Advanced Research in VLSI

Proceedings of the Fifth MIT Conference, March 1988

edited by
Jonathan Allen and
F. Thomson Leighton

The MIT Press
Cambridge, Massachusetts
London, England

PUBLISHER'S NOTE

This format is intended to reduce the cost of publishing certain works in book form and to shorten the gap between editorial preparation and final publication. Detailed editing and composition have been avoided by photographing the text of this book directly from the authors' prepared copy.

This book was printed and bound in the United States of America

Library of Congress Cataloging-in-Publication Data

Advanced Research in VLSI.

Papers presented at the Fifth MIT Conference on Advanced Research in VLSI, organized by the Microsystems Research Center of the Massachusetts Institute of Technology, held in Cambridge, Mass., Mar. 1988.
Includes bibliographies and index.
1. Integrated circuits--Very large scale integration--Congresses. I. Allen, Jonathan, 1934- . II. Leighton, Frank Thomson. III. Massachusetts Institute of Technology. Microsystems Research Center. IV. MIT Conference on Advanced Research in VLSI (5th: 1988: Cambridge, Mass.)
TK7874.A3352 1988 621.395 88-621
ISBN 0-262-01100-X

Advanced Research in VLSI

Proceedings of Previous Conferences

Proceedings of Caltech Conference on Very Large Scale Integration, 1979, Charles L. Seitz, Editor, Computer Science Department, California Institute of Technology, 256–80, Pasadena, CA 91125.

Proceedings of Second Caltech Conference on Very Large Scale Integration, 1981, Charles L. Seitz, Editor, Computer Science Department, California Institute of Technology, 256–80, Pasadena, CA 91125.

Proceedings, Conference on Advanced Research in VLSI, MIT, January 1982, Paul Penfield, Jr., Editor, Artech House, Inc., 610 Washington St., Dedham, MA 02026.

Third Caltech Conference on Very Large Scale Integration, 1983, Randal Bryant, Editor, Computer Science Press, Inc., 1803 Research Boulevard, Rockville, MD 20850.

Proceedings, Conference on Advanced Research in VLSI 1984, Paul Penfield, Jr., Editor, Artech House, Inc., 610 Washington St., Dedham, MA 02026.

1985 Chapel Hill Conference on Very Large Scale Integration, Henry Fuchs, Editor, Computer Science Press, Inc., 1803 Research Boulevard, Rockville, MD 20850.

Advanced Research in VLSI: Proceedings of the 4th MIT Conference, Charles Leiserson, Editor, MIT Press, 55 Hayward Street, Cambridge, MA 02142.

Advanced Research in VLSI: Proceedings of the 1987 Stanford Conference, Paul Losleben, Editor; The MIT Press, 55 Hayward St., Cambridge, MA 02142.

All of the above volumes are available directly from the publishers.

Preface

As the size and complexity of VLSI circuits has increased over the last decade, the emphasis among the many VLSI disciplines has changed from its intial focus on circuits and technology to a growing concern with design issues. Additional levels of abstraction have encouraged computer scientists to contribute in a complementary way with electrical engineers. While computer scientists were concerned with logic, architecture, and complexity issues, electrical engineers addressed technology and circuit issues. Early in the decade, there was a tendency for the two disciplines to produce designs in different representational domains. The meeting ground was at the artwork level, where both groups shared a concern for the details of layout and interconnection. In recent years, however, there has been increased integration as local optimizations within the individual representational domains were recognized as inadequate in terms of providing the best overall system. Performance for specific applications is acknowledged as the key motivator, and it has necessitated a greater awareness of the interactions of various design representations, such as layout, circuit, and architecture.

In this conference, the broad range of important contributions to the design of high-performance circuits (often targeted at novel applications) is demonstrated. New architectures aimed at improved system performance are discussed, and the use of novel analog circuits in connectionist architectures is described. Given the importance of speed, it is not surprising to find several papers on clock design methodologies, and the use of self-timed circuits. Fault tolerance, reconfiguration, testing, and verification are topics of growing concern due to the huge complexity of existing chips and systems which require fresh approaches to design and CAD tools. Design optimization continues to be a major theme, focusing on logic, circuit size, and speed. The specification of high-performance layout (both in theory and practice) is also addressed, and the integration of these design tools into easy-to-use and extendable CAD systems deals with the need to coherenly interrelate programs oriented to many facets of the overall design. We believe this broad range of interest illustrates the vitality in all aspects of contemporary VLSI research.

We gratefully acknowledge the strong contributions of those on the program committee, who collectively and carefully read the eighty submitted paper, and selected nineteen from an outstanding field. The desire to avoid parallel sessions, and the need to cover a substantial range of topics, prevented us from accepting many excellent papers. We also thank the authors of the nineteen contributed papers, and the nine invited speakers, whose expertise characterizes the ongoing progress in the many diverse aspects of VLSI as well as Robert Lucky who is the Conference banquet speaker. Thanks to Bernard Chern and Robert Grafton of the National Science Foundation for their support of the conference. Barbara Lory, Barbara Tilson, and the staff of the MIT Microsystems Research Center have contributed to all aspects of the Conference's organization and planning. Paul Penfield is the guiding hand who has set the standard for this continuing series of MIT conferences. Carver Mead and Chuck Seitz started these university-based conferences in 1979, and MIT is pleased to carry on the tradition during the even-numbered years. We hope you enjoy these papers and the view they provide of today's VLSI research.

Jonathan Allen
F. Thomson Leighton

Program Committee

Jonathan Allen, Co-chairman, MIT
Robert Brayton, University of California at Berkeley
William J. Dally, MIT
Stephen W. Director, Carnegie-Mellon University
Henry Fuchs, University of North Carolina
Andrea LaPaugh, Princeton University
F. Thomson Leighton, Co-chairman, MIT
Thomas Lengauer, University of Paderborn, West Germany
Gordon D. Robinson, GenRad, Inc.
Alberto L. Sangiovanni-Vincentelli, University of California at Berkeley
Thomas G. Szymanski, AT&T Bell Laboratories
Jeffrey D. Ullman, Stanford University

Contents

A Pulse-Width Modulation Design Approach and Path-Programmable Logic for Artificial Neural Networks

Neil E. Cotter, Kent Smith, and Martin Gaspar

Electrical Engineering Department
University of Utah
Salt Lake City, UT 84112

Simple circuits for performing mathematical operations on analog signals are described. Multiplication, addition, and nonlinear scaling of voltages are accomplished by a pulse-width modulation scheme that utilizes comparators and switched capacitor filters implemented in CMOS. A set of standard cells containing these circuit elements may be used in a Path-Programmable Logic scheme that provides for extremely rapid artificial neural network integrated circuit design. Design of a Kohonen self-organizing feature map is described.

1. Introduction

Biological neural networks rely on analog signal processing and massive connectivity to achieve low-power, high-speed computation. Artificial Neural Networks (ANNs) mimic this ability by performing computations electronically or optically. A new approach for designing ANN integrated circuits based on Pulse-Width Modulation (PWM) circuits is described here. The PWM approach has several advantages over other proposed approaches [1,2,5,7-11,19,20]: (1) it provides a reliable inter-chip communication scheme, (2) it provides a circuit for multiplication that has wide dynamic range and tolerates variations in component values, (3) it provides a nonlinear scaling circuit with a precisely controllable transfer function, (4) it provides a small number of cell types that can be used to implement almost any neural network algorithm, (5) it provides convenient digital PWM test signals, and (6) it provides design cells that require small numbers of components.

The Path-Programmable Logic (PPL) system, developed by Smith [21,26-29] at the University of of Utah, may be used to layout PWM circuits for ANNs. In PPL, layout starts with the definition of signal lines that are laid down in a grid on the integrated circuit. Circuit cells are then placed at appropriate locations on the grid. The primary advantage of PPL is that complete integrated circuits can be designed in as little as a few hours, but PPL has additional advantages for the design of ANNs: (1) circuits for simulation are automatically extracted from the layout, (2) analog and digital circuits are easily combined, and (3) mixed types of ANNs can be implemented on a single chip. In the last section of the paper the simplicity of the PPL approach is demonstrated for a somewhat complicated algorithm: Kohonen's self-organizing feature map [17].

2. Circuit building blocks for ANNs

CMOS circuit building blocks used in the PPL design of ANNs are shown in Figure 1. The first building block is a comparator with complementary outputs. As shown below, the comparator is useful in at least four roles: (1) in converting analog voltage levels to PWM signals, (2) in multiplying voltage levels, (3) in finding the smallest of a set of voltages, and (4) in scaling voltages by nonlinear functions.

The second building block is a summation and low pass filter circuit designed with switched capacitors. This circuit is useful as a stand-alone low pass filter or as a summation circuit. It accepts either PWM waves or analog levels as input, and it outputs a slowly varying signal that is proportional to the duty cycle of the sum of the input voltages. A complementary analog output (denoted by a solid circle) is provided for use in multiplication circuits. All resistances in the circuit have the same value and are simulated by switched capacitors that require a clock signal ($\phi 1$).

The third building block is a capacitive storage circuit for synaptic weights. Values stored as voltages on synaptic weight capacitors act as the memory elements in an ANN. The function of these synapses is to control the strength of connections between neurons, exactly paralleling the function of synaptic connections between biological neurons.

Changes in synaptic weights are effected by a voltage connected to a capacitor through a CMOS switch and a resistance R. If the switch is closed for a short time t_Δ, the capacitor starts charging from its initial voltage V_0 towards the input voltage V_I. The net change ΔV in voltage on the capacitor is proportional to t_Δ provided t_Δ is small:

$$\Delta V \approx \frac{t_\Delta}{RC}(V_I - V_0)$$

This form of synaptic weight change is useful in proportional increment training [4,22,25], backward error propagation [23], self-organizing feature maps [17], and adaptive resonance [6].

If VI » V0 then the dependence on V_0 is eliminated from the equation for ΔV, and the synaptic weight can be stepped by fixed increments proportional to t_Δ and V_I:

$$\Delta V \approx \frac{t_\Delta}{RC} V_I$$

This form of synaptic weight change is useful in outer product rule learning for Hopfield networks [13-16,30].

The synaptic weight can also be set to a desired voltage level by leaving the CMOS switch closed until the capacitor charges to the value of V_I. This is useful for initializing synaptic weights.

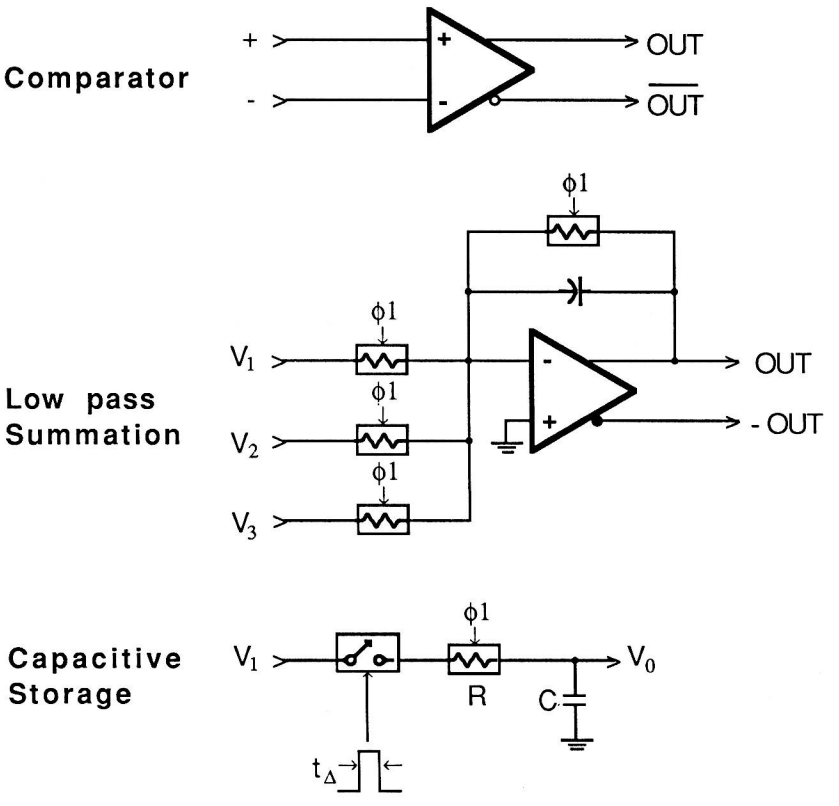

Figure 1 – Circuit building blocks for ANNs.

3. PWM interchip communication scheme

This and the following sections describe how the ANN circuit building blocks can be combined to provide useful functions. The first such function is the communication of signals between ANN integrated circuits, a central issue in the design of expandable ANNs.

Biological neurons use pulses called action potentials to communicate with one another. To a first approximation, pulse rates encode the activity of neurons. Because all action potentials are the same shape and size, the duty cycle of neural output is proportional to pulse rate. Hence, neurons convey information by altering the duty cycle of a pulse train. Equivalently, neurons may be said to be communicating sample values of an analog signal.

A PWM wave accomplishes the same task; the duty cycle of a PWM wave conveys information about an analog signal. By low pass filtering the PWM signal, the underlying analog signal is recovered. This scheme is

used in computer sound generators that sample a desired output waveform, convert it to PWM, and then low pass the PWM signal to produce a continuous output waveform.

There are three benefits of communicating with PWM rather than analog signals: (1) the PWM waveform is a digital signal, (2) the duty cycle of the PWM waveform can be controlled more precisely than the value of an analog signal, and (3) the information content of PWM signals is unaffected by moderate attenuation. These benefits permit communication of analog information to be accomplished with a minimum of analog circuitry. Consequently, circuit complexity is reduced and the reliability of communication is increased.

Figure 2 illustrates the PWM interchip communication scheme. In a transmitting chip, analog output signal V_o is converted into a PWM wave i(t) by a comparator whose output goes high whenever the analog signal is greater than the instantaneous height of a reference triangle wave v(t). With each cycle of the triangle wave a rectangular current pulse is produced. The width and DC component of the pulse are proportional to V_o. Current pulses are used so that output lines can be wired together to form the sum of signal strengths, a convenient operation for ANNs. If the input to the downstream ANN is a virtual ground, then a resistor may be used to convert voltages to current in the sending chip as shown in the figure.

The width (or duty cycle) of a pulse is equivalent to a sample value of the original analog signal. If the frequency of the triangle wave is much higher than the frequency of the analog signal, then the analog signal may be recovered by passing the PWM wave through a low pass filter [3,12,24], as shown in the receiving chip in Figure 2.

It is assumed that the reference triangle wave is provided by an external signal source. Although this requires additional circuitry, the same triangle wave can be used by every chip in a system, and triangle waves are easily generated. Furthermore, amplification or other analog processing of the triangle wave is unnecessary.

4. Multiplication

A common element of ANN algorithms is the need to compute weighted sums of voltages, where the weights are synaptic strengths and the voltages represent incoming activity from sensors or other neurons. Perceptrons provide a typical example:

$$V_i = \mathrm{sgn}\left(\sum_{j=0}^{N} w_{ij} V_j\right)$$

where $V_i \equiv$ output of ith neuron

$w_{ij} \equiv$ synaptic weight for jth input

$V_j \equiv j$th input

Since synaptic strengths and input signals are both represented by voltages, a voltage multiplier is needed. Figure 3 shows the block diagram of a PWM multiplier. Multiplication is accomplished by using the

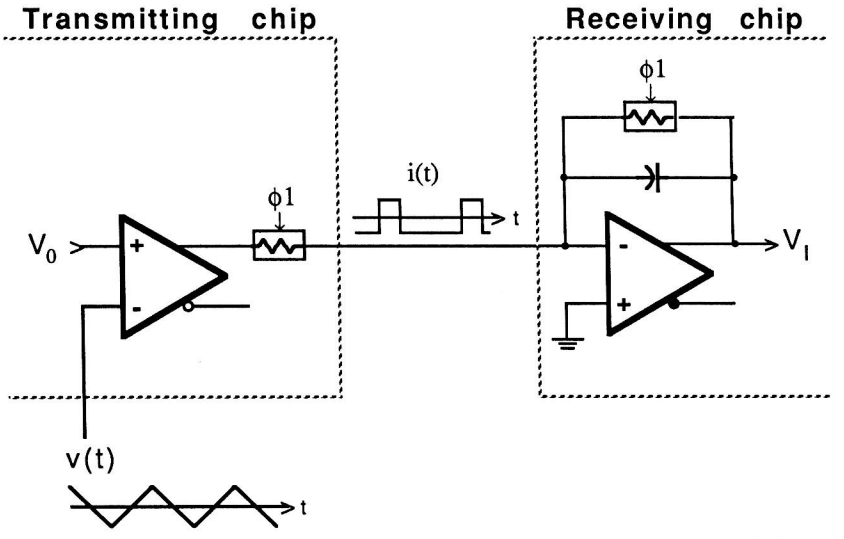

Figure 2 – PWM interchip communication scheme for ANNs.

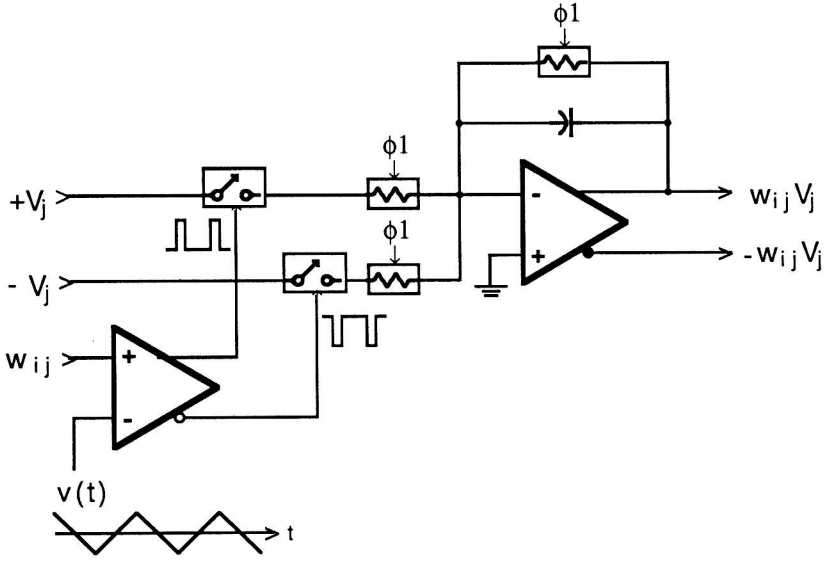

Figure 3 – Schematic diagram of PWM multiplier.

comparator described in the previous section to convert voltage w_{ij} into a PWM wave . (If the incoming signal is already a PWM wave this step is unnecessary.) The PWM wave is then used to control CMOS switches through which voltages $+V_j$ and $-V_j$ are passed. Dual signal polarities are needed for four quadrant multiplication. This is the reason for having op-amps with complementary outputs in the low pass summation circuit discussed earlier.

The output of the multiplication circuit is a PWM wave whose duty cycle is proportional to w_{ij} and whose height is proportional to V_j. Hence, the DC component of the output waveform represents the product $w_{ij}V_j$. A low pass filter can be used to extract the DC component, yielding a continuous analog signal.

5. Nonlinear scaling function

In most ANN algorithms a sharp threshold or nonlinear sigmoidal function is required. If the inputs to a comparator are ground and a signal, a sharp threshold function of the signal is obtained.

A nonlinear sigmoidal transfer function g() may also be obtained from a comparator. The scheme is illustrated in Figure 4. A signal V_i that is to be compressed by the g() function is connected to the + input of a comparator, and a reference signal $g_p^{(-1)}(t)$ is connected to the - input of the comparator. The output of the comparator is a PWM wave with DC component V_o equal to $g(V_i)$. A low pass filter extracts the DC level.

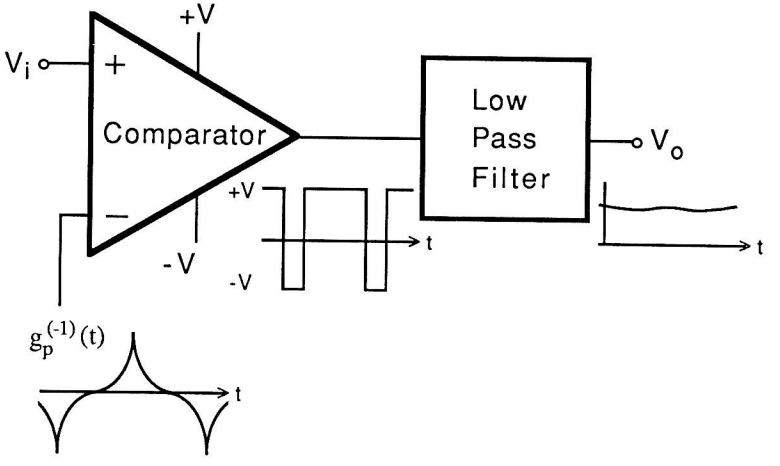

Figure 4 – Nonlinearity Circuit with arbitrary transfer function.

To obtain a given transfer function g(), the needed reference signal $g_p^{(-1)}(t)$ is calculated in four steps illustrated in Figure 5:

1. Find the function inverse $g^{(-1)}()$ of g(). (The function inverse $g^{(-1)}()$ satisfies $g^{(-1)}(g(V)) = V$ and should not be confused with the multiplicative inverse $\frac{1}{g(V)}$.) For $g^{(-1)}()$ to exist, g() must be invertible. This requirement is met by all strictly increasing or decreasing functions, a class of functions that includes sigmoids used in ANNs.
2. Scale the horizontal axis of $g^{(-1)}()$ so that its domain corresponds to one time interval T, where T is much longer than the clock $\phi 1$ used for switched capacitors. For this scaling to be possible, g() must have a restricted range. This requirement is met by sigmoids that are bounded above and below.
3. Create a periodic signal by replicating $g^{(-1)}()$ along the horizontal time axis.
4. To reduce the bandwidth required for $g_p^{(-1)}(t)$, flip every other copy of $g^{(-1)}()$ right-to-left, and.round off the infinitely high points on the signal. Start rounding off at voltages just above the comparator supply voltages. The resulting waveform is the desired reference signal $g_p^{(-1)}(t)$.

The DC level V_o of the comparator output is equal to the percentage of time the comparator output is high ($V_i > g_p^{(-1)}(t)$) minus the percentage of time the comparator output is low ($V_i \le g_p^{(-1)}(t)$). Voltage V_o may be computed from one interval of $g_p^{(1)}()$ as illustrated in Figure 6. The mathematical equation relating V_o and V_i involves integrals corresponding to the time intervals where $V_i > g_p^{(-1)}(t)$ and $V_i \le g_p^{(-1)}(t)$:

$$V_o = \frac{1}{2}\left(\int_{-1}^{t_i} V\,dt \; - \; \int_{t_i}^{-1} V\,dt \right) = g(V_i)$$

where t_i is implicitly defined by $g_p^{(-1)}(t_i) = V_i$

and $\pm V$ are the comparator supply voltages

A single external circuit may be used to generate and distribute the $g_p^{(-1)}(t)$ signal to any number of ANN integrated circuits. A stairstep waveform passed through a smoothing filter is a suitable way to generate $g_p^{(-1)}(t)$, and the shape of the waveform may be slowly varied over time if desired. This is useful in Hopfield networks where increasingly steep sigmoidal functions are used in solving optimization problems.