

基于 FPGA 的神经元相关性分析的设计与实现

刘培龙, 黄乐天, 林凌鹏

(电子科技大学 通信与信息工程学院, 四川 成都 611731)

摘 要: 利用 Verilog HDL 语言, Xilinx 的 ISE 平台实现了神经元相关性分析的设计. 首先对神经元相关性分析的理论 and 软件实现的方法进行了简单介绍, 然后对相关性分析的主要模块进行了设计, 最后用 ModelSim 进行了功能仿真和时序仿真, 用 ISE 做了逻辑综合与实现以及性能分析. 所选 FPGA 器件 xc5vlx220-2ff1760 逻辑资源消耗只占 7%, 最高时钟频率可以达到 240Mhz 左右. 只需要 48 个时钟周期就可以实现两个神经元之间相关性的计算, 也就是 200ns. 64 通道的情况下需要 0.4ms, 而用软件实现的方法至少需要几秒的时间, 这样可以对神经元之间的相关性进行实时性分析.

关键词: 相关性分析; 神经元; FPGA; Verilog HDL

中图分类号: TN4; R338

文献标识码: A

文章编号: 1000-7180(2012)05-0119-05

Design and Implementation of Neural Correlation Analysis Based on FPGA

LIU Pei-long, HUANG Le-tian, LIN Ling-peng

(School of Communication and Information Engineering of UESTC, Chengdu 611731, China)

Abstract: Implementing the design of neural correlation analysis using Verilog HDL and Xilinx ISE platform. Firstly, introduce the theory and software implementation method of neural correlation analysis. And then design the main module of the correlation analysis. Finally, do the functional simulation and timing simulation using ModelSim, synthesize and implement the design based on ISE platform. Only 7% logic resources are consumed of the FPGA device xc5vlx220-2ff1760. The maximum frequency can reach 240MHz. It only needs 48 clocks to implement the correlation analysis between two neurons that is 200ns. 64 channels need 4ms. However, if use the software method, it needs some seconds at least. So, it can realize the real time neural correlation analysis.

Key words: correlation analysis; neural; FPGA; Verilog HDL

1 引言

神经元之间的相关性包含大量有用信息, 分析神经元之间的相关性对研究神经系统的工作机制有重要的意义^[1-4].

目前主要是通过多电极阵列来采集脑神经系统的信息, 随着多电极阵列数目越来越大, 对采集的数据进行相关性网络的计算将是一个需要巨大计算量的工作. 所以很有必要设计一种相关性分析的硬件实现方法, 以提高计算速度. 用 FPGA 来实现神经

元相关性的分析, 在实时性、便携性以及低功耗方面将会有很好的表现.

2 神经元相关性分析软件实现

从多电极采集的数据是随时间变化的模拟信号, 经过处理可以得到锋电位序列 (spike trains). 然而, 通常对在某一时间点的单个锋电位不感兴趣, 而是对一个时间间隔内的锋电位数目也就是锋电位计数 (spike counts) 进行分析^[5]. 所以两个神经元之间的相关性在这里主要是指两个锋电位计数之间的相

收稿日期: 2011-09-20; 修回日期: 2011-10-25

基金项目: 国家自然科学基金项目(61006027)

关性. 锋电位计数序列 x, y 相关性计算公式如下:

$$\begin{aligned} \text{Corr}(x, y) &= \frac{\text{Cov}(x, y)}{\sigma_x \cdot \sigma_y} \\ &= \frac{E[(x_i - \bar{x})(y_i - \bar{y})]}{\sqrt{E(x_i - \bar{x})^2} \cdot \sqrt{E(y_i - \bar{y})^2}} \quad (1) \end{aligned}$$

这里 $\text{Cov}(x, y)$ 指的是序列 x, y 的协方差, σ_x, σ_y 分别是 x, y 的标准差, E 代表求均值. 可以对公式(1)化简

$$\text{Corr}(x, y) = \frac{\sum[(x_i - \bar{x})(y_i - \bar{y})]}{\sqrt{\sum(x_i - \bar{x})^2} \sqrt{\sum(y_i - \bar{y})^2}} \quad (2)$$

其中 $\sum$ 代表累加求和. 这样省去了三次除法运算, 在硬件实现时可以节省很多逻辑资源.

公式中涉及到乘法, 除法, 开方等运算, 都是比较占资源的, 尤其是开方运算. 在第 3 部分将对这些主要模块进行具体设计.

目前神经元相关性的分析大多是通过软件的方法实现的, 文献[2-4]中介绍了一些常用的实现方法. 其中文献[3]是通过一个 IPFM(Integral Pulse Frequency Modulation)模型来实现相关性分析的, 文献[4]阐述了一种定向的互相关分析的方法.

在这里主要介绍文献[2]所阐述的并行计算的方法, 后面硬件实现的性能也是主要和这个方法作比较. 一种能够节省相关性计算所需时间的方法就是利用并行计算的方法, 利用多台处理器联合起来对所需的计算同时执行. 这种并行计算的方法如图 1 所示, 共分为四个阶段, 数据划分, 任务提交, 任务执行以及结果收集.

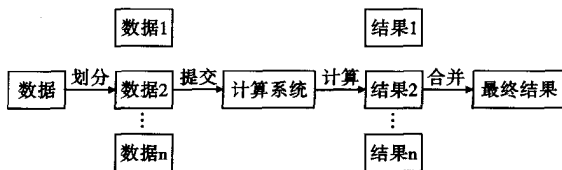


图 1 并行计算系统框图

第一个阶段根据用户的要求将所有数据划分成为特定的数据文件, 并且生成在这个系统上运行的任务提交文件. 第二个阶段是将任务文件提交给系统. 第三个阶段就是任务的执行, 完成相关性网络的计算. 最后一步就是收集和合并所得到的相关性网络的结果.

相关性计算所需时间如表 1 所示. 其中 C 是电极的数目, V 是所处理数据的个数. 当 $C=64, V=250\ 000$ 时, 计算时间是 7.71s, 和一般的方法相比计算时间已经得到了显著的减少. 但是还是在几秒的

数量级, 对于分析神经系统的实时性来说还有一定差距.

表 1 不同 C 和 V 下的相关性计算时间

C	V		
	250 000	500 000	1 000 000
64	7.71s	14s	30s
128	30s	59s	124s
256	122s	244s	498s
512	490s	988s	1978s

3 神经元相关性分析 FPGA 实现

按照 FPGA 自顶向下的基本设计流程. 自顶向下的设计框图如图 2 所示.

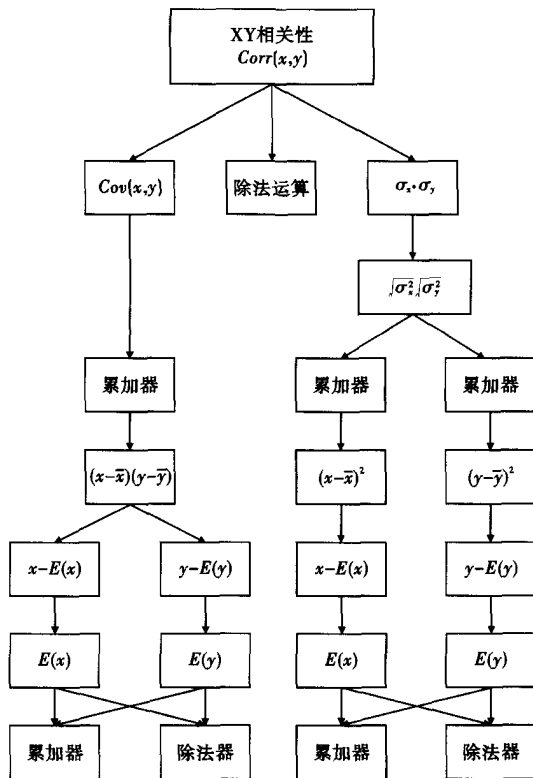


图 2 相关性计算自顶向下设计框图

由设计框图可以看出, 整个设计包含以下模块: 加减法模块、累加器模块、乘法器模块、除法器模块, 开平方模块等. 首先将通过 MATLAB 得到的锋电位计数序列存放在两个存储文件中, 然后通过程序调用这两个文件中的锋电位计数序列, 最终计算得到这两个序列之间的相关性. 对于不同神经元之间的相关性, 可以通过改变存储在这两个文件中的锋电位计数序列来实现. 下面对涉及到的三个主要模块, 乘法器模块、除法器模块和开方运算模块进行了

设计.

3.1 乘法器模块设计

本设计中所用到的锋电位计数序列是根据文献 [5],通过 MATLAB 仿真得到的. 某一时间间隔内

的锋电位数目完全可以由 8bit 二进制数来表示,所以采用 8bit 的有符号乘法器. 在这里使用一种 Booth 算法的乘法器^[6],对乘数进行比特对编码,或称为基 4 编码. 基 4 编码的编码规则如表 2 所示.

表 2 比特对(基 4)编码规则

m_{i+1}	m_i	m_{i-1}	Code	BRC_{i+1}	BRC_i	值	状况	效果
0	0	0	0	0	0	0	全 0 字串	移动两位
0	0	1	1	0	1	+1	末尾为 1 的字串	相加
0	1	0	2	0	1	+1	单个 1 的字串	相加
0	1	1	3	1	0	+2	末尾为 1 的字串	移一位,相加,移一位
1	0	0	4	<u>1</u>	0	-2	开始为 1 的字串	移一位,相减,移一位
1	0	1	5	0	<u>1</u>	-1	单个 0 的字串	相减
1	1	0	6	0	<u>1</u>	-1	开始为 1 的字串	相减
1	1	1	7	<u>0</u>	0	0	全 1 字串	移两位

表 2 中, $m_{i+1} m_i m_{i-1}$ 是乘数 2 补形式的连续三位, $BRC_{i+1}BRC_i$ 是 Booth 重编码位,根据 Booth 重编码位的值进行相应的操作,最终实现乘法运算. 在这里仅仅对乘数进行重新编码,而被乘数保持不变,之所以对乘数进行重新编码是为了减少乘法运算周期所需要的加法运算次数.

设计中将乘法器分为数据通道和控制器两部分,并且对这种乘法器进行了仿真和综合,仿真图如图 4(a)所示. 根据乘数和被乘数的不同所需要的时钟周期数也不同,大约都在 10 个时钟周期左右.

3.2 除法器模块设计

本设计中用到的除法器主要是实现求均值运算和最后协方差除以标准差的乘积. 由于除法器消耗逻辑资源比较多,所以有必要采用一种有效的方法来节省资源. 在这里采用一种精简寄存器时序除法器,它是基于在除法运算序列开始执行时,被除数寄存器的内容就要向最高位移动,同时为商的位留出存储空间. 这种结构在物理资源利用方面更为有效,因为它不再使用单独的寄存器保存商,结构如图 3 所示.

另外这种方法还有以下特性:

- (1) 移位和减法操作在同一时钟周期内进行;
- (2) 调整余数以对其在寄存器中的最后位置进行修正;
- (3) 使用溢出位检测不正确的结果.

图 3 中给出的框图是被除数为 8 位,除数为 4 位的情况,被除数与商共用一个寄存器. 该寄存器最左端包括一个扩展位用来处理被除数和除数所需的初始移位,并用来保存从被除数中减去除数所形成

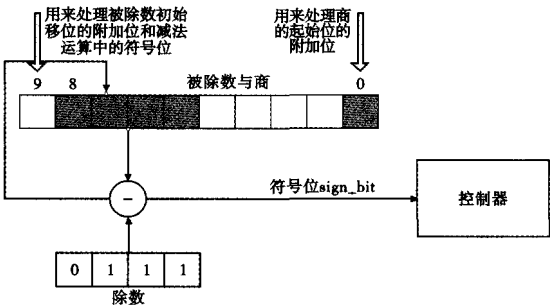


图 3 精简寄存器除法器结构图

的符号位. 另外,该寄存器最右端也需扩展一位以保存求商所形成的第一位.

对于 193 除以 2 的情况进行了仿真,如图 4(b)所示,只需要 8 个时钟周期就能求出商和余数,如果按照普通的除法则需要 96 个时钟周期. 利用 ISE 综合后,只需要 229 个四输入查找表即可实现除法运算.

3.3 开方运算模块设计

本文采用由 Li Yamin 和 Chu Wanming 提出的一种改进的不恢复余数的开方算法^[7]. 该开方算法主要创新点在于对部分余数的使用上,而且该算法能准确计算出平方根的最后一位,每次迭代只需做加减和移位操作,每次迭代产生的部分余数将用于下次迭代.

本设计中被开方数为 16 位,采用流水线的结构来实现. 设 D 为被开方数, $Q_k = q_7 q_6 \cdots q_k$ (总共 $8-k$ 位)为第 k 次迭代产生的平方根, $R_k = r_8 r_7 \cdots r_k$ (总共 $9-k$ 位)为第 k 次运算时开始的部分余数,具体步骤如下:

表 3 性能分析

Device	xc5vlx220-2ff1760
Number of Slice LUTs	9729/138240(7%)
Maximum Frequency	240.044MHz
Number of IOs	24/800(3%)
Total Power	2.596W

对于设计的性能分析主要关注逻辑资源消耗情况和最高时钟频率. 通过表 3 可以看出,所消耗的逻辑资源只占 7%. 最高时钟频率可以达到 240.044MHz,也就是说一个时钟周期是 4.166ns,由前面的仿真结果可知,完成一次相关性运算共需要 48 个时钟周期,也就是 200ns 的时间. 64 通道的情况下计算两两之间的相关性需要 0.4ms,而前面提到的并行计算的方法至少需要 7.71s. 可见用 FPGA 实现的方法可以很好的达到实时性的要求.

5 结束语

为了改进目前主要用软件实现相关性计算比较费时的问题,文中提出了一种基于 FPGA 的神经元相关性分析的方法. 通过 Verilog HDL 编程,用 ISE 10.1 对设计进行了综合和性能分析,计算两个神经元之间相关性的最高时钟频率可以达到 240MHz,只需要 200ns 的时间. 但是,还存在一些需要改进的地方,比如说采用定点计算精度不高的问题,还有如果扩展到将来 1000 个通道的情况,各个神经元之间通信的问题等,这些都是很值得去研究的问题.

参考文献:

[1] Michael Krumin, Inna Reutsky, Shy Shoham. Correlation-based analysis and generation of multiple spike trains using Hawkes models with an exogenous input [J]. Frontiers in Computational Neuroscience, 2010 (4): 147.

[2] Pedro Ribeiro, Jennifer Simonotto, Marcus Kaiser, et

al. Parallel calculation of multi-electrode array correlation networks[J]. Journal of Neuroscience Methods, 2009(184): 357-364.

[3] Gaetano Giunta, Alessandro Neri. Neural Correlation Based on the IPFM Model[J]. Transactions on Systems, Man, And Cybernetics, 1990(20): 262-268.

[4] Hidekazu Kaneko, Shinya S Suzuki, Jiro Okada, et al. Directional cross-correlation analysis of neuronal burst firing[C]// Proceedings of the 19th International Conference-IEEE/EMBS. Chicago USA: IEEE, 1997: 1425-1427.

[5] Jakob H Macke, Philipp Berens, Alexander S Ecker, et al. Generating spike trains with specified correlation coefficients[J]. Neural Computation, 2009(21): 397-423.

[6] 吴明森,李华旺,刘海涛. 一种 16x16 位高速低功耗流水线乘法器的设计[J]. 微电子学与计算机,2003(8): 151-153.

[7] Yamin Li, Wanming Chu. A new non-restoring square root algorithm and its VLSI implementations[C]// International Conference on Computer Design. Austin, Texas, USA: IEEE, 1996:538-544.

[8] 胡学良,张春,王志华. 开方运算单元的高层次综合设计[J]. 微电子学与计算机,2005,22(8):36-38.

作者简介:



刘培龙 男,(1987—),硕士研究生. 研究方向为神经网络、现代通信中的信号处理.

黄乐天 男,(1984—),博士研究生. 研究方向为神经网络、现代通信中的信号处理、通信专用集成电路设计.

林凌鹏 男,(1987—),硕士研究生. 研究方向为神经网络、现代通信中的信号处理.