The Use of FPGAs in a Novel Computing Subsystem

Tom Kean
Algotronix

Irene Buchanan
Algotronix

January 7, 1992

1 Introduction

The architecture of Algotronix's Configurable Array Logic (CAL) technology [Algotronix91] is particularly suited to the building of arrays of chips to form a single computing surface [Gray89]. The CHS2x4 and its associated software have been designed and built to explore this aspect of FPGA usage.

2 General Description

The CHS2x4 is a low cost add on card for the PC/AT bus. It is a custom computer that can be used for a variety of applications including:

- development system for products based on CAL chips
- ASIC emulation
- programmable systolic machines
- cellular automaton machines
- signal and image processing machines
- logic design training

The CHS2x4 contains up to 9 CAL1024 chips and 2M bytes of static RAM. Two boards can be connected together to form a system with a 4x4 array of CAL chips for computation and up to 4M bytes of static RAM. The CHS2x4 board is supported by the CLS software suite, including a graphical editor and an online debugger, and a library of interface routines which can be called from the user's C programs.

3 Hardware Architecture

Architecturally the CHS2x4 system consists of three hardware subsystems, see figure 1. These are:

1. PC Interface and Control Subsystem,
2. Computation Subsystem,
3. Memory Subsystem,

The bus architecture, figure 2 allows the memory subsystem and CAL configuration memory of the computation subsystem to be read and written from the host processor. Configuration memory and the memory subsystem share an address space local to the board. In addition, the computation array may execute memory operations, using the counter as an address register, synchronised to the host clock.

3.1 Bus Architecture

The four main busses on the CHS2x4 board interconnecting the major subsystems shown on these figures are:

1. Address Bus (A[0..21]).
   This bus is used to address the CAL control store, the static RAM data memory and the EPROM which contains the power up configuration for the control CAL.

2. Low Data Bus (LD[0..7]).
   This is the main data bus on the board and connects the PC data bus, the CAL control store, the data memory and the inital configuration EPROM.

3. High Data Bus (HD[0..7]).
   This bus is used in conjunction with LD[0..7] to allow 16 bit transfers between the PC, the data memory and the computation array inputs and outputs.

4. Control Bus (Mode[0..2]). The mode bits of the control bus determine the use to which the data busses will be put as detailed in table 1.

3.2 PC Interface and Control Subsystem

This subsystem has the task of handling communications with the host PC/AT and controlling transfers on the internal busses of the CHS2x4. For maximum flexibility this subsystem is based around a CAL1024 chip whose configuration is loaded from an EPROM when the PC/AT is reset or powered up.

The CHS2x4 is accessed through 16 locations in the PC/AT's IO space using a protocol implemented by the control CAL.

3.3 Computation Subsystem

The computation subsystem of the CHS2x4 board consists of a 2x4 array of CAL1024 chips and appropriate address decoding for their control store which provides random access to cells within the array for defining logic circuits and sensing function unit outputs. Access to the periphery of the array is available via the system busses and a number of connectors. At the north boundary the four CAL's are fully pinned out to a 128 way connector. This may be used to connect to a neighbouring CHS2x4 board allowing a large system containing a 4x4 array of CAL chips to be built. The west boundary of $CAL_{0,1}$ is fully pinned to the 37 pin DIN connector at the back panel edge of the PC/AT board. This may be used to directly connect external inputs and outputs. This bus is also connected to the west boundary of the control CAL to provide communication between the computation array and the control subsystem - this could be used, for example, to indicate the termination of an algorithm running on the computation array. In addition, at the east boundary, the High Data and Low Data busses connect to every fourth boundary cell. At the west boundary of $CAL_{0,0}$ both the High Data and Low Data busses are connected to boundary cells. These connections support transfers to and from the on board memory.

3.4 Memory Subsystem

The memory subsystem provides the computation array with high bandwidth access to a large local data memory thus supporting systolic and pipelined computations where data is streamed through the CAL array.

The memory subsystem contains four 32 pin sockets for static RAMs and appropriate address decoding. These sockets can be filled with 128kx8 SRAM chips, 128kx8 SRAM hybrids or 512kx8 SRAM hybrids. If all four sockets are filled with 512k hybrids a 2M byte

memory is provided.

3.5 External Connectors

There are two connectors on the CHS2x4 board to permit access to signals on the periphery of the CAL array.

1. Back Panel DB37 Connector.
   This externally accessible connector is the primary means of interfacing external devices with the CHS2x4 board. It provides 32 IO connections to the computation array.
2. Header Connector.
   This connector is intended mainly for connecting two CHS2x4 boards together to form larger arrays but it can also be used as a general purpose connector, for example in conjunction with a user designed daughterboard containing specialist interfacing circuitry.

4 Software Architecture

The software support for the CAL architecture and the CHS2x4 Custom Computer consists of a CAE environment and an applications environment. This is shown diagramatically in figure 3.

4.1 CAE Environment

The Configurable Logic Software (CLS) CAE environment contains components for both automated and manual CAL configuration. It consists of the following programs

- CLARE
  The Configurable Logic Array Editor (CLARE) is a powerful graphical tool for manipulating cellular designs. The user builds a design as a hierarchy of blocks, each of which may contain individual configurable cells and instances of other blocks. Point to point routing within blocks may be interactive or automatic via a maze router. The editor supports physical design, in a symbolic form, by allowing the user to allocate cells. At the same time, structural design is supported by allowing the user to display nets and manage inter-block and intra-cell wiring. Designs are saved in a simple text file, the *.cfg* file, that is read by other programs to produce the bit patterns to personalise a CAL chip or CHS board.
- LAURA and LIBCAL
  LAURA is the equivalent of a block generator in ASIC chip design. The purpose of LAURA is to provide a programming interface which designers can use to produce and manipulate *.cfg* files. LAURA is intended to complement CLARE by facilitating parameterisation of layouts. Basic library elements may be generated manually and then manipulated algorithmically. This should be of interest to those designers producing highly regular systolic designs. LIBCAL provides routines for reading and writing the Algotronix data file formats and defines the main data structures used by the Algotronix programs.
- LISA
  The *lisa* program generates CAL configuration .cfg files suitable for placement and routing with the CLARE graphical editor automatically from truth tables.
- edif2cfg
  The *edif2cfg* program generates CAL

configuration .cfg files suitable for placement and routing with CLARE automatically from EDIF netlists which can be created by the OrCAD or ViewLogic schematics packages.

- cfg2edif
  The *cfg2edif* program extracts netlist and timing information from placed and routed CAL designs.
- cfg2cal
  The *cfg2cal* program generates binary programming information for a given CAL board from a .cfg file, output by the CLARE graphics editor or another tool. *cfg2cal* can also generate hex files for use by EPROM programmers to allow CALs to be used in stand-alone applications.

4.2 Applications & Debugging Environment

The CAL architecture offers the designer unique facilities for monitoring and changing the internal state of an FPGA in real time. The *debbie* program provides an interactive debugging environment for users of the CHS2x4 board. Debbie provides windows to monitor the state of internal cells on the CAL array and the values in the memory and counter register on the CHS2x4 board. The values being output by individual cells are displayed as either a red (for logic 1) or blue (for logic 0) band above the corresponding gate. Cell functions and routing are determined by decoding the array control store. Cell output logic values are determined by reading back the cell output bit of the control store and not by simulation of the user's design.

The CHS2x4 board is controlled via an applications program running on the PC/AT which takes charge of disk access and displaying results as well as computations not suited for implementation on CALs. Thus, the support software for the CHS2x4 board takes the form of a library of C routines LIB2X4 which can be called from an applications program written by the user.

As well as allowing access to the local memory for transfer of data and results, these routines provide for access to the CAL's control store. This permits dynamic reconfiguration of individual resources such as routing multiplexers and function units and the monitoring of cell outputs. It is also possible to programmatically clear any latch in the circuit implemented on CALs. The software tracks the mapping of gate component names in the user's schematic to cell coordinates allowing reference to gates by name in the user's control program. This feature means the user programs which access the CAL control store need not be changed when the design placement changes. It is also possible to reconfigure the control CAL using the C library to provide new ways of accessing the board within a given application.

5 Conclusions

The CHS2x4 custom computer is an example of a powerful, effective and innovative system built from FPGA hardware and supported by both CAD tools and run-time software systems. In particular an on-line debugger provides interactive, graphically based, monitoring of the status of the computing surface in real time and fine detail.

At present the board has found application in a number of areas [Oldfield91], [Luk91]. Some of the most interesting applications of the board are in the area of dynamically reprogrammed logic either under program con-

trol e.g. to hard wire constants into operator logic or self reconfiguration in which the board reconfigures itself from a design stored in the RAM as a result of a computation.

References

[Algotronix91] Algotronix Ltd., *Configurable Array Logic User Manual* Edinburgh UK, 1991.

[Brebner] G. Brebner, *Configurable Array Logic Circuits for Computing Network Error Detection Codes* Department of Computer Science Report, University of Edinburgh, June 1991

[Gray89] J.P. Gray and T.A. Kean, *Configurable Hardware: A New Paradigm for Computation.*, Proc. Decennial Caltech Conference on VLSI, Pasadena CA, March 1989.

[Oldfield91] J.V. Oldfield and C.J. Kappler, *Implementing Self-Timed Systems: Comparison of a Configurable Logic Array with a Full-Custom VLSI Circuit* FPGA's: Proceedings of the International Workshop on Field Programmable Logic and Applications, Oxford UK 1991.

[Luk91] Wayne Luk and Ian Page, *Parameterising Design for FPGAs* FPGA's: Proceedings of the International Workshop on Field Programmable Logic and Applications, Oxford UK 1991.

Mode	*Operation*
0	Reload Control CAL
1	Write Counter
2	8 bit Write on LD[0..7], 8 bit Read on HD[0..7]
3	16 bit Write
4	16 bit Read
5	8 bit Write
6	8 bit Read
7	No Operation

Table 1: Bus Usage Modes

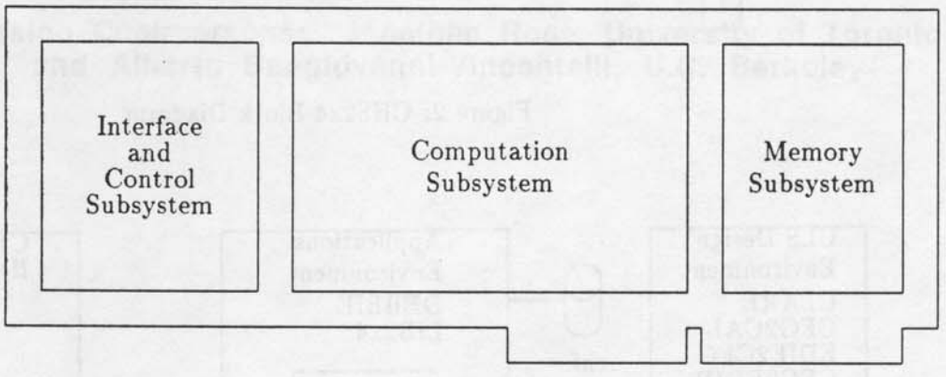

Figure 1: CHS2x4 Board

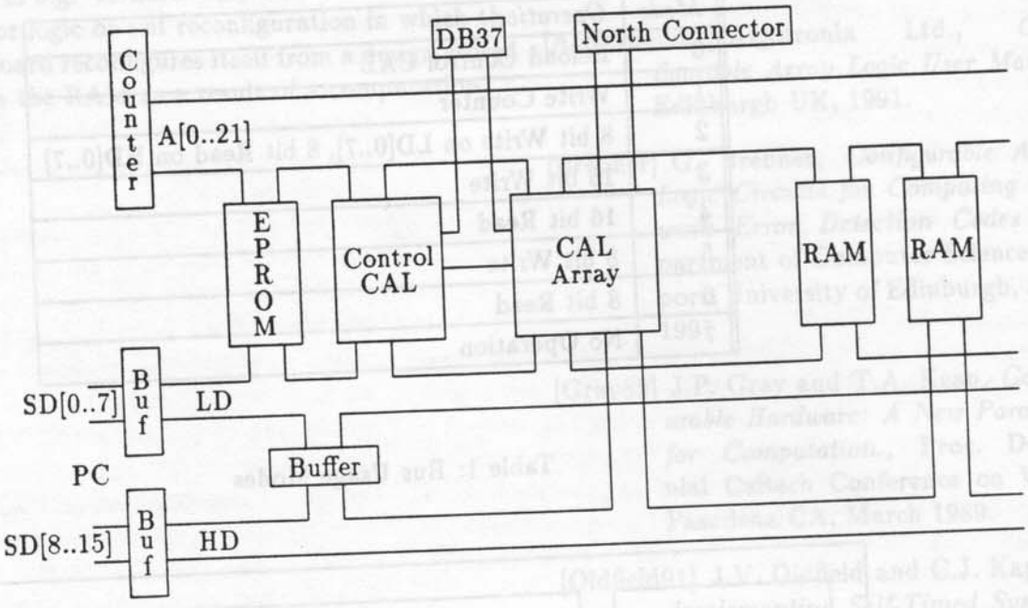

Figure 2: CHS2x4 Block Diagram

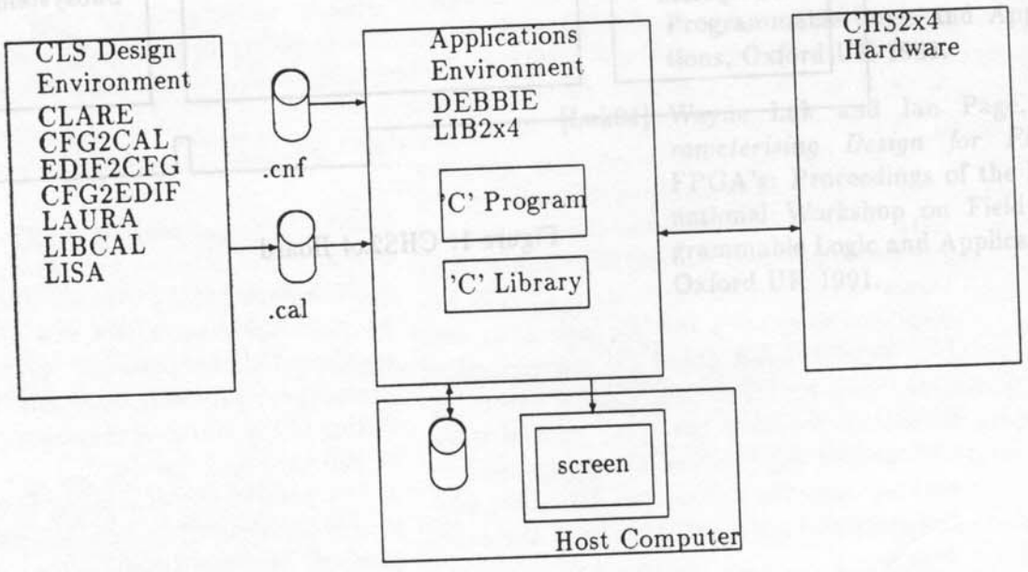

Figure 3: Software Architecture