

---

# 基于 FPGA 的高速 LDPC 码编码器的设计与实现

丁宏 杨帅

指导教师：罗武

(北京大学 卫星与无线通信实验室 北京 100871)

## 摘要：

LDPC 码是通信系统中一种性能十分优秀的信道编码。本文针对便于硬件实现的 QC-LDPC 码进行了编码器设计，采用多路并行、流水线结构、优化关键路径等多种手段，在 Altera 公司 FPGA 平台上实现了编码速率高达 1.6Gbps 的编码器，并使用逻辑分析仪验证了编码器在高速运行下结果的正确性。

**关键词：**LDPC，高速编码器，FPGA

## Design and Implementation of LDPC High Speed Encoder on FPGA

Ding Hong, Yang Shuai

Director: Luo Wu

(Satellite and Wireless Communication Lab, Peking University, Beijing, 100871)

## Abstract:

LDPC code is a kind of channel code with good performance in communication system. In this paper, we have design a high speed encoder with QC-LDPC structure. In our design, we have adopt many methods to increase the speed of encoder, such as partial parallel strategy, pipelined structure and optimization of the shortest critical path. Finally, we have verified the correction of the encoder when it works at high speed.

**Key Words:** LDPC, High Speed Encoder, FPGA

## 1 引言

信道编码，是现代通信系统中的编码方式之一，除此之外还有信源编码、密码编码和多址编码等。它是在信息序列上有目的地增加一些监督比特，使之具有检错或纠错的能力，从而提高通信系统传输的可靠性。信道编码器是完成从信息序列到编码码字之间映射的功能。

一种性能优秀的信道编码能够通过增加较少的冗余信息来获得尽可能强的纠错能力。LDPC 码是一种性能十分优秀的信道编码。它是 Gallager 于 1962 年发表的论文《Low-Density Parity- Check Code》中提出的[1]。在文章中，他证明了 LDPC 码性能接近于香农限，同时提出了构建 H 矩阵的一种方法，以及两种解码方法和及其电路原理图。限于当时的科技水平和硬件条件，LDPC 码没有得到重视和推广。直到 1996 年，D. Mackay 和 R. Neal 证明了 LDPC 码性能和成本都优于 Turbo 码[2]，使得 LDPC 码再次进入人们的视野，并且掀起了一番研究的热潮。在许多场合下 LDPC 码性能优于 Turbo 码，具有较大的参数灵活性和较低的误码率，译码复杂度更低，理论上可实现完全的并行操作，便于硬件实现，极具高速编译码潜力。因而 LDPC 码成为当今信道编码领域最瞩目的热点。

LDPC 码被广泛认为是下一代移动通信系统的信道编码方案。现在它已经被多个通信标准所采用，例如 WLAN、DVB-S2、WIMAX[3]等。随着通信系统的不断发展，人们对通信系统的速度要求也越来越高，高达上 Gbps 的传输速率将成为下一代移动通信的要求。因此

高速 LDPC 码编码器设计成为一个十分热门的研究方向。本文首先介绍了 LDPC 码编码原理，然后提出了码长为 4032，码率为 1/2 的 LDPC 码编码器设计方案，最后给出了设计和测试结果。

## 2 LDPC 码编码原理

LDPC 编码主要有两种方法：

1. 直接编码方法。它是将编码码字直接乘以生成矩阵完成编码，如下所示：

$$C = X * G$$

其中， $X = [x_1 \ x_2 \ \dots \ x_m]$  是信息序列， $C = [c_1 \ c_2 \ \dots \ c_n]$  是生成的 LDPC 码字， $n > m$ ， $G$  是  $m \times n$  的生成矩阵。

2. 间接编码的方法。它是利用编码码字和校验矩阵  $H$  的关系，采用间接运算的方法实现编码，如下所示：

$$H * C^T = 0$$

其中  $H$  是  $r \times n$  的矩阵 ( $r = n - m$ )，称为一致校验矩阵 (简称校验矩阵)。

本文采用第二种方法对 LDPC 码进行编码。

### 2.1 H 矩阵的结构

本文实现的编码器所对应的 LDPC 码的  $H$  矩阵采用 Block-PEG 算法[4]构造的 QC-LDPC 码，它可以用分块偏移矩阵的形式表示。这种分块偏移矩阵的结构如图 1 所示，其中  $H$  矩阵被划分为  $m_b \times n_b$  块，每一块都是一个大小为  $K \times K$  的矩阵。

$$H = \begin{bmatrix} H_{1,1} & H_{1,2} & \dots & H_{1,n_b} \\ H_{2,1} & H_{2,2} & \dots & H_{2,n_b} \\ \vdots & \vdots & \ddots & \vdots \\ H_{m_b,1} & H_{m_b,2} & \dots & H_{m_b,n_b} \end{bmatrix}$$

图 1 分块偏移形式的 H 矩阵

每一个  $H_{i,j}$  可以用 0 或者是一个正整数来简化地表示。如果是 0 则表示是一个  $K \times K$  的零矩阵。如果是一个正整数  $L (L \leq K)$ ，表示一个带偏移的单位矩阵。它的第一行的第  $L$  列是 1。下一行是由上一行循环右移一位得到的。例如若  $H_{i,j}$  是一个  $7 \times 7$  的矩阵， $L=4$ ，则矩阵的形式如图 2 所示。为了便于硬件设计，将上述的  $H$  矩阵进一步表示成 F-H 矩阵形式。F 矩阵

$$H_{ij} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

图 2 偏移矩阵示例

表示  $H$  矩阵中是否是一个零矩阵，如果是零矩阵则表示成 0，如果是带偏移量的单位矩阵则表示成 1。而  $H$  矩阵表示校验矩阵中非零矩阵的偏移量。由于每一行中只有有限的几个矩阵是非零的，所以可以把他们按顺序表示出来。下面以一个码长为 56 码率为 0.5，块大小为 7 的码字为例进行说明，其中  $H = [H_i H_p]$ 。  $H_i$  和  $H_p$  的 F-H 形式分别如图 3 和图 4 所示。

$$H_i: F = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \quad H = \begin{bmatrix} 0 & 6 \\ 1 & 4 \\ 2 & 5 \\ 1 & 0 \end{bmatrix}$$

图 3 分块偏移矩阵  $H_i$  的 F-H 表示

$$H_p: F = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad H = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

图 4 分块偏移矩阵  $H_p$  的 F-H 表示

其中  $H_p$  矩阵的  $H$  矩阵只有第一行第二列为 1。它的形式不同于其他的偏移矩阵，矩阵中右上角的 1 被 0 代替。其表示形式如图 5 所示，其中黑色方格代表 1，白色方格代表 0。

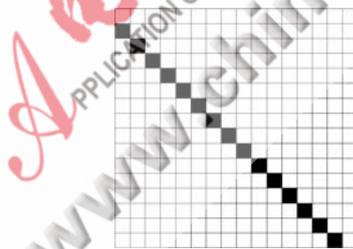


图 5  $H_p$  矩阵右上角的子矩阵形式

这样设计的目的是为了便于完成矩阵乘法的逆运算，下一节将具体介绍。

## 2.2 编码算法

在间接编码的方法中，设  $H = [H_i H_p]$ ，  $C = [X_i X_p]$ 。根据  $H * C^T = 0$ ，将上面两个式子代入得到如下式子：

$$H_i * X_i^T + H_p * X_p^T = 0$$

设  $H_i * X_i^T = X_b$  则  $X_b = H_p * X_p^T$ 。从而  $X_p^T = H_p^{-1} * X_b$ ，所以我们可以根据上面设计的  $H$  矩阵首先根据输入信息求出  $X_b$ 。然后利用矩阵的逆运算求出  $X_p$  从而完成编码。

**第一步求解：**  $X_b = H_i * X_i^T$

分块和偏移量描述的矩阵非常适合做矩阵和向量的乘法。设  $H_i$  为  $4 \times 4$  的分块矩阵，块的大小为  $7 \times 7$ ，对应的  $X_i^T$  和  $X_b$  可以分为 4 段，每段 7 比特。仅考虑  $H_i$  中第一行块与  $X_i^T$  的乘法， $X_b$  只涉及 4 段中的第一段，如图 6 所示，其中  $H_i$  矩阵为  $F = [1 0 1 0]$ ，  $H = [0 6]$ 。每当  $X_i$  送入

编码器一位，需要对 $X_b$ 的每一位进行一次累加更新，当 $X_i$ 一帧输入结束时，对应的 $X_b$ 就是

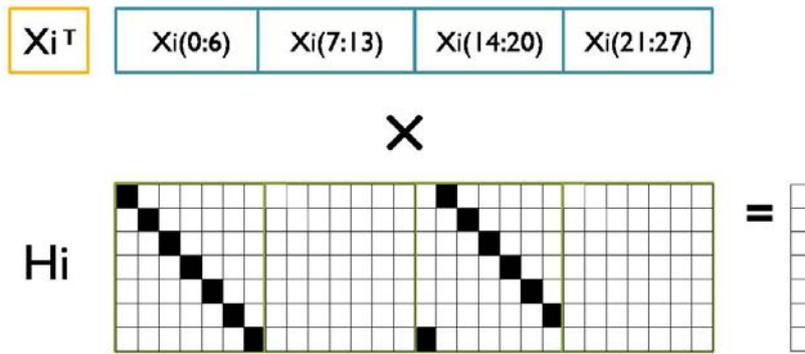


图 6  $H_i * X_i^T = X_b$  运算示例

相乘的结果了。由于 $H_i$ 具有分块偏移的特殊结构，并不需要对 $X_b$ 的每一位进行更新：当 $X_i$ 对应 0 矩阵时， $X_b$  每一位的值保持不变；对应非 0 矩阵时， $X_b$  中只有一位需要更新，而且下一个需要更新的位置在当前更新位置的下方（如果已经是最下方则循环到最上方）。所以，任何时候最多只有一位在进行累加更新。具体算法如下：

$X_i$  第 0 段输入时， $F[0]=1$  指示对应 $H_i$  第 0 块为非 0 矩阵；因为是第一个非 0 块，只需将  $X_i$  的值直接赋给 $X_b$ 。 $X_b$  中最先被赋值的位置为第 0 位，由  $H[0]=0$  指示，然后依次是第 1、2、3、4、5、6 位。 $X_i$  第 1 段输入时， $F[1]=0$  指示为 0 矩阵， $X_b$  的值保持不变。 $X_i$  第 2 段输入时， $F[2]=1$  指示为非 0 矩阵； $X_b$  中最先被累加的是第 6 位，由  $H[1]=6$  指示，将 $X_b(6)$  的值和当前的 $X_i(6)$  相加的结果赋给 $X_b(6)$ ，完成累加的运算，然后以同样的方法完成 $X_b$  中 0、1、2、3、4、5 位的累加更新。 $X_i$  第 3 段输入时， $F[3]=0$  指示为 0 矩阵， $X_b$  的值保持不变。 $X_i$  输入完毕时，此时的 $X_b$  即为最终结果。

如上所述，一个  $7 \times 28$  矩阵和  $28 \times 1$  向量的乘法只需要一个乘累加单元和少量比较逻辑实现。四个这样的模块就可以实现  $28 \times 28$  的矩阵与  $28 \times 1$  的向量间的乘法运算。

### 第二步求解： $X_p^T = H_p^{-1} * X_b$

这一步的是利用第一步中得到的 $X_b$ 和 $H_p$ 求解 $X_p$ 。由于 $H_p$ 的矩阵具有双对角分块的结构特点，因此不需要求解 $H_p$ 的逆矩阵就可以得到 $X_p$ ，求解方法如图 7 所示。由于 $H_p$ 的第一行只有一个 1。因此很容易就能得到 $X_p^T(0)$ 等于 $X_b(0)$ 。当求得了 $X_p^T(0)$ 以后对于矩阵 $H_p$ 的第八行只有一个未知数，因此很容易求出 $X_p^T(7) = X_p^T(0) \oplus X_b(7)$ 。这样按照 0,7,14,21,1,8,..... 的顺序就可以依次求出 $X_p^T$ 的所有比特。最后将信息比特 $X_i$ 和校验比特 $X_p$ 依次输出就得到 LDPC 码的系统码编码。

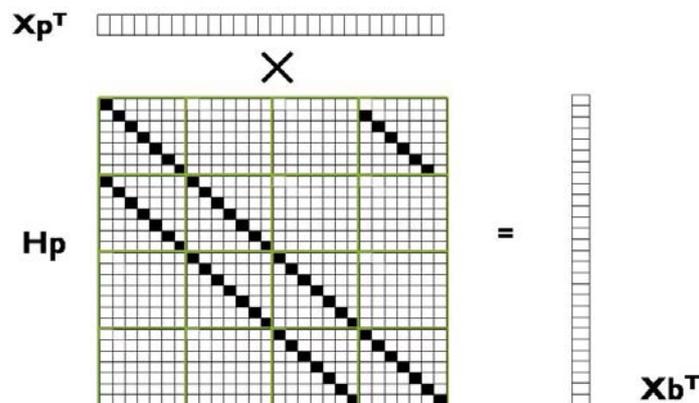


图 7  $X_p^T = H_p^{-1} * X_b$  运算示例

### 3 高速编码器设计方案

#### 3.1 整体方案

我们实现的高速编码器与 16QAM 调制技术相结合，采用四路并行编码的方案。编码器的整体结构如图 8 所示：

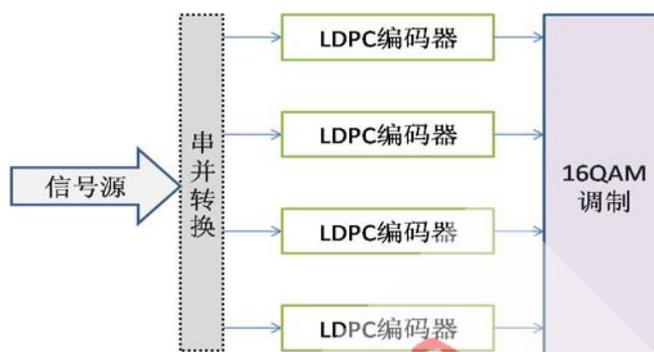


图 8 LDPC 编码器的整体设计方案

信号通过串并转换分成四路送入 LDPC 编码器，然后将编码器的输出结果进行 16QAM 调制，将调制结果发送到信道。为了达到编码速率上 Gbps 的要求，每一路编码器的编码速率要达到 250M bps 以上。为了实现这一目标，我们采用了多路并行运算，多级流水线结构，以及优化关键路径等方法，完成了整个系统的设计。设计码长为 4032，码率为 0.5，采用 QC-LDPC 码结构。

#### 3.2 LDPC 编码器的结构

LDPC 高速编码器的内部结构如图 9 所示。

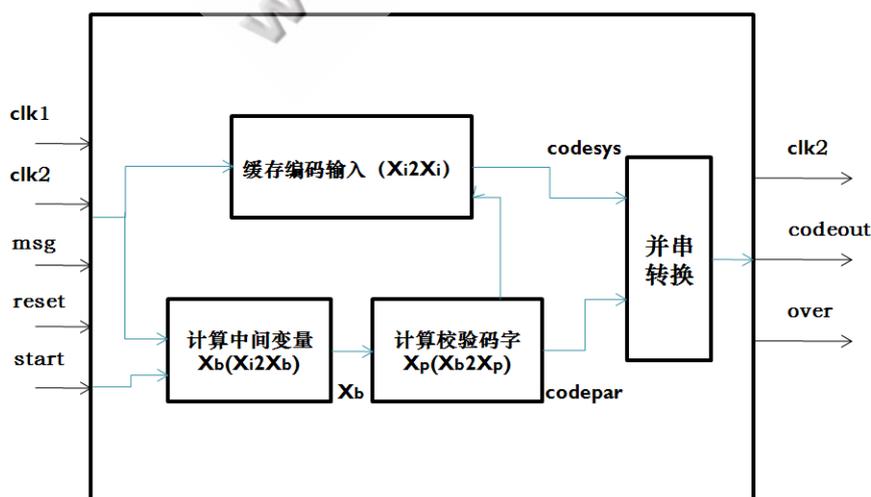


图 9 LDPC 编码器设计框图

编码器对外接口信号包括输入信号和输出信号两部分。输入信号包括系统工作时钟  $clk_1$ ,  $clk_2$ , 其中  $clk_2$  的时钟频率是  $clk_1$  的两倍, 系统复位信号  $reset$ , 输入开始指示信号  $start$  和输入信息序列  $msg$ 。输出信号包括输出开始指示信号  $over$ 、输出编码结果  $codeout$ 。在  $start$  指示有效后的 2016 个时钟周期内,  $msg$  信号输入码长为 2016 的信息序列, 完成一帧输入后可以立即接收下一帧, 输入和编码运算由  $clk_1$  驱动。 $codeout$  是由信息序列  $codesys$  和校验序列  $codepar$  经过并串转换以后得到的结果, 在  $over$  指示有效后的 4032 个时钟周期内输出编码结果, 由时钟  $clk_2$  驱动。

以下简要介绍编码器中的几个关键模块:

#### 计算中间变量 $X_b$ 的模块:

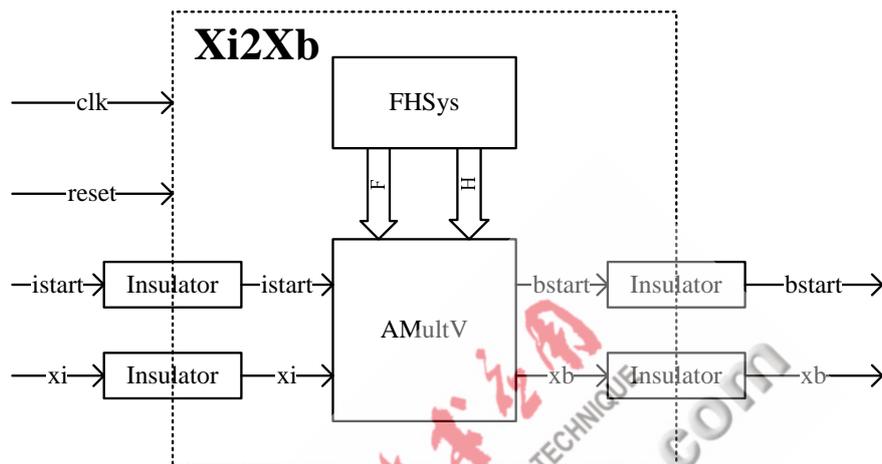


图 10 计算中间变量  $X_b$  的模块框图

本模块对外接口信号同样包括输入和输出信号两部分。输入信号包括工作时钟  $clk$ , 系统复位信号  $reset$ , 信息序列输入启动指示信号  $istart$  和信息序列  $xi$ , 在  $istart$  有效后的 2016 个时钟内输入码长为 2016 的信息序列  $xi$ 。输出信号包括中间节点输出启动指示信号  $bstart$ 、中间变量  $xb$ , 在  $bstart$  有效后的 2016 个时钟内  $xb$  输出 2016 比特的中间变量信息。 $FHSys$  存储系统码对应的校验矩阵  $H_i$ , 包括分块矩阵  $F$  和偏移量矩阵  $H$ 。 $AMultV$  则完成乘累加运算。 $Insulator$  是一个一级缓冲寄存器模块, 其作用是减小关键路径延时。

#### 计算校验码字 $X_p$ 模块:

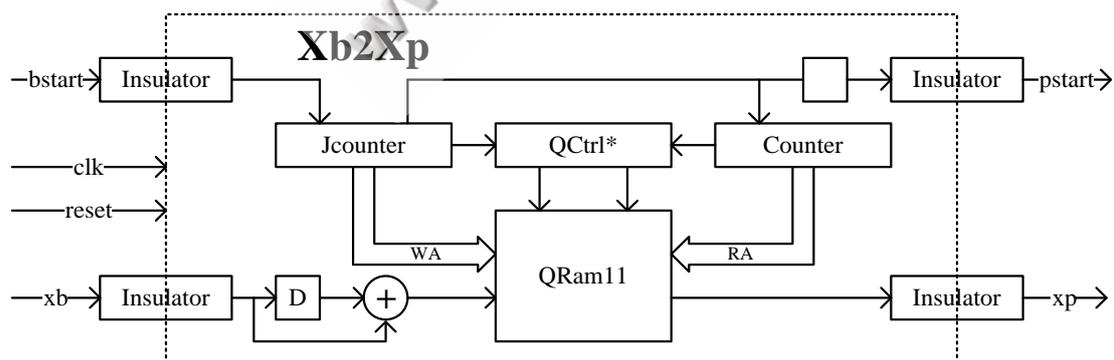


图 11 计算校验码字  $X_p$  模块框图

本模块的输入端  $xb$  不是通常的顺序输入, 而是以间隔 144 的跳越顺序输入: 0、144、288、.....、2016-144、1、145、289、.....、2015。 $Jcounter$  是一个 11 位计数器, 完成上述循序地址的产生。然后将输入的  $xb$  和上一拍的  $xp$  相加后写入到存储  $xp$  的 RAM 的对应位置。 $QRam11$  和  $QCtrl$  分别是存储和控制校验码字模块。11 位计数器  $Counter$  产生 0 到 2015 的地址信号, 从  $QRam11$  中读取  $xp$  向外输出。

### 缓存编码输入模块:

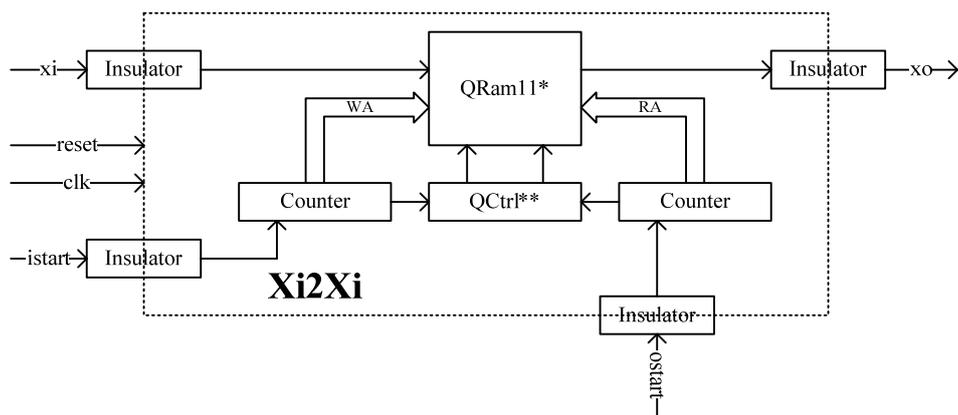


图 12 编码缓存模块框图

输入和输出的控制计数器都是 0~2015 的加 1 计数器，在外部帧启动信号到来后开始计数，写入或读出相应的数据。由于 Xi2Xb 和 Xb2Xp 每一个模块都有 1 帧左右的延时，所以 Xi2Xi 需要将信息码缓存 2 帧。QRam11 由 3 个能存储  $2^{11}$  位的 RAM 以及相应的 RAM 切换控制逻辑组成，完成缓存功能。

## 4 设计结果与测试

### 4.1 设计平台

本文采用的开发平台是 Altera 公司的 StratixII\_DSP 开发套件，它包括 StratixII\_DSP 的开发板一套，用于开发用户逻辑部分的 QUARTUS II（包括 SOPC BUILDER）软件以及用于软件开发的 NIOS II IDE。

StratixII\_DSP 开发板的核心是一片 StratixII 2S60C4ES 的 FPGA，约 600 万门左右。板上还有 FLASH, SDRAM 等存储器，串口和 JTAG 等对外接口，还有 ADC 和 DAC 各两片。

利用这套开发平台，我们可以把设计出的 LDPC 高速编码器载入到开发板的 NIOS 系统中，对开发板输入系统命令，使编码器能够正常的运行起来。通过逻辑分析仪，测试编码输出结果从而验证系统的正确性。

开发板上的 FPGA 是 Altera 公司的 FPGA 芯片 StratixII EP2S60F1020C4ES, Stratix® II 器件建立在 1.2V、90nm、SRAM 工艺之上。芯片的 1020 个输出端可以配置成不同的输出标准，比如 LVTTTL 方式，LVDS 方式，HyperTransport 方式等。其中 LVTTTL 方式在 4mA 电流驱动下，最快可以输出 225Mbps 的数据，LVDS 和 HyperTransport 方式可以输出高达 1Gbps 的数据速率。

表 1 StratixII EP2S60F1020C4ES 资源[5]

Name	Fast PLLs	Enhanced PLLs	M512 RAM 512bits	M4K RAM 4kbits	M-RAM 512kbits	RAM Total(bits)
Number	8	4	329	255	2	2,544,192

LEs	DSP Blocks	Pins
60,440	36	1020

## 4.2 设计结果

在 ModelSim 中进行功能仿真的输出波形如图 13 所示，

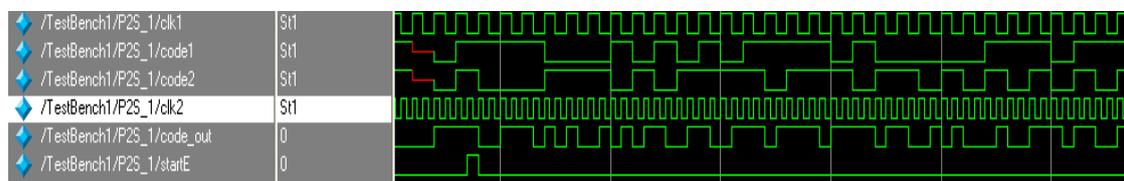


图 13 编码器输出结果

其中 code1 和 code2 分别对应编码器的系统码和校验码。经过并串转换以后得到的输出为 code\_out，其结果与 MATLAB 中的计算结果一致。使用 Quatus5.0 软件进行综合，得到的综合结果如表 2 所示。

表 2 资源使用情况

Family	Stratix II
Device	EP2S60F1020C3ES
Total ALUTs	8,944 / 48,352 ( 18 % )
Total pins	15 / 719 ( 2 % )
Total virtual pins	0
Total memory bits	69,632 / 2,544,192 ( 3 % )
DSP block 9-bit elements	0 / 288 ( 0 % )
Total PLLs	0 / 12 ( 0 % )
Total DLLs	0 / 2 ( 0 % )

综合结果显示，经过并串转换模块后输出信号的时钟速率可以达到 500MHz。实际测试时设置系统时钟速率为 400MHz，四路并行输出的信息速率为 1.6Gbps，系统可以正常工作。下一节给出了详细的测试方案。

## 4.3 测试方案

硬件测试时，将编码器输出的数据进行串并转换，转换的数据通过 16 路并行输出，使用逻辑分析仪存储编码器输出的数据，然后将逻辑分析仪和 PC 连接，与 MATLAB 计算的结果进行比较从而验证了编码器的编码正确性。

这里采用 16 路并行输出，主要是基于以下几方面的考虑，首先是要保证逻辑分析仪能够正确的采样输出数据，由于高速编码器的每一路编码输出的时钟速率都是 400M 的时钟，4 路并行达到 1.6Gbps 的速率。而逻辑分析仪的采样时钟最多只能达到 300MHz，采用 16 路并行可以使逻辑分析仪的采样时钟工作在 100MHz。另外，逻辑分析仪的一个测试端口可以引出的采样数据线有 16 根，而且 16 位的二进制数方便逻辑分析仪存储为 4 个 16 进制数，有利于后期和软件仿真的结果作比较。这种方案对 LDPC 编码器的编码输出进行了全比特的验证，更有力地验证了编码器在高速条件下编码的正确性。

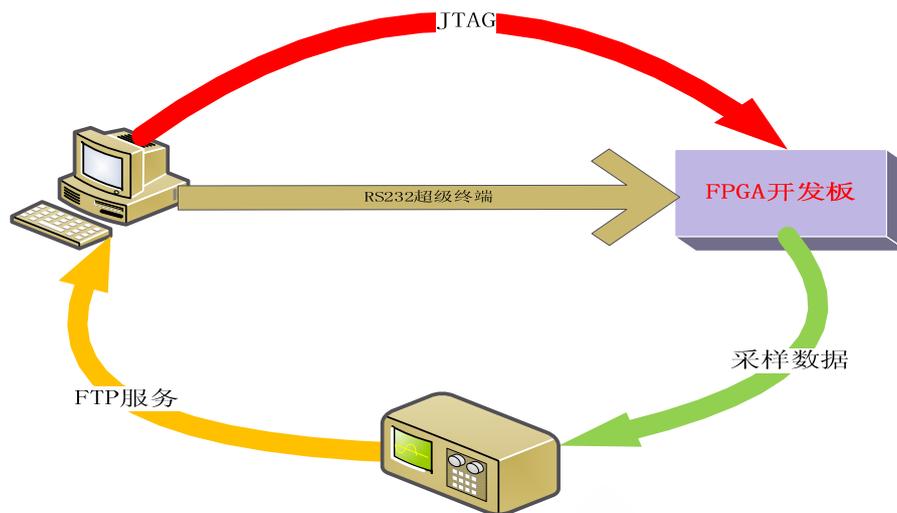


图 14 测试系统图

如图 14 所示开发板和 PC 机有 3 种通信手段，首先 PC 机通过 JTAG 接口把程序写入到 FPGA 开发板中，然后 FPGA 开发板就装载了我们设计的高速编码器模块。通过一个 RS232 接口，我们可以通过 PC 向串口写入控制命令控制开发板上的程序运行。当开发板上的程序开始运行以后，通过逻辑分析仪观测 FPGA 的输出数据，将输出数据进行保存，通过 FTP 将逻辑分析仪的数据转移到 PC 机上，然后将实测的采样数据和软件仿真的数据进行比较从而验证编码器的正确。在实际测试中对逻辑分析仪进行编程，通过编程控制逻辑分析仪存储需要的编码输出数据，并将输出数据通过逻辑分析仪的监听功能保存到逻辑分析仪的硬盘上。利用逻辑分析仪的网络接口，将保存的结果通过 NFS 文件系统将结果传输到计算机上，通过计算机处理测试数据，并且将测试数据和软件仿真的结果进行比较。

如图 15 所示是从逻辑分析仪上下载到计算机上的数据。输出数据共有两个：一个是 OVER 位宽是 1 位，另一个是 CHECK 位宽是 16 位，用十六进制数表示。数据部分的第一列表示的是逻辑分析仪内部的采样计数，由于逻辑分析仪内部的存储空间有限，所以每一次最多存储 8K 个内部计数的二进制数据。经过计算如果用 16 路并行输出数据，那么每一帧输出要 252 个时钟计数，加上一些码字间隔保护，最多输出 30 帧的数据。因此，设置信号源使之循环输出 30 帧的信号信息。信号源是由一个 53 阶的移位寄存器组成的，产生伪随机序列。将逻辑分析仪的输出信息和 MATLAB 运算的结果进行比较，可以验证高速运行条件下编码器的编码正确性。

MACHINE 1 - State Listing			
Label	>	OVER	CHECK
Base	>	Hex	Hex
0		1	8431
1		1	CA23
2		1	4CEC
3		1	68E5
4		1	E17D
5		1	0C03
6		1	7403
7		1	945C
8		0	945C

图 15 逻辑分析仪输出测试数据

---

测试结果表明,当输出时钟频率为 400MHZ,编码速率达到 1.6Gbps 时,编码器可以得到正确的编码结果。

## 5 结语

LDPC 码是一种性能优秀的信道编码,是下一代移动通信系统的备选方案。本文在 Altera 公司的 FPGA 开发平台上实现了编码速率高达 1.6Gbps 的 LDPC 编码器设计,并采用逻辑分析仪验证了设计的可靠性。本设计具有编码速率高,占用逻辑资源少的优点。高速的编码器必须要有高速的译码器与之相匹配,高速 LDPC 译码器的设计也是当今另一个热门的研究领域。

## 6 参考文献

- [1] R.G.Gallager, "Low-density parity-check codes", IEEE Transactions on Information Theory, Jan.1962, pp.21-28
- [2] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low-density parity-check codes," Electron. Lett., vol. 32, Aug. 1996, pp. 1645-1646
- [3] IEEE Std. 802 16e, "IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layer for Combined Fixed and Mobile Operation in Licensed Bands," Feb. 2006
- [4] Sharon, E, "Constructing LDPC codes by error minimization progressive edge growth", Communications, IEEE Transactions on, March 2008, pp.359-368.
- [5] Stratix II Device Handbook, [www.altera.com](http://www.altera.com)
- [6] EDA 先锋工作室, Altera FPGA/CPLD 设计(高级篇), 人民邮电出版社, 2005
- [7] ShuLin Danie J.Costello,Jr.著. 晏坚,何元智,潘亚汉等译. 差错控制编码 机械工业出版社,2007
- [8] 袁俊泉,孙敏琪,曹瑞著. Verilog HDL 数字系统设计及其应用. 西安电子科技大学出版社. 2003

### 论文原创性声明

本人郑重声明,该论文是本人在导师的指导下进行研究工作所取得的成果。据本人所知,论文中除了特别加以标注的地方外,不包含他人已经发表或撰写过的研究成果。本声明的一切法律结果由作者承担。

作者签名: 丁宏 杨帅  
日期: 2008 年 8 月 8 日