

Aplicaciones sobre Plataformas de Redes Neuronales en Tiempo Real

Tosini, Marcelo - Acosta Nelson

INTIA/INCA - Fac. de Ciencias Exactas - Universidad Nacional del Centro de la Prov. de Bs. As.

Paraje Arroyo Seco s/n – Tandil - Argentina

{mtosini, nacosta}@exa.unicen.edu.ar

Resumen

El presente trabajo pretende el diseño de una metodología para la construcción de aplicaciones basadas en redes neuronales sobre una plataforma Muren. Las aplicaciones se restringen a sistemas de control y reconocimiento de patrones por imágenes. Se describe la arquitectura del sistema de desarrollo Muren, basado en 2 procesadores ZISC de 78 neuronas cada uno, una FPGA Spartan II, bancos de memoria y lógica adicional de comunicación.

Palabras clave

Redes neuronales, Tiempo real, Procesador dedicado, FPGA.

1. Introducción

Muchas de las aplicaciones de control industrial presentan un comportamiento no lineal y sus modelos correspondientes habitualmente se hallan repletos de incertidumbre. Esto dificulta la materialización de controladores adecuados para tales aplicaciones. Con el propósito de solucionar los problemas en el diseño de controladores de sistemas no lineales, en las últimas dos décadas se han propuesto una buena cantidad de enfoques satisfactorios basados principalmente en lógica difusa o en redes neuronales.

Todas estas aplicaciones cuya solución se realiza aplicando sistemas no basados en modelos requieren de controladores cada vez más veloces. Para satisfacer esta necesidad de reducir los tiempos de respuesta se trabaja con dos grandes enfoques:

- a. Optimización de los algoritmos. Se busca reducir el tiempo de cálculo de cada una de las etapas [2][4][8].
- b. Mejoramiento de las plataformas de cálculo. Trabajos de investigación y diversas materializaciones se agrupan en aplicaciones basadas en microcontroladores dedicados [1][3], software sobre plataformas genéricas [10] y circuitos a medida de la aplicación [5].

Los controladores basados en redes neuronales pueden ser materializados por software sobre hardware de propósito general, sobre microcontroladores dedicados [7] o mediante un circuito específico (ASIC¹, FPGA², PLD³) [6]. Para cualquier forma de materialización es conveniente realizar una optimización del sistema que involucre operaciones a medida de la aplicación, tamaños de datos adecuados, módulos de comunicación rápidos, etc.

El requerimiento de velocidad para el cálculo en un controlador exige la migración de software a hardware, por ejemplo mediante técnicas de codiseño hardware/software. En un hardware a medida se pueden aprovechar características propias de la aplicación (paralelismo, tamaños de rutas de datos y operadores de cálculo a medida de los operandos) y técnicas de altas prestaciones.

El objetivo global de éste proyecto es el desarrollo de una metodología y sus herramientas para agilizar el diseño de aplicaciones automáticamente. La metodología incluye el diseño

¹ Application Specific Integrated Circuit

² Field Programmable Gate Array

³ Programmable Logic Device

de las máquinas de estado que permitan comunicar el módulo de control con los módulos que realizan el acceso a los distintos bloques (componentes) de la placa Muren (cámara, los dos ZISC, los bancos de RAM, y comunicaciones RS232, paralela, USB). La sección 2 presenta la arquitectura de la placa utilizada como procesador acoplado. La arquitectura del ZISC es descrita en la sección 3, mientras que la metodología se propone en la sección 4. La sección 5 y 6 muestran los alcances del proyecto y las conclusiones.

2. Arquitectura Muren

Muren_EMB [11] es un sistema embebido diseñado para manejar señales digitales de entrada, reconocer patrones y transmitir resultados de salida sin necesidad de modelado de datos ni intervención de una PC.

La placa (figura 1) contiene: a) soporte para comunicación a través de puertos serie y paralelo, b) una red neuronal digital para reconocimiento de patrones de alta velocidad (2 procesadores ZISC78), c) un banco de memoria y d) una FPGA (Spartan II de Xilinx [9]) que puede ser programada para realizar operaciones específicas como adquisición y procesamiento de datos en tiempo real, extracción de características y comunicación con otros dispositivos periféricos.

Los resultados de salida obtenidos pueden ser: posición de determinados objetos, sus categorías, pertenencia a contextos y otras. Otra característica importante de Muren es su capacidad de reconfiguración. En cualquier momento puede cargarse un nuevo programa en la FPGA y un nuevo motor de reconocimiento en los ZISC.

Además, la placa está provista de una cámara digital para captura de imágenes en tiempo real, lo que permite la realización de sistemas de análisis de imágenes inteligentes como reconocimiento de rostros, huellas digitales, seguimiento de blancos y otros.

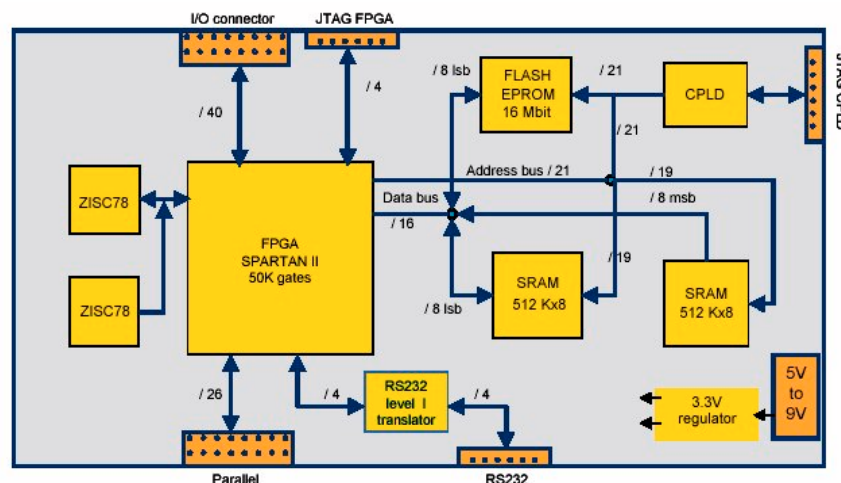
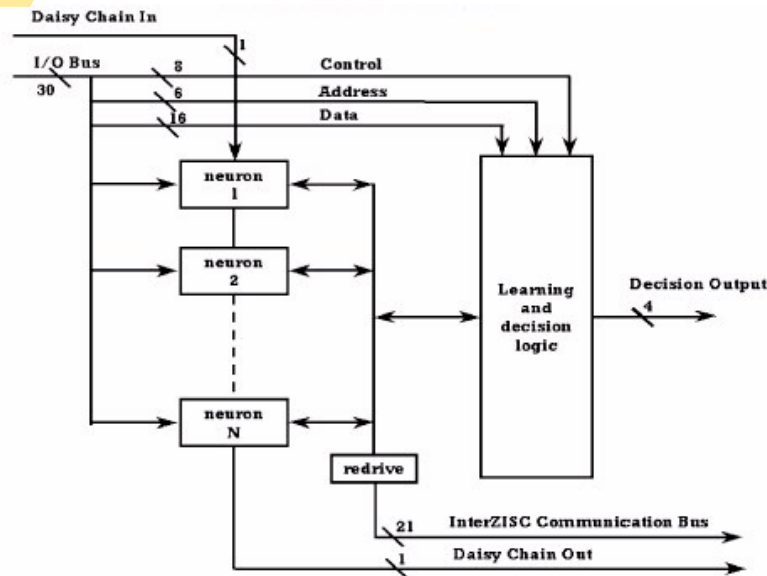


Figura 1: Diagrama de componentes del sistema Muren_Emb

3. Tecnología ZISC

ZISC (Zero Instructions Set Computer) [12] es una unidad de proceso en paralelo basada en redes neuronales y dotada de potencia informática ultra rápida para entornos de trabajo que requieren clasificar o reconocer patrones de información (figura 2). Su nombre se debe a que no es necesaria una programación meticulosa con instrucciones específicas para llevar a cabo una tarea determinada. ZISC sólo requiere que se alimente cada neurona con ejemplos representativos y que se los clasifique de acuerdo al trabajo.

Cada procesador ZISC contiene 78 neuronas independientes que pueden pensarse como neuronas de procesadores paralelos. Cada una de estas células trabaja utilizando dos técnicas de representación espacial, Radial Basis Function (RBF) y K-Nearest Neighbor (KNN), a partir de las cuales compara un vector de entrada de 64 bytes con un vector similar almacenado en la memoria de la célula. Si el vector de entrada es semejante al vector almacenado en la memoria de la célula se dispara una señal, de lo contrario no. Al trabajar en forma paralela, el chip ZISC le indica a sus 78 células comparar su memoria con el vector de entrada al mismo tiempo. Como resultado, la velocidad del sistema se incrementa drásticamente.



Gentileza Silicon Recognition Inc

Figura 2: Diagrama de bloques del chip ZISC78

Los chips ZISC pueden usarse también para encontrar correspondencias aproximadas. En lugar de preguntar por la correspondencia exacta, se puede preguntar por la más parecida. Por lo tanto, las células que están por sobre un cierto umbral se disparan simultáneamente y el controlador en el chip se fija cual devuelve el valor más cercano.

La tecnología ZISC es capaz de realizar una evaluación detallada en menos de cuatro microsegundos, aspecto que la convierte en ideal para aplicaciones en tiempo real. Sin embargo, la potencia real de este chip está en su escalabilidad. Una red ZISC puede expandirse agregando más dispositivos ZISC, sin sufrir una merma en la velocidad de reconocimiento. No hay un límite teórico para el número de células que pueden formar una red neuronal, existiendo hoy día productos comerciales provistos por Silicon Recognition con 10.000 células o más.

Hay un sinnúmero de aplicaciones que son particularmente apropiadas para el ZISC. Entre ellas se encuentran el reconocimiento de rostros, el control en tiempo real, telecomunicaciones, automoción, industria del juguete, automatizaciones industriales, bioinformática, etc. En los casos de defensa, seguridad, control de acceso y vigilancia inteligente existen dispositivos y desarrollos para: Reconocimiento de personas, vehículos y otros objetos. Identificación y seguimiento de objetivos en ambientes complejos (por ejemplo, seguimiento de objetivos en misiles). Rastreo de personas o vehículos, Identificación de huellas digitales o iris oculares. Identificación y procesamiento de sonidos, entre otros.

4. Ciclo de Diseño de Aplicaciones Autónomas

Debido a la conexión entre los módulos y la ubicación central de la FPGA, ésta es la encargada de implementar la interfaz entre todos ellos y además realizar su sincronización. Cabe destacar que los módulos provistos por Silicon Recognition (en VHDL) son la interfaz entre la FPGA y cada componente. El código VHDL de interfaz permite realizar ejemplos para aplicaciones que usen la Muren como co-procesador de una computadora.

Para una aplicación autónoma sobre la Muren se requiere migrar el programa principal de la aplicación desde la PC hasta la FPGA; tarea no imposible, mas sí muy costosa. Se propone la descomposición en una máquina de estados del programa, para así realizar el control de todas las tareas desde la FPGA.

La metodología incluye la siguiente secuencia (con las retroalimentaciones necesarias):

- a) Definición del problema.
- b) Diseño de una posible solución software.
- c) Implementación de la misma en el simulador Muren. Esta herramienta provee un sistema de simulación de la placa, que permite construir la aplicación software.
- d) Entrenamiento de la red neuronal en el simulador Muren. Se entrenan las redes neuronales utilizando el simulador para poner a punto la aplicación.
- e) Grabación del entrenamiento de la red neuronal en el simulador Muren. Se baja a un archivo el entrenamiento de las redes neuronales para luego ser cargada en los procesadores ZISC de la Muren.
- f) Diseño de la posible solución hardware basada en la solución software. Se realiza un rediseño de la aplicación para permitir su migración a la FPGA de la Muren.
- g) Implementación de la solución sobre la Muren. La solución debe ser implementada utilizando VHDL en el nivel de síntesis por medio de RTL (transferencia de registros).
- h) Configuración de la solución hardware en la FPGA.
- i) Carga del entrenamiento de la red neuronal en los procesadores ZISC.
- j) Prueba del sistema completo y de forma autónoma.

En esta propuesta hay muchos problemas a resolver. Entre ellos se destacan la necesidad de herramientas para el diseño de la red neuronal y la traducción del software a la máquina de estados hardware en VHDL-RTL. En el mercado existen herramientas que asisten al diseñador de sistemas para la puesta a punto de la red neuronal (por ejemplo: NN toolkit para Matlab, NNGen, Neural Connections, entre otros). El problema de la traducción de la aplicación software al hardware que la implemente no tiene herramientas de ayuda.

Esta traducción debe ser realizada de forma manual; donde cada uno de las instrucciones del lenguaje debe ser representada con un bloque VHDL que realice el cálculo, mientras que todos estos bloques son sincronizados por medio de una máquina de estados general.

5. Aplicaciones enfocadas

Las aplicaciones finales de ésta tecnología son:

- Un sistema de control para una cámara giro-estabilizada que tiene como función calcular las aceleraciones que sufre la cámara y realizar los movimientos para mantener su posición de enfoque. Se tiene en cuenta tanto movimientos del vehículo sobre el cual está colocada como las vibraciones. La aplicación requiere las funciones para cada eje del sistema, lectura del acelerómetro (incluida la digitalización de la señal), proceso de las señales de entrada (dependiendo del estado actual y el valor de la lectura), cálculo de la

función de respuesta y próximo estado, y envío de las señales de salida a los actuadores de estabilización.

- Un sistema de análisis de imágenes para el reconocimiento de patrones. Requiere la lectura del CCD, su almacenamiento en una memoria, el recorrido de la imagen completa para armar la sección a analizar, cálculo del vector de representación de la imagen, envío de ese vector al ZISC, y filtro de las categorías más cercanas.

6. Conclusiones

La metodología descrita en este artículo permite (permitirá) generar controladores basados en redes neuronales de altas prestaciones en tiempos bastante cortos. Estos diseños se limitan a redes neuronales del tipo K nearest neighbor (KNN) y Radial Basis Function (RBF). Los resultados obtenidos son muy preliminares, debido al estado actual (inicial) del proyecto. De cualquier forma, se puede asegurar que ya se han encontrado los problemas principales a resolver.

Actualmente se está trabajando en la generación automática de circuitos que materialicen los diseños de las máquinas de estados sobre plataformas FPGA utilizando VHDL como lenguaje de síntesis.

Referencias

- [1] Acosta, G. y Tosini, M., “*A Firmware Digital Neural Network for Climate Prediction Applications*”, 2001 CCA/ISIC conference, IEEE Conference on Control Applications and IEEE International Symposium on Intelligent Control, Mexico D.F., Mexico, 2001.
- [2] Beiu, V., “*How to build VLSI-efficient neural chips*”, Proceeding of the International ICSC Symposium on Engineering of Intelligent Systems, EIS’98, Vol 2, 66-75, 1998.
- [3] Burr, J., “*Digital neural networks implementations*”, in Neural Networks, Concepts, Applications, and Implementations, Vol III. Englewood Cliffs, New Jersey: Prentice Hall, 1991.
- [4] Gadea Gironés, Rafael, “*Implementación mediante FPGA de un perceptrón multicapa entrenado con el algoritmo de backpropagation aplicando técnicas de computación paralela, pipeline y arquitecturas sistólicas*”, Tesis doctoral, Universidad Politécnica de Valencia, 1999.
- [5] Guccione, S. & González, Mario, “*A neural network implementation using reconfigurable architectures*”, in: W. Moore, W. Luk (eds.), More FPGAs, Abingdon, Oxford, UK, 443-451, 1994.
- [6] Lee, J., Sheu, B., Fang, W. & Chellappa, R., “*VLSI neuroprocessors for video motion detection*”, IEEE Transactions on Neural Networks, Vol 4, Nro 2, 1993.
- [7] Schönauer, T., Jahnke, A., Roth, U. & Klar, H., “*Digital Neurohardware: Principles and Perspectives*”, Proc. Neuronale Netze in der Anwendung (NN’98), Magdeburg, invited paper, 1998.
- [8] Tosini, M., “*Implementación en FPGA de una arquitectura programable para redes neuronales digitales*”, VI Congreso Iberchip IWS’2000, Sao Paulo, Brasil, 169-180, 2000.
- [9] <http://www.xilinx.com/partinfo/databook.htm>
- [10] Rubio, M., Pérez, M., “*Técnicas de redes neuronales aplicadas al reconocimiento y clasificación de píxeles de imágenes de satélite*”, IV Simposio Nacional de Predicción, 1996.
- [11] “*MUREN: High-Speed Pattern Recognition Embedded Board Using a ZISC Neural Network*”, Manual revision 3.3, Silicon Recognition, Inc.
- [12] “*ZISC: Zero Instruction Set Computer*”, Version 4.2, Silicon Recognition, Inc., 2002