DIAGONALIZATION of the CIRCUIT CONFIGURATION ARRAY

William Bricken
November 2001

CONTENTS

Diagonalization
Reconfiguration
    Spread
    Row Density

A Circuit Configuration Array (CCA) is a set of connections that define the functionality of a particular circuit. Like the logic graph structure of an ASIC, the CCA can be rearranged into different patterns of connectivity while keeping functionality invariant. The CCA pattern is defined by a dnet, a network of distinction nodes, with logical structure defined by the links.

DIAGONALIZATION

Many different CCA layout techniques are available. The simplest is a square CCA (Figure 1). Since connectivity is sparse, this wastes much of the potential computational area, while keeping infrastructure wiring and resource management to an absolute minimum.

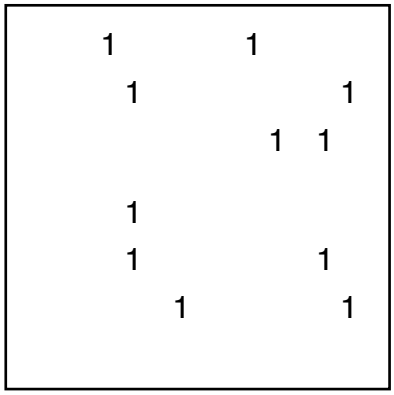

Figure 1: A Square CCA Layout

Rearranging connections can help to compress their distribution in array space, and thus reduce the area of the array required for the particular functionality. A *diagonalized configuration* restricts all connections to one-half of the array, the upper-right-triangle (Figure 2). Diagonalization is

easily achieved in software during compilation, and requires no additional infrastructure logic and wiring.

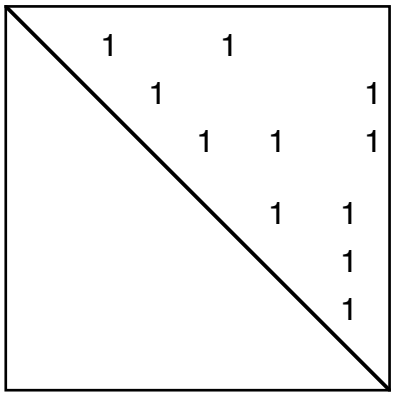

Figure 2: A Diagonalized CCA Layout

Each array mark represents a connection between two dnodes. The location of that mark identifies two coordinates, a row-distance and a column-distance, both measured from the diagonal (Figure 3). These distances are always equal. Thus, twice the row distance specifies the length of wire along which an evaluative signal must propagate (at 200ps/mm).

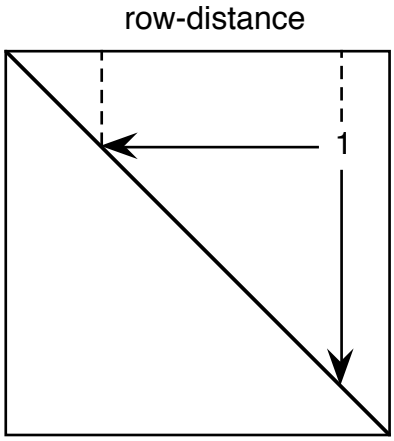

Figure 3: Wire-distance from the diagonal to a mark

RECONFIGURATION

A second step in diagonalization is to minimize the distance to the diagonal of the furthest away connections. Mathematically, minimizing this distance is the same a minimizing a topological sort of a Directional Acyclic Graph

(DAG). Since structural operators in the compiler can modify the graph which is being sorted, the minimal topological sort can vary across different graph structures.

Figure 4 shows the diagonalized configuration in two-level (SOP) form. Here, the array processing time is minimal; the trade-off is that the length of wires which propagate signals is maximal. All free array space is pushed into another triangle inside the larger upper-right-triangle.

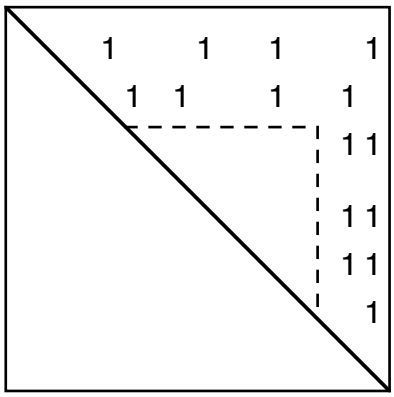

Figure 4: Two-level diagonal form

The most deeply nested form of a configuration (the Implicate Normal Form INF) has the longest evaluation path, but the shortest connecting wires. In Figure 5, marks are clustered closely to the diagonal. Free array space is pushed into another triangle, in the most upper-right corner.

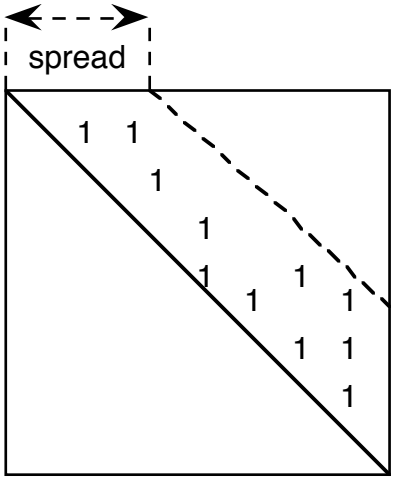

Figure 5: Deeply-nested diagonal form

Spread

The row-distance of the furthest mark is the *spread* of the INF-array. Several techniques exist for reducing the spread of an INF-array. These include:

Reduce the number of marks in a particular column (split columns)

Move a particular row or column (align internal inputs)

Change the structure of the dnet (structural rearrangement)

In all circuits, the spread can be reduced to a reasonable target distance. Spread is a function of the bit-width of the computation, and reducing some row-distances creates extra rows, so spread does have a lower bound. Here we assume a nominal spread of 140 rows.

Row Density

With .25um feature size, approximately 1000 rows fit into one mm of silicon. The length of the diagonal is 1.4 that of the side of a square array (side * sqrt 2). Similarly, the linear distance of the furthest connection from the diagonal is .7 spread (spread / sqrt 2).

Thus, when CCA hardware for N rows is reoriented, we have a total area of (rows * spread) (Figure 6):

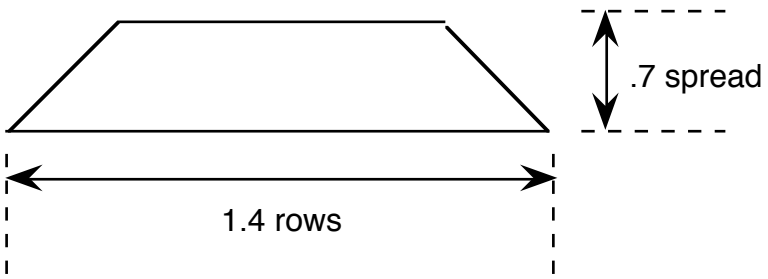

Figure 6: Reorientation of a diagonal form

We assume that the row size will be much greater than the spread. A width of 1 mm in the reoriented CCA would support 700 rows. Choose the spread to be 140 rows, making the height of the reoriented CCA .1 mm (140*.7 = 100 rows at 1000 rows/mm).

Thus one mm^2 of silicon would contain 700*10 = 7000 rows.

Each row of a CCA can contain multiple marks, making each row equivalent to an n-ary gate, where n is the number of marks. This would convert to (N-1)

two-input gates. We assume that the average circuit would permit a minimal average of 3 marks per row. This would double the effective gate size of the CCA, and is listed under ACE.

Row densities are summarized below:

feature size	*rows/mm*	*gate-equivalents/mm^2*	
		WCE	*ACE*
.25	1000	7000	14000
.18	2000	14000	28000
.13	4500	31500	63000

In general, let D be the density in rows/mm for a given feature size.

WCE gate-equivalents/mm^2 = 1/2 spread * D

A nominal 100 mm CCA chip would incur a fabrication cost of around $10. It would accommodate around 1/2 million gates, while providing easier programming and greater flexibility than current FPGAs. For $10 CoG, current PLD and FPGA technologies offer around 20K gates. The performance improvement, for the same cost, is 25:1.

Note: The CCA idea is currently abstract, so that physical size and performance estimates are not anchored in a physical implementation.