School of Electrical, Electronic & Computer Engineering

UNIVERSITY OF
NEWCASTLE UPON TYNE

Modeling Asynchronous ANNs for Energy Efficient Implementation

Yuan Chen, Fei Xia, Alex Yakovlev

Technical Report Series

NCL-EECE-MSD-TR-2005-105

May 2005

Contact:
Fei.Xia@newcastle.ac.uk

NCL-EECE-MSD-TR-2005-105

School of Electrical, Electronic & Computer Engineering,
Merz Court,
University of Newcastle upon Tyne,
Newcastle upon Tyne, NE1 7RU, UK

http://async.org.uk/

Modeling Asynchronous ANNs for Energy Efficient Implementation

Yuan Chen, Fei Xia, Alex Yakovlev

Abstract – **Artificial Neural Networks (ANNs) have been widely used in embedded systems to provide multi-point control. We use Asynchronous communication mechanisms (ACMs) to provide a means to enable ANN neurons to be activated only when necessary, so that systems' energy consumption can be greatly reduced.**

1 INTRODUCTION

Artificial Neural Networks (ANNs) have been introduced into the field of embedded systems in order to provide an effective means of multi-point control. Whatever differences these ANNs may have, all neurons in these networks are activated under one global clock, whether necessary or not. Although it may not cause much trouble in desktop systems, the high energy cost of data exchange and processing may not be realistic for embedded systems. Even worse, the high speed global clock switching all over such ANNs at approximately the same time is the largest source of EMI (Electromagnetic Interference) [1]. All the problems caused by synchronous ANNs call for new ANN models.

2 BASIC STRUCTURE OF ASYNCHRONOUS ANNS

Biological neural networks do not have such global clocks and their neurons are activated when new data is ready in their input neurons. The intrinsic feature of biological neurons' asynchrony characteristic enlightens us to integrate Asynchronous Communication Mechanisms (ACMs) [2] with traditional ANN design.

2.1 Asynchronous Neuron Design

Jeff Hawkins and Dileep George's recent research work [3,4] about human neural cortex give us fresh ideas about ANNs design. When they explore the Invariance Property [4] about the human retina, they pointed out that the prediction characteristic was essential for human neural cortex to make quick reactions. Every neuron will make its prediction about the new data it may receive. If the received data agrees with its prediction, neurons do not output its calculation result in detail, but only send signals which mean OK or NO VARIATION outside. Otherwise, ALARM signal will be generated and sent out to others. Detailed data will only be provided when ALARM signals are answered. On the other hand, neurons stay inactive until ALARM signals are received. By this way, neurons act asynchronously and can focus their limited resources to solving the key part of the entire problem and react quickly.

If artificial neurons can be designed in the same way, the limited energy of embedded systems can be optimally used. To this end, we introduce ACMs into our ANNs design. An ACM, as defined here, is a connector between two asynchronous processes, a writer and a reader, through which a sequence of data items can be passed without necessitating the writer and reader to be locked for synchronization under a global clock [2]. In asynchronous ANNs, all data processing happens in neurons; therefore every neuron can play the role of a writer or a reader in different situations, based on the usage of its memory. When it is activated, a neuron works as a reader to fetch data from the memory of other connected neurons. After that, the neuron will do its calculation and write the calculation result into its memory, in which it plays the role as a writer.

2.2 Asynchronous Neural Regions

The entire biological neuron network can be seen as a hierarchy which contains multiple layers. Values used by particular neurons to make prediction are called **Basic Patterns**, such as the 256 grey scales for visual neurons. Biological networks incline to combine their neurons' predictions together to make the uniform pattern recognition of the entire object or some regions of it. The joint patterns used by neurons of one layer are called **Advanced Patterns**. Every layer contains its unique advanced patterns and there are deterministic relationships between advanced patterns in neighboring layers. If the new data matches one of the advanced patterns in the present layer, the layer will not output the advanced pattern, but only the pattern's number, in which way the network can convey the same information with very low overhead.

In order to provide the similar performance for ANNs, we integrate the idea from asynchronous circuits into our design. Asynchronous circuits use a pair of handshake signals such as Req (Request) and Ack (Acknowledge) to control a group of data wires, which are called Bundled Data [5]. In our design, a

group of neurons with the same bundle of inputs and outputs are called a **neuron region**. Neurons in the same region are activated synchronously, and make the entire pattern recognition of the region. They share the same handshake signals to communicate with neurons in other regions. The more neurons one region contains, the lower rate the overhead of handshake signals the region has.

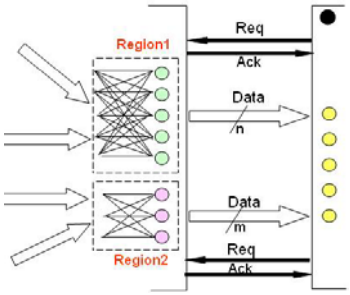

Figure 1: Asynchronous Neuron Region

The entire network can be seen as a GALS (Global Asynchronous Local Synchronous) [6] system, since neurons within a region work synchronously while exchange data asynchronously outside the region. In this way, we can greatly preserve characteristics of traditional synchronous ANN models in our new ANN design. Furthermore, the asynchronous neural region is a good solution for EMI because the switching of the handshake signals is not correlated in time [1]. Recently, the world's first asynchronous microprocessor made by EPSON reduced energy consumption by 70% and electromagnetic radiation by 20dB [7], which shown a promising future for asynchronous ANN design.

2.3 Signal and Data Processing

Neuron regions use both signals and data to communicate. Signals can only convey simple information but cost far fewer energy than data which contains plenty of information. According to the type of information processing, each neuron region of the type discussed in this paper consists of three parts.

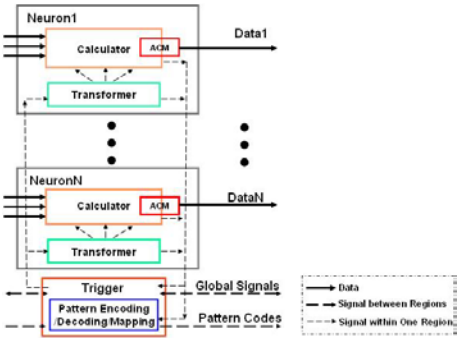

Figure 2: Data Processing vs. Signal Processing

Although every region contains more than one neuron, it has only one **trigger**. The trigger receives handshake signals from outside world to decide whether to activate neurons in the region, or only do pattern mapping. Every neuron in the region can be divided into a **transformer** and a **calculator**. Activated by the trigger, the transformer generates all kinds of signals for data processing, which is performed by the calculator.

The trigger is the most active part of the region but consumes extremely little energy because it only processes very simple signals. On the other hand, the greater part of system energy is consumed by the calculator, which does large amounts of data processing every time. Since most of the time a region only has the trigger alert to receive outside signals, asynchronous ANNs can perform the same tasks as but with much less energy than synchronous ones.

3 A MATLAB MODEL OF ASYCHRONOUS ANNS

We have developed the method to model and simulate Asynchronous ANNs using MATLAB 7.0 for Windows. As a first example, an asynchronous neuron with five main parts has been built using MATLAB, which is shown in Figure 2.

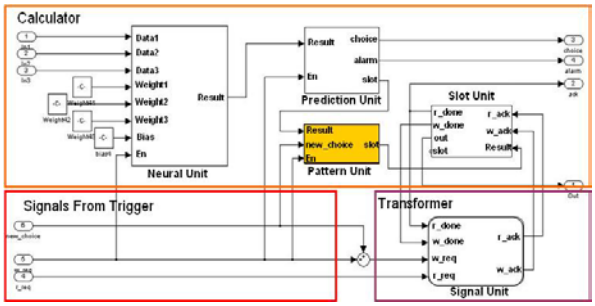

Figure 3: Asynchronous Neuron Built in MATLAB

In hardware design, dual-rail protocol [5] is used, in which two wires are used to carry a single bit of information. In this paper, two numbers 1 and -1 are used in the simulation to denote the binary inputs while 0 represents the idle state.

Neural Unit is the same as the traditional synchronous BP (Back-propagation) neuron with three inputs and one output. This unit and the following Prediction Unit, Pattern Unit, Slot Unit constitute the calculator of the neuron.

Prediction Unit: In our toy example, neurons use two basic patterns: {-1, 1} as the prediction values. The **Maximum Likelihood Rule** is used to make judgment. If the present result is close enough to one of the basic patterns, the neuron will treat the received data as the chosen pattern and write the pattern value into its memory, which will avoid the accumulation of system noise. At the same time,

Alarm signal outputs "-1" to inform that no new data has been detected, and the Choice signal is used to inform the trigger of the region about the basic pattern chosen by the present neuron. However, if the calculation result is far from both the two basic patterns, the Choice signal keeps idle and Alarm signal will become "1", and the new result will be written into the memory as the new data.

Pattern Unit: Pattern Unit is highlighted because it only appears in neurons in the highest layer. As mentioned before, if the trigger part of the present region receives no signal to inform new data, it will make pattern mapping without data calculation. It means data in the memory of neurons does not need to be updated every time. Things are different in neurons in the highest layer because final results will be read from their memories. Therefore, Pattern Unit will write the chosen pattern into the memory when the trigger gives the indication, while let the result of Prediction Unit access the memory when no indication is received. In neurons which do not give the final result, this unit is omitted.

Slot Unit: In asynchronous ANNs, every neuron works as writer or reader in different situations. In our design, the writer may overwrite the data in its memory since it does not know when the data will be read by neurons in other region, and the reader may reread the same data because every neuron connect with more than one neurons. Because the memory must satisfy the write and overwrite requests from the writer, and read and reread ones from the readers, the POOL type of ACMs [2] is chosen. In this paper, only one memory unit or slot is used in every neuron and all the data actions are treated as atomic. This is because of the relatively small size of the data items.

Signal Unit: Signal Unit is the transformer of the neuron, and will give local signals used by the calculator part. Since every neuron's writing and reading actions are atomic and mutually exclusive, an additional local variable "K" is used as a "key" to the slot. K=0 means the slot is locked and K=1 means unlocked. Whoever grasps the key can access the slot in the fashion of the classical MUTEX protection scheme. The algorithm of slot control is given as:

Writer		Reader	
w0:	wait until k=1	r0:	wait until k=1
w:	k:=0 write slot	r:	k:=0 read slot
wd:	k:=1	rd:	k:=1

Trigger Unit: Trigger unit belongs to the entire region, not any particular neuron. The trigger will activate its neurons when global Alarm signals are received. It has two main tasks. The first one is signal conversion. Global handshake signals such as R_req and Ack will be turned into regional ones, or vice versa. More complex control can be provided here if necessary. The other task is pattern recognition. It can be divided into three parts: pattern encoding, pattern mapping and pattern decoding. **Pattern encoding** will turn advanced patterns of the region into corresponding codes. If any neuron in the region sends alarm signal "1" to inform new data, a global Alarm signal "1" will be sent. The global alarm signal will also be created when the trigger may not find proper code of the present pattern, which is called **pattern miss-shot**. In other cases, the trigger will generate the pattern code to feed other regions. When no valid global alarms are received, the trigger will only do **pattern mapping** to generate its pattern codes based on the received ones from its inputs. **Pattern decoding** is only available in the regions in the highest layer, which decode pattern codes into advanced patterns, and inform the pattern unit of its neurons to choose the correct basic patterns.

In our method, most parts of the neurons are modularized so that designers can easily use them to build their own systems. However, the trigger unit is not fixed in order to give designers enough freedom to realize their systems' unique requirements and pattern maps.

4 SIMULATIONS

In this paper, we want to simulate a one bit full-adder (two operands A0, B0 with carryin C0 to generate sum A1 and carryout C1). A three-input-two-output neural network (Figure 3) has been built using the techniques described in section 3. MATLAB NNTOOL is used to obtain the weights &bias of the system.

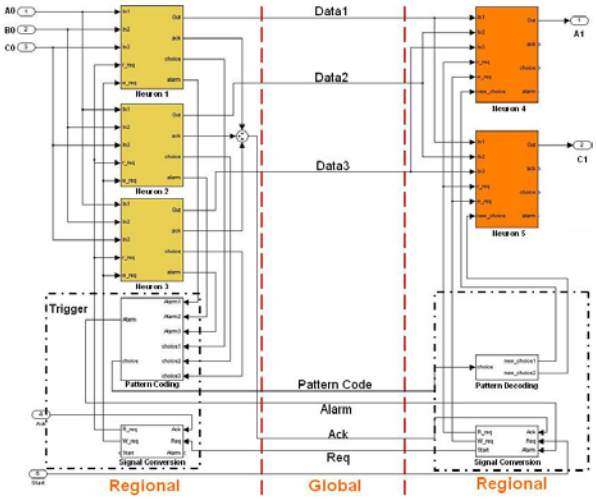

Figure 4: 3-input-2-output ANN

ANN is fed with data evenly generated from {-1 -1 -1} to {1 1 1} have been added with random

numbers which follow **Gaussian Distribution**. They are used to simulate the noise-integrated data fed to the network. Neurons 1, 2, 3 are fed with the same original data from the outside world, and constitute the first region of the network. The second region contains neurons 4, 5.

Although each neuron in the first region can recognize two basic patterns: "-1" and "1", there are only six valid advanced patterns. In this simple toy example, patterns {-1 1 1} and {1 -1 -1} are chosen to be encoded since they happen most frequently. Consequently, one wire "Pattern Code" is used to send pattern codes ("1" for {-1 1 1} and "-1" for {1 -1 -1}) to the second region.

Figure 5 gives the data traffic between the two regions. In almost half of the time, one wire of pattern code is used instead of three wires of detailed data. Consequently, neurons in the second region are activated to do data calculation only half of the time. The more neurons one region contains, or the more patterns are encoded, the fewer detailed data items are transmitted in the network and the fewer times neurons in the next region are activated.

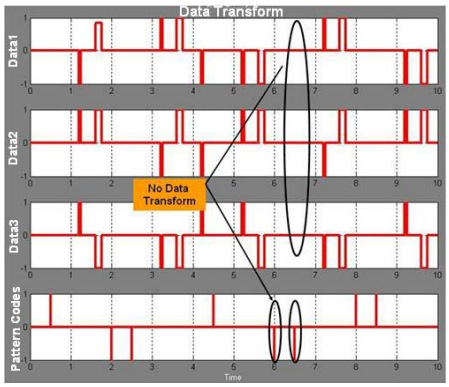

Figure 5: Pattern Codes vs. Detailed Data

Furthermore, we fed the same data into both our asynchronous ANN and a synchronous ANN built with the same parameters. The result of asynchronous ANN is functionally the same as its synchronous counterpart with some minor delay. It is because data is fetched by the asynchronous neurons when the alarm signals are answered while data items are sent directly to synchronous neurons unconditionally without delay. At the same time, the level of switching activity in an asynchronous ANN is significantly lower owing to the use of asynchronous triggering mechanism and self-timing. The result demonstrates that asynchronous ANNs can successfully carry out tasks traditionally fulfilled by synchronous ANNs.

5 CONCLUSIONS

New biological suppositions and ACMs have been integrated with traditional ANNs design. Neurons are grouped into different regions to make pattern recognition according to their predictions, and send pattern codes or data to communicate with others in different situations. This may offer a more adequate implementation technology than synchronous in terms of using timing and power resources. Reduced EMI is another potential advantage.

Acknowledgments

The authors would like to thank Jack Scannell, Malcolm Young and Fei Hao for enlightening and extensive discussions. Yuan Chen is supported by Newcastle University's International Research and department scholarships.

References

[1] The University of UTAH, *Asynchronous Circuit and System Design* http://www.cs.utah.edu/research/factsheets/asyncprt.pdf

[2] Fei Xia and Ian Clark; *Algorithms for Signal and message Asynchronous Communication mechanisms and their Analysis*; Fundamenta Informaticae Volume 50, Issue 2 Pages: 205-222

[3] Jeff Hawkins and Sandra Blakeslee. 2004. *On Intelligence*. Times Books, Henry Holt and Company, New York, NY 10011

[4] Dileep George, Jeff Hawkins; *Invariant Pattern Recognition using Bayesian Inference on Hierarchical Sequences* http://www.stanford.edu/~dil/RNI/DilJeffTechReport.pdf

[5] J.Sparsø and S. Furber (eds.) 2001, *Principles of asynchronous circuit design- A systems perspective,* kluwer Academic Publishers

[6] Hemani, A.; Meincke, T.; Kumar, S.; Postula, A.; Olsson, T.; Nilsson, P.; Oberg, J.; Ellervee, P.; Lundqvist, D.1999,;*Lowering power consumption in clock by using globally asynchronous locally synchronous design style* Design Automation Conference, 1999. Proceedings. 36th, Pages:873 - 878

[7] http://www.epson.co.jp/e/newsroom/2005/news_2005_02_09.htm