收端将指示发送端可以继续发送数据
LxBCMP0	链路口数据块传输完毕, 用来指示 DMA 传输时接收端已经接收完毕发送端发送的数据块
LxDATI3-0P	链路口接收通道数据线, LVDS P 端
LxDATI3-0N	链路口接收通道数据线, LVDS N 端

LxCLKINP	链路口接收通道时钟，LVDS P 端
LxCLKINN	链路口接收通道时钟，LVDS N 端
LxACKO	链路口接收通道确认输出信号，通过该信号，发送端将指示接收端可以继续发送数据
LxBCMPI	链路口数据块传输完毕，用来指示 DMA 传输时接收端已经接收完毕发送端发送的数据块

每片 TS201 有 4 个全双工链路口。每一个链路口独立的进行接收和发送操作，可以与其他链路口连接也可以不连。每个链路口可采用 1 位或 4 位传输模式；TS201 链路口结构图 4.1 所示。

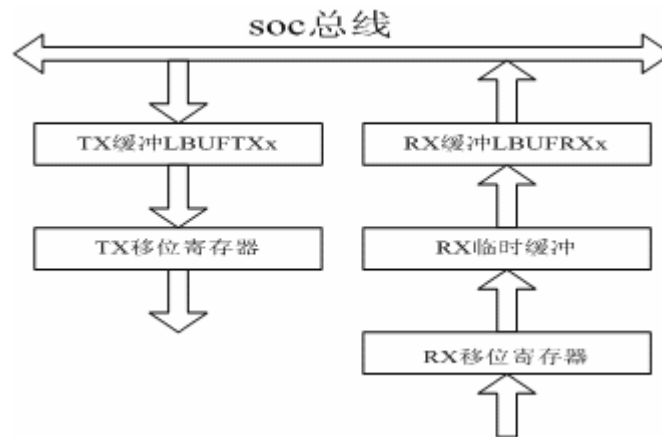


图 4.1 TS201 链路口结构

链路口传输过程主要包括以下几步：

- 1) 当发送方检测到 LxACKI 为高电平时，意味着接收缓冲器为空，发送方可以发送数据；
- 2) 第一个数据（1 位或 4 位）总是在链路口时钟（LxCLOUTP）的上升沿发送；
- 3) 最后一个数据（1 位或 4 位）总是在链路口时钟（LxCLKOUTP）的下降沿发送；
- 4) 当链路口停止时 LxCLKOUTP 为低；
- 5) 每次传输最少为 4 个 32 位字。每传输 4 字当采用 1 位模式时需要 64 个链路口时钟周期，采用 4 位模式时需要 16 个链路口时钟周期。

4.2 FPGA 向 DSP 发数的设计

链路口发送模块主要包块数据缓存、数据转化、数据输出、控制和时钟模块构成，其结构如图 4.2 所示。

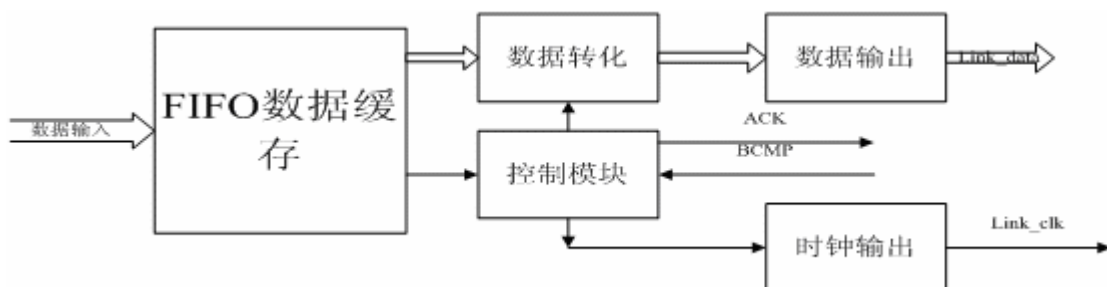


图 4.2 FPGA 发送模块结构

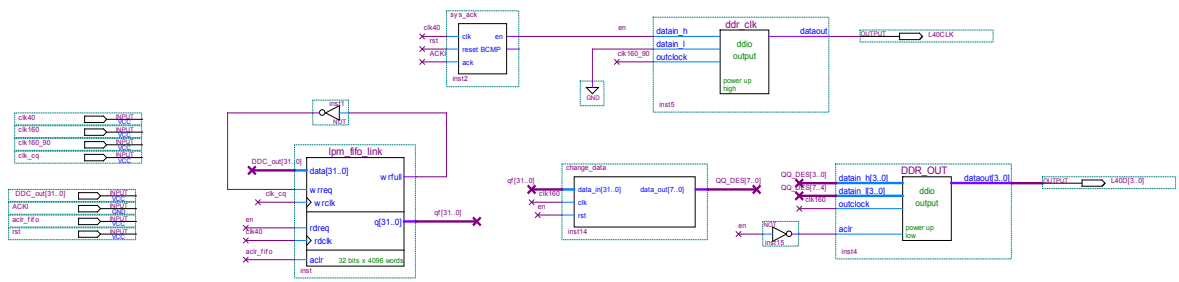


图 4.3 FPGA 发送模块

在本系统中 DDC 处理后的数据以抽取时钟的速率写入 FIFO 中；当控制模块检测到 ACK 信号为高时，整个系统开始工作，以 40Mhz 的速率从 FIFO 读出 DDC 的处理结果；数据转化模块负责把 32 位数据转化为 8 个 4 位数，其数据率为 320Mhz。下面是各部分的具体实现。

4.2.1 FIFO 数据缓存

由于前面 DDC 的处理后的数据数据率为抽取时钟的速率，而链路口的最低传输速率为 1/4 核时钟(DSP 的核时钟为 600Mhz)，在该系统中发送速率为 320MHZ，为了解决写入数据与读出数据的速度不匹配，在这里加一个 FIFO 作为数据缓存。FIFO 的深度为 32bit*4098 字的，在该系统中每次向 FIFO 写入 4000 个字后 FIFO 清零一次，及每次向 DSP 传 4000 个字 32 位字。

4.2.2 控制模块

该模块主要控制整个链路口发送模块各部分的工作状态，通过判断 ACK，当检测到 ACK 信号为高时，用 40Mhz 时钟延时两个时钟周期后，将 en 置高，整个系统开始工作，同时将 BCMP 置高，因为在这里对 FPGA 而言没有用 DMA 传输，所以不必考虑 BCMP 信号。

4.2.3 数据转化模块

主要是负责将 FIFO 读出的数据 32 为宽的数据转化为 8 个 4 位宽的，数据通过链路口送到 DSP。在 ACK 来后的第一个时钟的上升沿开始工作。其输入时钟为 160Mhz，数据率为 320Mhz，相当于 DDR 模块，在时钟上升沿和下降沿同时工作。

4.2.4 时钟模块

当 en 信号有效后开始输出时钟送往 DSP，DSP 用该时钟接收链路口送来的数据，这里为了保证时钟能准确地采到数据，将该时钟移相 90 度。送往 DSP 的时钟为 160Mhz。

4.2.5 结果验证

将 DDC 处理后的数据用采样时钟写到数据缓存 FIFO 中，这里做 4 倍抽取，其抽取时钟为 10Mhz，以 40Mhz 的数率读出经过链路口发送模块发到 DSP。经过分析，FPGA 通过链路口可以准确地向 DSP 发送数据。

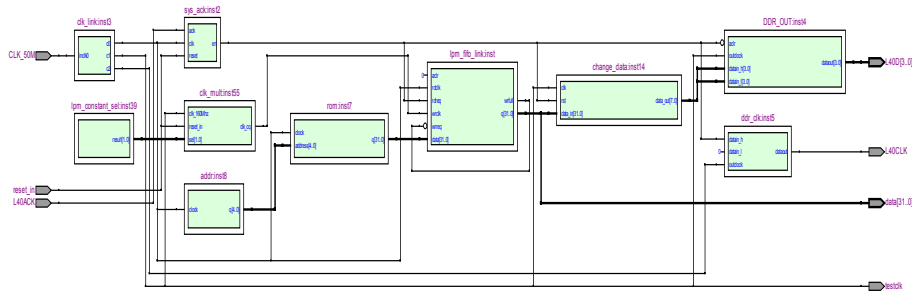


图 4.4 FPGA 发送模块的 RTL 结构

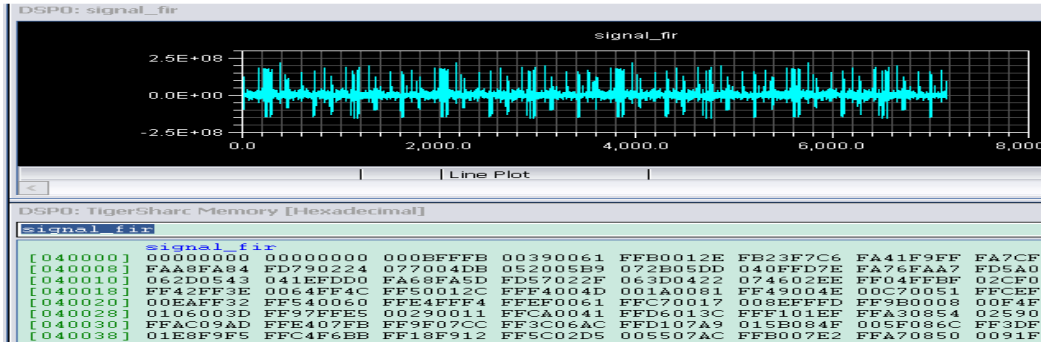


图 4.5 DSP 接收到 FPGA 处理的数据

4.3 FPGA 链路口接收 DSP 的数据

链路口接收模块主要包括数据转化模块、数据缓存模块、控制模块。当控制模块收到发送请求时，产生 ACK 送往 DSP，同时启动数据转化模块和 FIFO 数据缓存模块。数据转化模块将 DSP 发过来的 4 位宽的数据转化为 32 为宽的数据写到 FIFO 中。同时 FIFO 以 20MHZ 的速率向外板发送数据，当数据发送完毕后系统产生清零信号，接收模块准备接收下一批数据。

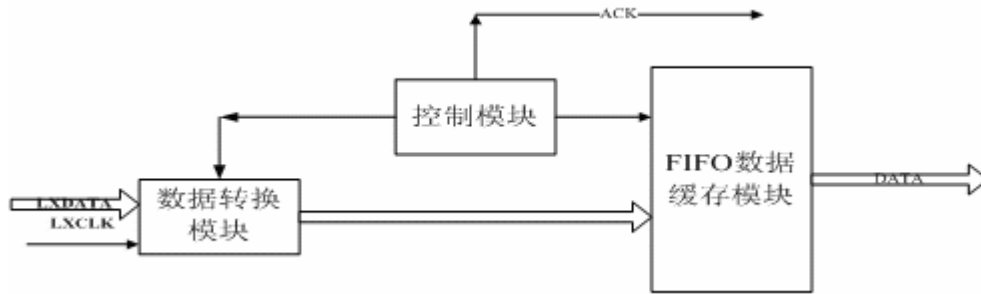


图 4.6 FPGA 链路口接收模块结构

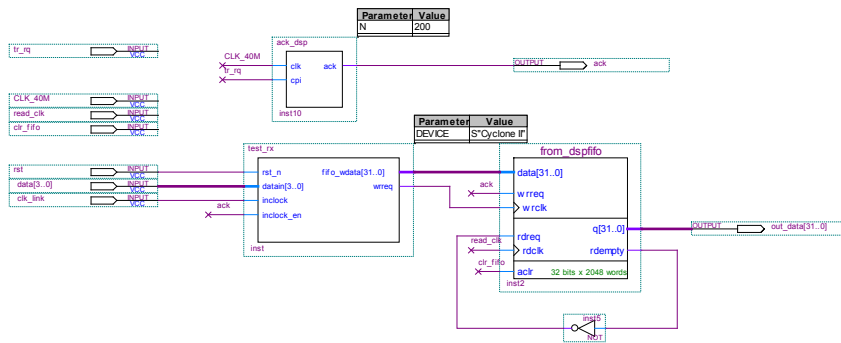


图 4.7 FPGA 接收模块

4.3.1 数据转化模块

主要是负责将 DSP 传来的 4 位宽的数据转化为 32 位宽的数据写到 FIFO 中；当 ACK 有效时，开始转化，每 8 个 4 位数据转化为 1 个 32 位宽的数据，每转化一个数据需要 4 个链路口时钟周期，第一个时钟的上升沿是低 4 位，第四个时钟的下降沿是高 4 位。链路口输入时钟为 150Mhz，数据率为 300Mhz，相当于 DDR 模块，在时钟上升沿和下降沿同时工作。

4.3.2 控制模块

主要控制接收部分的工作状态。当受到外部发送请求信号时，输出 ACK 信号，通知 DSP 可以发送数据，同时启动接收模块准备接收数据。

4.3.3 FIFO 数据缓存

由于 FPGA 向外板发送的速率与数据转化的速率不一致在这里需要加一个 FIFO，当 ACK 有效时数据以四分之一链路口时钟的速率写入 FIFO 中，同时以 20MHZ 的速率向外板发送，当一批数据发送完毕后 FIFO 清零，接收模块准备接收下一批数据。

4.3.4 仿真与验证

在 DSP 中放一批数据，这里放的是 32 位全 1 到 F 的数据。当 DSP 收到 FPGA 的 ACK 信号后开始向 FPGA 发数，DSP 以 150Mhz 的数据率发送，FPGA 将接收的数据放到 FIFO 中再以 20Mhz 的速度读出。图 4.8 是用 signal Tap II 观测到从 FIFO 读出数据，由观测结果可知 FPGA 收到的数与 DSP 发送的数据一致。通过分析可知该模块可以准确接收 DSP 传来的数据。

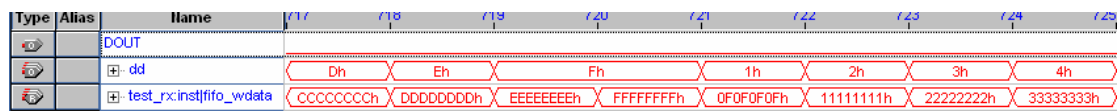


图 4.8 FPGA 接收到的数据

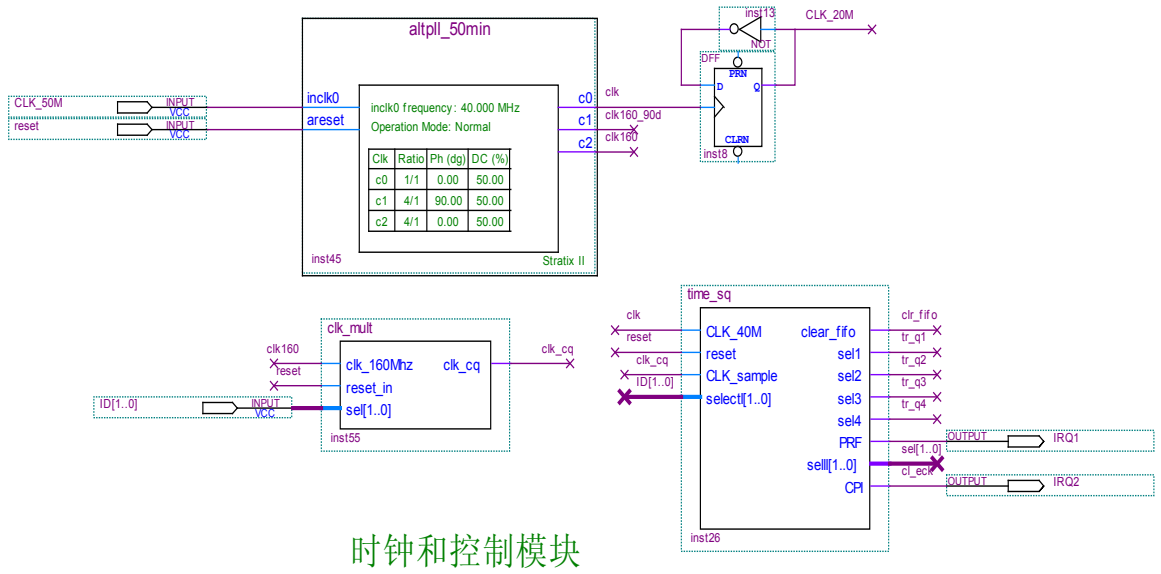
5. 系统总体设计与结果分析

5.1 系统整体设计

经过前面的分析设计，信号处理模块，链路口发送接收模块均已实现，现在只需将上面设计的模块组合在一起，再通过控制模块，就可以构成该信号处理系统。我们把该系统分为时钟和控制模块，信号处理模块，数据发送模块，数据接收转发模块。

5.1.1 时钟和控制模块

该模块主要产生各模块所需的时钟主要有 40Mhz，160Mhz，20Mhz，10Mhz，5Mhz。同时控制 DSP 接收和发送数据，控制 FPGA 信号处理和数据通信。其结构如图 5.1 所示。

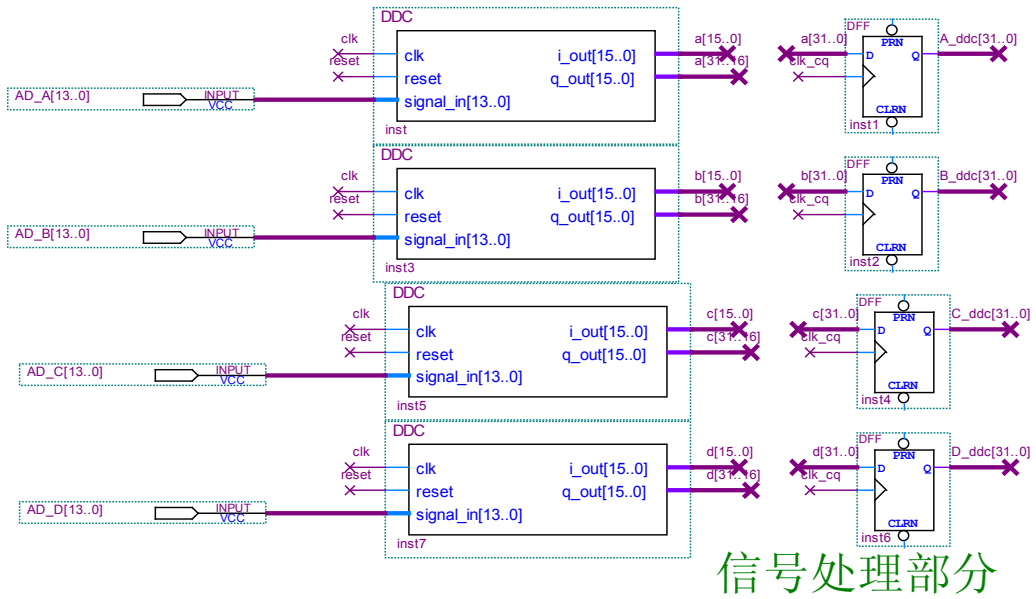


时钟和控制模块

图 5.1 时钟和控制模块

5.1.2 信号处理模块

该模块主要接收 4 路 AD 采样的信号，把接收到的信号进行数字下变频，4 路信号处理都以 40Mhz 的速度并行处理。其结构如图 5.2 所示。

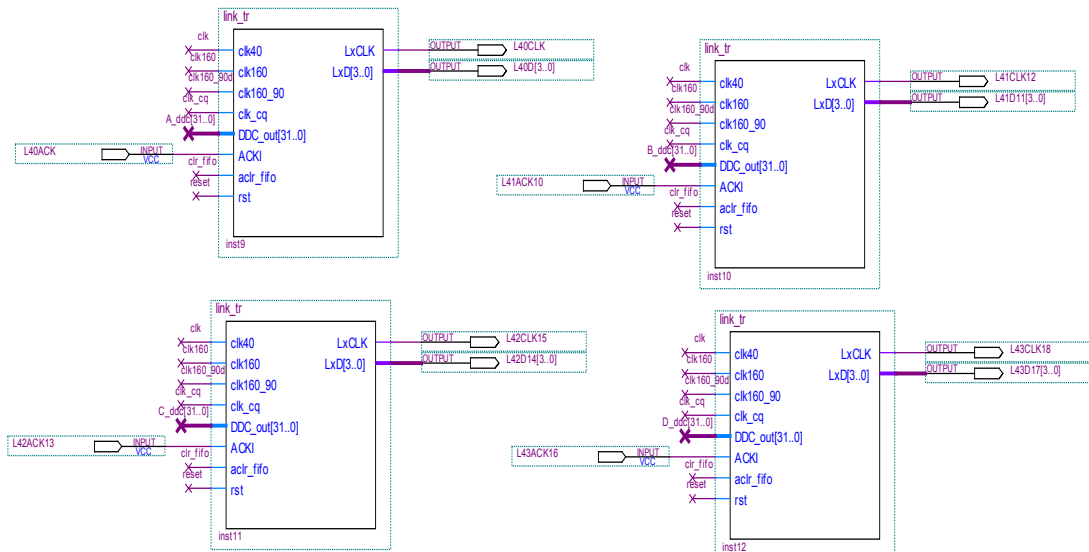


信号处理部分

图 5.2 信号处理模块

5.1.3 数据发送模块

4 路 DDC 的处理结果通向 4 个链路口发送模块，链路口发送模块再把数据分别送给 4 片 DSP 进行后续信号处理，该模块并行处理，每个模块的输出数据率为 320Mhz。其结构如图 5.3 所示。

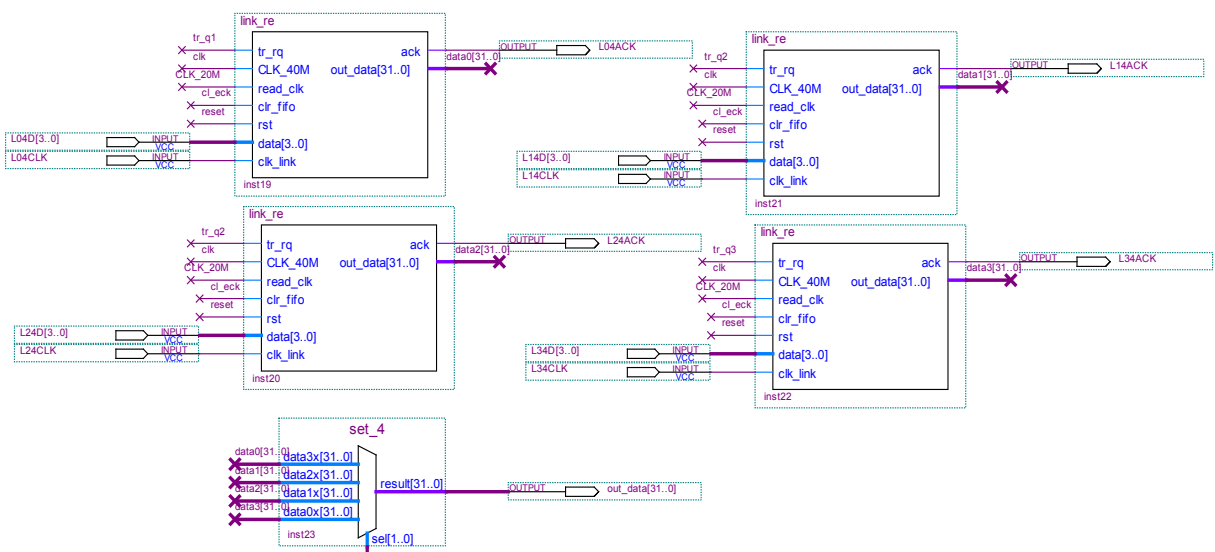


FPGA向DSP发数

图 5.3 数据发送模块

5.1.4 数据接收转发模块

4 片 DSP 的处理结果，通过链路口回传给 FPGA，经过 FPGA 汇总后再传给其他信号处理板。该模块以 150Mhz 的速率接收 4 片 DSP 的处理结果，再经过汇总以 20Mhz 的速度传给后板。其结构如图 5.4 所示。



FPGA接受DSP的处理结果同时向外板发数

图 5.4 数据接收转发模块

5.1.5 结果验证

将以上的四个模块建成为 FPGA 的顶层程序。经过 Quartus II 软件的全编译后，当然在编译前要做一些时序约束，保证布线时高速时钟线和数据线的稳定、可靠。生成的编译报告如图 5.5，整个 FPGA 应用程序占用片上资源的情况是：锁相环 25%，逻辑单元 7%，乘法器 64%，引脚 39%，存储器 47%。片上资源足够而且留有很大的裕量，方便以后程序升级。

Flow Status	Successful - Fri Jun 20 15:36:03 2008
Quartus II Version	7.1 Build 156 04/30/2007 SJ Full Version
Revision Name	quadrature_sampling
Top-level Entity Name	top_threeDDC
Family	Cyclone II
Device	EP2C70F672I8
Timing Models	Final
Met timing requirements	No
Total logic elements	4,663 / 68,416 (7 %)
Total combinational functions	3,831 / 68,416 (6 %)
Dedicated logic registers	3,583 / 68,416 (5 %)
Total registers	3591
Total pins	166 / 422 (39 %)
Total virtual pins	0
Total memory bits	537,216 / 1,152,000 (47 %)
Embedded Multiplier 9-bit elements	192 / 300 (64 %)
Total PLLs	1 / 4 (25 %)

图 5.5 系统资源占用情况

FPGA 在 40MHz 下可以工作稳定，很好的对 4 片 DSP 进行控制，对输入信号进行数字下变频的实测结果图 5.6。将 DSP 收到的 FPGA 处理结果与 Matlab 处理结果对比可以看出，除了一些暂态部分有些差异之外，余下部分相减的误差均小于 10^{-8} ，在误差允许范围内，说明该设计满足系统要求，可以完成相应的任务。

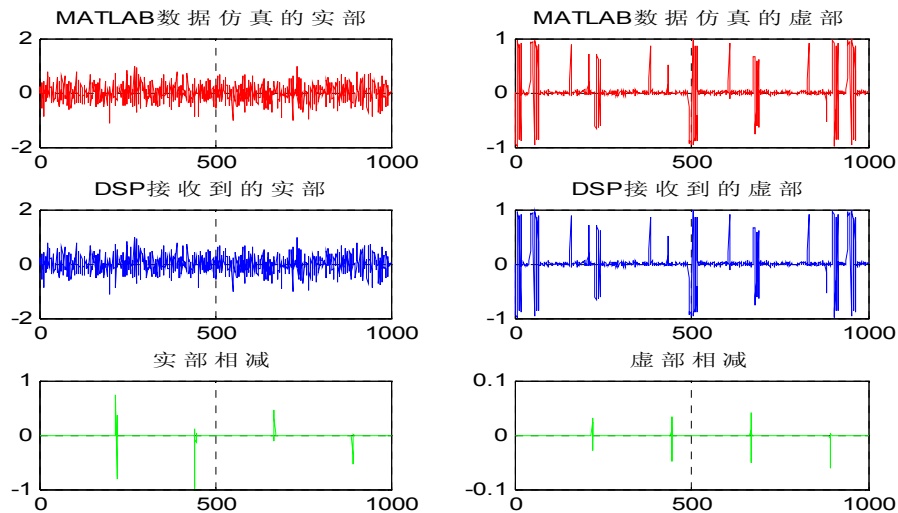


图 5.6 结果分析

5.2 结论

本文结合具体高速信号处理板，该信号处理板采用 DSP+FPGA 的结构。结合 FPGA 与 DSP 的各自优点，在 FPGA 上实现数字下变频和链路口通信以及 FPGA 对系统整体控制。其相应的算法和接口通信均可以实现，该系统可以完成相应的信号处理任务，对其他高速实时信号处理系统设计有一定的参考价值。

由于 FPGA 并行处理的优势，在很多高速处理系统都希望更多的算法在 FPGA 中进行，这对提高系统的实时性很有必要。很多信号处理系统中，速度的瓶颈已不是芯片的处理速度而是数据传输。因此

将更多的信号处理算法在 FPGA 上实现和提高数据传输速率有必要再做进一步的研究，也是未来高速信号处理的研究热点。

参考文献

- [1] 苏涛, 何学辉, 吕林夏, 《实时信号处理系统设计》。西安电子科技大学出版社, 2006.
- [2] Uwe Meyer-Baese, Digital Signal Processing with Field Programmable Gate Arrays (Second Edition) , 2003.10.
- [3] San Jose. Tool Flow for Design of Digital IF for Wireless Systems. Altera Corporation, May 2007.
- [4] San Jose. Analog Devices Link-Port Reference Design. Altera Corporation, February 2005.
- [5] Analog Devices. ADSP-TS201 Tigers ARC® Processor Hardware Reference, Revision 1.1, December 2004.
- [6] 苏涛, 吴顺君, 李真芳等, 《高性能 DSP 与高速实时信号处理》(第二版), 西安电子科技大学出版社, 2002.
- [7] San Jose. Cyclone II Device Handbook . Altera Corporation , 2004.

原创性声明:

本人郑重声明, 此处提交的论文《基于 FPGA 的多通道高速实时信号处理系统设计》, 是本人在导师指导下, 在西安电子科技大学期间进行研究工作所取得的成果。据本人所知, 论文中除已注明部分外不包含他人已发表或撰写过的研究成果。本声明的法律效果将完全由本人承担。

作者签字: 冼友伦

日期: 2008 年 7 月 25 日

作者简介:

冼友伦 男, 1983 年生, 现在西安电子科技大学电子工程研究所攻读硕士学位。研究方向为高速实时信号处理。

卢护林 男, 1984 年生, 现在西安电子科技大学电子工程研究所攻读硕士学位。研究方向为高速实时信号处理。

苏涛 男, 1968 年生于陕西, 博士, 教授, 博士生导师。主要从事 DSP 和高速实时信号处理方面的研究。

联系方式:

通信地址: 西安电子科技大学电子工程研究所

邮编 : 710071

电话: 13572282641

Email: jiack163@163.com