Hardware Intelligent Control System

By

Brett Ferlin

Submitted to:

Dr. Donald Schertz

EE 451 Senior Laboratory

December 13, 2001
Summary

My project, the Hardware Artificial Intelligent Control System, is proposed to provide a cheap, fast alternative to expensive learning systems based on neural networks. By implementing a model of a biological neuron, it is hoped that audio processing and phoneme recognition can be achieved. Once this small task has been conquered, the system should be adaptable to most control applications.

Timeline

The project will take place in four phases, each planned to take approximately a month.

- Phase 1 - Design and build current implementation.
- Phase 2 - Rework Design for learning accuracy and then Rebuild.
- Phase 3 - Rework Design for speed and efficiency, then Rebuild.
- Phase 4 - Finalize Design

The design for the first phase has been completed this semester, and will be discussed in this proposal.

Phase 1 Block Diagram

![Fig 1. Block Diagram of Proposed System](image)

Input sources write directly to the memory space: when some external input has data ready, it is written to the memory space at some predefined memory location.

Output sources write directly from memory space: when some external output needs data, it just reads some predefined area of memory.
The Microprocessor reads the ‘consciousness stack’ and acts according to the ‘command byte’ at the beginning of the present action. The entire stack is then moved back to retain a short memory of the last few actions. The microprocessor also has a few pins for programming the neuron chips.

The Memory space can contain anything from inputs and output data being readied for processing, to actual banks of saved neuron chips which can be read and written to the chips dynamically. This allows for the use of far more chips than are physically available, taking a performance hit for added accuracy. This banking will be further researched in phase 3.
The single ‘neuron’ chips will have a block diagram similar to the proposed one given below in Figure 3.

The flip-flops are simply storage devices that will hold the current test and response values for this ‘neuron’.

The comparitor circuit simply compares the flip flop bus with the current input bus to decide what to respond with.

The switch is simply a circuit which given the input will either respond with the input bus, or the response value from the flip-flop.

And the Microprocessor will have a flow diagram much like:

![Flow Diagram](Fig 4. Proposed Flow Diagram)
Circuit Diagrams and Simulation Results

- Flip Flop Bank
  The flip-flop bank is used to hold the data until it’s ready to be used in processing.

This flip-flop bank is used twice in the phase 1 implementation of the neuron chip (shown in fig 6) to hold the testing and response data.

This chip was set up and tested using the circuit shown in fig 7, however a ‘disable’ pin was added to the design for implementation in programming. If one chip is being programmed, it would be disadvantageous to have that value altered by a previous chip.

This chip was tested for the ability to accomplish the task for which it was designed, and not for actual system integration response. Such data will be accumulated when the system is completed.
Datasheets

A few programmable logic devices considered for use in this project are shown in figure 9.

Fig 8. Datasheets for PLD Devices
Features

- HIGH PERFORMANCE E*CMOS® TECHNOLOGY
  - 4 ns Maximum Propagation Delay
  - \( F_{\text{max}} = 250 \, \text{MHz} \)
  - 3.5 ns Maximum from Clock Input to Data Output
  - UltraMOS® Advanced CMOS Technology
- ACTIVE PULL-UPS ON ALL PINS
- COMPATIBLE WITH STANDARD 22V10 DEVICES
  - Fully Function/Fuse-Map/Parametric Compatible with Bipolar and UVCMOS 22V10 Devices
- 50% to 75% REDUCTION IN POWER VERSUS BIPOLAR
  - 90mA Typical Icc on Low Power Device
  - 45mA Typical Icc on Quarter Power Device
- E* CELL TECHNOLOGY
  - Reconfigurable Logic
  - Reprogrammable Cells
  - 100% Tested/100% Yields
  - High Speed Electrical Erasure (<100ms)
  - 20 Year Data Retention
- TEN OUTPUT LOGIC MACROCELLS
  - Maximum Flexibility for Complex Logic Designs
- PRELOAD AND POWER-ON RESET OF REGISTERS
  - 100% Functional Testability
- APPLICATIONS INCLUDE:
  - DMA Control
  - State Machine Control
  - High Speed Graphics Processing
  - Standard Logic Speed Upgrade
- ELECTRONIC SIGNATURE FOR IDENTIFICATION

Description

The GAL22V10, at 4ns maximum propagation delay time, combines a high performance CMOS process with Electrically Erasable (E*) floating gate technology to provide the highest performance available of any 22V10 device on the market. CMOS circuitry allows the GAL22V10 to consume much less power when compared to bipolar 22V10 devices. E* technology offers high speed (<100ms) erase times, providing the ability to reprogram or reconfigure the device quickly and efficiently.

The generic architecture provides maximum design flexibility by allowing the Output Logic Macorecell (OLMC) to be configured by the user. The GAL22V10 is fully function/fuse map/parametric compatible with standard bipolar and CMOS 22V10 devices.

Unique test circuitry and reprogrammable cells allow complete AC, DC, and functional testing during manufacture. As a result, Lattice Semiconductor delivers 100% field programmability and functionality of all GAL products. In addition, 100 erase/write cycles and data retention in excess of 20 years are specified.

Copyright © 2000 Lattice Semiconductor Corp. All brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. (503) 268-8000; 1-800-LATTICE; FAX (503) 268-8556; http://www.latticesemi.com

August 2000

22v10_06
# MACH 4 CPLD Family

**High Performance EE CMOS Programmable Logic**

## FEATURES

- High-performance, EE CMOS 3.3-V & 5-V CPLD families
- Flexible architecture for rapid logic designs
  - Excellent First-Time-Fit™ and refit
  - SpeedLocking™ for guaranteed fixed timing
  - Central, input and output switch matrices for 100% routability and 100% pin-out retention
- High speed
  - 5.0ns $t_{PD}$ Commercial and 7.5ns $t_{PD}$ Industrial
  - 182MHz $f_{MAX}$
- 32 to 256 macrocells; 32 to 384 registers
- 44 to 256 pins in PLCC, PQFP, TQFP and BGA packages
- Advanced capabilities for easy system integration
  - 3.3-V & 5-V JEDEC-compliant operations
  - JTAG (IEEE 1149.1) compliant for boundary scan testing
  - 3.3-V & 5-V JTAG in-system programming
  - PCI compliant (-50/-55/-60/-65/-7/-10/-12 speed grades)
  - Safe for mixed supply voltage system designs
  - Programmable pull-up or Bus-Friendly™ inputs and I/Os
  - Hot-socketing
  - Programmable security bit
  - Individual output slew rate control
- Flexible architecture for a wide range of design styles
  - D/T registers and latches
  - Synchronous or asynchronous mode
  - Dedicated input registers
  - Programmable polarity
  - Reset/ preset swapping
- Advanced EE CMOS process provides high-performance, cost-effective solutions
- Supported by Vantis DesignDirect™ software for rapid logic development
  - Supports HDL design methodologies with results optimized for Vantis
  - Flexibility to adapt to user requirements
  - Software partnerships that ensure customer success
- Vantis and third-party hardware programming support
  - VantisPRO™ (formerly known as MACHPRO®) software for in-system programmability support on PCs and automated test equipment
  - Programming support on all major programmers including Data I/O, BP Microsystems, Advin, and System General
Equipment List

For this project, I will need the following:

- A microprocessor of some sort, a PC with a sound card would be adequate.
- Microphone/amplifier/speakers
- A collection (over 30) programmable logic devices which are here at Bradley.
- Testing boards and components