PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H03K 19/177, 19/173

A1

(11) International Publication Number: **WO 98/43354**

(43) International Publication Date: 1 October 1998 (01.10.98)

(21) International Application Number: PCT/US97/15382

(22) International Filing Date: 28 August 1997 (28.08.97)

(30) Priority Data:
08/823,265 24 March 1997 (24.03.97) US

(71) Applicant: XILINX, INC. [US/US]; 2100 Logic Drive, San Jose, CA 95124 (US).

(72) Inventors: YOUNG, Steven, P.; 6131 Paso Los Cerritos, San Jose, CA 95120 (US). NEW, Bernard, J.; 142 Stacia Street, Los Gatos, CA 95030 (US). CAMILLERI, Nicolas, J.; 6589 Boston Post Court, San Jose, CA 95120 (US). BAUER, Trevor, J.; 2252 Angela Court #5, Campbell, CA 95008 (US). BAPAT, Shekhar; 900 Pepper Tree Lane #514, Santa Clara, CA 95051 (US). CHAUDHARY, Kamal; 1775 Milmont Drive, D-105, Milpitas, CA 95035 (US). KRISHNAMURTHY, Sridhar; 4321 Verdigris Circle, San Jose, CA 95134 (US).

(74) Agents: YOUNG, Edel, M. et al.; Xilinx, Inc., 2100 Logic Drive, San Jose, CA 95124 (US).

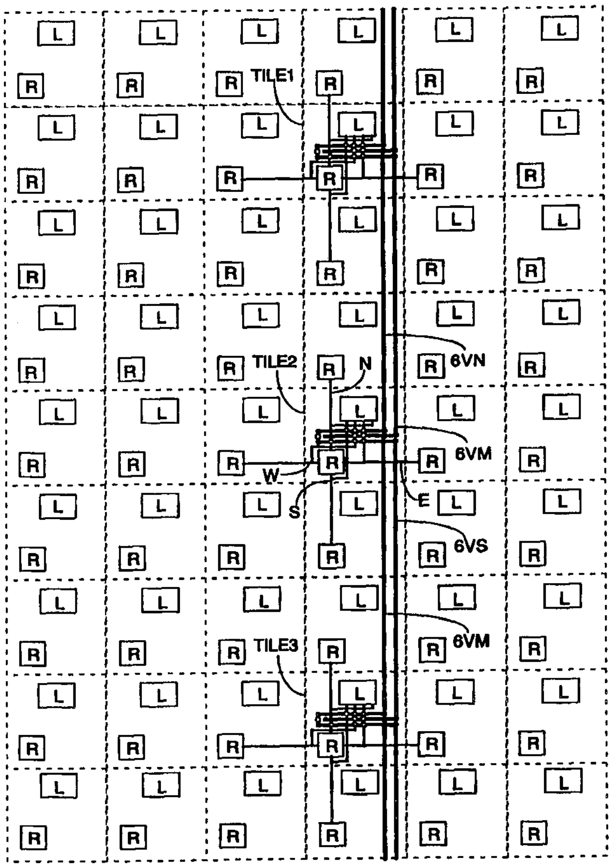
(81) Designated States: JP, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
With amended claims.

(54) Title: FPGA REPEATABLE INTERCONNECT STRUCTURE

(57) Abstract

The invention provides an FPGA interconnect structure preferably included in an array of identical tiles. According to a first aspect of the invention, a combination of single-length lines (S, N, E, W) connecting to adjacent tiles and intermediate-length lines (6VM, 6VN, 6VS) connecting to tiles several tiles away creates an interconnect hierarchy which allows any logic block to be connected to any other logic block, yet also allows for fast paths to both adjacent tiles and tiles some distance away. According to a second aspect of the invention, each tile comprises a logic block that includes a Configurable Logic Element (CLE) and an output multiplexer. Fast feedback paths are provided within the logic block to connect the CLE outputs to the CLE inputs, bypassing the output multiplexer and therefore providing faster feedback than can be obtained in most conventional FPGA logic blocks. According to a third aspect of the invention, high fanout signals can be distributed to any tile in the array.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

FPGA REPEATABLE INTERCONNECT STRUCTURE

BACKGROUND OF THE INVENTION

Field of the Invention

The invention relates to programmable integrated circuit devices, more particularly to the interconnect structure in a field programmable logic device.

Description of the Background Art

Field programmable gate arrays (FPGAs) include logic blocks connectable through a programmable interconnect structure. The interconnect structure typically provides for connecting each logic block to each other logic block. Early FPGAs accomplished this by providing short interconnect segments that could be joined to each other and to input and output terminals of the logic blocks at programmable interconnection points (PIPs). As these FPGAs become larger and more complex, the interconnect structure must also become both larger and more complex. In order to improve speed (performance), direct connections to adjacent logic blocks have been provided, and for transmitting a signal the distance of many logic blocks, longer lines have been provided. In order to save silicon area, less frequent PIPs have been provided. With fewer PIPs present, the routing is less flexible (for the same number of routing lines), but typically faster due to reduced loading. By removing only those PIPs which are least often used, routing flexibility can be minimally affected. Thus, there is a trade-off between performance, silicon area, number of routing lines, and routing flexibility.

Several U.S. Patents show such structures for interconnecting logic blocks in FPGAs. Freeman in U.S. Reissue Patent Re 34,363 describes the first FPGA interconnect structure, and includes short routing segments and flexible connections as well as global lines for signals such as clock signals. Carter in U.S. Patent 4,642,487 shows the addition of direct connections between adjacent logic blocks to the interconnect structure of Freeman.

These direct connections provide fast paths between adjacent logic blocks. Greene et al in U.S. Patent 5,073,729 shows a segmented interconnect structure with routing lines of varied lengths. Kean in U.S. Patent 5,469,003 shows a hierarchical interconnect structure having lines of a short length connectable at boundaries to lines of a longer length extending between the boundaries, and larger boundaries with lines of even longer length extending between those boundaries. Kean shows in particular lines the length of one logic block connecting each logic block to the next, lines the length of four logic blocks connectable to each logic block they pass, and lines the length of sixteen logic blocks connectable at the length-four boundaries to the length-four lines but not connectable directly to the logic blocks. In Kean's architecture, adjacent logic blocks in two different hierarchical blocks (i.e., on either side of the boundaries) connect to each other differently than adjacent logic blocks in the same hierarchical block.

Pierce et al in U.S. Patent 5,581,199 shows a tile-based interconnect structure with lines of varying lengths in which each tile in a rectangular array may be identical to each other tile. In the Pierce et al architecture, an interconnect line is part of the output structure of a logic block. Output lines of more than one length extend past other logic block input lines to which the logic block output lines can be connected. All of the above-referenced patents can be reviewed for more understanding of prior art routing structures in FPGAs.

In the interconnect structures described by Freeman and Greene et al, each path is formed by traversing a series of programmably concatenated interconnect lines, i.e., a series of relatively short interconnect lines are programmably connected end to end to form a longer path. The relatively large number of programmable connections on a given signal path introduces delay into the signal path and therefore reduces the performance of the FPGA. Such interconnect structures are called "general interconnect".

The direct connections first described by Carter and included in the architecture of Kean provide fast paths between adjacent logic blocks, but in Carter's structure general interconnect must still be used to traverse the distance between any two blocks that are not adjacent. Therefore, circuits large enough or complex enough to require interconnecting signals between non-adjacent blocks (which frequently occur) must use the general interconnect to make these connections. For short paths, general interconnect is slower than direct interconnect, because general interconnect must be connected through several PIPs, or, if long lines are used, must be buffered to accommodate long or heavily loaded signals, introducing delay. Additionally, it is inefficient in terms of silicon area to use long lines for short paths that may be traversing only a few logic blocks, since the long lines can otherwise be used for longer paths. Further, since software that implements a logic design in an FPGA typically places interconnected logic in close proximity, structures that take advantage of this placement strategy will work well with the software, resulting in shorter compilation times for routing software and more efficient circuit implementations.

Interconnect lines called "quad lines" are included in the XC4000EX FPGAs from Xilinx, Inc., and described on pages 4-32 through 4-37 of the Xilinx 1996 Data Book entitled "The Programmable Logic Data Book", available from Xilinx, Inc., 2100 Logic Drive, San Jose, California 95124, hereinafter referred to as "the Xilinx 1996 Data Book". (Xilinx, Inc., owner of the copyright, has no objection to copying these and other pages referenced herein but otherwise reserves all copyright rights whatsoever.) However, since each quad line contacts every tile that it traverses, these lines have a large number of PIPs, each of which adds RC delay.

Pierce et al provides fast paths between both adjacent logic blocks and logic blocks several tiles apart. The output lines of the Pierce et al architecture can each drive the inputs of a limited set of other logic blocks. However, the possible destinations are limited to selected logic blocks, and the interconnect lines can only access certain specific inputs of the destination logic blocks.

In each of the prior art structures recited above, each interconnect line has programmable connections to the inputs of other logic blocks. However, in the structures of Freeman, Carter, and Pierce et al, a given logic block input can be driven from either horizontal interconnect lines, or vertical interconnect lines, but not both. An alternative approach is to separate the interconnect lines from the logic block inputs by way of a routing matrix, which gives each interconnect line more flexible access to the logic block inputs. Such an architecture is used in the Xilinx XC5200 family of FPGAs from Xilinx, Inc., and described on pages 4-184, 4-185, and 4-192 through 4-196 of the Xilinx 1996 Data Book. In the XC5200 family interconnect structure, most interconnect lines entering the tile connect to a routing matrix within the tile, rather than directly to logic block inputs or outputs. Connections between pairs of interconnect lines and between interconnect lines and logic block inputs are made through lines called "tile interconnect lines" that do not leave the tile. The advantage of having an extra interconnect line in a path from the edge of a tile to the logic block in the tile is that the routing matrix is flexible but consumes a relatively small amount of silicon area. A combination of PIPs can allow access from any line entering the tile to any desired input of a destination logic block. Yet the total number of PIPs is smaller than in many other interconnect structures. The disadvantage is that getting on and off the tile interconnect lines inserts a certain amount of delay into the path for each tile traversed. This delay inhibits the fast propagation of signals through the FPGA. Tavana et al have therefore provided long lines connectable to every tile they pass and double-length lines that bypass the tile interconnect lines in one tile. These lines can be used for signals that are traversing one or more tiles without accessing the logic blocks in the traversed tiles.

Kean separates the interconnect lines from the logic block inputs using input multiplexer switches, which provide routing flexibility to the inputs.

Since the slowest signal path between logic blocks typically determines the performance of a circuit, it is advantageous to make the slowest path as fast as possible. One way to accomplish this is to design the interconnect structure such that there is a relatively uniform delay on all signal paths throughout an FPGA. In the above routing structures, a typical distribution of delays on signal paths shows a few signal paths with significantly greater delay than the average. These signal paths are typically those with large "RC trees", i.e., signal paths which traverse a resistor (such as an unbuffered PIP), then have a large capacitance on the destination side of the resistor. An interconnect structure with relatively uniform delay could be better realized if large capacitances on a signal path (e.g., longer interconnect lines) were predictably placed on the source side of the resistor, or as close as possible to the source end of the signal path.

High fanout signals have large capacitance and are often slower than low fanout signals. Prior art routing structures had high-fanout signal routing with relatively large RC delay. An interconnect structure should ideally provide high-fanout signal routing with a delay comparable to that of other signals.

It is therefore desirable to find an interconnect structure that allows: 1) uniformly fast propagation of signals, including high-fanout signals, throughout the FPGA; 2) implementation of localized circuits in non-adjacent as well as adjacent blocks using fast paths; 3) ease of use by software; 4) efficient implementation of commonly used logic functions; and 5) a high degree of routing flexibility per silicon area consumed.

One method of improving the performance of localized circuits is to provide feedback paths from the outputs of a given logic block to the inputs of the same logic block. Such fast feedback paths are useful to speed up combinational logic spanning successive function generators in the same CLE. One such feedback path is implemented in the ORCA™ OR2C FPGAs from Lucent Technologies Inc. ("ORCA" is a trademark owned by Lucent Technologies, Inc.) The ORCA logic block is described in pages 2-9 through 2-28 of the Lucent Technologies October 1996 Data Book entitled "Field-Programmable Gate Arrays", available from Microelectronics Group, Lucent Technologies Inc., 555 Union Boulevard, Room 30L-15P-BA, Allentown, PA 18103. Figure 1A shows a simplified diagram of ORCA OR2C logic block 100 with output multiplexer 101. Figure 1B shows the programmable feedback paths provided for logic block 100 of Figure 1A. The feedback paths extend from logic block 100 outputs O4, O3, O2, O1, O0 to inputs A4, A3, A2, A1, A0, B4, B3, B2, B1, B0, C0, WD3, WD2, WD1, WD0 of the same logic block. For example, one such feedback path extends from output O0 through output line 102, PIP 103, line 104, buffer 105, line 106, PIP 107, and line 108 to logic block input A0. In the ORCA OR2C device, the outputs of the output multiplexer in the logic block feed back to the logic block inputs.

A feedback path from a Configurable Logic Element (CLE), through an output multiplexer, and back into the CLE through an input multiplexer is incorporated in the XC5200 family of FPGAs from Xilinx, Inc. The XC5200 family feedback path is described in pages 4-192 and 4-193 of the Xilinx 1996 Data Book.

The ORCA OR2C and the XC5200 family have the advantage of added flexibility gained by routing feedback paths through the output multiplexer. However, this approach also has an associated speed penalty caused by the additional delay of passing through the output multiplexer.

Another feedback technique is described in pages 4-32 through 4-37 of the Xilinx 1996 Data Book. This technique is used in the XC4000EX family of FPGAs, as shown in Figure 27 on page 4-34 of the Xilinx 1996 Data Book. The feedback paths exit the XC4000EX tile on lines labeled "DIRECT" and reenter the tile on lines labeled "FEEDBACK" to complete the fast feedback paths. However, the XC4000EX CLE (labeled "CLB" in Figure 27) does not include an output multiplexer. (The term "output multiplexer" as used herein means more than two multiplexers each generating a single logic block output, where each multiplexer has as inputs more than two function generator outputs.)

Yet another feedback technique is used in the FLEX 10K™ FPGA from Altera Corporation, as disclosed in pages 31-53 of the "FLEX 10K Embedded Programmable Logic Family Data Sheet" from the Altera Digital Library 1996, available from Altera Corporation, 2610 Orchard Parkway, San Jose, CA 95134-2020. ("FLEX 10K" is a trademark owned by Altera Corporation.) In the FLEX 10K logic block, eight feedback paths are provided in a logic block with eight 4-input function generators. Therefore, it is impossible for each of the function generator outputs to simultaneously drive one input of each of the function generators. Software mapping of logic into function generators is thus complicated by the need to place logic in a particular function generator having a feedback path to related logic in another particular function generator within the same logic block. When two or more function generators are feeding a single function generator in the same logic block, placement of logic into specific function generators of the FLEX 10K logic block may be required. If a sufficiently large number of PIPs is provided in the FLEX 10K input multiplexer, this limitation can be overcome. However, the Altera solution carries an implicit trade-off between a large number of PIPs (and a resulting larger silicon area) and placement software complexity.

SUMMARY OF THE INVENTION

According to one aspect of the invention, an FPGA interconnect structure includes a combination of wiring segment lengths and connections to logic blocks such that a given logic block can be connected through fast paths both to adjacent logic blocks and to logic blocks several tiles away. In the preferred mode, the FPGA includes a two-dimensional array of identical tiles. In each tile is a logic block. Also in each tile are programmable interconnection points (PIPs) and segments of interconnect lines that adjoin segments of interconnect lines in adjacent tiles. The adjoined segments form interconnect lines extending through several tiles to PIPs in other tiles some distance away. A combination of lines connecting to adjacent tiles (called single-length lines) and lines at least three tiles long connecting a first tile to at least second and third tiles at least two and three tiles away (called intermediate-length lines) creates an interconnect hierarchy which allows any logic block to be connected to any other logic block, and yet also allows for fast paths both to adjacent tiles and to tiles some distance away. Longer interconnect lines (called long lines) may be included as a third level of hierarchy to permit efficient interconnection of widely separated tiles. Long lines can span the entire width of the tile array, or can be programmably segmented into two or more shorter long lines. In one embodiment, long lines are distinguished from intermediate-length lines in that a pattern of PIPs spanning two or more tiles is repeated along the length of the long line. When the size of the tile array is increased, more instances of the pattern occur. By contrast, intermediate-length lines according to the invention are of a predetermined length that does not alter when the size of the array is increased. When the size of the tile array is increased, additional intermediate-length lines are added.

A unique aspect of the invention is having an interconnect line (specifically an intermediate-length line or a long line) that programmably connects to logic blocks in at least three separate tiles, while extending through at least one tile in which no PIPs connect to the interconnect line. Such an interconnect line is said to be "unconnectable" to the tile in which no PIPs connect to the interconnect line and also "unconnectable" to the logic block in the tile, although connections can be made from the interconnect line to the logic block through PIPs in other tiles. A programmable connection from an interconnect line to a logic block in a given tile may be made: a) directly to a logic block input; b) through one or more PIPs connected to the interconnect line in the same tile; or c) through one or more PIPs connected to the interconnect line in the same tile and through one or more single-length lines. If any such programmable connection from an interconnect line to a logic block in a given tile can be made, the interconnect line is said to be "connectable" to the logic block and to the tile.

In a preferred embodiment of one aspect of the invention, from an originating tile an intermediate-length line connects to the tile three tiles away (i.e., separated by two tiles from the originating tile), then continues and connects to the tile six tiles away (i.e., separated by five tiles from the originating tile). This intermediate-length line (called a "hex line") does not connect to the intervening tiles one, two, four, and five tiles away. Instead, these tiles are reached indirectly by using single-length lines from the originating tile, the third tile, or the sixth tile. Connecting to only three of the seven tiles traversed by the hex line reduces the number of PIPs, and therefore reduces the silicon area required to form the interconnect line (thereby lowering the cost of the FPGA) and also reduces the capacitance added by PIPs (thereby increasing FPGA performance). Yet the combination of this intermediate-length routing, direct connections to neighboring tiles, and long lines to distant tiles allows highly flexible routing of signals.

Many modifications of the interconnect structure of this aspect of the invention are possible. One such modification is the use of asymmetrical PIP patterns on interconnect lines (intermediate-length lines or long lines) such that the interconnect line connects three logic blocks, with two of the three logic blocks being in adjacent tiles, while the third logic block is in a tile separated from the other two tiles.

As FPGAs grow larger, the amount of routing required per tile grows larger. Therefore, the silicon area required to implement the necessary PIPs tends to grow larger, and the silicon area per tile increases as the number of tiles in an FPGA increases. It is desirable to reduce the number of PIPs required per tile.

In one embodiment, only the single-length lines have connections to the logic block inputs. The intermediate-length lines have connections to each other and to single-length lines, but not to the logic block inputs. The long lines have connections to the intermediate-length lines, but not to the single-length lines or to the logic block inputs. Therefore, the number of PIPs in a tile is reduced. As with the intermediate-length lines, reducing the number of PIPs on a long line reduces both silicon area and capacitance on the long line, thereby reducing RC delay. In some embodiments, special tiles include additional connections between the various interconnect lines to facilitate distribution of high-fanout signals. In some embodiments, global lines having direct access to the logic block inputs are available for global signals such as clocks or other high fanout control signals. In some embodiments, the long lines have connections to each other in the repeatable tile instead of in special tiles.

In some embodiments, the single-length lines are driven by unbuffered PIPs and the intermediate-length and long lines are driven by buffered PIPs. Signals are typically routed on longer buffered interconnect lines first, then fanned out on shorter unbuffered lines. In this manner, large unbuffered RC trees are avoided, making delays on signal paths throughout the FPGA more uniform and improving performance. More uniform delays also make it easier to predict the performance of a circuit earlier in the design cycle.

According to a second aspect of the invention, the logic block comprises a Configurable Logic Element (CLE) and an output multiplexer. Fast feedback paths are provided within the logic block that connect some CLE outputs to some

CLE inputs, either directly or through an input multiplexer. The fast feedback paths bypass the output multiplexer and therefore provide faster connections than can be obtained in conventional FPGA logic blocks. The CLE outputs driving the fast feedback paths may be buffered or unbuffered. The fast feedback paths may be used to drive function generator inputs or other CLE inputs.

In one embodiment, the fast feedback interconnect lines are taken directly from the outputs of the function generators. In other embodiments, multiplexers are inserted to allow other signals from the CLE to use the fast feedback paths. In either case, the same outputs drive both the fast feedback paths and at least some of the inputs to an output multiplexer in the same logic block. By bypassing the output multiplexer, the feedback paths lose some flexibility but eliminate the delay that would be incurred by traversing the output multiplexer.

In one embodiment having four 4-input function generators per logic block, four fast feedback paths are provided, each of which goes to a different input of each function generator (i.e., each function generator has four inputs, each of which can be driven by a different one of the four function generators). Therefore, regardless of how the four functions are mapped into the CLE, all possible combinations of function generators feeding each other can be implemented entirely with fast feedback paths. Therefore, a variety of wide functions can be implemented using the high-performance fast feedback paths. CLE implementations comprising up to five function generators can take advantage of this aspect of the invention, with each of five function generator outputs feeding a different one of four inputs of each of the other four function generators.

According to a third aspect of the invention, high fanout signals can be distributed to any tile in the array. A signal on a horizontal long line traverses a row of tiles, in which it makes contact with the logic block in each tile through hex lines and single-length lines. The horizontal single-length lines connected to some horizontal hex lines can programmably drive vertical long lines. Using these programmable connections, the signal on the horizontal long line bus is transferred to the vertical long lines. From the vertical long lines, a high-fanout signal is delivered to an array of tiles.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1A shows a simplified diagram of the prior art ORCA OR2C logic block (100).

Figure 1B shows the programmable feedback paths from the outputs of ORCA OR2C logic block 100 to the inputs of the same logic block.

Figure 2 shows some of the single-length and intermediate-length interconnect lines according to one embodiment of the invention. Figure 2 is a simplified representation of the embodiment of Figure 3A.

Figure 3A shows a bus representation of a single tile in which intermediate-length lines are connected to every third and sixth tile (i.e., hex buses are used).

Figure 3B is a slightly modified version of Figure 3A showing a different representation of the single-length buses and switching structure 403.

Figure 3C shows a detailed representation of a single tile in the preferred embodiment. All features shown in Figure 3A are represented, and some additional features are included.

Figure 4 shows the region of Figure 3C labeled INTERCONNECT.

Figure 5 shows the output multiplexer (OMUX) of Figure 3C.

Figure 6 shows the input and output signals for the configurable logic element (CLE) of Figure 3C.

Figures 6A and 6B show the internal logic for the CLE of Figure 3C. To view the entire CLE, Figures 6A and 6B must be viewed together as shown in the key at the top of Figure 6A.

Figure 7 shows the input multiplexer (IMUX) of Figure

3C.

Figure 8 shows a routing example of a signal path using hex and single-length lines.

Figure 9 shows a routing example of a signal path using long lines, hex lines, and single-length lines.

Figure 10 shows a routing example of a high fanout control signal delivered to a column of tiles.

Figure 11 shows a routing example of a high fanout function generator input signal delivered to a column of tiles.

Figure 12 shows an example of how to combine the interconnect of Figures 10 and/or 11 to deliver a high-fanout signal to an array of tiles.

Figure 13 is a simplified representation of a portion of a logic block in the embodiment of Figure 3C, showing the fast feedback paths from the outputs of a given CLE to the inputs of the same CLE.

DETAILED DESCRIPTION OF THE DRAWINGS

The following drawing conventions are used throughout the figures. A small solid black dot at the intersection of two lines indicates a permanent electrical connection between the crossing lines. An open circle enclosing an intersection between two lines, or separating a single line into two line segments, indicates a programmable bidirectional connection between the lines or line segments (for example, a pass transistor which is turned on to make the connection). An open triangle at an intersection of two lines indicates a programmable connection with signal flow going onto the line pointed to by the apex of the triangle. (The signal is of course then present on the full length of the line. Thus, a triangle pointing in the opposite direction would have the same signal flow because the triangle points to the same line.) A triangle showing a connection between two buses indicates that at least one connection exists between the lines making up the buses. In accordance with one embodiment of the present invention, programmable connections are provided by using programmable interconnection points (PIPs), wherein each PIP includes at least one transistor.

A line which ends within a tile or block structure (i.e., does not extend to the border of a tile or block) is physically terminated within the tile or block. A line which extends to the border of the tile connects to a line on the next tile, which it contacts when two tiles are abutted together. Note that some lines which extend to an edge of a tile and thus into an adjacent tile change names at the tile boundary.

Simplified Representation of the Inventive Concept

Figure 2 shows in simplified form some of the single-length and intermediate-length interconnect lines according to one embodiment of the invention. Figure 2 illustrates part of an array of tiles in an FPGA. In each tile is logic block L and a routing or switching structure R. Also in each tile are line segments which connect to line segments in adjacent tiles to form interconnect lines. Most of the interconnect lines in the FPGA are not shown in Figure 2 so as not to obscure the connectivity of the invention. Three of the tiles, labeled TILE1, TILE2, and TILE3 are shown in more detail, and short- and intermediate-length interconnect lines extending from TILE1, TILE2, and TILE3 are shown. (However, intermediate-length lines extending horizontally are not shown.) Single-length lines N, S, E, and W extend north, south, east, and west, respectively, from TILE2 to the four neighboring tiles. Equivalent lines not labeled extend from TILE1 and TILE3 as well. Identical structures are also present in all other tiles but are not shown in Figure 2. These single-length lines N, S, E, and W extend between switching structures R in neighboring tiles; and have programmable connections to logic block L.

Intermediate-length lines also extend north, south, east and west from each tile to connect to other tiles three and six tiles away. Therefore in this embodiment the intermediate-length lines are "hex lines" as previously defined. Only the vertically extending (north and south) hex lines are shown, to avoid obscuring the connections. (Later figures show the complete connection pattern.)

From TILE2, hex line 6VN extends to the north a length of six tiles, connecting to TILE1 three tiles to the north and to another tile (not shown) six tiles to the north. Also from TILE2, hex line 6VS extends to the south six tiles, connecting to TILE3 and to another tile (not shown) six tiles to the south. Hex line 6VM connects tile TILE2 to tiles TILE1 and TILE3. Thus, three hex lines extend vertically and connect to TILE2. Also in TILE2, line 6VS is programmably connectable to line 6VN, as indicated by the PIP joining the ends of these two lines. In this manner an easy connection is available to a tile twelve tiles away from the starting tile. Only one additional connection is needed to extend the connection to another interconnect line six tiles long.

Note that within TILE2, only the single-length lines N, S, E, and W have access to logic block L. Thus a signal on a hex line must transfer to a single-length line to access logic block L in the destination tile. A signal entering TILE2 on line 6VS can connect to lines W, N, or E and thereby to logic block L in TILE2. Further, the signal now on single-length lines W, N or E can also access the logic block L in the tiles to the west, north or east of TILE2. In this embodiment, no connection from hex line 6VS to single-length line S is provided, as eliminating this connection saves silicon area and the tile to the south of TILE2 can be accessed through another path. In the embodiment of Figure 2, a signal in TILE3 can reach the logic block of the tile to the south of TILE2 either by: 1) traversing two single-length lines to the north; or 2) taking hex line 6VM to TILE2, connecting to single-length line W or E, then entering switching structure R in TILE2 and "bouncing" (moving onto and off of an interconnect line in the same tile) from single-length line W or E to single-length line S inside the switching structure. The second alternative has one more PIP in the signal path than the first alternative, the PIP in switching structure R. In another embodiment, a connection to the south can of course be provided.

Repeatable Tile Representation

Figure 3A more completely illustrates a single tile. Figure 3A shows bus segments of various widths, as indicated by diagonal slash lines having a number indicating the bus width. The number of lines in each bus segment can of course vary in different embodiments. In the embodiment of Figure 3A, the tiles are identical. Since the illustration is of a single tile, complete interconnect lines must be understood by realizing that bus segments extending to the top of Figure 3A will contact bus segments at the bottom of another copy of Figure 3A placed adjacent to the north. Similarly, bus segments extending to the right of Figure 3A will contact bus segments of another copy of Figure 3A placed adjacent to the right, and so forth. An array such as present in an FPGA is represented by placing many copies of Figure 3A together.

Hex Buses

In Figure 3A, 12-segment bus 6N extends horizontally through programmable region PROG-2, then turns north to become bus segment 47. Bus segment 47 extends to the top edge of the tile. Another copy of Figure 3A placed adjacent and to the north of Figure 3A will have a bus segment 46 extending to the bottom of the copy that contacts bus segment 47. Bus segment 46 extends vertically through the tile, but is offset right by one bus before it reaches the top of the tile, such that bus segment 46 in a given tile contacts bus segment 45 in the tile adjacent to the north. Similarly, at the tile boundary, bus segment 45 contacts bus segment 44, bus segment 44 contacts bus segment 43, bus segment 43 contacts bus segment 42, and bus segment 42 contacts bus segment 41. This offset arrangement results in a hex length bus extending through six adjacent tiles. Bus segment 41 extends into the tile from the bottom, then turns west and extends horizontally through programmable region

PROG-2 as bus segment 6S. Bus segment 6V also extends horizontally through programmable region PROG-2, and connects to bus segment 44. Each tile therefore includes three bus segments 6N, 6V, and 6S extending into programmable region PROG-2. These bus segments enable the connections from a given tile to tiles three and six tiles away, which makes these intermediate-length lines hex lines as defined above. Bus segments 41, 42, 43, 44, 45, 46 and 47 together form a 72-line vertical "routing track" designated 6VN at the north end and 6VS at the south end of Figure 3A. Bus segments 42, 43, 44, 45, and 46 are called "interior segments", and bus segments 41 and 47 are called "end segments".

Horizontal 12-segment hex buses are similarly provided, each tile having portions 6W, 6H, and 6E extending into programmable region PROG-2. Bus segment 6E extends from programmable region PROG-2, then turns east and extends as bus segment 37 to the east edge of the tile. Bus segment 37 of a given tile contacts bus segment 36 in the adjacent identical tile to the east. Similarly to bus segments 41-47, bus segments 31-37 provide a hex length bus extending through six adjacent tiles. Bus segment 31 extends into programmable region PROG-2 as bus segment 6W. Bus segment 34 extends into programmable region PROG-2 as bus segment 6H. Bus segments 31, 32, 33, 34, 35, 36 and 37 together form a 72-line horizontal routing track designated 6HW at the west end and 6HE at the east end of Figure 3A.

Note that 72-line hex buses 6VN, 6VS, 6HW, 6HE are not typical buses in that the hex lines making up the buses do not all start in the same tile and end in the same tile, i.e., there is no fixed boundary for the bus. Twelve hex lines end, and another twelve hex lines begin, in each tile. This feature, among others, distinguishes the inventive structure over that of Kean and other hierarchical interconnect structures. In the hierarchical interconnect structures, an FPGA is divided hierarchically with periodic routing structures at the hierarchical boundaries. The present invention uses identical tiles, each of which tiles includes end segments of some of the lines.

Other Buses

North, south, east, and west single-length buses (N, S, E, and W respectively), each 24 lines wide, are also provided. These single-length buses connect to each other within switching structure 403 located in programmable region PROG-2. In Figure 3A, single-length buses S, W and E are shown entering switching structure 403 in two positions, in order to correspond to Figure 4. However, the bus segments labeled S,W, and E pass continuously through switching structure 403. The way these bus segments pass through switching structure 403 is clarified in Figure 3B, which shows a different representation of switching structure 403, but is otherwise the same as Figure 3A.

In this embodiment, each single-length line can be connected to the input multiplexer IMUX-2. Each single-length line can also be connected to some of the hex lines.

Figure 3A includes another feature: horizontal and vertical long line buses. Each of the two long line buses is 12 lines wide. Two-line buses 437P, 438 (corresponding to lines LV0 and LV6, LH0 and LH6, respectively, in Figures 4 and 7) taken from each long line bus extend into programmable region PROG-2 where they can be programmably connected to drive hex lines. The two accessed long lines in each long line bus are separated from each other by six lines, and each long line is accessed every sixth tile. Two-line buses 437I, 438 taken from each long line bus extend from input multiplexer IMUX-2, which can provide signals to the long lines. (Two-line bus 437I includes the same two signals as two-line bus 437P.) The horizontal long line bus is designated LHW at the west end and LHE at the east end of Figure 3A. LHW and LHE extend through an offset region 422 (shown in Figure 3C) which offsets the bus by one line per tile. In this manner, each identical tile accesses two different lines of horizontal long line bus LHW/LHE, and each long line is accessed every sixth tile. Similarly, the vertical long line bus is designated LVN at the north end and LVS at the south end of Figure 3A. LVN and LVS extend through an offset region 421 similar to offset region 422. The long line buses extend the entire length or width of the tile array.

Another bus extends horizontally through the programmable region PROG-2, a tristate buffer bus, 4 lines wide, designated TW at the west end and TE at the east end of Figure 3A. TW and TE are connected by programmable offset region 423. Programmable offset region 423 is similar to offset regions 421 and 422 in that it offsets the bus by one line; however, programmable offset region 423 is only four lines wide and also includes one programmable bidirectional PIP for extending the length of bus TW or TE. (The internal structure of programmable offset region 423 is shown in Figure 3C.) This PIP can be used either to "break" each line of the tristate bus every fourth tile, or to connect lines together every fourth tile to continue a tristate line up to the full width of the tile array.

Programmable Bus Connections

Programmable bus connections are located in region INTERCONNECT-2, which is part of programmable region PROG-2. Where hex buses cross single-length buses, open triangles indicate programmable connections in which one of the hex lines at the base of the triangle can drive one of the single-length lines at the tip of the triangle. Triangles pointing to hex lines represent buffered connections, and triangles pointing to single-length lines represent unbuffered connections such as simple pass transistors.

Note that not all intersections have triangles. For example, hex bus segment 6V can connect to east and west single-length buses E and W, but not to north and south buses N and S. Similarly, hex bus segment 6H can connect to north and south single-length buses N and S, but not to east and west buses E and W. Hex bus segment 6S coming from the south can connect to east and west single-length buses E and W and also to north bus N. This partial connectivity saves silicon area while still providing useful connections for good routability and performance. Similarly, hex bus segment 6N coming from the north can connect to east and west single-length buses E and W and also to south bus S; hex bus segment 6W coming from the west can connect to north and south single-length buses N and S and also to east bus E; and hex bus segment 6E coming from the east can connect to north and south single-length buses N and S and also to west bus W. In each of these cases, the connection may be limited by hardware or software to being unidirectional such that the hex buses can drive the single-length buses, but the single-length buses cannot drive the hex buses. (In the actual hardware of this embodiment, the connection is a bidirectional pass-gate; whereby the single-length buses can drive the hex buses. However, doing so reduces the speed of the resulting signal path dramatically, and therefore software for programming the FPGA does not select such a path.)

All of the single-length buses can be driven by output multiplexer OMUX-2 by way of output bus OUT. Hex bus segments 6E, 6W, 6N, and 6S can also be driven by output bus OUT.

Vertical hex bus segments 6N and 6S can be connected together via programmable bus connection 404. (The internal structure of programmable bus connection 404 is shown in Figure 4.) In this fashion, a line in a vertical hex bus can be continued 12, 18 or more tiles, rather than terminating after 6 tiles. Similarly, horizontal hex bus segments 6W and 6E can be connected together via programmable bus connection 405. Additionally, vertical hex bus segments 6N and 6S can be connected to horizontal hex bus segments 6W and 6E via programmable bus connections 406, 407, 408, and 409. The connectivity of these bus connections is shown in Figure 4. Note also that hex bus segment 6V (which implements the connection to the third tile in the vertical direction) can drive horizontal hex bus segments 6W and 6E. Similarly, hex bus segment 6H (which implements the connection to the third tile in the horizontal direction) can drive vertical hex bus segments 6N and 6S. By this means, the direction of a signal path is easily changed between horizontal and vertical.

Horizontal hex bus segments 6E and 6W in a given tile can be driven by the two horizontal long lines accessed in the same tile. Vertical hex bus segments 6N and 6S in a given tile can be driven by the two vertical long lines accessed in the same tile. This capability allows a signal on a long line to fan out to an entire row or column of tiles with high speed, as later shown in Figures 10-12, by routing from the long line to each hex line (and hence to single lines) rather than propagating serially along the hex lines.

Switching structure 403 allows each single-length bus (N, S, E, W) to be connected to each of the other single-length buses. The connectivity in switching structure 403 is shown in Figure 4.

Significance of Programmable Bus Connections

The bus connections in Figure 3A implement an interconnect structure for FPGAs according to the invention. The buses of this embodiment interact in the following ways: 1) long lines can drive hex buses parallel thereto, but cannot drive either perpendicular hex buses or single-length buses except through the parallel hex buses; 2) hex buses can drive single-length buses both parallel and perpendicular thereto, but single-length buses cannot drive hex buses (although in one embodiment hex buses and single-length buses are connected through bidirectional PIPs, so a software limitation is imposed to make these connections unidirectional); 3) hex buses can further drive other hex buses both parallel and perpendicular thereto; and 4) most hex buses cannot connect to the inputs of input multiplexer IMUX-2, although a limited number of lines from the hex buses can make such connections, as is later described. Therefore, in the interconnect structure of Figure 3A, it is preferred to make connections between tiles by placing a signal first on the longest suitable line and from there moving the signal to a shorter line. For example: 1) a connection to a logic block three or six tiles away would be made through a hex line, switching to a single-length line to reach the input multiplexer; 2) a connection to a logic block seven tiles away would be made through first a hex line and then a single-length line (this connection requires no more PIPs than the first example); 3) a connection to a logic block on the other side of the FPGA would be made through first a long line, then a hex line, then transferred to one or more single-length lines before entering the destination input multiplexer. In this manner, large RC trees are avoided, by ensuring that unbuffered PIPs do not drive heavily loaded lines. Further, the various types of signal paths throughout the interconnect structure have similar delays, increasing the overall performance of the circuit.

Of course, many signal paths do not travel only within one column or one row of tiles. To turn a corner, a signal path can move from a given single-length line to a perpendicular single-length line through the switching structure 403, or from a given hex line to a perpendicular hex line.

Programmable Logic Portion of the Tile

The interconnect portion INTERCONNECT-2 of the repeatable tile of Figure 3A exists only for the purpose of delivering signals between the logic blocks in different tiles. The logic block LB of this embodiment comprises a configurable logic element CLE-2, an output multiplexer OMUX-2, an input multiplexer IMUX-2, and two tristate buffers 445. (The number of tristate buffers can of course vary in other embodiments of the invention.) Configurable logic element CLE-2, input multiplexer IMUX-2, and output multiplexer OMUX-2 in Figure 3A have small solid black triangles on each input and output. The direction of the apex of the triangle indicates the direction of the signal flow. Thus, a triangle with its apex inside the rectangular box indicates an input line or bus and a triangle with its apex outside the rectangular box indicates an output line or bus.

Input multiplexer IMUX-2 of this embodiment accepts inputs from: 1) each of single-length buses N, S, E, W; 2) four lines 428 of each 12-line vertical hex bus 42, 43, 44, 45, 46, 47; 3) four lines 429 from 8-line output bus OUT from output multiplexer OMUX-2; and 4) a total of four fast-feedback lines 430 from configurable logic element CLE-2. Input multiplexer IMUX-2 can programmably create the following output signals: 1) two 13-line buses 436 that provide input signals to configurable logic element CLE-2; 2) two signal lines 431 that drive the data inputs of tristate buffers 445; 3) two signal lines 432 that enable tristate buffers 445; 4) two lines of vertical long line bus LVS; and 5) two lines of horizontal long line bus LHE.

Configurable logic element CLE-2 accepts only inputs from input multiplexer IMUX-2, and drives 12 output lines shown in Figure 3A as two 6-line buses 433. Four of these output lines form fast-feedback paths 430 to input multiplexer IMUX-2. Outputs 433 from configurable logic element CLE-2 drive output multiplexer OMUX-2. OMUX-2 gets one additional input 434 from tristate buffer bus TW, and drives 8-line output bus OUT. Each of the two tristate buffers 445 drives two of the tristate lines in tristate buffer bus TW via 2-line bus 435. The connection of tristate buffers 445 to tristate buffer bus TW is shown in more detail in Figure 3C.

Detailed Repeatable Tile Representation

Figures 3C-12 show the preferred embodiment of a repeatable interconnect structure according to the invention, which embodiment is similar to the embodiment of Figure 3A. Figure 3C shows a detailed representation of a single tile in the preferred embodiment. All features shown in Figure 3A are represented, and some additional features are included. Labels in Figure 3C correspond to the same labels in Figure 3A; thus corresponding structures are easily identified. There is one difference in representation: in Figure 3C, 72-line vertical routing track

6VN/6VS of Figure 3A is represented as two 36-line vertical routing tracks, 6RN/6RS on the right and 6LN/6LS on the left. Half of the lines of each 12-line hex bus are drawn on each side of programmable region PROG, which corresponds to programmable region PROG-2 in Figure 3A. This representational difference has no effect on the functionality of the tile. In Figure 3C, the horizontal and vertical hex buses are shown with first lines of all buses are grouped together. Only the first lines are labeled. For example, labels 41 through 47 represent the first lines of hex bus segments 41 through 47 of Figure 3A. Hex bus segments 31 through 37 of Figure 3A are similarly represented, but are not labeled in Figure 3C.

Interconnect region INTERCONNECT of Figure 3C is functionally the same as interconnect region INTERCONNECT-2 of Figure 3A. Features in the embodiment of Figure 3C that are not shown in Figure 3A include: 1) horizontal four-line global bus G along the top of the tile provides four global inputs to input multiplexer IMUX, which global inputs can be used to distribute high-fanout or high-speed global clock signals; 2) two direct output connection lines DOW from output bus OUT (driven by output multiplexer OMUX) exit the tile of Figure 3C on the west edge, where they are connected to input lines DIE of an adjacent tile to the west; 3) two direct input connection lines DIE from the adjacent tile to the east enter the tile of Figure 3C on the east edge and go to input multiplexer IMUX; 4) two direct output connection lines DOE from output bus OUT exit the tile of Figure 3C on the east edge, where they are connected to lines DIW of an adjacent tile to the east; 5) two direct input connection lines DIW from the adjacent tile to the west enter the tile of Figure 3C on the west edge and go to input multiplexer IMUX; and 6) two carry chains are included in configurable logic element CLE, each carry chain having an input on the south edge of the tile and an output on the north edge of the tile, the inputs designated CIN0 and CIN1 and the outputs designated COUT0 and COUT1, respectively. Some embodiments of the interconnect structure according to the invention do not have these features.

Interconnect Region

Figure 4 shows interconnect region INTERCONNECT of Figure 3C in detail. All input and output lines are located in the same relative positions in Figures 3C and 4. Input and output lines to interconnect region INTERCONNECT in Figure 3C can be identified by looking at the corresponding labels in Figure 4. With the exception of the division of the vertical routing track, as described above, Figure 4 also corresponds directly to the interconnect region INTERCONNECT-2 of Figure 3A. Numerical labels identify features of the implementation previously shown and described.

Programmable bus connections 404 and 405 in Figure 4 demonstrate an additional feature of the invention. In programmable bus connections 404 and 405, vertical and horizontal hex buses (respectively) are programmably connected together to enable the continuation of a signal path in the same direction from one hex bus to the next. As shown in Figure 4, some of these connections can be bidirectional (for example, implemented as pass transistors) while others are unidirectional (such as buffered connections). In one embodiment, the bidirectional hex buses (hex buses that can be programmably driven from either end) have more programmable connections to perpendicular buses than do the unidirectional hex buses (hex buses that can be programmably driven from only one end). Such bidirectional connections will be most flexible, while such unidirectional connections will be faster, since the PIP can be implemented as a single buffer without a programmable enable.

Vertical long lines LV0 and LV6 identify the first and seventh lines, respectively, of vertical long line bus LVS that are accessed in each repeatable tile, as previously described. Similarly, horizontal long lines LH0 and LH6 identify the first and seventh lines, respectively, of the horizontal long line bus LHE.

As previously described, hex bus segments 6N, 6V, and 6S of Figure 3A are divided in Figure 3C into hex bus segments 6NR, 6VR, and 6SR, which exit the INTERCONNECT region on the east edge to form 36-line vertical routing track 6RN/6RS of Figure 3C, and hex bus segments 6NL, 6VL, and 6SL, which exit the INTERCONNECT region on the west edge to form 36-line vertical routing track 6LN/6LS of Figure 3C.

Output Multiplexer OMUX

Figures 5, 6 and 7 show the output multiplexer (OMUX), configurable logic element (CLE) and input multiplexer (IMUX), respectively, of the embodiment shown in Figure 3C. All input and output lines are physically located in the same relative positions in Figures 3C, 5, 6 and 7, thus input and output lines may readily be traced between the figures. The PIP patterns shown in Figures 5 and 7 form only one of many possible patterns. The implementation of these three blocks, and the connections between the blocks, may differ widely in different embodiments of the invention, provided that the three blocks are compatible with each other. For example, the number of connections between the blocks may be increased or decreased. Further, two or more of these blocks may be combined, or one or both of the input or output multiplexers may be eliminated. Although these variations are not further described herein, such modifications are encompassed by the inventive concept. It will be apparent to one skilled in the art after reading this specification that the present invention may be practiced within these and other architectural variations.

Figure 5 shows details of the interior of output multiplexer OMUX. As shown in Figure 5, the input lines (the outputs of logic block CLE) are selectively programmed to drive a series of eight outputs (OUT0 to OUT7). Not all inputs can drive all outputs; a selection of how many PIPs to provide and where they should be located is typically made based on the trade-off constraints of performance, silicon area, and routing flexibility.

Configurable Logic Element CLE

Figure 6 shows the input and output signals for the configurable logic element (CLE) of the embodiment shown in Figure 3C. Many configurable logic elements can be used with the interconnect structure of the invention. For example, pages 4-11 through 4-23 of the Xilinx 1996 Data Book describe a configurable logic element used in the XC4000 Series FPGA architecture which is compatible with the invention. Pages 4-294, 4-295, and 13-13 through 13-15 of the same document describe another compatible configurable logic element, used in the XC3000 Series FPGA architecture. Pages 4-188 through 4-190 of the same document describe yet another compatible configurable logic element, used in the XC5200 Family FPGA architecture.

One CLE that can be used with the embodiment of Figure 3C is shown in Figures 6A and 6B. In this embodiment, the CLE is implemented in two similar portions called "slices". To view the entire CLE, Figures 6A and 6B, each showing one slice, must be viewed together as shown in the key at the top of Figure 6A. Programmable (configurable) bits controlling the function of the CLE are shown as small boxes containing an "x". The CLE of Figures 6A and 6B includes four function generators: F and G in Figure 6B, and H and J in Figure 6A. Function generator F has four data inputs F1, F2, F3, F4; function generator G has four data inputs G1, G2, G3, G4; function generator H has four data inputs H1, H2, H3, H4; and function generator J has four data inputs J1, J2, J3, J4. Each of function generators F, G, H, J also has a data-in input DI and an output O. In addition, each function generator F, G has a write strobe WS input WSF, WSG, respectively, generated in write strobe generator WB from clock input signal CK0 optionally inverted by multiplexer 69B, set/reset input signal SR0 optionally inverted by multiplexer 60B, and input signal BF optionally inverted by multiplexer VF. Each function generator H, J has a write strobe WS input WSH, WSJ, respectively, generated in write strobe generator WA from clock input signal CK1 optionally inverted by multiplexer 69A, set/reset input signal SR1 optionally inverted by multiplexer 60B, and input signal BH optionally inverted by multiplexer VH.

Function generators F, G, H, J of Figures 6A and 6B have the ability to operate as shift registers as well as small RAMs and lookup tables. Write strobe signal WS controls the shifting function and the RAM write cycle. Data-in port DI carries a data signal (in this embodiment derived from input signals BF, BH, BG, BJ) for writing to an addressed memory cell. When function generators F, G, H, J are configured as shift registers, the shift register data input signal is taken from BF, BG, BH, BJ, respectively. Write strobe generators WB, WA generate write strobe signals WSF and WSG, WSH and WSJ, respectively. Signals WSF, WSG, WSH, WSJ strobe data into their respective function generators during a write cycle when the function generators are configured as RAM, and cause a shift when the function generators are configured as shift registers. The use of function generators as RAM is described in pages 4-11 through 4-23 of the Xilinx 1996 Data Book.

The function generator of this embodiment can therefore be configured as a look-up table, a shift register, a 16x1 RAM, half of a 16x1 dual-ported RAM (when the two function generators in a slice are paired together), half of a 16x2 RAM, or half of a 32x1 RAM (when the two function generators in a slice are paired together). Not all combinations of function generator configurations are supported in a single slice. The supported combinations are detailed below.

In this embodiment, write strobe signals WSF and WSG are controlled by write strobe generator WB. A High (logic 1) pulse on function generator input WS causes a write to occur to the function generator look-up table, to either the first memory cell of the look-up table (if configured as a shift register) or to an addressed cell (if configured as a RAM).

Write strobe generator WB functions as follows in the supported combinations of function generator configurations: a) when function generators F and G are both configured as look-up tables, write strobe signals WSF and WSG are held

Low (logic 0); b) when function generator G is configured as a 16x1 RAM or shift register and function generator F is configured as a look-up table, write strobe signal WSF remains Low while write strobe signal WSG pulses High when SR0 is active (Low or High, depending on whether multiplexer 60B is inverting or not) and CK0 makes the proper transition (falling or rising, depending on whether multiplexer 69B is inverting CK0 or not); c) when function generators F and G are configured as a dual-ported 16x1 RAM, a 16x2 RAM, or a pair of shift registers, write strobe signals WSF and WSG pulse High together when SR0 is active and CK0 makes the proper transition; d) when function generators F and G are configured as a single 32x1 RAM, one but not both of write strobe signals WSF and WSG pulses High, when SR0 is active and CK0 makes the proper transition. The generated High pulse occurs on write strobe signal WSF if multiplexer VF is non-inverting and input signal BF is High or if multiplexer VF is inverting and input signal BF is low, otherwise the High pulse occurs on write strobe signal WSG. As can be seen from the above description, SR0 acts as a write enable and CK0 acts as a clock to a synchronous RAM or shift register, while BF serves as a fifth address bit when the two function generators in a slice are configured as a 32-bit RAM.

Write strobe signals WSH and WSJ are generated by write strobe generator WA and controlled by input signals SR1, CK1, and BH, in a corresponding fashion.

AND gates 61F, 61G, 61H, 61J and multiplexers 81F, 81G, 81H, 81J, CF, CG, CH, CJ implement multiplier and carry logic functions. In Figures 6A and 6B, each of multiplexers 81F, 81G, 81H, 81J has separate programmable memory cells that determine the multiplexer output. In another embodiment, the two multiplexers in each slice (81F and 81G, 81H and 81J) share memory cells and therefore have the same programmed functionality.

In the embodiment of Figures 6A and 6B, each slice has a separate carry chain. One carry chain is from input signal CIN0 to output signal COUT0 and incorporates the outputs of function generators F and G. The other carry chain is from input signal CIN1 to output signal COUT1 and incorporates the outputs of function generators H and J.

Each slice further includes five-input-function multiplexer F5A, F5B and six-input-function multiplexer F6A, F6B. In Figure 6A, multiplexer F6A receives an input signal on west-going line F5W from the output of multiplexer F5B in Figure 6B. In Figure 6B, six-input-function multiplexer F6B receives an input signal on east-going line F5E from the output of multiplexer F5A in Figure 6A. Multiplexer F6A receives a second input signal from multiplexer F5A, and multiplexer F6B receives a second input signal from multiplexer F5B. Multiplexer F5A receives inputs from outputs O of function generators H and J. Multiplexer F5B receives inputs from outputs O of function generators F and G.

The CLE of this embodiment receives four extra control signals on lines BF, BH, BG, and BJ. These lines can be programmably inverted using multiplexers VF, VH, VG, and VJ, respectively, and control several useful functions. The five-input function multiplexers F5A, F5B are controlled by lines BH, BF, respectively. The six-input function multiplexers F6A, F6B are controlled by lines BJ, BG, respectively. A user can form a four-input multiplexer using multiplexer F5A and the outputs of function generators H and J, with function generators H and J each configured as a two-input multiplexer. Similarly, a four-input multiplexer can be formed using multiplexer F5B and the outputs of function generators F and G. A user can form an eight-input multiplexer using either multiplexer F6A or multiplexer F6B receiving inputs from both multiplexers F5A and F5B, each of which is receiving outputs from its respective function generators. Further, with multiplexer F5A, the memory cells in function generators H and J can operate as a 32-bit RAM. With multiplexer F5B, the memory cells in function generators F and G can operate as a 32-bit RAM. Lines BF, BH, BG, BJ serve as multiplexer control lines or RAM address lines, in addition to other functions as later described.

Registers RX, RY, RZ, RV generate outputs XQ, YQ, ZQ, VQ, respectively. Clock inputs to registers RX, RY, RZ, RV are designated by open triangles with apexes pointing into these registers. Clock enable inputs CE can be disabled using AND-gates 62A, 62B. (AND-gates 62A, 62B are controlled by a programmable memory cell that provides one of the inputs to each AND-gate.) Registers RX, RY, RZ, RV can be set or reset either synchronously or asynchronously through set port S or reset port R. (The choice of synchronous or asynchronous is made by a programmable memory cell shared between the two registers in each slice.) Set/reset input signals SR0, SR1 are controlled by multiplexers 60A, 60B and AND-gates 67A, 67B. Inputs SR0, SR1 (if enabled by AND-gates 67A, 67B) can set or reset the registers. Set/reset control units R81B, R82B, R81A, R82A are programmably controlled to either set or reset registers RX, RY, RZ, RV, respectively. Registers RX, RY, RZ, RV can also be configured as latches, the choice being made by a programmable memory cell shared between the two registers in each slice.

XOR gates SF, SG, SH, SJ generate the sum portion of an addition or multiplication.

Multiplexers MF, MG, MH, MJ select between signals to place onto output lines X, Y, Z, V, respectively. Multiplexers MF and MH select between the related function generator output signal, sum output signal, and five-input function multiplexer output signal. Multiplexers MG and MJ select between the related function generator output signal, sum output signal, and six-input function multiplexer output signal. The function generator F, G, H, J output signal is selected when a logical function of up to four inputs is chosen. The five-input function multiplexer F5A, F5B output signal is selected when one of a limited set of nine-input functions is chosen, including any logical function of five inputs, a 4-to-1 multiplexer, and any other function that can be implemented as two four-input functions feeding a 2-to-1 multiplexer. The six-input function multiplexer F6A, F6B output signal is selected when one of a limited set of 18-input functions is chosen, including any logical function of six inputs, an 8-to-1 multiplexer, and any other function that can be implemented as the output of two five-input function multiplexers F5A, F5B feeding a 2-to-1 multiplexer. The sum output signal of XOR gates SF, SG, SH, SJ is selected when an appropriate arithmetic operation such as addition, subtraction or multiplication is implemented.

Multiplexers OF, OG, OH, OJ allow the carry chain to bypass the related function generator. Multiplexers OF, OG, OH, OJ select whether the respective carry multiplexer CF, CG, CH, CJ is controlled by the function generator output signal O or whether the carry chain bypasses the function generator by propagating the carry-in signal to the next stage along the carry chain. Multiplexers CA, CB allow additional access onto their respective carry chains by selecting either a carry-in signal CIN0, CIN1 (buffered in this embodiment by inverters I121B and I122B, I121A and I122A, respectively) or an input signal BF, BH as the starting point for the carry chain in the CLE.

Multiplexers DF, DH select a signal to be loaded into related registers RX, RZ either from the outputs of multiplexers MF, MH, or from input signals BF, BH, respectively. Multiplexers DG, DJ select a signal to be loaded into related registers RY, RV either from the outputs of multiplexers MG, MJ, or from input signal lines BG, BJ, respectively.

Multiplexers BB, BA are bypass multiplexers that can apply carry-out signals COUT0, COUT1 to output lines YB, VB, or can forward input signals BG, BJ to output lines YB, VB to route a signal generated on one side of the CLE to a destination on the other side of the CLE.

AND-gates BRB, BRA allow input signals BG, BJ to serve as set/reset signals to registers RX, RY, RZ, RV, or alternatively allows input signals BG, BJ to be decoupled from registers RX, RY, RZ, RV so that input signals BG, BJ can be used for other purposes, such as controlling multiplexers F6B, F6A or providing a DI signal to function generators G, J, without setting or resetting the registers. A logic 0 in the memory cells controlling AND-gates BRB, BRA decouples lines BG, BJ from set/reset control units R81B, R82B, R81A, R82A; while a logic 1 in the associated memory cell enables AND-gates BRB, BRA to use the signals on BG, BJ to set or reset registers RX, RY, RZ, RV as determined by set/reset control units R81B, R82B, R81A, R82A.

Input Multiplexer IMUX

Figure 7 shows details of the interior of input multiplexer IMUX of the embodiment shown in Figure 3C. As shown in Figure 7, the input lines in this embodiment (G, 430, DIW, 428, DIE, 429) are selectively programmed to drive a series of 34 outputs (431, 432, 436, LV0, LV6, LH0, LH6). 26 of these outputs drive CLE inputs 436, two drive the data inputs 431 and two drive the enable inputs 432 of tristate buffers 445, two drive LV0 and LV6 (two lines of vertical long line bus LVS), and two drive LH0 and LH6 (two lines of horizontal long line bus LHE). Not all inputs can drive all outputs; a selection of how many PIPs to provide and where they should be located is typically made based on the trade-off constraints of performance, silicon area, and routing flexibility. In one embodiment, any IMUX output not connected to any IMUX input line is connected through a pullup to the positive voltage supply.

Routing Examples

Figures 8 and 9 show examples of signal paths implemented in an array of tiles according to the embodiment shown in Figure 3C. The labeling convention used in Figures 8 and 9 is that tiles are identified by a row and column number based on the portion of the array shown in the figure. Within a tile the line labels correspond to the bus labels indicated in Figure 3A. However, in Figures 8 and 9, the label indicates a single line rather than the entire bus. Only lines used in the examples are shown and labeled. With this labeling convention, some labels are used more than once. But no ambiguity exists, because all tiles are identical and each tile has a different tile number. The tile numbers distinguish between labels used more than once in the same drawing.

Figure 8 shows a routing example of a signal path using hex and single-length lines. A signal is sourced by source logic block L0 of tile 8-2, and programmably connected within the tile through line OUT to hex segment 6N. Hex segment 6N turns north and becomes vertical hex segment 47. This signal must fan out to logic blocks at three destinations in tiles 5-3, 4-3, and 1-2. The signal passes through tiles 7-2 through 3-2 on segments 46 through 42. At tile 5-2, the signal is on segment 44, where it fans out to segment 6V. In tile 5-2, segment 6V is programmably connected to single-length line E, which continues east into tile 5-3 as line W. In tile 5-3, line W is an input to the first destination, logic block L1, and also to switching structure 403, where it is programmably connected to line N. Line N of tile 5-3 continues north into tile 4-3 as line S, which is an input to the second destination, logic block L2. The northern end of the vertical hex line is reached in tile 2-2, where the signal is on segment 41. Segment 41 is programmably connected to single-length line N, which continues north into tile 1-2 as line S. In tile 1-2, line S is an input to the third destination, logic block L3.

Thus in Figure 8 a signal net is formed from source logic block LO to destination logic blocks L1, L2 and L3. The connections are fast because the PIP onto the hex line is buffered, there is a low capacitance on the hex line due to the small number of PIPs on the hex line, and each path from source to destination has a small total number of PIPs inserting delay into the signal path. In the example of Figure 8, each signal path between source and destination logic blocks requires at most three PIPs: one buffered PIP onto the hex line and one or two unbuffered PIPs onto single-length lines. (A logic block is the structure designated LB in Figure 3A.)

Figure 9 shows a routing example of a signal path using a long line, hex lines, and single-length lines. A signal is sourced by source logic block L8 of tile 1-1, and must be routed to logic blocks L4, L5, L6, and L7. The signal from logic block L8 is programmably connected within tile 1-1 to one of the vertical long lines LV accessible from that particular tile. Long line LV extends southward for 18 tiles to tile 19-1, where long line LV is programmably connected to hex segment 6S, then continues southward to drive other hex lines (not shown) south of the portion of the array shown in Figure 9. (The gap of 18 tiles in Figure 9 is included to demonstrate that long line LV must carry the signal far enough that a long line is more efficient than concatenated hex lines.) From hex segment 6S in tile 19-1, the signal turns south and becomes vertical hex segment 41. The signal passes through tiles 20-1 through 24-1 on segments 42 through 46. At tile 22-1, the signal is on segment 44, where it fans out to segment 6V and continues south to tile 25-1. From segment 6V in tile 22-1, the signal makes a programmable connection to segment 6E (demonstrating the connection of two perpendicular hex lines). The signal then moves east on segment 37, traversing tiles 22-2 through 22-6 on segments 36 through 32. At tile 22-4, the signal is on segment 34, where it fans out to segment 6H and continues east to tile 22-5. In tile 22-4, the signal makes a programmable connection between segment 6H and single-length line S, which continues south into tile 23-4 as line N. In tile 23-4, line N enters switching structure 403 and connects to line E, which continues east into tile 23-5 on line W, an input to the first destination, logic block L4. From segment 32 in tile 22-6, the signal continues east into tile 22-7 on segment 31, then turns north onto segment 6W. Segment 6W is programmably connected to line N, which continues north into tile 21-7 as line S and is an input to the second destination, logic block L5.

Returning to the vertical hex line, the signal on segment 46 in tile 24-1 continues south into tile 25-1 on segment 47, which turns west on segment 6N. From segment 6N in tile 25-1, the signal makes a programmable connection to segment 6E. The signal then moves east on segment 37, traversing tiles 25-2 through 25-6 on segments 36 through 32. From segment 32 in tile 25-6, the signal continues east into tile 25-7 on segment 31, then turns north onto segment 6W. Segment 6W is programmably connected to line S, which continues south into tile 26-7 as line N and is an input to the third destination, logic block L6. One further connection remains to be made. From the south end of the vertical hex line, in tile 25-1, the signal on segment 6V makes an additional programmable connection to line E. Line E continues east into tile 25-2 on line W. In tile 25-2, line W enters switching structure 403 and connects to line S, which continues south into tile 26-2 on line N, an input to the fourth destination, logic block L7.

Thus in Figure 9 a signal net is formed from source logic block L8 to destination logic blocks L4, L5, L6 and L7. As in the example of Figure 8, the connections to long lines and hex lines are buffered and the RC delay of the unbuffered connection is minimized. In this example, these destinations required at most four PIPs in the interconnect path between logic blocks: one or two buffered PIPs onto hex lines, and one or two unbuffered PIPs onto single-length lines. Yet 31 tiles were traversed between source logic block L8 and destination logic block L6.

In all of the above routing examples, alternative paths are available, many of which require more PIPs than the signal paths used in these examples. If some of the interconnect lines on one path are already used by other signals, an alternative path may be used.

High Fanout Routing Examples

Figure 10 shows a routing example of a high fanout control signal, such as a clock (CK), clock enable (CE), set/reset (SR), or tristate enable signal (432) delivered from a vertical long line to a column of tiles. A signal can be placed onto a vertical long line either from outside the tile array, or by using one of lines 437I of Figure 3A. From vertical long line bus LV, in every twelfth tile

(TILE4) the high fanout control signal emerges on line 600. Line 600 drives two vertical hex lines, one of which (H1) extends north six tiles from TILE4 and one of which (H2) extends south six tiles from TILE4. From hex lines H1 and H2, lines 428 (shown in Figure 3A) permit four bits of each vertical hex bus to drive the logic block. The input multiplexer (IMUX) of Figure 7 permits vertical hex lines, through lines 428, to drive control signals CK, CE, SR, and 432 without passing through a single-length line, although signals on other CLE inputs, such as function generator inputs, must always pass through a single-length line to reach the CLE. Hex line H1 drives the logic block in TILE4, plus the five tiles adjacent to the north. Note that in the northernmost tile reached by hex line H1, line 428 does not make contact with hex line H1. Instead, contact is made with hex line H3, which originates 12 tiles north of TILE4. Hex line H2 drives the six logic blocks adjacent to the south of TILE4. North of hex line H1, hex line H3 drives the adjacent six logic blocks. South of hex line H2, hex line H4 drives the adjacent six logic blocks. Each hex line drives a high fanout control signal in six logic blocks. The tiles accessed by hex lines H1 and H2 (labeled 610 in Figure 10) form a column of 12 tiles that can be vertically repeated to access an entire column of an array of repeating tiles.

Figure 10 shows that for a high fanout control signal, there is only one PIP between a signal on a vertical long line and any destination logic block in the same column. (However, in the preferred embodiment, there is an additional PIP added to the path in input multiplexer IMUX.) Therefore, there is very little delay on this path. Additionally, this distribution method has the same number of PIPs and the same number of interconnect lines (one long line, one hex line) in the path to each logic block input. Therefore, there is very low skew on this path. Although the global lines (G in Figure 3C) would typically be used for up to four clocks, the distribution method of Figure 10, by taking advantage of the hierarchical nature of the interconnect scheme according to the invention, produces an excellent additional clock distribution network with high performance and low skew.

Figure 11 shows a routing example of a high fanout function generator input signal (or other non-control signal) delivered from a vertical long line to a column of tiles. A signal can be placed onto a vertical long line either from outside the tile array, or by using one of lines 437I of Figure 3A. From vertical long line bus LV, in every twelfth tile (TILE5-6) the high fanout function generator input signal emerges on line 601. Line 601 drives two vertical hex lines, one of which (H5) extends north six tiles from TILE4 and one of which (H6) extends south six tiles from TILE4. From hex lines H5 and H6, line segments 6S, 6N and 6V are used in conjunction with single-length lines N, S and switching structures 403 to drive the logic blocks, similar to the interconnections in Figures 8 and 9. In one embodiment, signals connecting to function generator inputs inside the CLE must always pass through a single-length line to reach the CLE. Hex line H5 drives, through single-length lines, the logic blocks in tiles TILE5-7, TILE5-8, and TILE5-3. In addition, hex line H5 drives, through single-length lines, the logic blocks in the three southernmost tiles of the 12-tile column to the north of column 611 (corresponding to tiles TILE5-10, TILE5-11, and TILE5-12 of column 611). Hex line H6 drives, through single-length lines, the logic blocks in tiles TILE5-4, TILE5-5, TILE5-6, and TILE5-9. In addition, hex line H6 drives, through single-length lines, the logic blocks in the two northernmost tiles of the 12-tile column to the south of column 611 (corresponding to tiles TILE5-1 and TILE5-2 of column 611). Tiles TILE5-1 and TILE5-2 are driven, through single-length lines, by hex line H7, which originates 12 tiles north of TILE5-6. Tiles TILE5-10, TILE5-11, and TILE5-12 are driven, through single-length lines, by hex line H8, which originates 12 tiles south of TILE5-6. Each hex line drives a high fanout function generator input signal in six logic blocks. Thus, by cooperation between adjacent 12-tile columns 611, a repeating pattern is formed that enables the distribution of non-control signals to an entire column of an array of repeating tiles.

Note that other connections of hex lines and single-length lines are possible. For example, the logic block in TILE5-2 can be driven by hex line 6V through a switching structure and single-length line in TILE5-3 (not shown) and hence through an additional switching structure and single-length line in TILE5-2 to drive the logic block in TILE5-1. This or a similar arrangement can be used in the northernmost implementation of 12-tile column 611, to access the two northernmost tiles. Similar connections can be made in the southernmost implementation of 12-tile column 611 to access the three southernmost tiles.

Figure 11 shows that for a high fanout function generator input signal, there are either two or three PIPs between a signal on a vertical long line and any destination logic block in the same column: one from the long line to a hex line, one from the hex line to a first single-length line, and sometimes one to a second single-length line through the switching structure. (However, in the preferred embodiment, there is an additional PIP added to the path in input multiplexer IMUX.) Only the last one or two PIPs are unbuffered. Therefore, there are no large unbuffered RC trees on this net, no signal path has more than two unbuffered PIPs in series, and the delay on each signal path is minimized.

Figure 12 shows an example of how to combine the interconnect of Figures 10 and/or 11 to deliver a high-fanout signal to an array of tiles. A signal can be placed onto a horizontal long line either from outside the tile array, or by using one of lines 438 of Figure 3A. The signal enters the portion of the array shown in Figure 12 on a line in horizontal long line bus LH. Long line bus LH traverses a row of 12 tiles 612, in which it makes contact with 12 logic blocks through hex lines and single-length lines. The horizontal single-length lines connected to some horizontal hex lines can drive vertical long lines through

PIPs in the IMUX. From the logic blocks in the 12 tiles of Figure 12, lines 437I extend from the input multiplexer (see Figure 3A) in the logic block to drive the vertical long lines LV. Using lines 437I in each of the 12 tiles 612, the signal on horizontal long line bus LH is transferred to 12 vertical long lines LV. Columns of 12 vertical tiles (613) can be implemented by either 12 tiles 610 of Figure 10 (for control signals) or 12 tiles 611 of Figure 11 (for function generator or other non-control inputs). Multiple 12-tile columns 613 can be placed to increase the number of accessible tiles. In this manner, high fanout signals can be distributed from an originating tile to any tile in the array.

As an alternative to the distribution scheme of Figure 12, special tiles can be created that permit, for example, hex lines to drive long lines, thus bypassing the single-length lines used in Figure 12. Such special tiles can be advantageously placed in rows along the top and bottom edges of the tile array, or at intervals within the array.

Fast Feedback Paths

A logic path in an FPGA typically comprises flip-flops or latches alternating with a series of combinational logic gates. These combinational logic gates are implemented in cascaded function generators. Critical paths are often those paths having a large number of function generators cascaded in series between such flip-flops or latches. Thus, such a critical path has one connection from a flip-flop to a function generator, and one connection from a function generator to a flip-flop, but many connections between two function generators. Therefore, a significant performance benefit for critical paths is gained from fast feedback paths that speed up connections between function generators. Hence, in one embodiment only the CLE outputs driven by function generator outputs have connections to the fast feedback paths. Registered outputs do not have such connections. Similarly, the CLE inputs connected to the fast feedback paths are function generator inputs and not control inputs. One such embodiment is the embodiment of Figure 3A, wherein unregistered CLE outputs have connections to the fast feedback paths. (Unregistered CLE outputs X and Y, Z and V are shown in Figures 6B, 6A, respectively.) In the embodiment of Figure 3A, fast feedback paths are routed through fast feedback lines 430 and input multiplexer IMUX-2. Registered CLE outputs are routed into output multiplexer OMUX-2 of Figure 3A and do not have connections to the fast feedback paths. (Registered CLE outputs XQ and YQ, ZQ and VQ are shown in Figures 6B and 6A, respectively.)

Figure 13 shows in simplified form the fast feedback paths of the embodiment of Figure 3C from the outputs of the CLE to the inputs of the same CLE. Labels in Figure 13 that are the same as labels in Figures 6A and 6B denote the same components or signal lines. For example, in Figure 13 CLE output V is driven by the output of function generator J. (Intervening logic in region 135 is not shown in Figure 13, but is shown in Figure 6A.) CLE output V also feeds back into region 134, which in the embodiment of Figure 3C is implemented as part of input multiplexer IMUX. In region 134, CLE output V connects through PIPs 130, 131, 132, 133 to one input (J3, H3, G2, F2, respectively) of each function generator (J, H, G, F, respectively) in the CLE. Similarly, each of the other unregistered CLE outputs Z, Y, X connects through PIPs to one input of each function generator in the CLE. Each function generator output connects to a different input of each of the function generators in the CLE. Registered CLE outputs VQ, ZQ, YQ, XQ do not connect to the fast feedback paths.

In another embodiment of the invention, the fast feedback paths can drive CLE inputs other than the function generator inputs. For example, the fast feedback paths can drive CLE inputs BJ, BH of Figure 6A, and BG, BF of Figure 6B. These connections can be used to provide the data input for function generators configured in shift register mode. Wide high performance shift registers can be implemented using such feedback paths.

Summary

The interconnect structure of the first aspect of the invention offers uniformly fast signals throughout the FPGA, due to the reduced loading (limited number of PIPs) on each interconnect line, the lack of large RC trees on signal nets, and the relatively small number of PIPs between source and destination on each path. The nearly uniform delays on various interconnect paths greatly mitigate a common situation in FPGAs where a few signal paths, much slower than the average signal path, significantly reduce the overall performance of the circuit. Intermediate-length lines enable the efficient implementation of localized circuits and ease the burden on routing software. The limited number of PIPs, and the judicious arrangement thereof, imparts a high degree of routing flexibility per silicon area consumed. Further routing flexibility is provided by the ability of logic block inputs to contact interconnect lines located on any edge of the logic block.

The fast feedback paths of the second aspect of the invention offer a method of bypassing the output multiplexers to provide a significantly faster feedback path from CLE outputs to inputs of the same CLE. Such fast feedback paths have a significant impact on the speed of critical paths in which many function generators are connected together in series. The fast feedback paths increase the performance of small circuits (or subsets of larger circuits) that fit into one CLE and reduce the number of routing resources outside the logic block that must be used to implement a given circuit. In one embodiment, all combinations of function generators in one CLE can drive each other through fast feedback paths, regardless of how logic is mapped into the function generators of the CLE. In another embodiment, each function generator in one CLE can simultaneously drive all of the other function generators (other than itself) through fast feedback paths.

Thus it will be understood that the present invention provides a new and useful interconnect structure for FPGAs. Those having skill in the relevant arts of the invention will now perceive various modifications and additions which may be made as a result of the disclosure herein. Accordingly, all such modifications and additions are deemed to be within the scope of the invention, which is to be limited only by the appended claims and their equivalents.

CLAIMS

What is claimed is:

1. An FPGA comprising:

an array of tiles, each tile including:

a logic block;

a plurality of PIPs;

a plurality of line segments, each such line segment abutting another such line segment in an adjacent such tile to form a plurality of interconnect lines;

some of said interconnect lines being single-length lines programmably connecting two such logic blocks in two adjacent such tiles; and

at least one of said interconnect lines being an intermediate-length line programmably connecting a first such logic block in a first such tile to a second such logic block in a second such tile through at least one such single-length line and one such PIP in said second tile, said second tile being separated from said first tile by at least a third such tile, said intermediate-length line not connecting to any PIPs in said third tile.

2. An interconnect structure for interconnecting logic blocks in an FPGA, comprising:

an array of logic blocks arranged in rows and columns, said logic blocks each having inputs and outputs;

a plurality of interconnect lines comprising a plurality of single-length lines and a plurality of intermediate-length lines,

each such single-length line being of sufficient length to connect to said inputs and outputs of two such adjacent logic blocks; and

each such intermediate-length line being of sufficient length to connect to said outputs of three such logic blocks separated from each other by at least one such logic block, each such intermediate-length line unconnectable to outputs of any two adjacent logic blocks; and

a plurality of PIPs, each such PIP programmably connecting one of said interconnect lines to another of said interconnect lines.

3. An interconnect structure for interconnecting an array of logic blocks, comprising:

an interconnect line;

at least five logic blocks arranged sequentially along said interconnect line; and

at least three PIPs for connecting said interconnect lines to said logic blocks;

said PIPs being located so as to leave at least one logic block unconnectable to said interconnect line between each pair of logic blocks connectable to said interconnect line.

4. An FPGA logic block comprising:

a configurable logic element (CLE) having fast feedback outputs, said CLE comprising more than two function generators each having inputs and at least one output configurably driving at least one of said CLE fast feedback outputs; and

an output multiplexer comprising more than two single-output multiplexers each having as inputs more than two of said outputs of said function generators;

and wherein:

each of said CLE fast feedback outputs configurably drives both at least one of said single-output multiplexer inputs and at least one of said function generator inputs.

5. An FPGA logic block comprising:

a plurality of function generators having data inputs and at least one output; and

an output multiplexer having inputs;

and wherein:

each of said function generator outputs configurably drives at least one of said output multiplexer inputs; and

each said data input of a given said function generator is configurably driven by said output of a different one of the others of said function generators.

6. An FPGA logic block comprising:

a configurable logic element (CLE) having fast feedback outputs, said CLE comprising more than two function generators, each such function generator having inputs and each such function generator further having at least one output configurably driving at least one of said CLE fast feedback outputs;

an output multiplexer structure comprising more that two single-output multiplexers each having more than two inputs;

means for programmably connecting each of said CLE fast feedback outputs to at least one of said single-output multiplexer inputs; and

means for programmably connecting each of said CLE fast feedback outputs to at least one of said function generator inputs.

7. An interconnect structure for distributing high-fanout signals to logic blocks in an FPGA, comprising:

an array of logic blocks arranged in rows and columns, said logic blocks each having inputs and at least one output;

a plurality of intermediate-length lines;

a plurality of long lines extending a length greater than a length of said intermediate-length lines, a first plurality of said long lines extending in a first direction and a second plurality of said long lines extending in a direction perpendicular to said first direction; and

means for programmably interconnecting one of said first plurality of said long lines and one of said second plurality of said long lines through one of said plurality of intermediate-length lines, said interconnecting means comprising a first PIP connecting said one of said first plurality of said long lines to said one of said plurality of intermediate-length lines.

8. An interconnect structure for distributing high-fanout signals in an FPGA, comprising:

a column of tiles, each such tile comprising a logic block having at least one input;

a plurality of intermediate-length lines extending through at least a portion of said column of tiles, each such intermediate-length line being at least three such tiles long, each such intermediate-length line connecting a first such tile to at least second and third such tiles at least two and three tiles away;

at least one long line extending through at least a portion of said column of tiles, said long line having a length greater than a length of said intermediate-length lines; and

means for programmably interconnecting said one long line to at least one such input in each said logic block, said interconnecting means comprising:

at least one such intermediate-length line; and

a PIP connecting said long line to said intermediate-length line.

AMENDED CLAIMS

[received by the International Bureau on 27 April 1998 (27.04.98); original claims 5 and 6 amended; remaining claims unchanged (2 pages)]

line unconnectable to outputs of any two adjacent logic blocks; and

a plurality of PIPs, each such PIP programmably connecting one of said interconnect lines to another of said interconnect lines.

3. An interconnect structure for interconnecting an array of logic blocks, comprising:

an interconnect line;

at least five logic blocks arranged sequentially along said interconnect line; and

at least three PIPs for connecting said interconnect lines to said logic blocks;

said PIPs being located so as to leave at least one logic block unconnectable to said interconnect line between each pair of logic blocks connectable to said interconnect line.

4. An FPGA logic block comprising:

a configurable logic element (CLE) having fast feedback outputs, said CLE comprising more than two function generators each having inputs and at least one output configurably driving at least one of said CLE fast feedback outputs; and

an output multiplexer comprising more than two single-output multiplexers each having as inputs more than two of said outputs of said function generators;

and wherein:

each of said CLE fast feedback outputs configurably drives both at least one of said single-output multiplexer inputs and at least one of said function generator inputs.

5. An FPGA logic block comprising:

a plurality of function generators having data inputs and at least one output; and

an output multiplexer having inputs;

and wherein:

each of said function generator outputs configurably drives at least one of said output multiplexer inputs; and each said data input of a given said function generator is configurably driven by said output of a different one of said function generators via a fast feedback path that bypasses said output multiplexer.

6. An FPGA logic block comprising:

a configurable logic element (CLE) having fast feedback outputs, said CLE comprising more than two function generators, each such function generator having inputs and each such function generator further having at least one output configurably driving at least one of said CLE fast feedback outputs;

an output multiplexer structure comprising more than two single-output multiplexers each having more than two inputs;

means for programmably connecting each of said CLE fast feedback outputs to at least one of said single-output multiplexer inputs; and

means for programmably connecting each of said CLE fast feedback outputs to at least one of said function generator inputs.

7. An interconnect structure for distributing high-fanout signals to logic blocks in an FPGA, comprising:

an array of logic blocks arranged in rows and columns, said logic blocks each having inputs and at least one output;

a plurality of intermediate-length lines;

a plurality of long lines extending a length greater than a length of said intermediate-length lines, a first plurality of said long lines extending in a first direction and a second plurality of said long lines extending in a direction perpendicular to said first direction; and

means for programmably interconnecting one of said first plurality of said long lines and one of said second plurality of said long lines through one of said plurality of intermediate-length lines, said interconnecting means comprising a first PIP connecting said one of said first

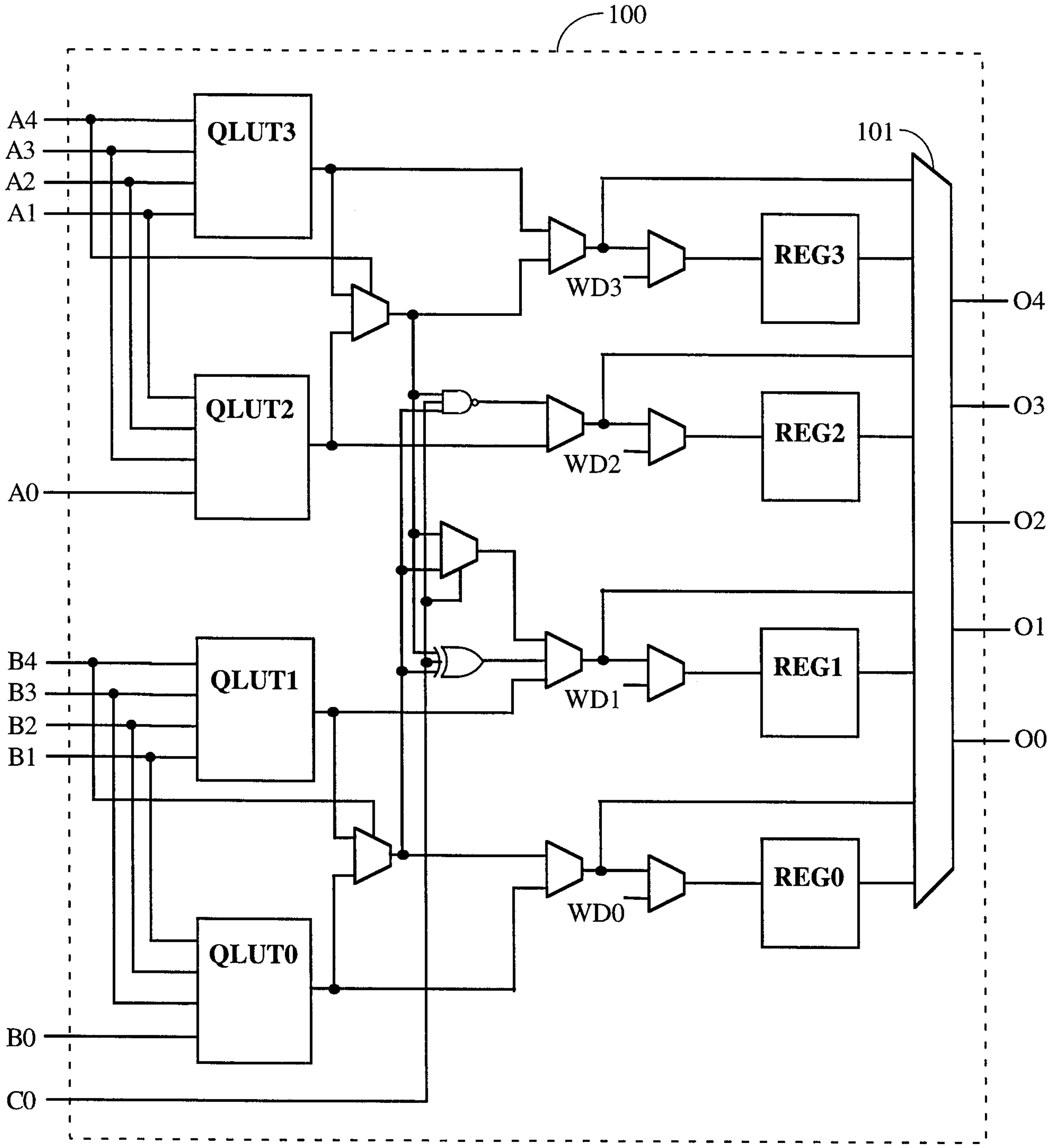


FIG. 1A
(Prior Art)

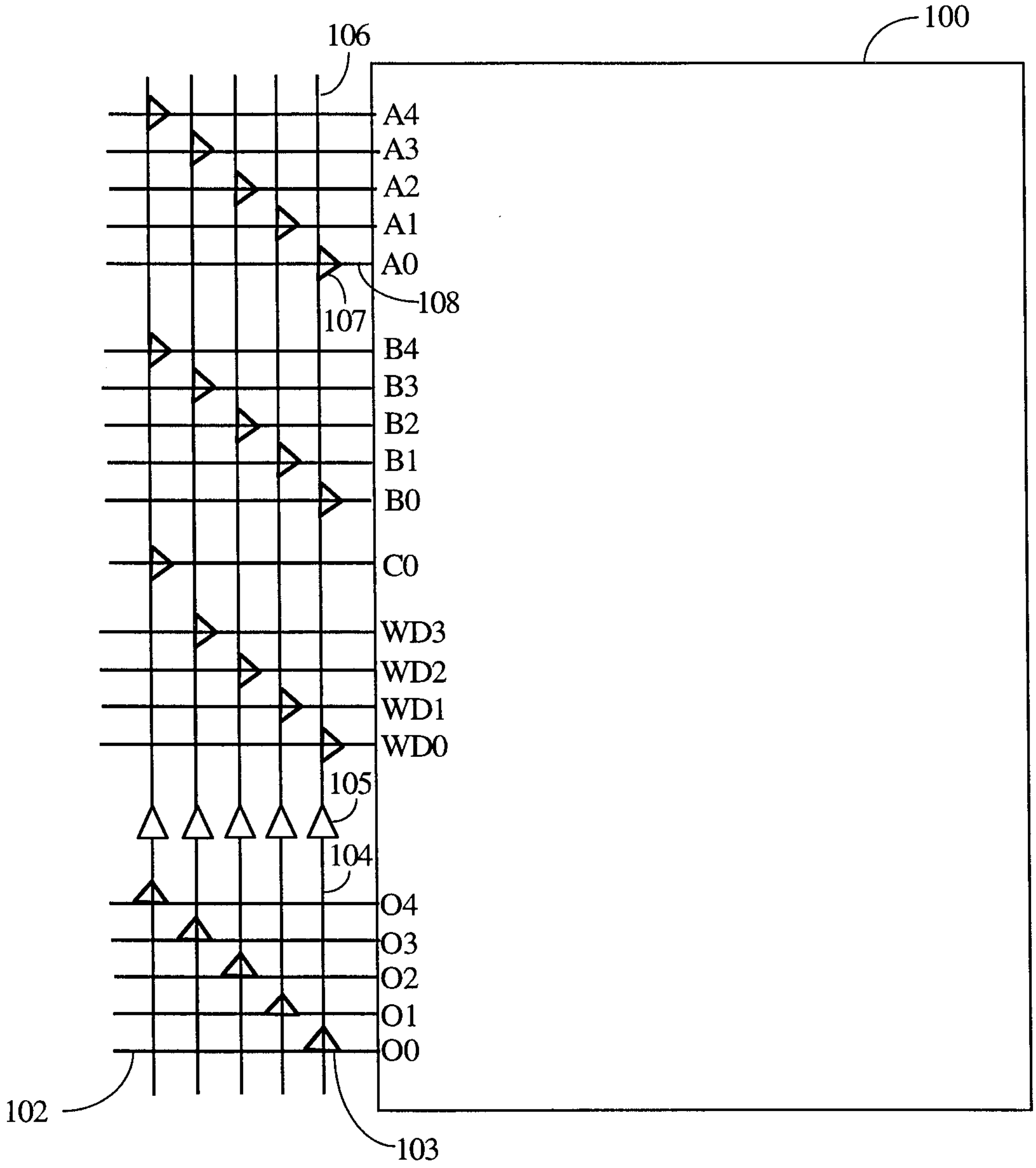


FIG. 1B
(Prior Art)

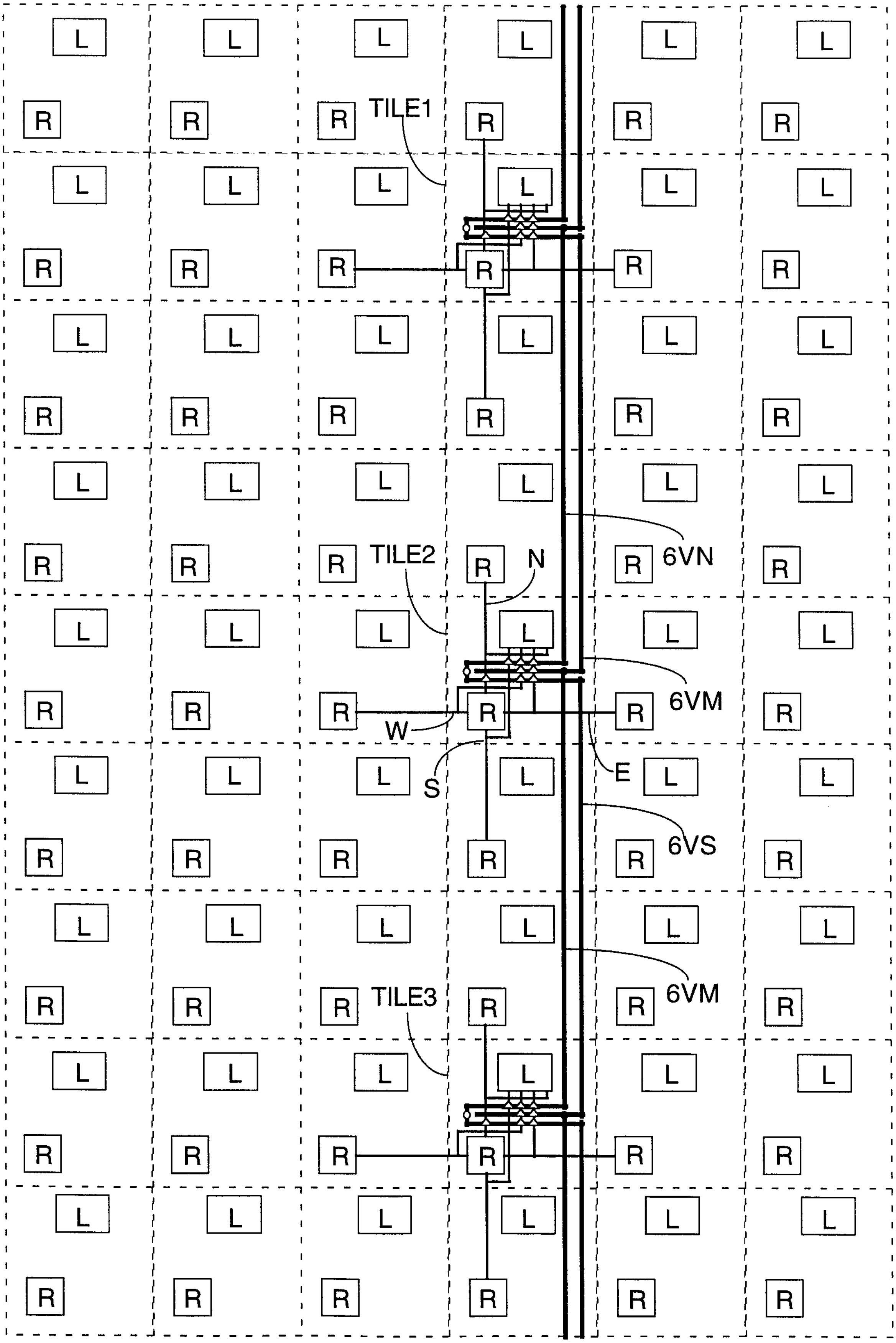


FIG. 2

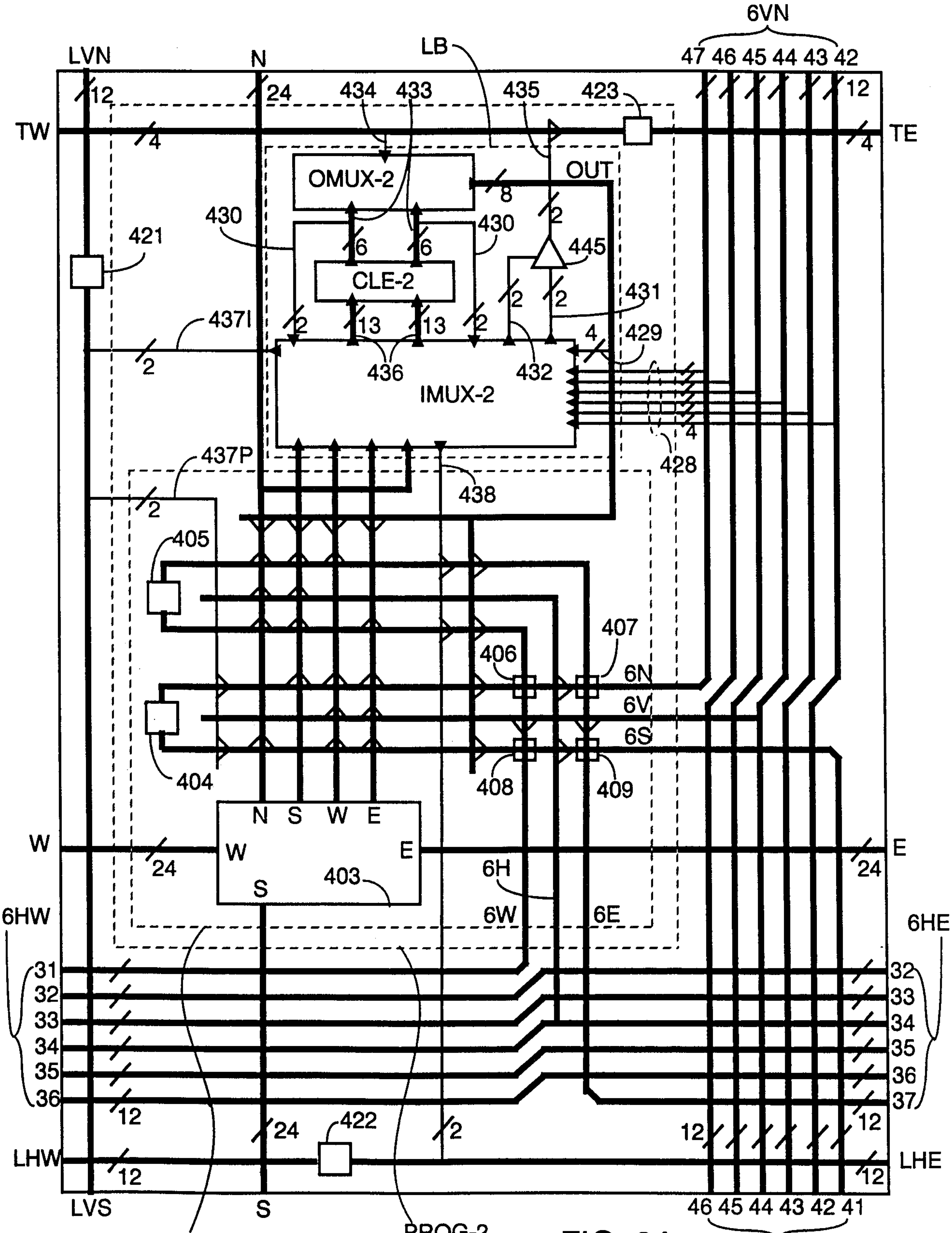

6VN
LVN
N
LB
47 46 45 44 43 42
12
24
434
433
435
423
12
TW
4
TE
4
OUT
OMUX-2
8
2
430
421
430
6
6
445
CLE-2
2
2
431
437I
2
13
13
2
4
429
2
436
432
IMUX-2
4
428
437P
438
2
405
407
406
6N
6V
6S
404
408
409
N S W E
W
W
E
E
24
24
403
6H
S
6HW
6W
6E
6HE
31
32
33
34
35
36
32
33
34
35
36
37
12
24
422
2
12
12
LHW
12
LHE
12
LVS
S
46 45 44 43 42 41
INTERCONNECT-2
PROG-2
6VS

FIG. 3A

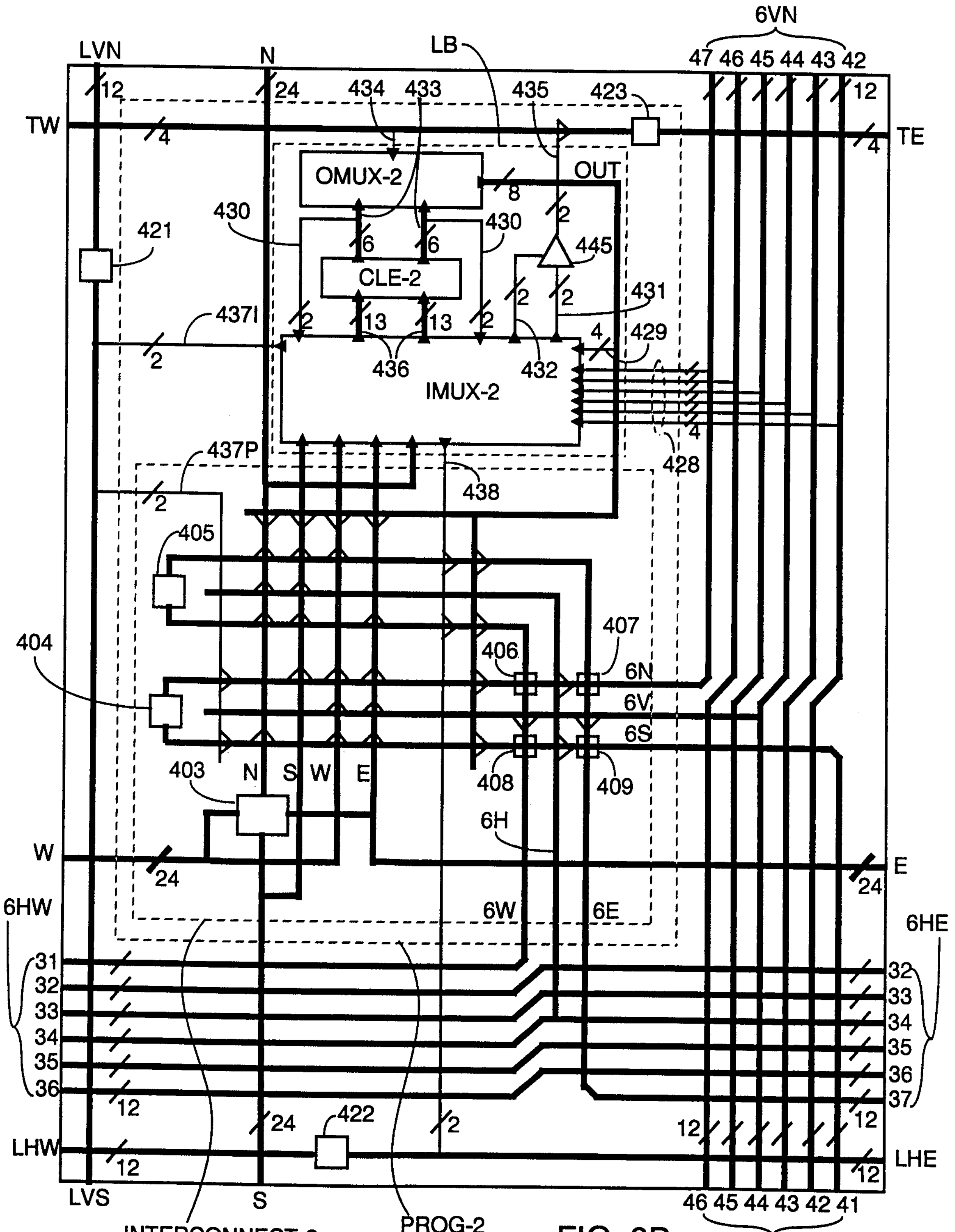

6VN
LVN
N
LB
47 46 45 44 43 42
12
24
434
433
435
423
12
TW
4
TE
4
OMUX-2
OUT
8
430
430
421
6
6
2
445
CLE-2
2
2
431
437I
2
13
13
2
4
429
2
436
432
IMUX-2
4
437P
428
2
438
405
407
404
406
6N
6V
6S
403
N S W E
408
409
6H
W
24
E
24
6HW
6W
6E
6HE
31
32
33
34
35
36
32
33
34
35
36
37
12
12
24
422
2
12
12
LHW
12
LHE
12
LVS
S
46 45 44 43 42 41
6VS
INTERCONNECT-2
PROG-2

FIG. 3B

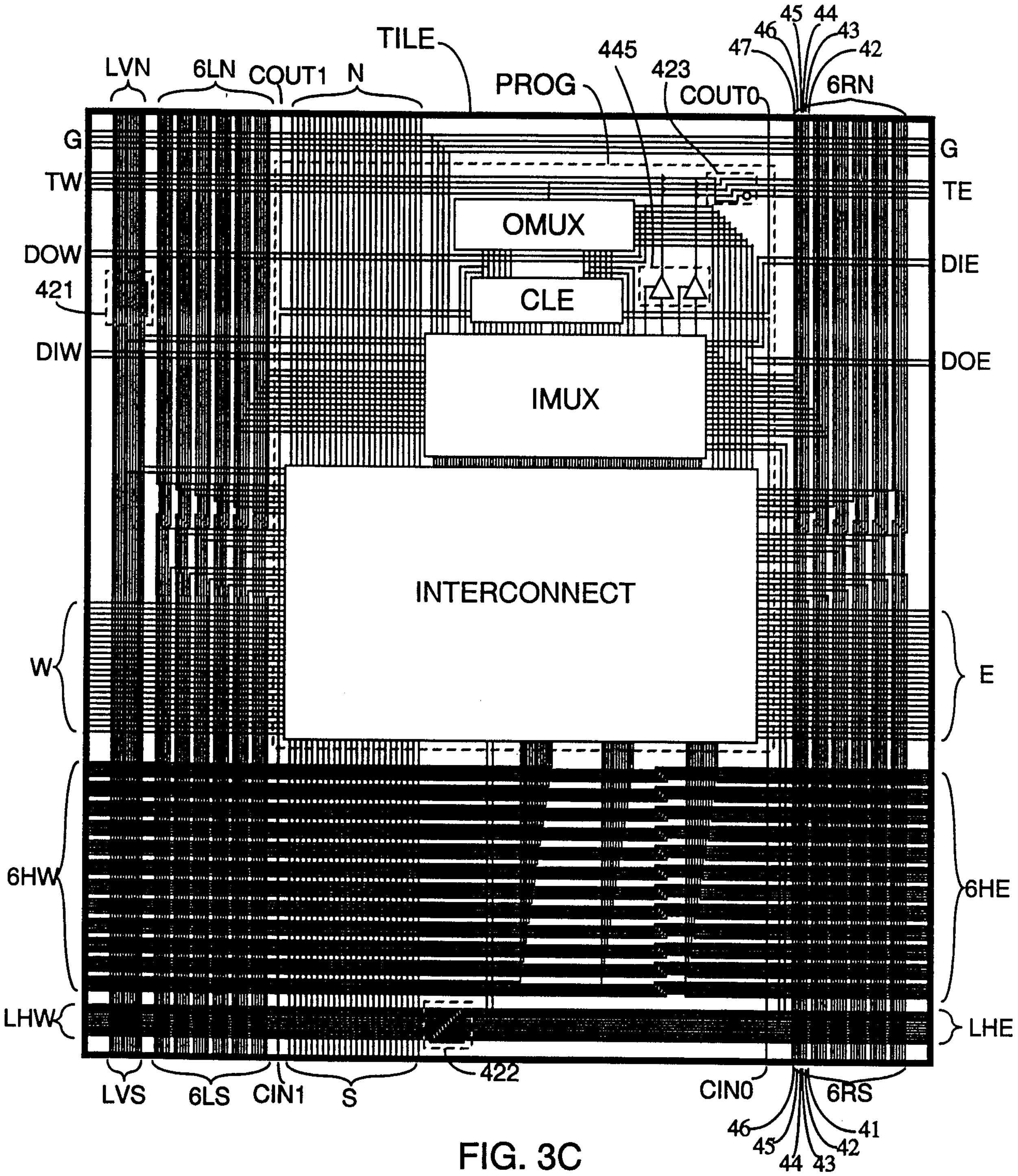

LVN
6LN
COUT1
N
TILE
PROG
445
423
COUT0
47
46
45
44
43
42
6RN
G
TW
DOW
421
DIW
OMUX
CLE
IMUX
INTERCONNECT
G
TE
DIE
DOE
W
E
6HW
6HE
LHW
LHE
LVS
6LS
CIN1
S
422
CIN0
6RS
46
45
44
43
42
41

FIG. 3C

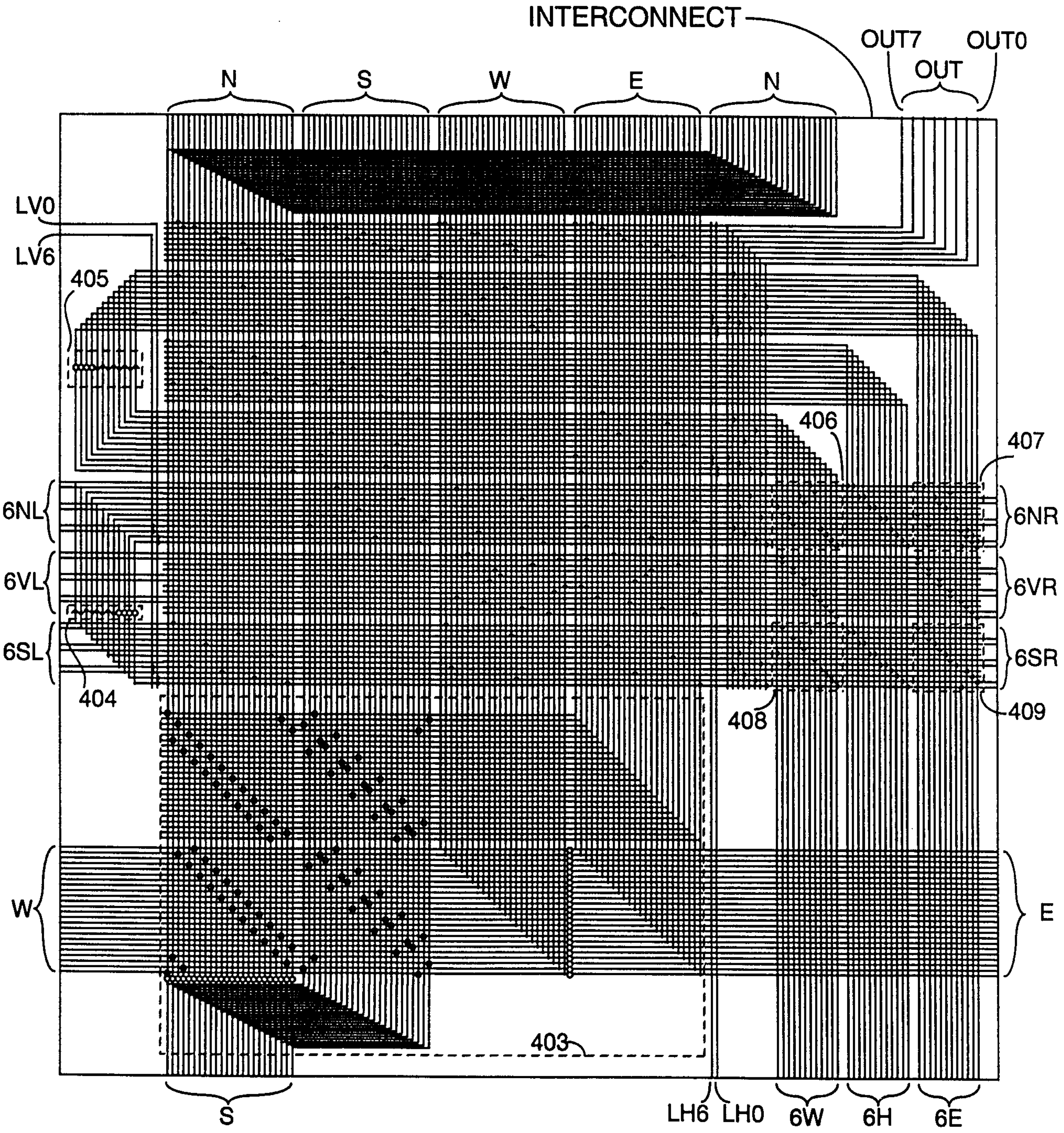

INTERCONNECT
OUT7
OUT
OUT0
N
S
W
E
N
LV0
LV6
405
406
407
6NL
6NR
6VL
6VR
6SL
6SR
404
408
409
W
E
403
S
LH6
LH0
6W
6H
6E

FIG. 4

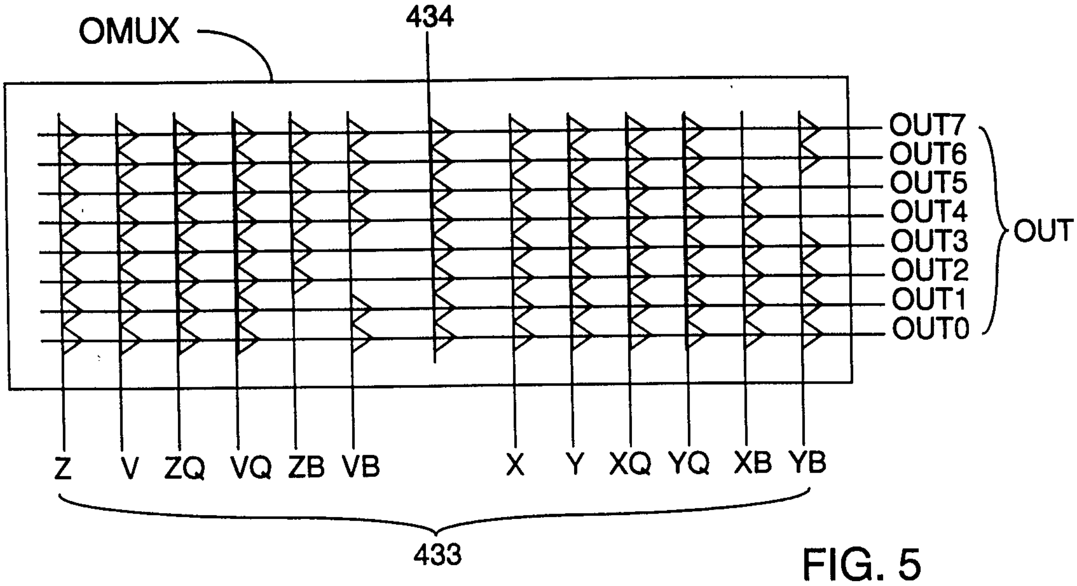

OMUX
434
OUT7
OUT6
OUT5
OUT4
OUT3
OUT2
OUT1
OUT0
OUT
Z
V
ZQ
VQ
ZB
VB
X
Y
XQ
YQ
XB
YB
433

FIG. 5

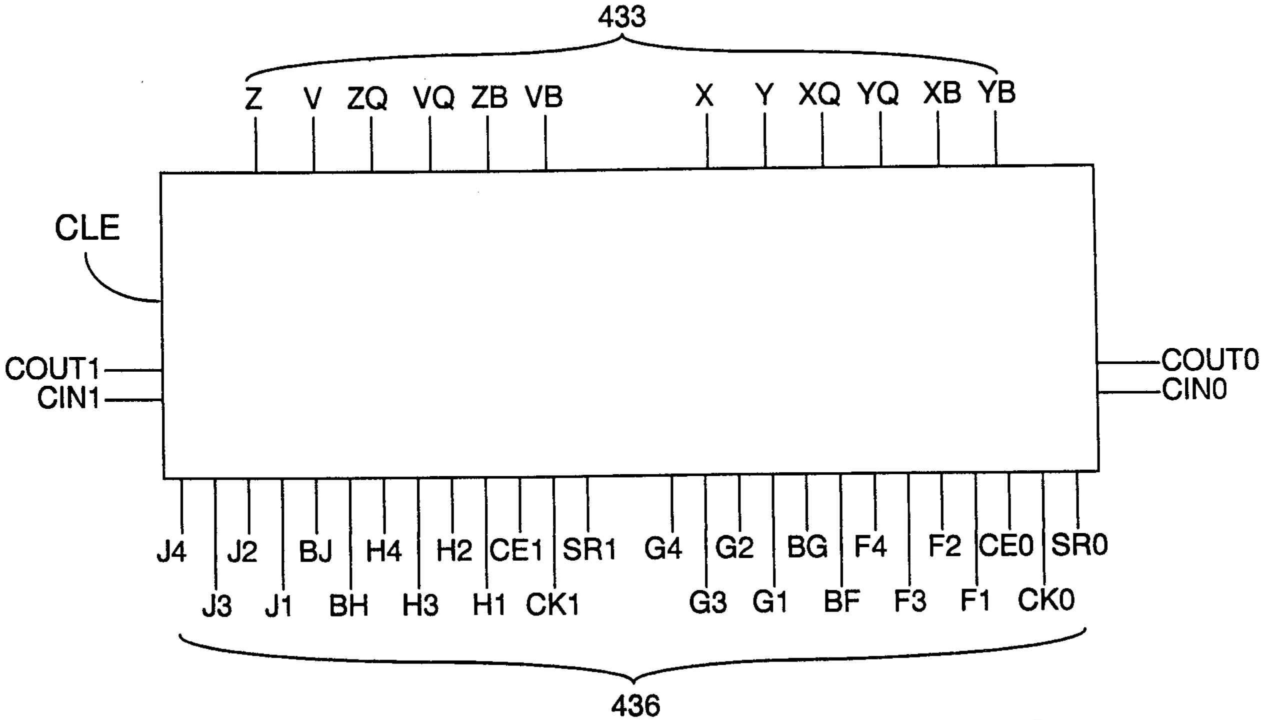
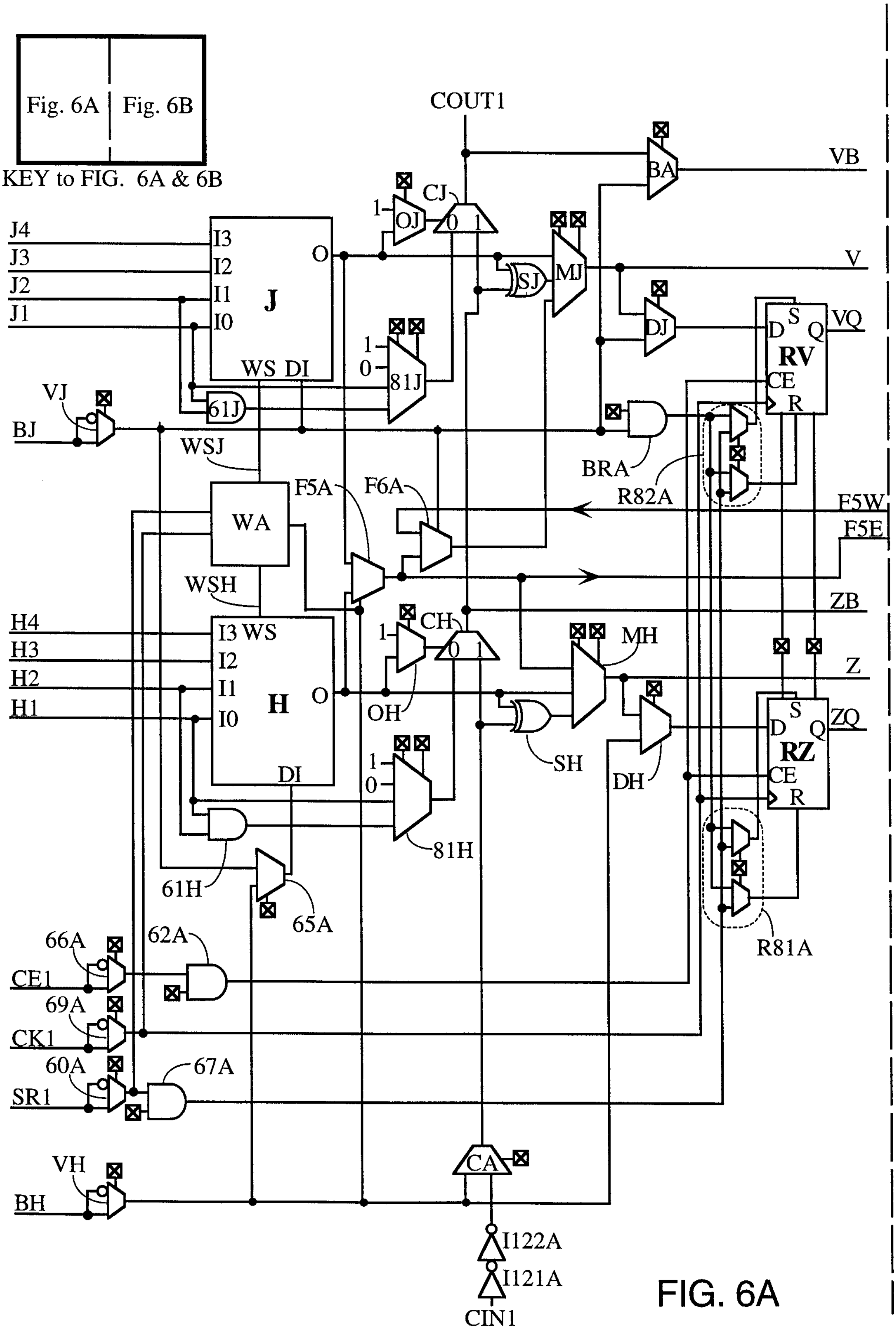

433
Z
V
ZQ
VQ
ZB
VB
X
Y
XQ
YQ
XB
YB
CLE
COUT1
CIN1
COUT0
CIN0
J4
J2
BJ
H4
H2
CE1
SR1
G4
G2
BG
F4
F2
CE0
SR0
J3
J1
BH
H3
H1
CK1
G3
G1
BF
F3
F1
CK0
436

FIG. 6



KEY to FIG. 6A & 6B

FIG. 6A

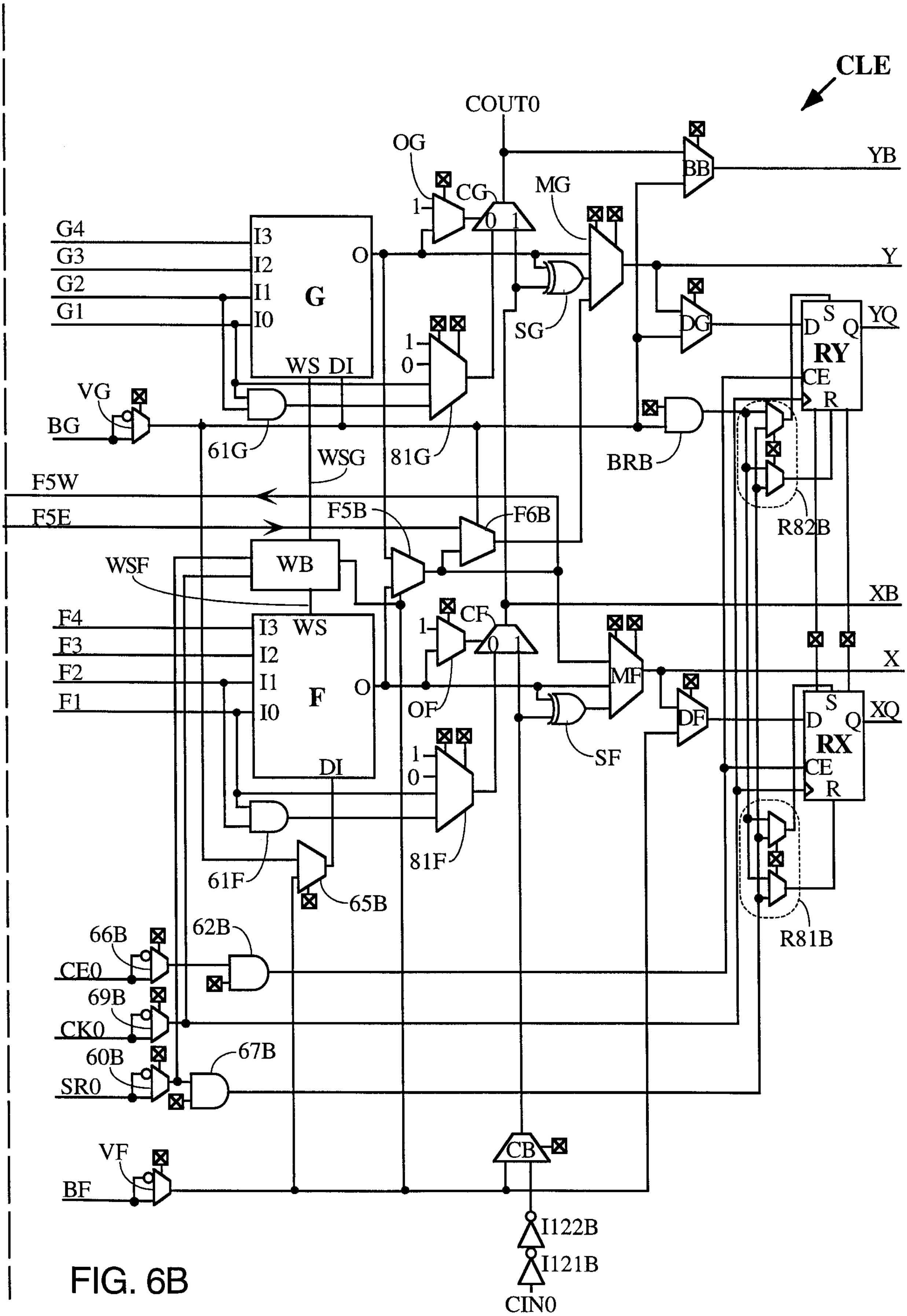

CLE
COUT0
OG
CG
MG
YB
BB
Y
G4
G3
G2
G1
I3
I2
I1
I0
O
G
WS DI
1
0
SG
DG
D S Q
YQ
RY
CE
R
VG
BG
61G
WSG
81G
BRB
R82B
F5W
F5E
F5B
F6B
WSF
WB
XB
F4
F3
F2
F1
WS
F
CF
OF
MF
X
SF
DF
XQ
RX
DI
61F
81F
65B
R81B
66B
62B
CE0
69B
CK0
60B
67B
SR0
VF
BF
CB
I122B
I121B
CIN0

FIG. 6B

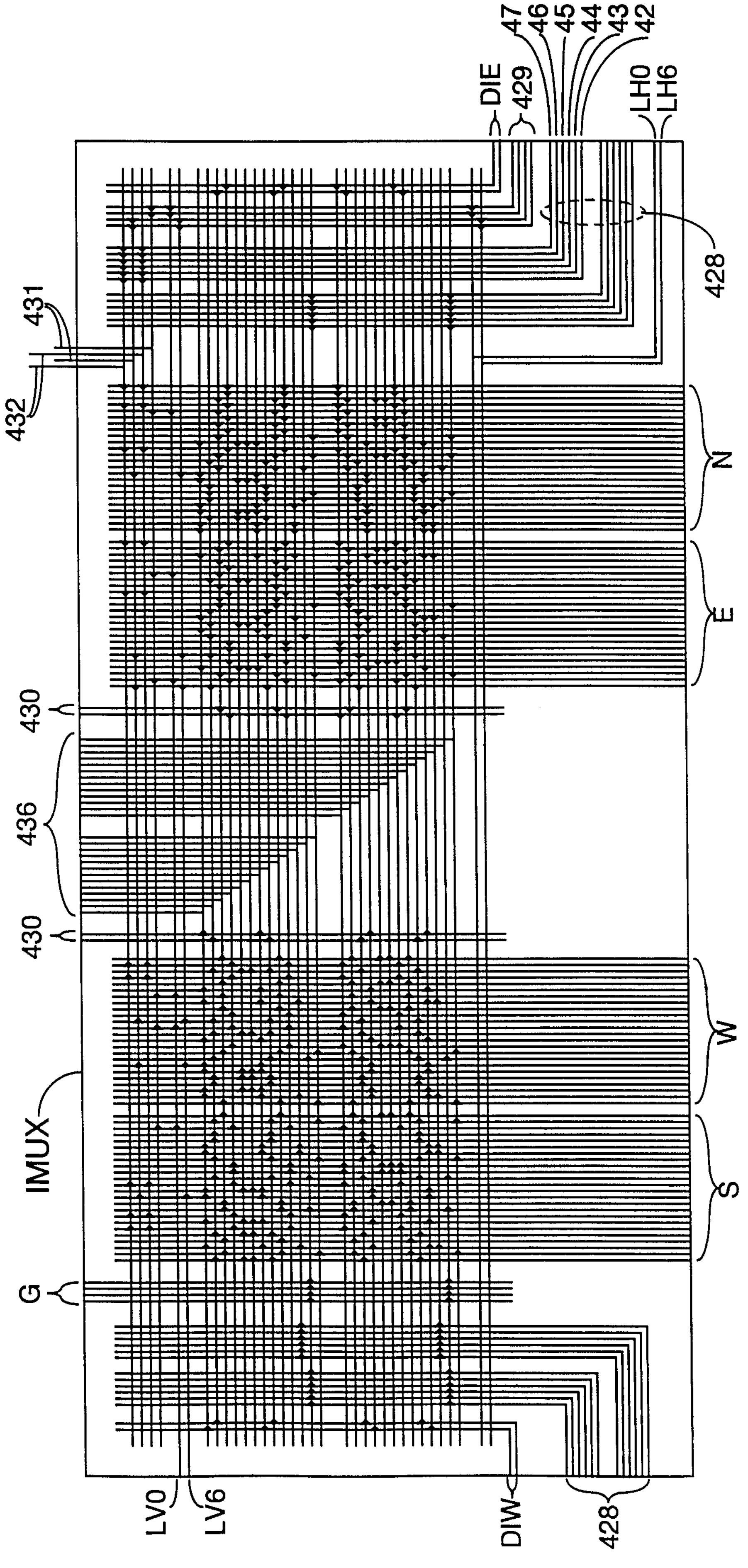

DIE
429
47
46
45
44
43
42
LH0
LH6
428
431
432
N
E
430
436
430
W
S
IMUX
G
LV0
LV6
DIW
428

FIG. 7

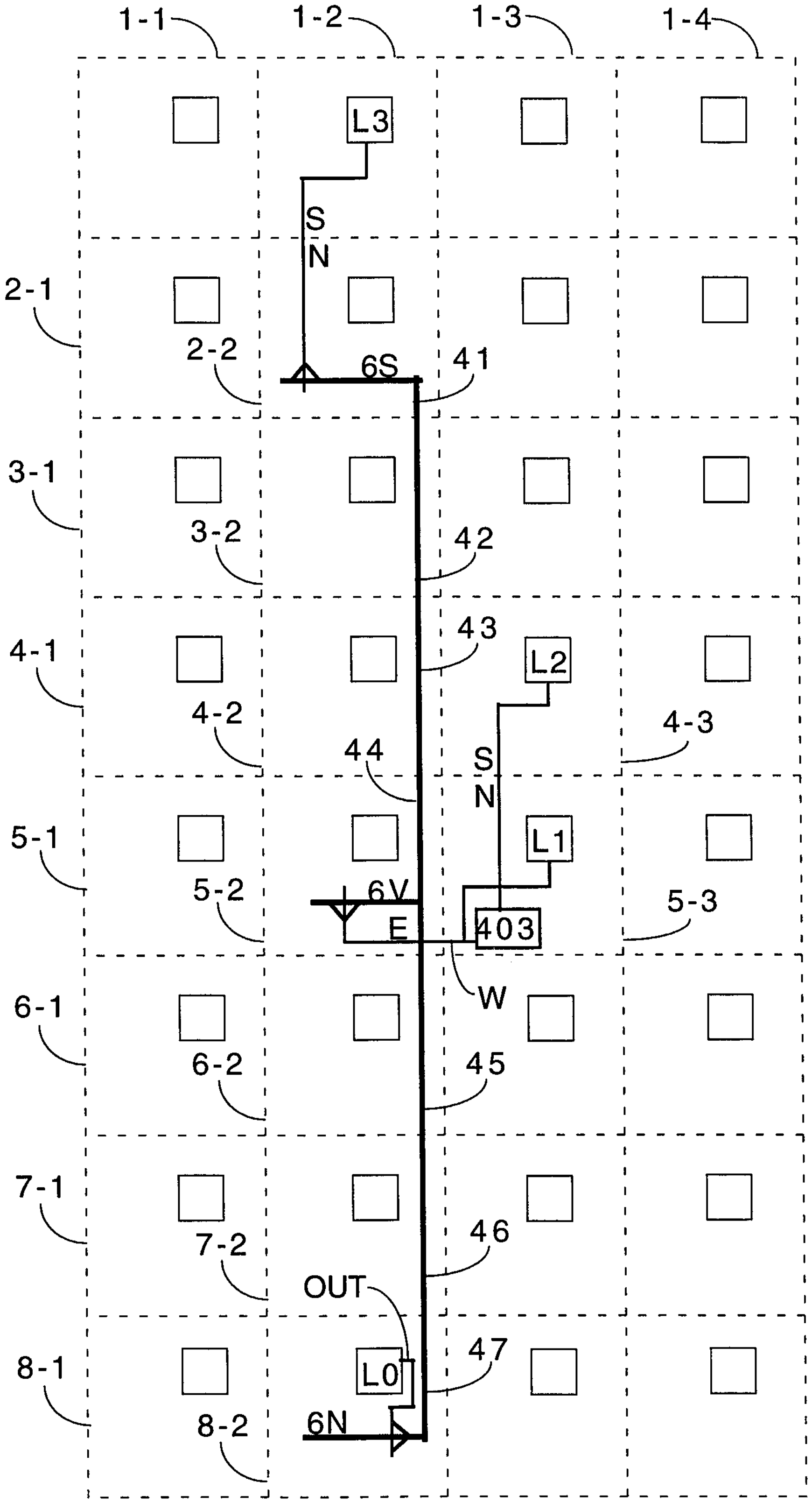

1-1
1-2
1-3
1-4
L3
S
N
2-1
2-2
6S
41
3-1
3-2
42
43
L2
4-1
4-2
4-3
44
S
N
5-1
L1
6V
5-2
E
403
5-3
W
6-1
6-2
45
7-1
7-2
46
OUT
8-1
L0
47
6N
8-2

FIG. 8

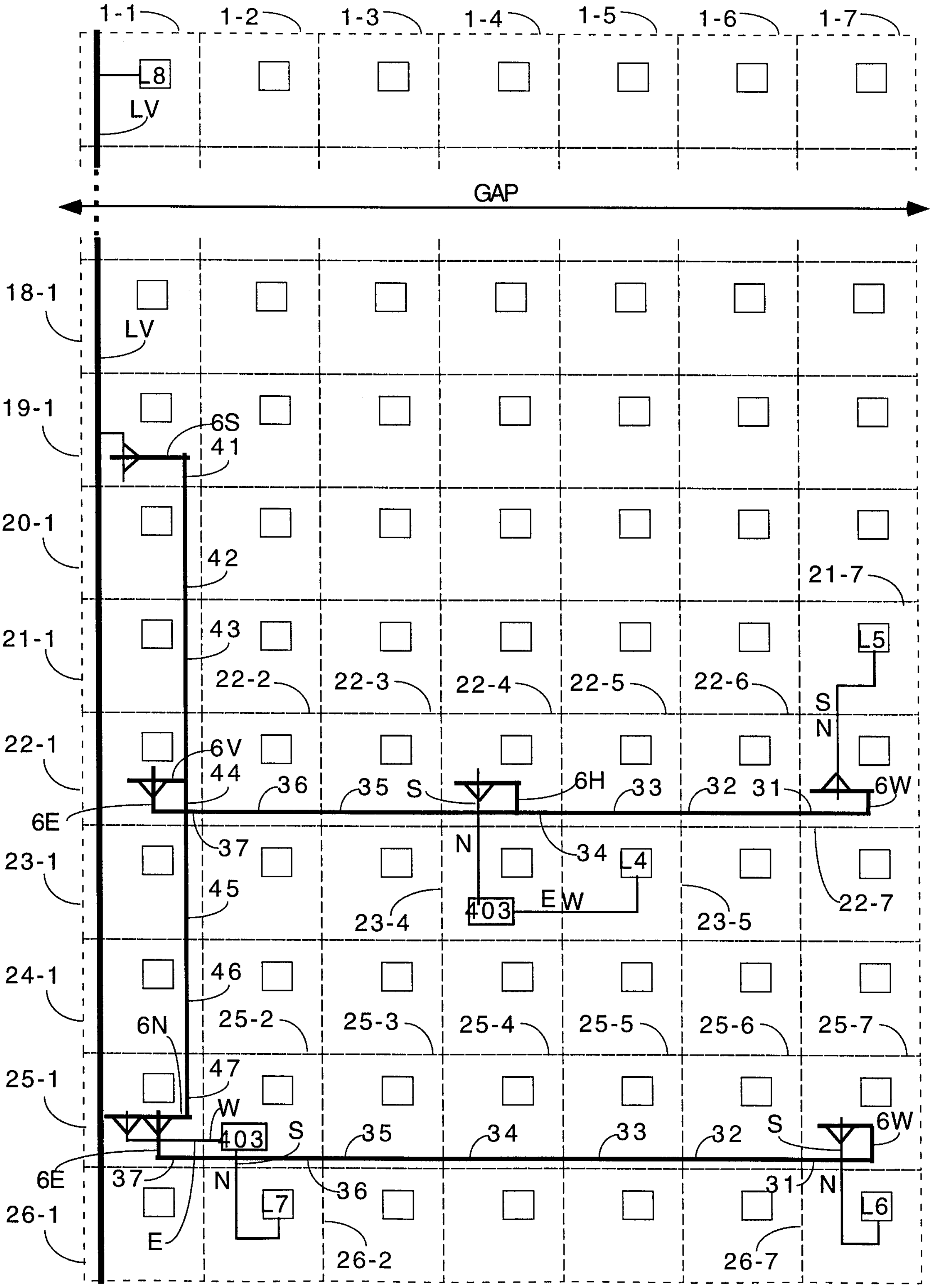

1-1
1-2
1-3
1-4
1-5
1-6
1-7
L8
LV
GAP
18-1
LV
19-1
6S
41
20-1
42
21-7
21-1
43
L5
22-2
22-3
22-4
22-5
22-6
S
N
22-1
6V
44
36
35
S
6H
33
32
31
6W
6E
37
N
34
L4
23-1
45
23-4
403
E
W
23-5
22-7
24-1
46
6N
25-2
25-3
25-4
25-5
25-6
25-7
25-1
47
W
403
S
35
34
33
32
S
6W
6E
37
36
31
N
N
L7
L6
26-1
E
26-2
26-7

FIG. 9

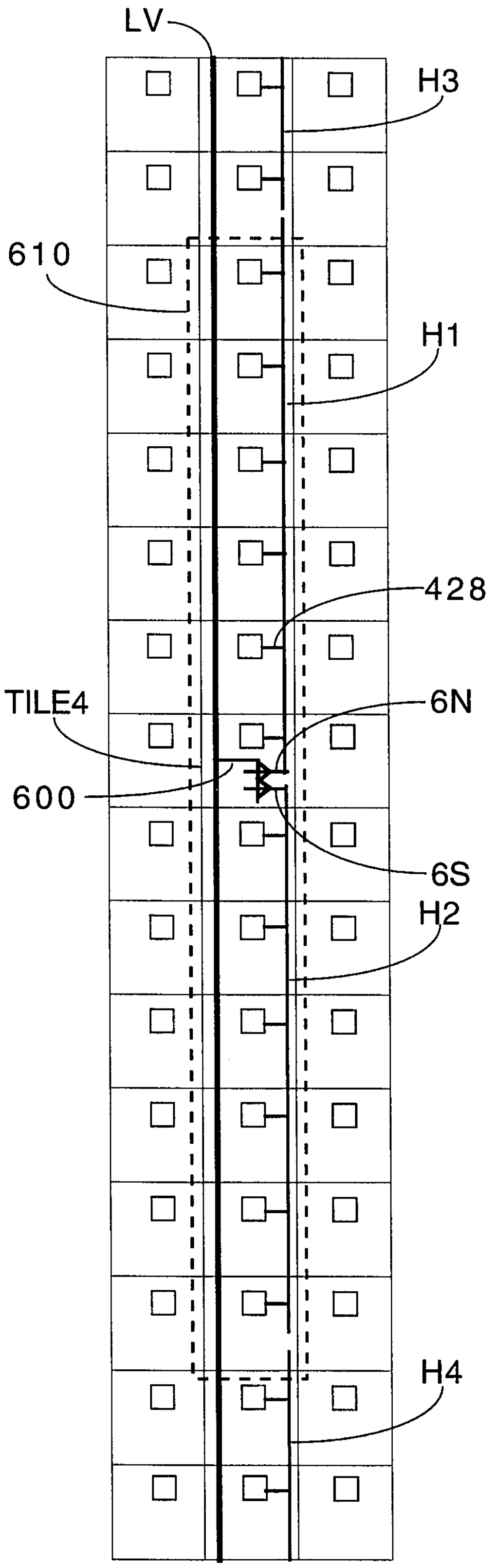

LV
H3
610
H1
428
TILE4
6N
600
6S
H2
H4

FIG. 10

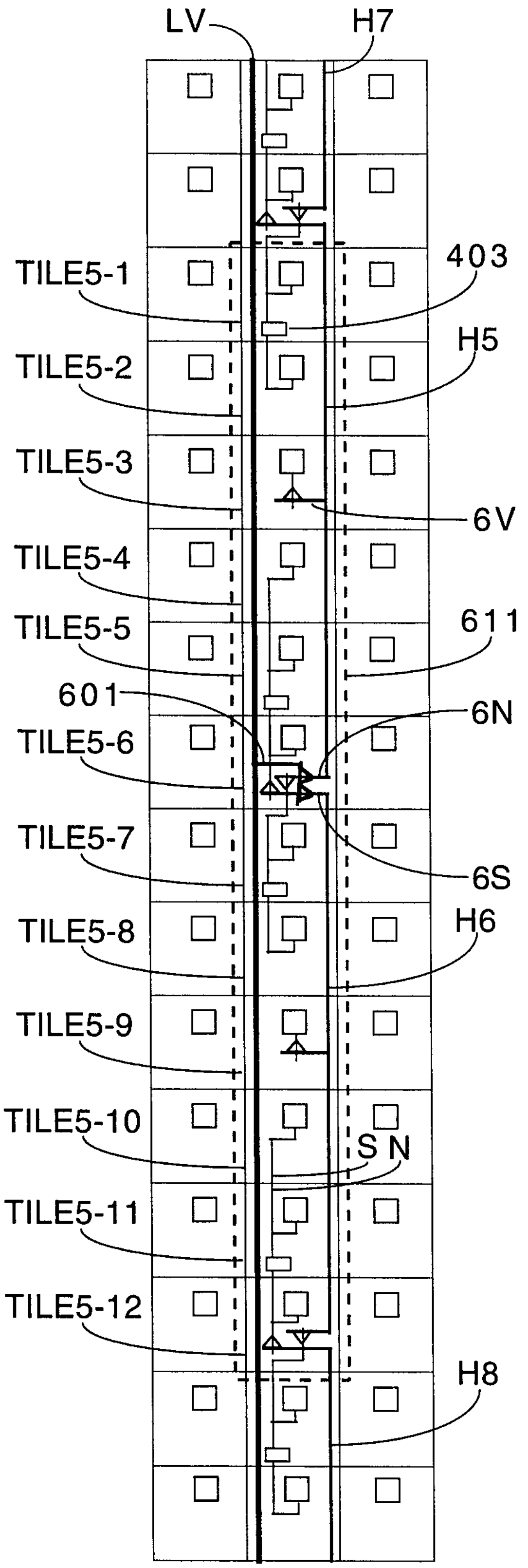

LV
H7
TILE5-1
403
TILE5-2
H5
TILE5-3
6V
TILE5-4
TILE5-5
611
601
6N
TILE5-6
TILE5-7
6S
TILE5-8
H6
TILE5-9
TILE5-10
S N
TILE5-11
TILE5-12
H8

FIG. 11

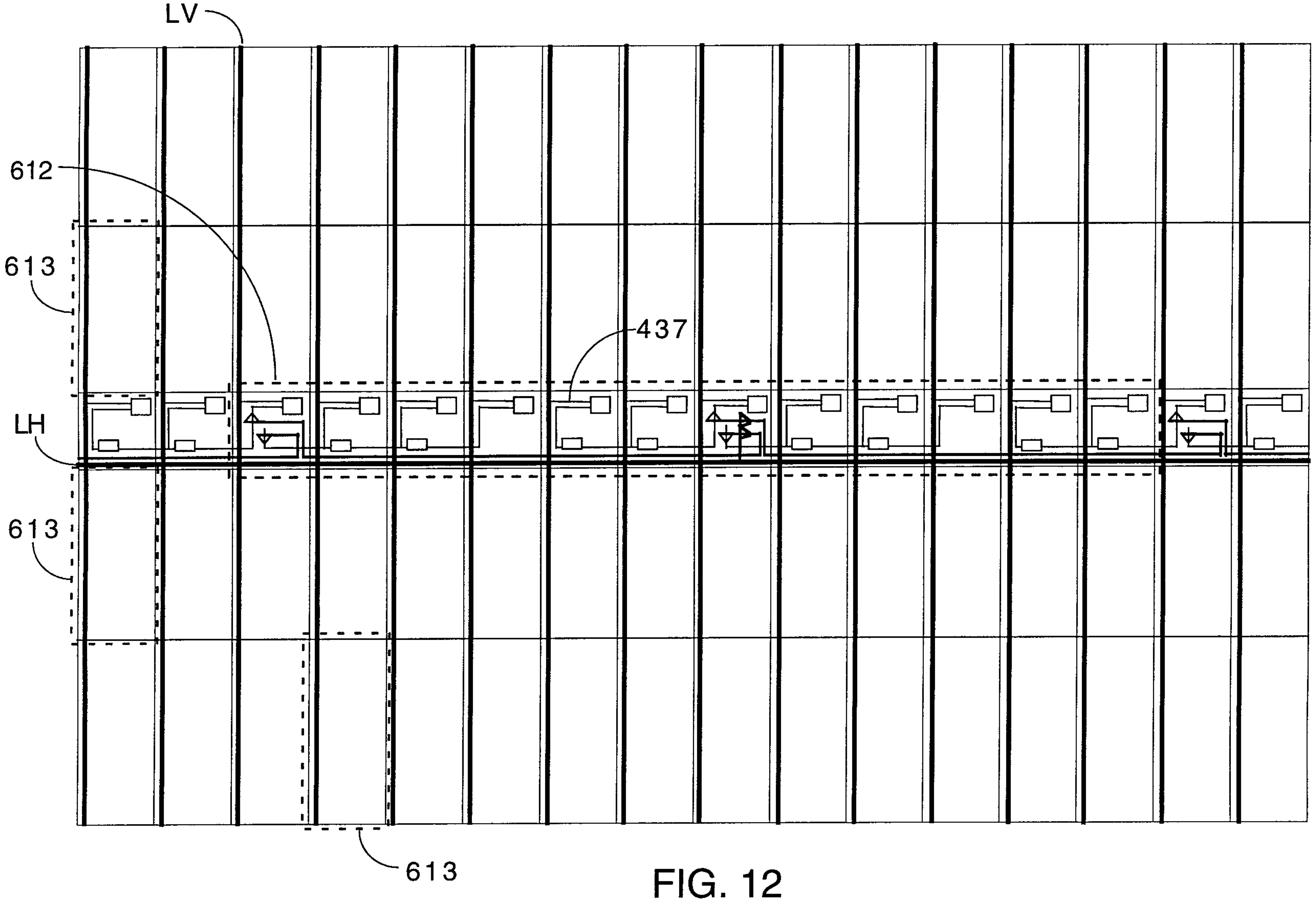


FIG. 12

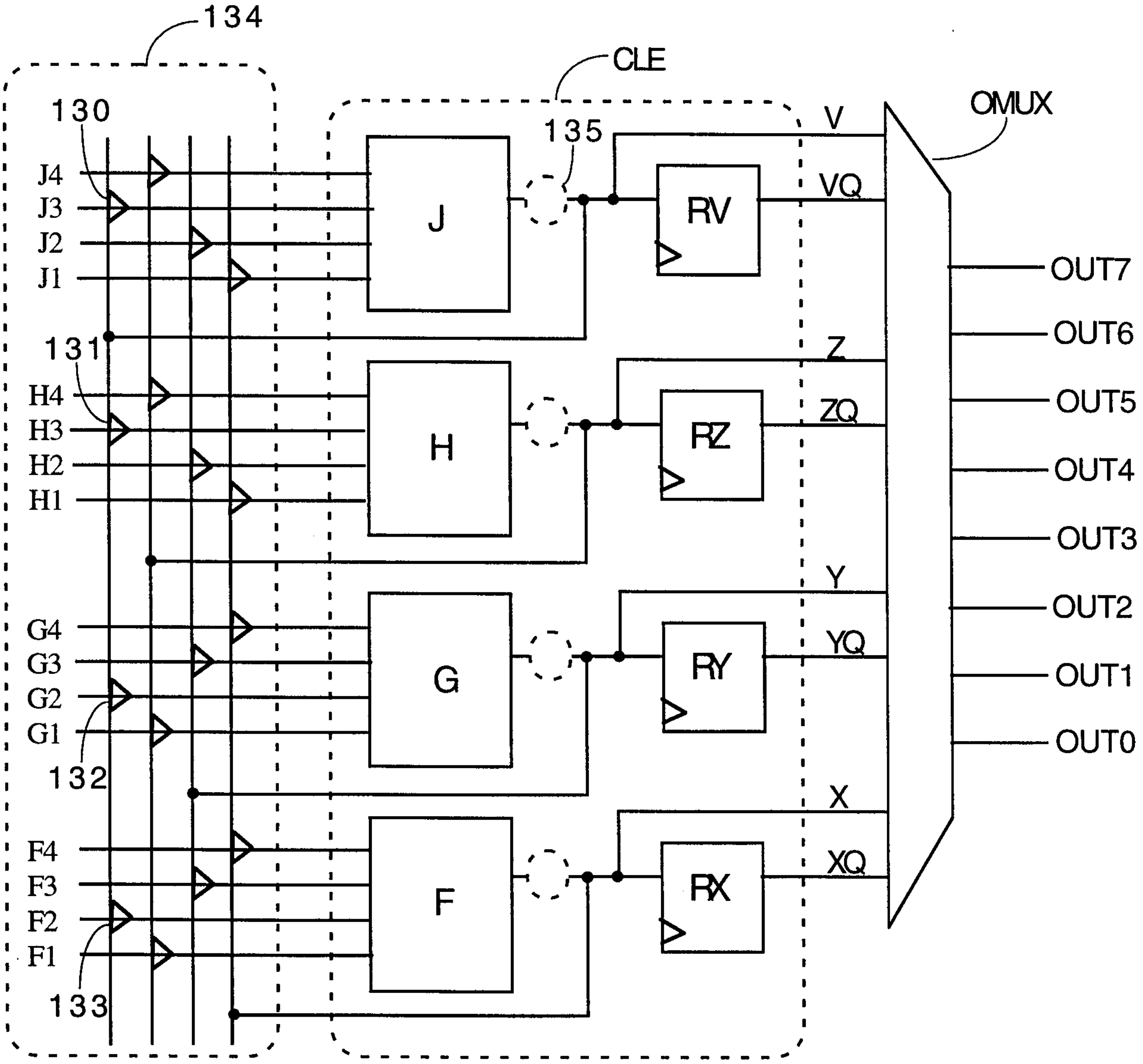

134
CLE
OMUX
130
135
J4
J3
J2
J1
J
RV
V
VQ
131
H4
H3
H2
H1
H
RZ
Z
ZQ
G4
G3
G2
G1
G
RY
Y
YQ
132
F4
F3
F2
F1
F
RX
X
XQ
133
OUT7
OUT6
OUT5
OUT4
OUT3
OUT2
OUT1
OUT0

FIG. 13

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 97/15382

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H03K19/177 H03K19/173

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H03K

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 260 881 A (AGRAWAL OM P ET AL) 9 November 1993	2
A	see column 2, line 65 - column 4, line 24; figures 6,15,50-55	1,3,7,8

A	WO 94 10754 A (XILINX INC) 11 May 1994 see page 21, line 10 - page 25, line 17; figures 1-7,10,12	1-8

A	US 5 543 732 A (MCCLINTOCK CAMERON ET AL) 6 August 1996 see column 3, line 11 - line 49	1-3,7,8

A	GB 2 300 951 A (ALTERA CORP) 20 November 1996 see abstract	1-3,7,8

	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

5

Date of the actual completion of the international search 6 April 1998	Date of mailing of the international search report 14.04.98
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Blaas, D-L

Form PCT/ISA/210 (second sheet) (July 1992)

Inte .ional Application No
PCT/US 97/15382

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	EP 0 746 107 A (IBM) 4 December 1996 see abstract ---	4-6
Y	US 5 581 199 A (PIERCE KERRY M ET AL) 3 December 1996 cited in the application	4-6
A	see column 2, line 8 - line 15; figures 1B,3,4A,4B,5C ---	1-3,7,8
A	US 5 537 057 A (LEONG WILLIAM W ET AL) 16 July 1996 see abstract -----	4-6

5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 97/ 15382

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. claims 1-3,7,8 :
Interconnection of logic blocks within a FPGA
2. claims 4-6:
FPGA logic block with logic elements, output multiplexer and fast feedback

1. ☒ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☒ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 97/15382

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5260881 A	09-11-93	US 5233539 A	03-08-93
		US 5644496 A	01-07-97
		US 5185706 A	09-02-93
		US 5231588 A	27-07-93
		US 5621650 A	15-04-97
WO 9410754 A	11-05-94	CA 2147363 A	11-05-94
		EP 0669056 A	30-08-95
		JP 8503111 T	02-04-96
		US 5469003 A	21-11-95
		US 5500609 A	19-03-96
		US 5528176 A	18-06-96
		US 5552722 A	03-09-96
		US 5670897 A	23-09-97
US 5543732 A	06-08-96	GB 2300949 A	20-11-96
GB 2300951 A	20-11-96	US 5592106 A	07-01-97
EP 0746107 A	04-12-96	US 5646546 A	08-07-97
		JP 8340248 A	24-12-96
US 5581199 A	03-12-96	EP 0721256 A	10-07-96
		JP 8265136 A	11-10-96
US 5537057 A	16-07-96	US 5598109 A	28-01-97