Using CAL to Accelerate Maze Routing of CAL Designs

Tom Kean
Algotronix

April 16, 1992

Abstract

This paper describes a novel application of the dynamic reconfiguration abilities of CAL chips in which the eight CAL devices on a CHS2x4 custom computer board are used to emulate the routing available within a single CAL chip. This provides a hardware 'routing oracle' which determines almost instantaneously whether a given net can be routed and functions as an accelerator for CAD software attempting to place and route a user design on a CAL chip.

1 Introduction

CAL devices [1, 3] are composed of a rectangular array of identical cells with the same orientation and nearest neighbour connections. Each cell has a simple function unit capable of implementing all the combinational functions of two input variables or D type latches with true or inverted clock and data signals. Cells also contain a permutation network (figure 1). Non-local connections must be routed through intermediate cells. The control store for the programmable structure is randomly accessible through a byte wide parallel interface: partial reconfiguration of arrays is possible. In addition the output of cell function units is mapped into the control store and can be read back. Two global wires are provided for low skew clock distribution to cells in the array. Special pad drivers are provided which allow an input and an output signal to share a single pin using three logic levels: this reduces the pin count required for cascading the devices without reducing I/O performance by time-sharing pads. The I/O circuits can also be programmed as normal inputs/outputs or three-state bidirectional buffers to interface with TTL or CMOS logic.

The place and route subsystem of Algotronix's CAD software uses a simulated annealing algorithm to repeatedly generate and evaluate potential placements for a user's design. In the later stages of this process the computer attempts to completely route the current placement. This step of the procedure is computation intensive and dominates run times for the software. This paper presents a hardware acceleration tool for this application which runs on the CHS2x4 board.

2 The Global Routing Algorithm

The global routing problem for CAL's is to successfully route a set of nets each of which has a single source and one or more destinations using the multiplexer routing resources of the CAL chip. The nets contend for multiplexer resources and the algorithm will fail if more than one net requires a particular multiplexer to route. In this case the current placement will be rejected.

Each net is routed individually using a wavefront expansion algorithm [2 so the order in which nets are routed is of critical importance and it is usuall necessary to rip up previously routed nets and retry nets in a different orde to successfully route a design. The hardware accelerator described here an swers the question: given the present allocation of multiplexers is it possibl to complete a connection between cell (x1,y1) and (x2,y2). Figure 3 show a graphical representation of the output of the algorithm when run on th CHS2x4 PC/AT bus interface CAL design of figure 2 for two different sourc positions. The 'S' represents the source cell of the net, '.' represents reach able cells and 'U' unreachable cells which cannot be wired to because of block caused by multiplexers assigned to other nets.

3 Hardware Architecture

Figure 1 shows the input and output connections to a CAL cell. Each outpu from the cell has a dedicated multiplexer which can select any of the othe inputs to the cell or the function unit output. If we think of logic one a representing 'reachable' and logic zero as representing 'not-reachable' the we can model this routing function with a block containing an OR gate ove the four neighbour inputs whose output is connected to all four neighbou outputs. We can build an array of these blocks in one-to-one correspondenc with the array of CAL cells we are attempting to route, then if and we inser a single logic one into the array at the source point of the net of interest the logic ones will propagate to all reachable cells.

There are two remaining considerations:

1. It is necessary to force all the model circuits outputs low before insertin the single logic high. If there are any spurious logic 1's elsewhere in th circuit before starting the test then these will propagate at the same tim as the 'real' 1 potentially resulting in unreachable cells being marked a reachable. Initialisation is achieved by inserting an AND gate on th output of the OR gates with its input controlled by the global signa G1: thus when G1 is low all outputs will be forced low.

2. The model at present represents an array prior to any configuration: w need to represent multiplexers presently assigned to the user's design This is done by dynamically reconfiguring the logic within the blocks We can represent a multiplexer assigned to another net by deleting th corresponding input to the OR gate in the adjacent cell. This is don by reconfiguring the function unit from OR to X1 or X2, or from X or X2 to constant ZERO according to which input is to be deleted Similarly, inputs at the edge of the array corresponding to unused of chip connections must be deleted.

3.1 CAL Layout

The CAL layout for a single repeating block is shown in figure 4. Since it is 4 cells wide and 2 cells high we can model a single CAL chip on the 4 by 2 array of CAL chips contained on a CHS2x4 board. This layout is not optimised for density: the goal was to model a single CAL device on the board and leaving extra space made some implementation details easier.

4 Software

The software required to implement the maze routing accelerator is in two parts:

1. A simple (40 line) LAURA program builds the 32x32 array of cell models based on the basic repeating cell design above. This corresponds to the routing available in a completely unused CAL chip.

2. Run time software loads the base configuration from step 1 then dynamically reconfigures the cells within the array to remove inputs to the OR gates where multiplexers are already assigned to nets. It then uses dynamic reconfiguration to insert a logic one at the trial nets source and reads back cell outputs to determine which model cells are reachable.

5 Conclusions

The hardware architecture will propagate a signal across one of the model cells in about 30ns. This is sufficiently fast that the limitation on performance is completely within the program which accesses the board (which must calculate the memory addresses to access to add the logic one at the nets source and check whether its destinations have been reached). It is expected that integration of the hardware accelerator within the place and route system would result in substantial overall performance increases.

This paper shows the potential benefits of a general technique where FPGA logic is used to directly model a situation of interest using direct access to the control store to set up initial conditions and read back results.

References

[1] Algotronix Ltd. *The Configurable Logic Data Book.* Edinburgh UK, 1990

[2] Lee C.Y. *An Algorithm for Path Connections and its Applications* IRE Trans. Electron. Computer, vol. EC-10, September 1961.

[3] Tom Kean. *Configurable Logic: A Dynamically Programmable Cellular Architecture and its VLSI Implementation.* PhD Thesis. University of Edinburgh, Dept. of Computer Science, 1989.

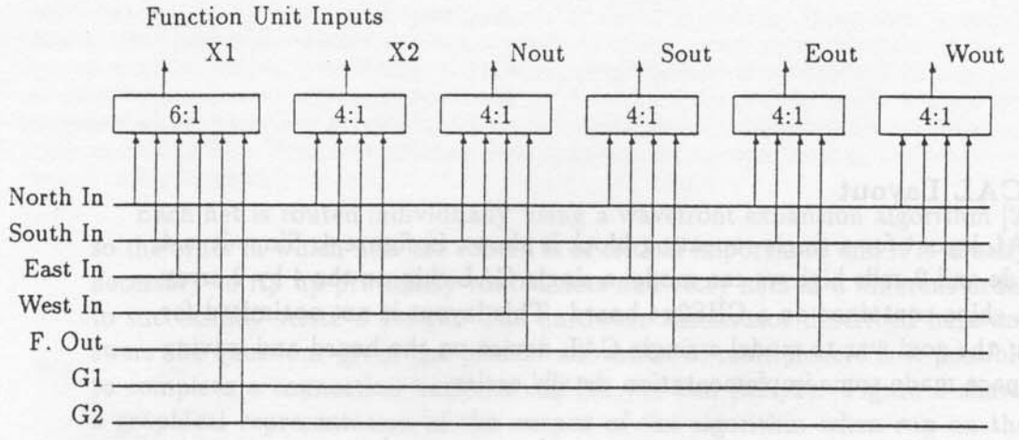

Figure 1: Cell Routing.

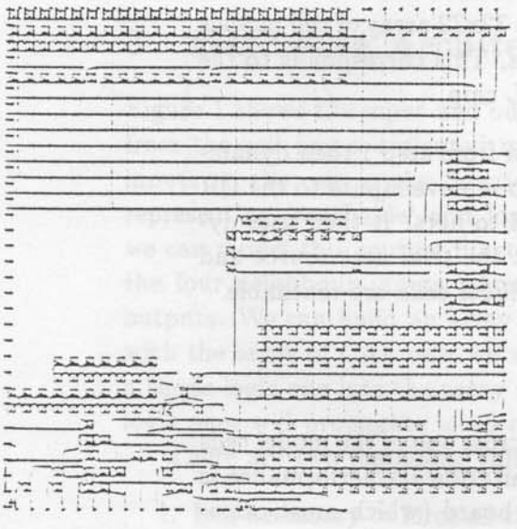

Figure 2: Control CAL Layout.

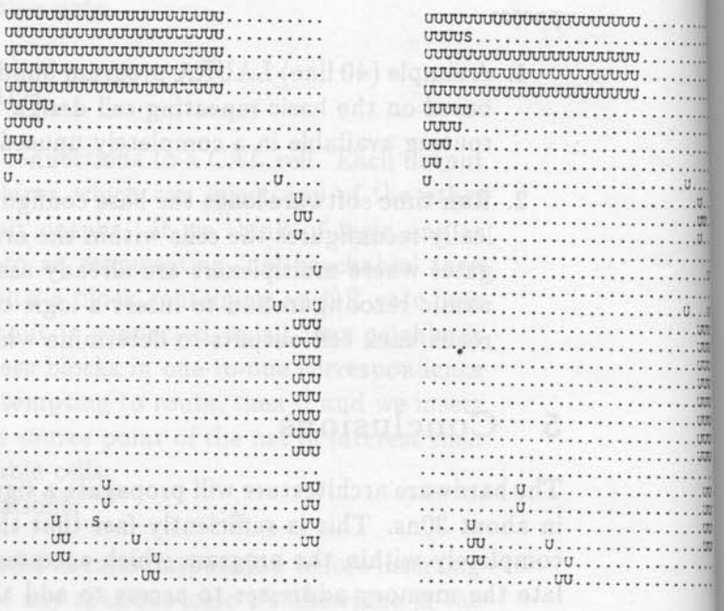

Figure 3: Reachability Data for Control CAL.

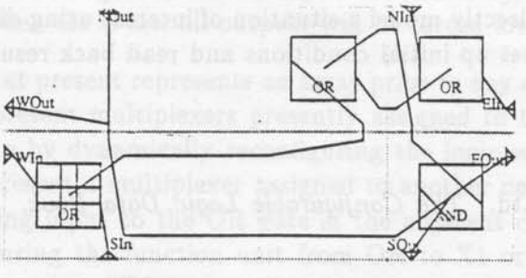

Figure 4: CAL Layout for Cell Routing Model.