Journal of Southeast University (English Edition) Vol. 28, No. 3, pp. 282 - 286 Sept. 2012 ISSN 1003—7985

FPGA implementation of bit-stream neuron and perceptron based on sigma delta modulation

Liang Yong Wang Zhigong Meng Qiao Guo Xiaodan

(Institute of RF- & OE-ICs, Southeast University, Nanjing 210096, China)

Abstract: To solve the excessive huge scale problem of the traditional multi-bit digital artificial neural network (ANN) hardware implementation methods, a bit-stream ANN hardware implementation method based on sigma delta ($\Sigma\Delta$) modulation is presented. The bit-stream adder, multiplier, threshold function unit and fully digital $\Sigma\Delta$ modulator are implemented in a field programmable gate array (FPGA), and these bit-stream arithmetical units are employed to build the bit-stream artificial neuron. The function of the bit-stream artificial neuron is verified through the realization of the logic function and a linear classifier. The bit-stream perceptron based on the bit-stream artificial neuron with the pre-processed structure is proved to have the ability of nonlinear classification. The FPGA resource utilization of the bit-stream artificial neuron shows that the bit-stream ANN hardware implementation method can significantly reduce the demand of the ANN hardware resources.

Key words: bit-stream; artificial neuron; perceptron; sigma delta; field programmable gate array (FPGA)

doi: 10.3969/j.issn.1003-7985.2012.03.005

The hardware implementation of the ANN is an important issue in ANN research. However, the traditional multi-bit digital signal processing (DSP) approach requires too many resources to implement the massive parallel calculation of the neural network and thus makes the chip unaffordable[1-2]. With the multi-bit DSP method, all the transmissions go through the multi-bit buses, and therefore, the mutual connections between the artificial neurons and the DSP units inside the artificial neuron are vast. For example, even in a simple 8-bit 3-5-3 structure ANN, there are 240 data lines between the neurons and about 2 000 data lines inside the 11 neurons. The huge scale brought by the vast connections and complicated multi-bit DSP units obstructs the hardware implementation of a general ANN. Normally, the hardware implemented ANN reported in the literature only has several neurons[3].

In recent years, a bit-stream DSP method has been proposed[4-10]. With this method, only one line is needed for one signal, and the bit-stream DSP units are very simple. Therefore, the problems of the connections and DSP units scale in the ANN are easy to solve.

According to this method, the bit-stream digital signals which come from a $\Sigma\Delta$ modulator[4-10] are processed. The $\Sigma\Delta$ modulator is a section of the $\Sigma\Delta$ analog-to-digital converter (ADC), which has become quite popular for achieving high resolution. The $\Sigma\Delta$ modulator converts the analog signal to a bit-stream digital signal, which is only an inner signal in a $\Sigma\Delta$ ADC. However, the bit-stream signal is directly processed in the new DSP method now.

A novel DSP method is introduced into an artificial neuron implementation in this paper. Four fundamental bit-stream modules and a kind of bit-stream artificial neuron are implemented in an FPGA and the function of bit-stream artificial neuron is verified. The total number of the data lines in a bit-stream 3-5-3 ANN is only about 300.

1 Bit-Stream Artificial Neuron Structure

Artificial neurons are the cells of the ANN. A bit-stream artificial neuron consists of a few bit-stream adders, multipliers, an activation function unit, and some parameters generators or memorizers. The biases and weights can be generated by the $\Sigma\Delta$ modulator or stored in the shift registers.

A three-input bit-stream artificial neuron can be realized according to the structure shown in Fig. 1. Each bit-stream adder can only process two input signals. So, the summation of the multi-input signals is operated by multi-stage bit-stream adders. All the inputs, outputs and inner signals of the artificial neuron are $\Sigma\Delta$ modulated signals.

When the bit-stream artificial neuron is implemented, the simplest neural network, the perceptron which is a linear classifier, can also be implemented. To implement the bit-stream artificial neuron, the design and implementation of the bit-stream adder, multiplier, activation function unit and digital $\Sigma\Delta$ modulator are necessary.

Received 2011-10-25.

Biographies: Liang Yong (1971—), male, doctor, associate professor; Wang Zhigong (corresponding author), male, doctor, professor, zgwang@seu.edu.cn.

Foundation items: The National Natural Science Foundation of China (No. 60576028), the Natural Science Foundation of Higher Education Institutions of Jiangsu Province (No. 11KJB510004).

Citation: Liang Yong, Wang Zhigong, Meng Qiao, et al. FPGA implementation of bit-stream neuron and perceptron based on sigma delta modulation[J]. Journal of Southeast University (English Edition), 2012, 28(3): 282 - 286. [doi: 10.3969/j.issn.1003-7985.2012.03.005]

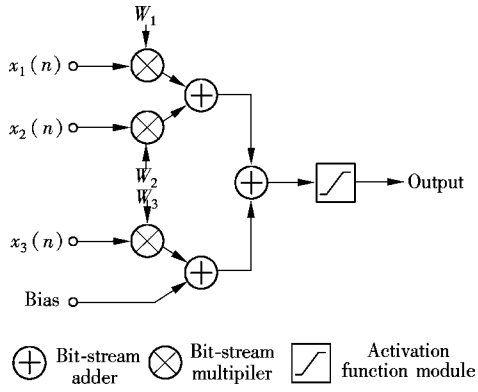

Fig. 1 Structure of bit-stream artificial neuron

2 Bit-Stream Modules Used in Bit-Stream Artificial Neurons

2.1 Bit-stream adder

There are four kinds of bit-stream adder circuits reported in the literature[8–10]. The first circuit is to perform addition through interleaving. This circuit wastes half input information, which leads to a very bad signal-to-noise ratio (SNR) of the output signal[9]. Both the second and the third circuits have a big loop structure inside, which limits the operation frequency[8–10]. The structure of the bit-stream adder used in this paper is shown in Fig. 2[9].

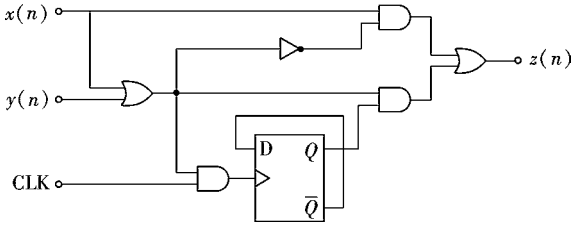

Fig. 2 Circuit of the bit-stream adder

In the circuit, when $x(n) \neq y(n)$, the CLK signal is connected to the D-type flip-flop (DFF) and the DFF generates the desired signal, a sequence of alternant 0 and 1. When $x(n) = y(n)$, $x(n)$ is directly transferred to the output terminal. All the information is used and the big loop structure is avoided in this circuit.

Compared with the other bit-stream adder circuits, the selected circuit can achieve much better SNR performance or the same SNR performance with several times operation frequency and about 20% hardware saving[9].

2.2 Bit-stream multiplier

The original structure of the bit-stream multiplier is given in Ref. [10]. Fig. 3 shows the equivalent circuit of the bit-stream multiplier. Exclusive-OR gates are applied to calculate the sub-product $x(i)y(j)$, and the bit-stream adder is applied to obtain the summation of $x(i)y(j)$.

This bit-stream multiplier has a problem. It always provides the reverse value of the correct product. To solve this problem, all the exclusive-OR gates in the bit-stream multiplier are replaced with exclusive-NOR gates. At the same time, the original bit-stream adders are replaced with the bit-stream adders given in Ref. [9] to increase the operation frequency and reduce the source utilization.

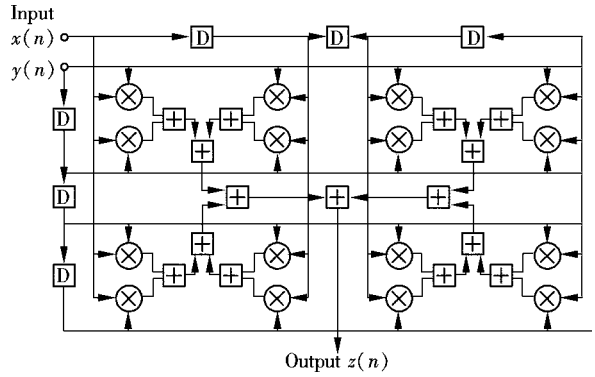

Fig. 3 Equivalent circuit of the bit-stream multiplier

The bit-stream multiplier implemented in an FPGA is proved by the measured power spectrum density (PSD). The PSD has two peak signal frequencies, which are the summation and the difference of the input signal frequencies, $f_y + f_x$ and $f_y - f_x$.

2.3 Bit-stream activation function unit

Several kinds of activation functions can be used in the artificial neurons. They are the threshold function, the sigmoid function, and so on. Among all the activation functions, the threshold function is the simplest and the most basic one. It is difficult to implement some activation functions with the bit-stream method now. So, the threshold function is employed to explore the implementation of the bit-stream artificial neuron first.

To realize the threshold function, an N-bit shift register is used as a moving window. The output signals of the last bit-stream adder in the artificial neuron are sent to the N-bit shift register. All the data staying in the shift register will be sent to an accumulator. The value of the accumulator denotes that the short term signal is positive or negative. The comparator gives the output according to the accumulated value. So, the threshold function unit is realized.

In each clock period, among all the data stored in the shift registers, only $x(i)$ is replaced with $x(i+N)$. To simplify the circuit, the accumulator is calculated by the following formula:

$$Z(n) = \begin{cases} Z(n-1) & x(i) = x(i+N) \\ Z(n-1)+1 & x(i) = 0, x(i+N) = 1 \\ Z(n-1)-1 & x(i) = 1, x(\mathrm{i}+N) = 0 \end{cases} \quad (1)$$

It means that only two input signals are needed for the accumulator. Fig. 4 shows the diagram of the bit-stream threshold function unit.

The simulation waveforms of the bit-stream threshold

function unit are shown in Fig. 5.

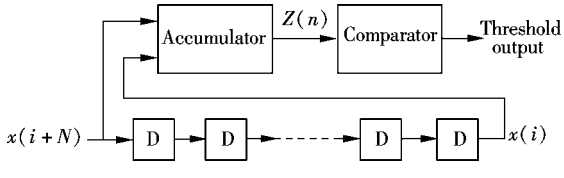

Fig. 4 Bit-stream threshold function unit

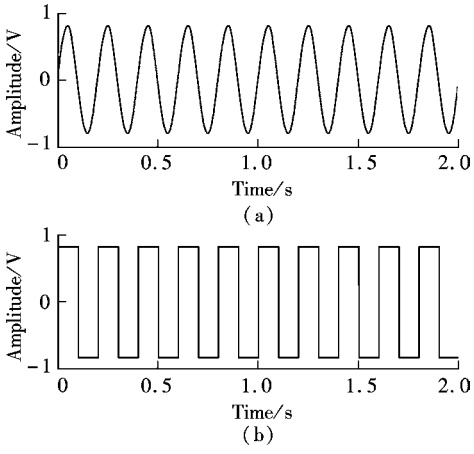

Fig. 5 Waveform of the threshold function. (a) Input signal; (b) Output signal

2.4 Digital ΣΔ modulator and signal generator

The ΣΔ modulators are needed to provide the biases and the weights in the bit-stream signal style. The existing ΣΔ modulator is an analog and digital mixed-signal module. It has analog sections inside and the input signal of the ΣΔ modulator is also an analog signal.

To realize the bit-stream artificial neuron in an FPGA, a new type of fully digital ΣΔ modulator is designed to generate the desired biases and weights. In this digital ΣΔ modulator, the analog input signal is replaced by a multi-bit digital signal, and the analog adder, the integrator, and the comparator are replaced by multi-bit digital ones. At the same time, the digital signal is denoted as a complementary code to convert the subtracting operation to the addition one.

A direct digital synthesizer (DDS) curve generator is designed in an FPGA. The DDS curve generator is combined with the digital ΣΔ modulator to build a digital ΣΔ signal generator. The ΣΔ signal generator can provide arbitrary waveforms to verify the function of the bit-stream artificial neuron or generate the biases and weights.

Fig. 6 shows the block diagram of the ΣΔ signal generator in an FPGA. The upper one is a DDS curve generator, and the lower one is a one-order digital ΣΔ modulator. The DDS curve generator provides a multi-bit digital waveform, while the digital ΣΔ modulator converts the multi-bit digital wave signal to the bit-stream signal.

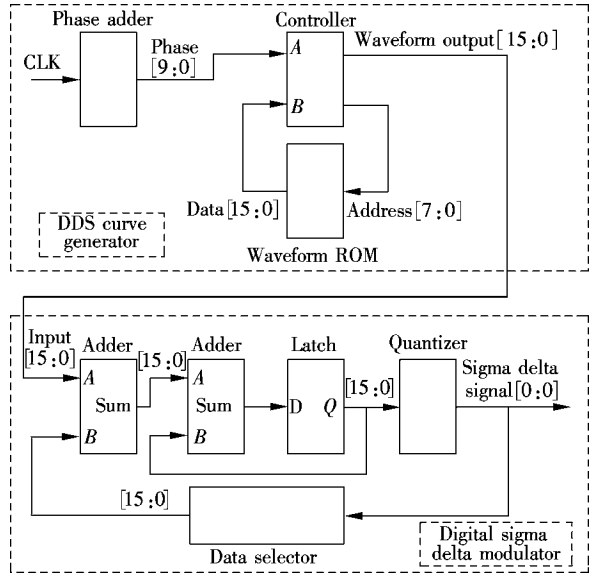

Fig. 6 Block diagram of the digital ΣΔ modulator

Fig. 7 shows the frequency spectrum of the ΣΔ signal generator output signal with a sine wave input signal. The spectrum has a single frequency component and a distinct characteristic of noise shaping. The data of the output signal is obtained from the actual running FPGA through the online logic analyzer, Chipscope. The fast Fourier transform (FFT) analysis of the data is carried out in Matlab.

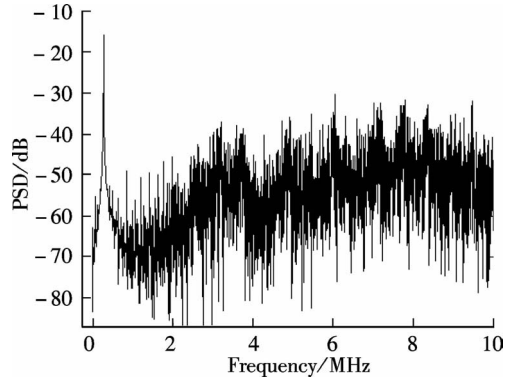

Fig. 7 Spectrum of the output signal

3 Bit-Stream Artificial Neuron Implementation and Verification

Based on the bit-stream modules mentioned above, the bit-stream artificial neuron is implemented in an FPGA. The target device is XC3S200. Tab. 1 shows the source utilization of one bit-stream artificial neuron. It indicates that one XC3S200 can afford about 25 bit-stream artificial neurons. If we use a higher grade FPGA, such as Virtex-6 family, we can obtain more than 1 000 neurons in one FPGA. Therefore, a relative complicated ANN is possible to realize.

Tab. 1 Source utilization of the bit-stream artificial neuron

Logic utilization	Used	Available	Utilization/%
Number of slices	87	1 920	4
Number of slice flip flops	148	3 840	3
Number of 4 input LUTs	98	3 840	2

To verify the function of the bit-stream artificial neuron, the neuron is configured to realize some logic functions. The training process is executed in Matlab according to the Widrow-Hoff algorithm.

Fig. 8 shows the simulated waveform of the artificial neuron when it performs the "NAND" logic with the Modelsim simulator.

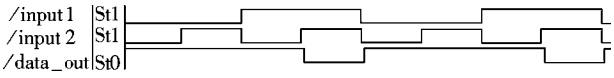

Fig. 8 Simulation waveform of the "NAND" logic

The second case is a bit-stream neural network which is built with two bit-stream artificial neurons. After training, it is proved that the simple bit-stream neural network can classify four kinds of points.

The normal perceptron consisting of the artificial neurons is only a linear classifier. To implement the nonlinear classification function, the input signals of the perceptron are pre-processed according to the following formula:

$$y = w_1 x_1 + w_{11} x_1^2 + w_{12} x_1 x_2 + w_{22} x_2^2 + w_2 x_2 + b \quad (2)$$

The structure of the bit-stream nonlinear classification perceptron is shown in Fig. 9. Since the bit-stream adders always output a half of the summation, the parameters w_2 and b are divided by 2 to maintain the balance within different signals.

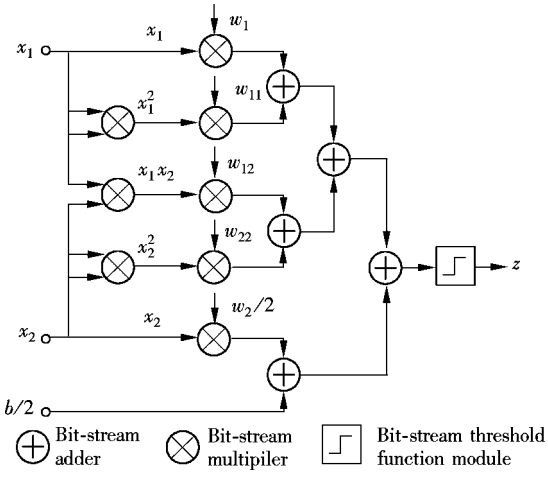

Fig. 9 Bit-stream nonlinear classification perceptron

The model of the bit-stream nonlinear classification perceptron is designed with Simulink and the model simulation is performed in Matlab. When the weights and bias take certain parameters, such as $w_1 = w_2 = -0.05$, $w_{11} = w_{22} = 0.6$, $w_{12} = -0.55$ and $b = -0.9$, all the area is divided into two parts by an ellipse curve, as shown in Fig. 10. It means that the bit-stream perceptron with the pre-processed structure has the ability of nonlinear classification. Actually, the two objects classification shown in Fig. 10 is the result of "exclusive-OR" logic, which is a typical nonlinear classification case.

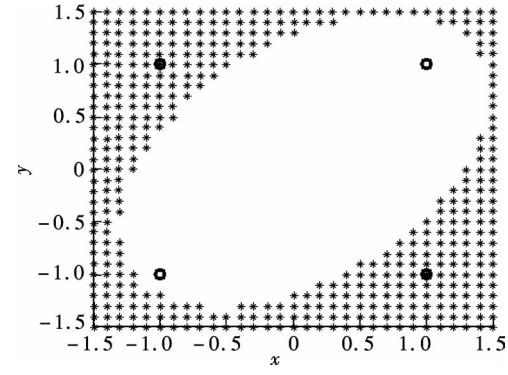

Fig. 10 Nonlinear classification

4 Conclusion

Four fundamental bit-stream modules and one kind of bit-stream artificial neuron are implemented in an FPGA. The problems of the connections and DSP units scale in the multi-bit ANN hardware implementation can be easily solved with the bit-stream DSP method. The function of the bit-stream artificial neuron is verified through the realization of logic functions, linear and nonlinear classifiers.

In further research, the bit-stream ANN will be constructed with the bit-stream artificial neuron to realize complicated functions, such as pattern discrimination. Another bit-stream activation function unit will be explored to build more kinds of artificial neurons and ANNs.

References

[1] Zhang D, Li H, Foo S Y. A simplified FPGA implementation of neural network algorithms integrated with stochastic theory for power electronics applications [C]// 31*st Annual Conference of the IEEE Industrial Electronics Society*. Raleigh, USA, 2005: 1018 - 1023.

[2] Hu H, Huang J, Xing J G, et al. Key issues of FPGA implementation of neural networks [C]//*Second International Symposium on Intelligent Information Technology Application*. Shanghai, China, 2008: 259 - 263.

[3] Himavathi S, Anitha D, Muthuramalingam A. Feedforward neural network implementation in FPGA using layer multiplexing for effective resource utilization [J]. *IEEE Transactions on Neural Networks*, 2007, **18**(3): 880 - 888.

[4] Katao T, Hayashi K, Fujisaka H, et al. Sorter-based sigma-delta domain arithmetic circuits [C]//18*th European Conference on Circuit Theory and Design*. Seville, Spain, 2007: 679 - 682.

[5] Pneumatikakis A, Deliyannis T. Direct processing of sigma-delta signals [C]//*Proceedings of the Third IEEE In-*

ternational Conference on Electronics, Circuits, and Systems. Rodos, Greece, 1996: 13 - 16

[6] Katao T, Suzuki Y, Fujisaka H, et al. Single-electron arithmetic circuits for sigma-delta domain signal processing [C]//8*th IEEE Conference on Nanotechnology*. Arlington, USA, 2008: 729 - 732

[7] Ng C W, Wong N, Ng T S. Bit-stream adders and multipliers for tri-level sigma-delta modulators [J]. *IEEE Transactions on Circuits and Systems* Ⅱ: *Express Briefs*, 2007, **54**(12): 1082 - 1086.

[8] O'Leary P, Maloberti F. Bit stream adder for oversampling coded data [J]. *Electronics Letters*, 1990, **26**(20): 1708 - 1709.

[9] Liang Y, Wang Z G, Meng Q, et al. Design of high speed high SNR bit-stream adder based on ΣΔ modulation [J]. *Electronic Letters*, 2010, **46**(11): 752 - 753.

[10] Fujisaka H, Kurata R, Sakamoto M, et al. Bit-stream signal processing and its application to communication systems [J]. *IEE Proc Circuits Devices Syst*, 2002, **149**(3): 159 - 166.

基于总和增量调制的比特流人工神经元和感知器的 FPGA 实现

梁 勇 王志功 孟 桥 郭晓丹

(东南大学射频与光电集成电路研究所,南京 210096)

摘要:为了解决传统的多比特数字方法在实现人工神经网络时所存在的硬件规模过于庞大的问题,提出了一种基于总和增量(ΣΔ)调制的比特流人工神经网络硬件实现方法. 在 FPGA 中实现了比特流加法器、乘法器、阈值函数单元和全数字 ΣΔ 调制器,并采用这些比特流运算单元构建了比特流人工神经元. 用实现逻辑函数功能和线性分类器的方法验证了比特流人工神经元的功能. 基于比特流人工神经元的带预处理结构的比特流感知器被证明具有非线性分类的能力. 比特流人工神经元实现所使用的 FPGA 资源表明比特流人工神经网络硬件实现技术可以显著地减少人工神经网络对硬件资源的需求.

关键词:比特流; 人工神经元; 感知器; ΣΔ; FPGA

中图分类号:TN710