US00RE45110E

(19) **United States**

(12) **Reissued Patent**

Madurawe et al.

(10) **Patent Number: US RE45,110 E**

(45) **Date of Reissued Patent: Sep. 2, 2014**

(54) **MPGA PRODUCTS BASED ON A PROTOTYPE FPGA**

(76) Inventors: **Raminda Udaya Madurawe**, Sunnyvale, CA (US); **Peter Ramyalal Suaris**, Danville, CA (US); **Thomas Henry White**, Santa Clara, CA (US)

(21) Appl. No.: **13/411,486**

(22) Filed: **Mar. 2, 2012**
(Under 37 CFR 1.47)

Related U.S. Patent Documents

Reissue of:

(64) Patent No.: **7,673,273**
Issued: **Mar. 2, 2010**
Appl. No.: **11/712,380**
Filed: **Mar. 1, 2007**

U.S. Applications:

(63) Continuation-in-part of application No. 11/645,313, filed on Dec. 26, 2006, now Pat. No. 7,627,848, which is a continuation of application No. 11/384,116, filed on Mar. 20, 2006, now Pat. No. 7,356,799.

(51) **Int. Cl.**
G06F 17/50 (2006.01)

(52) **U.S. Cl.**
USPC **716/138**; 716/116; 716/117

(58) **Field of Classification Search**
CPC G06G 17/505; G06F 17/5045; G06F 17/5068; G06F 17/5072; G06F 17/5027
USPC 716/116, 117, 128, 138; 326/37–41
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,609,986 A 9/1986 Hartmann et al.
4,706,216 A 11/1987 Carter
4,761,768 A 8/1988 Turner et al.
4,864,161 A 9/1989 Norman et al.
4,870,302 A 9/1989 Freeman
4,873,459 A 10/1989 El Gamal et al.
5,164,612 A 11/1992 Kaplinsky
5,191,241 A 3/1993 McCollum et al.
5,216,636 A 6/1993 Runaldue
5,343,406 A 8/1994 Freeman et al.
5,347,519 A 9/1994 Cooke et al.
5,488,316 A 1/1996 Freeman et al.
5,563,526 A 10/1996 Hastings et al.
5,581,501 A 12/1996 Sansbury et al.

(Continued)

OTHER PUBLICATIONS

Ashok K. Sharma, "Programmable Logic Handbook—PLDs, CPLDs, & FPGAs", 1998, pp. 99-171, McGraw-Hill, USA.

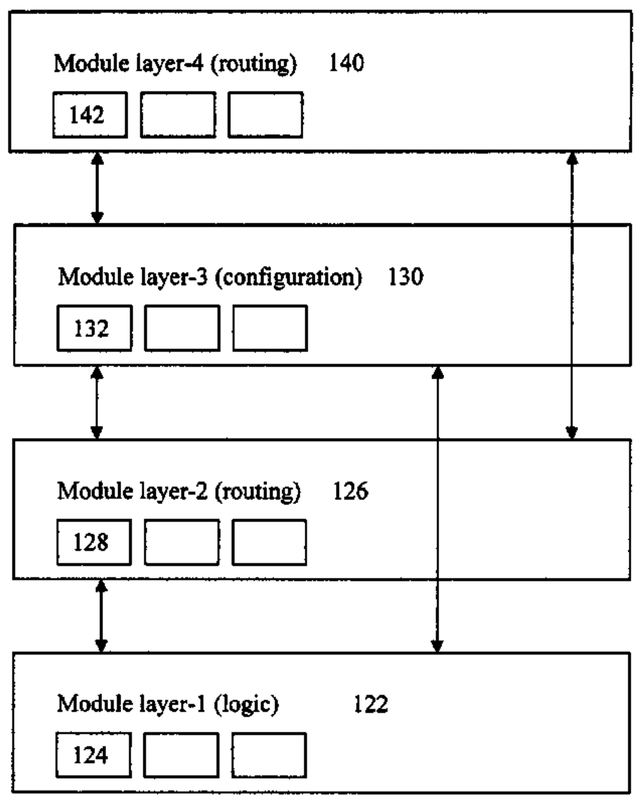
(Continued)

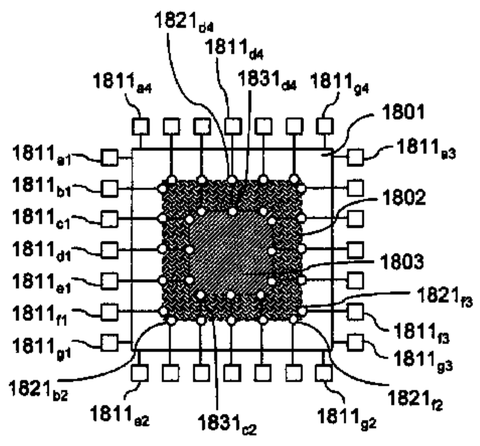
Primary Examiner — Vuthe Siek

(57) **ABSTRACT**

A smaller mask programmable gate array (MPGA) device derived from a larger field programmable gate array (FPGA), comprising: a layout of transistors and a plurality of interconnect layers substantially identical to a smaller region of the FPGA; and input/output pads matching a subset of the input/output pads of the FPGA; wherein, a design that is mapped to said smaller region of the FPGA device using said subset of input/output pads by a user programmable means can be identically mapped to the MPGA by a hard-wire circuit. Such a gate array further comprises a mask programmable metal-circuit in lieu of a user programmable configuration circuit of the FPGA; and a logic block to input/output pad connection in lieu of a logic block to a register at the boundary of said smaller region to an input/output pad connection of the FPGA.

31 Claims, 11 Drawing Sheets





(56) **References Cited**

U.S. PATENT DOCUMENTS

5,612,631	A		3/1997	Agrawal et al.
5,625,221	A		4/1997	Kim et al.
5,679,967	A		10/1997	Janai et al.
5,684,744	A		11/1997	Orgill et al.
5,701,233	A		12/1997	Carson et al.
5,754,826	A	*	5/1998	Gamal et al. 703/14
5,781,031	A		7/1998	Carson et al.
5,793,115	A		8/1998	Zavracky et al.
5,835,405	A		11/1998	Tsui et al.
5,844,422	A		12/1998	Trimberger et al.
5,880,598	A		3/1999	Duong
5,943,574	A		8/1999	Tehrani et al.
5,949,710	A		9/1999	Pass et al.
5,949,719	A		9/1999	Clinton et al.
6,005,806	A		12/1999	Madurawe et al.
6,018,476	A		1/2000	Madurawe et al.
6,097,211	A		8/2000	Couts-Martin et al.
6,134,171	A		10/2000	Yamagata et al.
6,134,173	A		10/2000	Cliff et al.
6,191,614	B1		2/2001	Schultz et al.
6,242,767	B1		6/2001	How et al.
6,259,271	B1		7/2001	Couts-Martin et al.
6,262,596	B1		7/2001	Schultz et al.
6,275,064	B1		8/2001	Agrawal et al.
6,275,065	B1		8/2001	Mendel
6,331,784	B1		12/2001	Mason et al.
6,331,789	B2		12/2001	Or-Bach
6,337,579	B1		1/2002	Mochida
6,340,830	B1		1/2002	Takemura
6,353,562	B2		3/2002	Bohm et al.
6,420,925	B1		7/2002	Fifield et al.
6,426,649	B1		7/2002	Fu et al.
6,445,065	B1	*	9/2002	Gheewala et al. 257/691
6,448,808	B2		9/2002	Young et al.
6,480,027	B1		11/2002	Ngai et al.
6,480,954	B2		11/2002	Trimberger et al.
6,496,887	B1		12/2002	Plants
6,504,398	B1		1/2003	Lien et al.
6,504,742	B1		1/2003	Tran et al.
6,515,511	B2		2/2003	Sugibayashi et al.
6,525,953	B1		2/2003	Johnson
6,551,857	B2		4/2003	Leedy
6,582,980	B2		6/2003	Feldman et al.
6,613,611	B1		9/2003	How et al.
6,614,259	B2		9/2003	Couts-Martin et al.
6,617,621	B1	*	9/2003	Gheewala et al. 257/207
6,627,985	B2		9/2003	Huppenthal et al.
6,633,181	B1		10/2003	Rupp
6,737,675	B2		5/2004	Patel et al.
6,738,962	B1		5/2004	Flaherty et al.
6,798,240	B1		9/2004	Pedersen
6,812,731	B1		11/2004	Trimberger
6,911,730	B1		6/2005	New
6,946,330	B2		9/2005	Yamazaki et al.
6,954,084	B2		10/2005	Islam
6,992,503	B2		1/2006	Madurawe
6,998,722	B2		2/2006	Madurawe
7,019,557	B2		3/2006	Madurawe
7,030,651	B2		4/2006	Madurawe
7,064,018	B2		6/2006	Madurawe
7,064,579	B2		6/2006	Madurawe
7,084,666	B2		8/2006	Madurawe
7,112,994	B2		9/2006	Madurawe
7,176,713	B2		2/2007	Madurawe
7,253,659	B2		8/2007	Madurawe
7,268,580	B2		9/2007	Madurawe
2001/0003428	A1		6/2001	Or-Bach
2001/0019155	A1		9/2001	Warashina et al.
2001/0028059	A1		10/2001	Emma et al.
2001/0047509	A1		11/2001	Mason et al.
2002/0073380	A1	*	6/2002	Cooke et al. 716/1
2002/0177260	A1		11/2002	Matsumoto
2002/0186044	A1		12/2002	Agrawal et al.
2003/0001615	A1		1/2003	Sueyoshi et al.
2003/0023762	A1		1/2003	Dhir et al.
2003/0085733	A1		5/2003	Pugh et al.
2003/0227056	A1		12/2003	Wang et al.
2004/0178819	A1		9/2004	New
2004/0268286	A1	*	12/2004	New et al. 716/14
2005/0023656	A1		2/2005	Leedy
2006/0195729	A1		8/2006	Huppenthal

OTHER PUBLICATIONS

Alexander, et al., "Three-Dimensional Field-Programmable Gate Arrays", Proceedings of the 8.sup.th Annual IEEE International ASIC Conference and Exhibit, 1995, pp. 253-256.

Chen Dong et al., "3-D nFPGA: A reconfigurable architecture for 3-D CMOS/Nanomaterial Hybrid Digital Circuits", IEEE Trans. Circuits and Systems, vol. 54, No. 11, Nov. 1, 2007 (pp. 2489-2501).

* cited by examiner

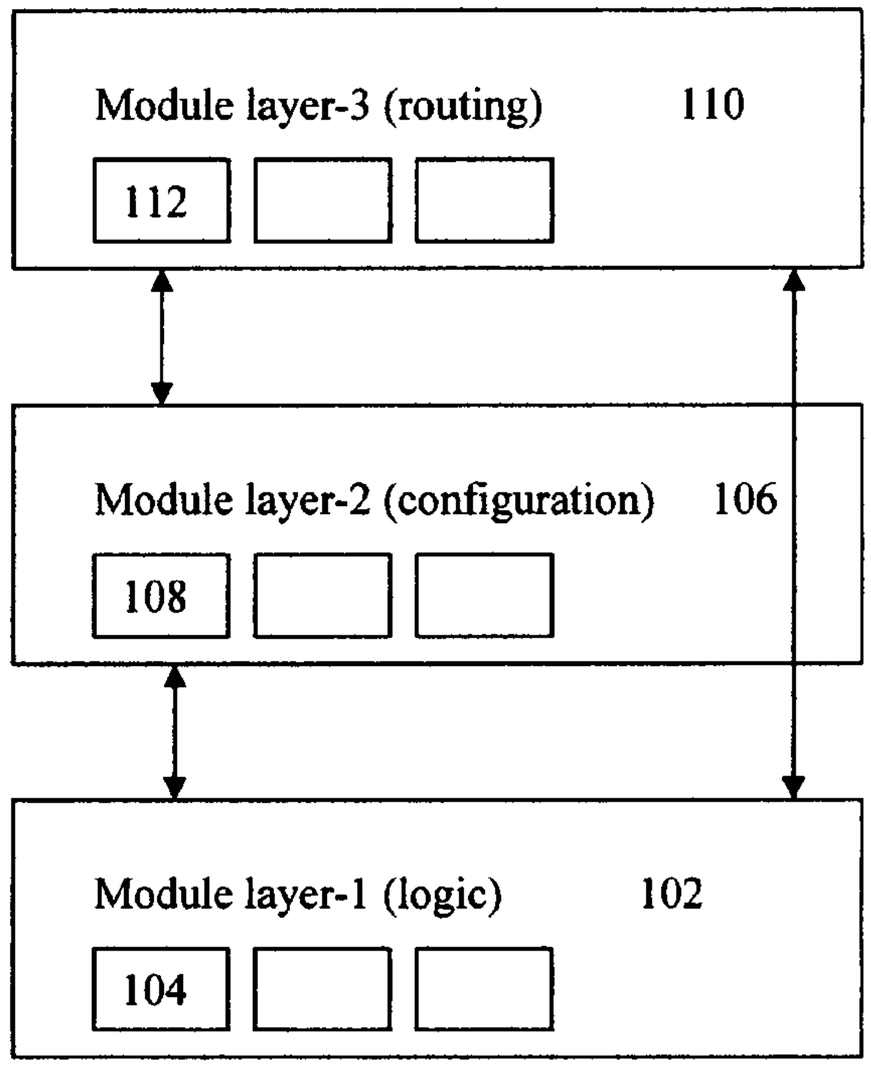

Module layer-3 (routing) 110
112
Module layer-2 (configuration) 106
108
Module layer-1 (logic) 102
104

Fig. 1

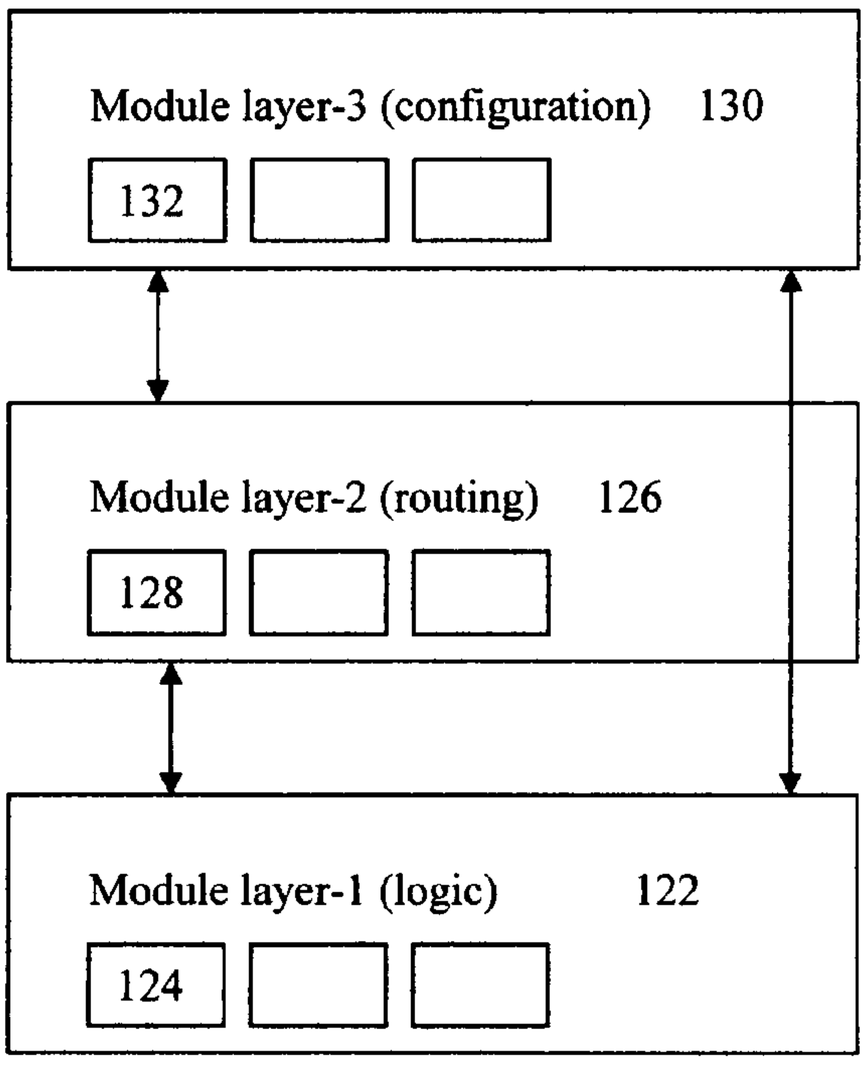

Module layer-3 (configuration) 130
132
Module layer-2 (routing) 126
128
Module layer-1 (logic) 122
124

Fig. 2

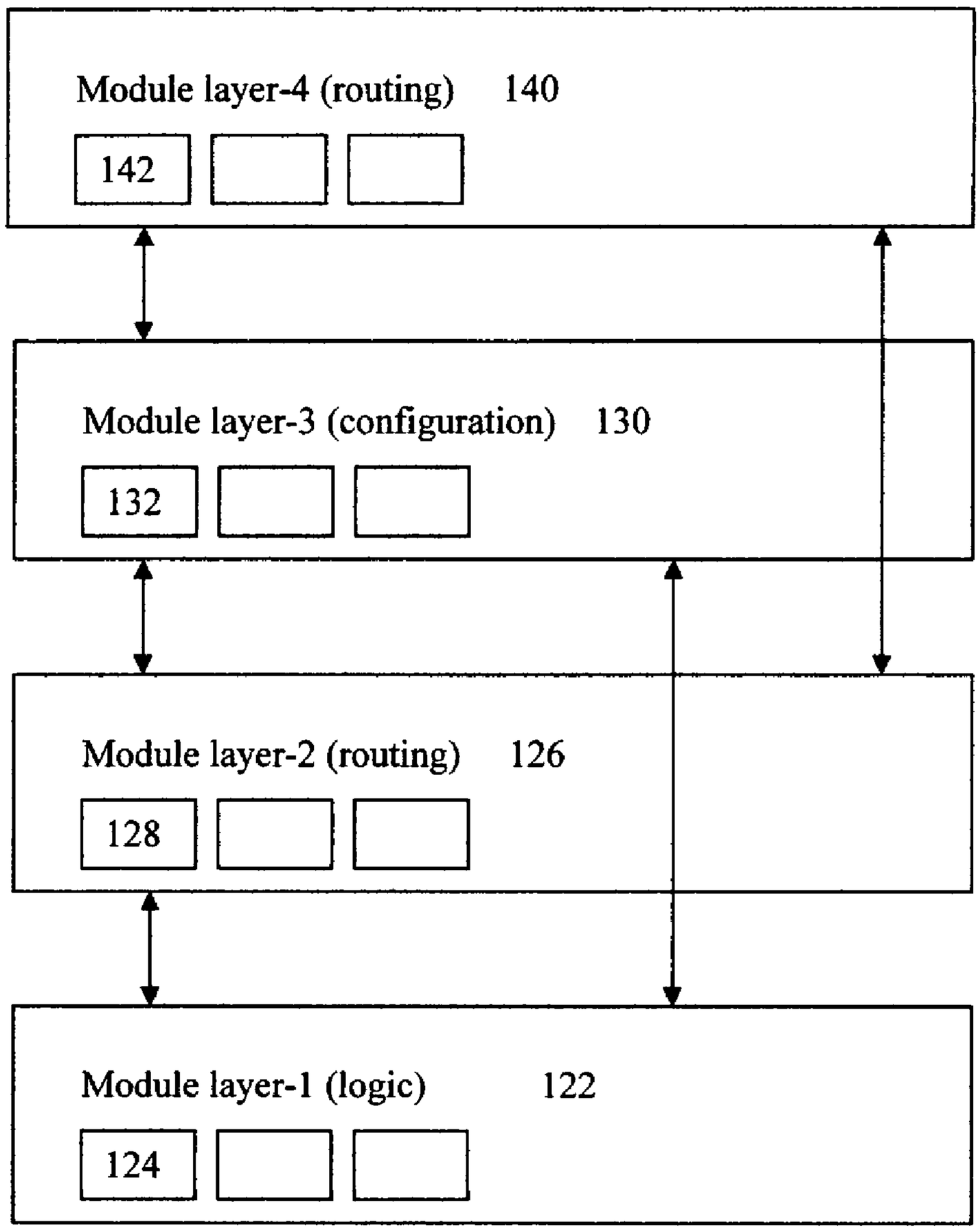

Module layer-4 (routing) 140
142
Module layer-3 (configuration) 130
132
Module layer-2 (routing) 126
128
Module layer-1 (logic) 122
124

Fig. 3

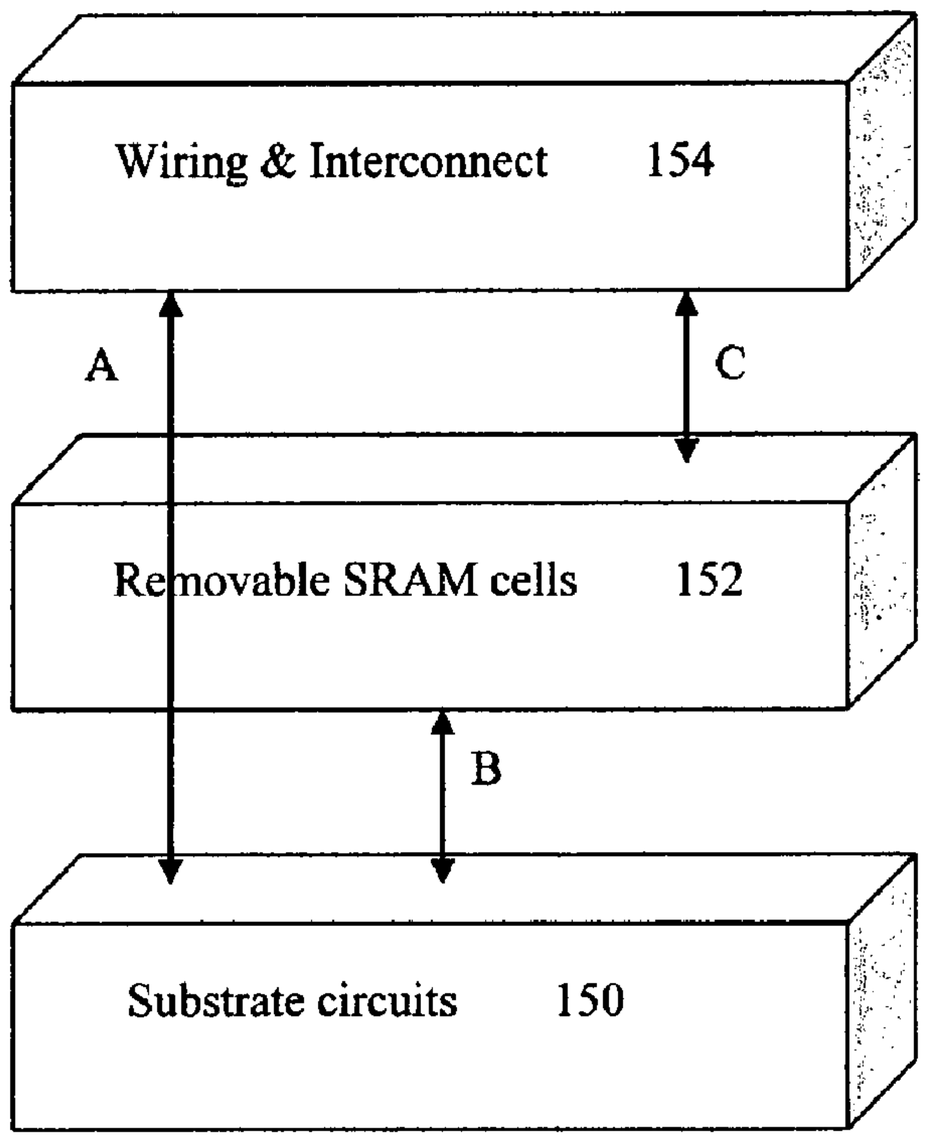

Wiring & Interconnect 154
A
C
Removable SRAM cells 152
B
Substrate circuits 150

Fig. 4

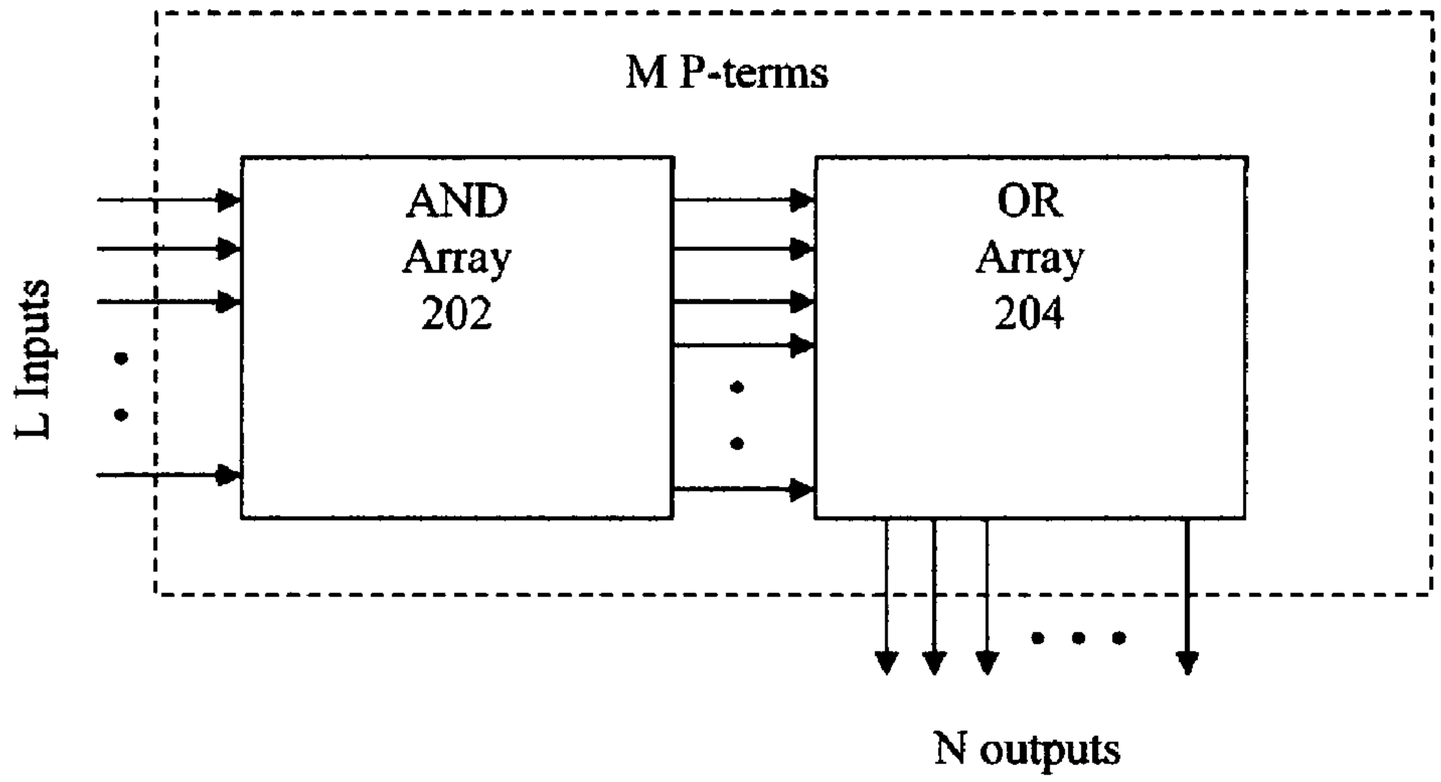
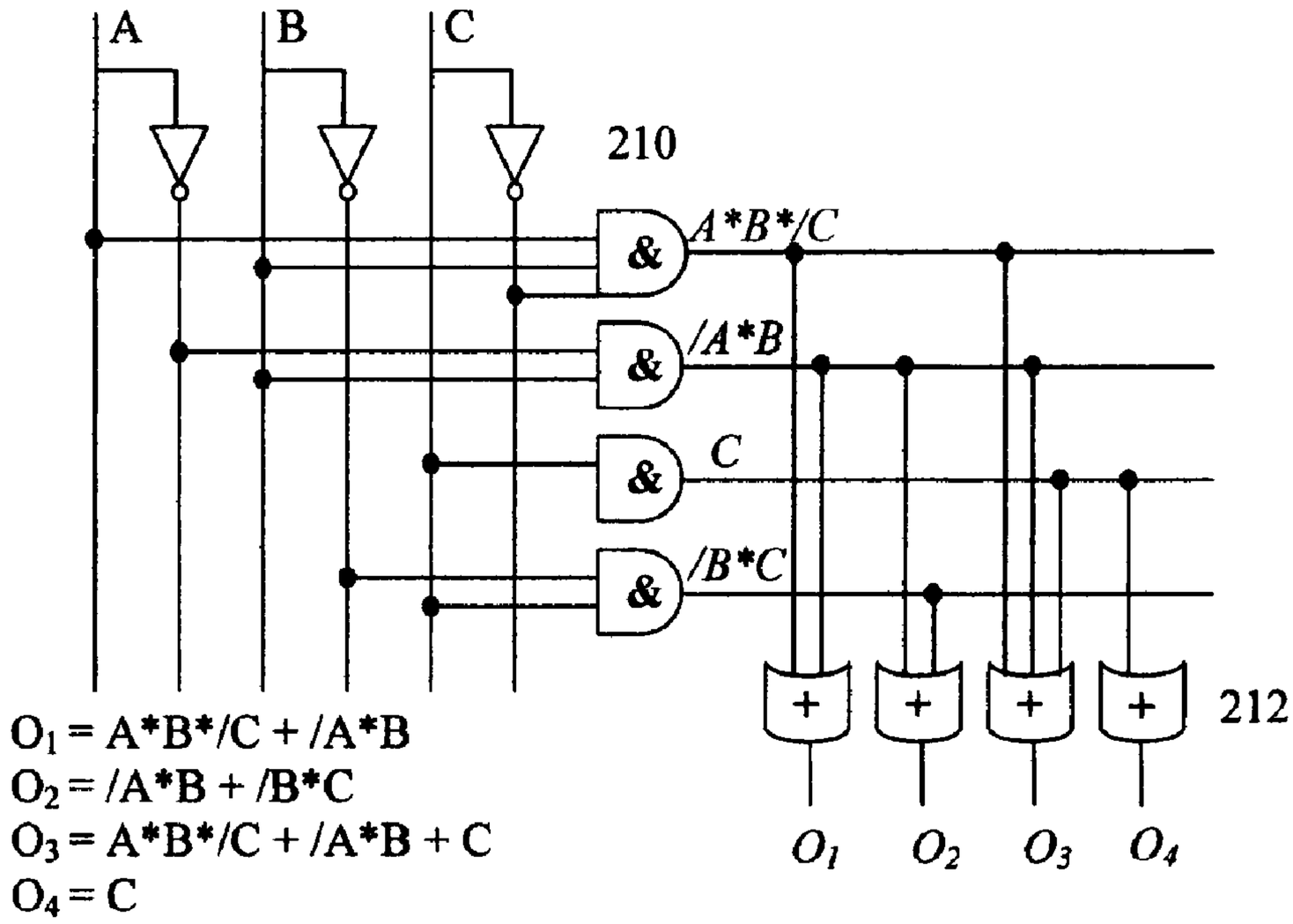

M P-terms
L Inputs
AND
Array
202
OR
Array
204
N outputs

Fig. 5



$O_1 = A*B*/C + /A*B$
$O_2 = /A*B + /B*C$
$O_3 = A*B*/C + /A*B + C$
$O_4 = C$

Fig. 6

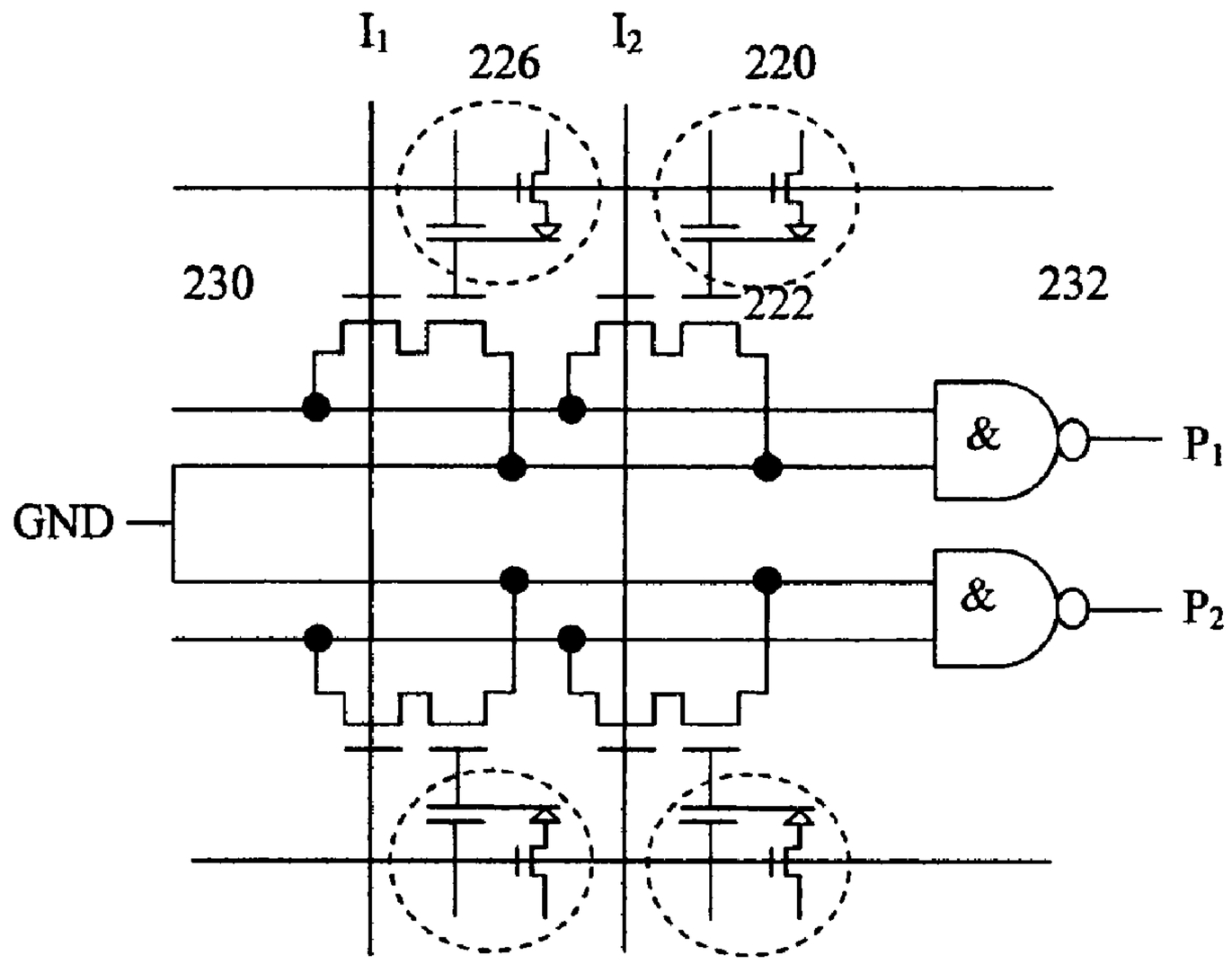


Fig. 7

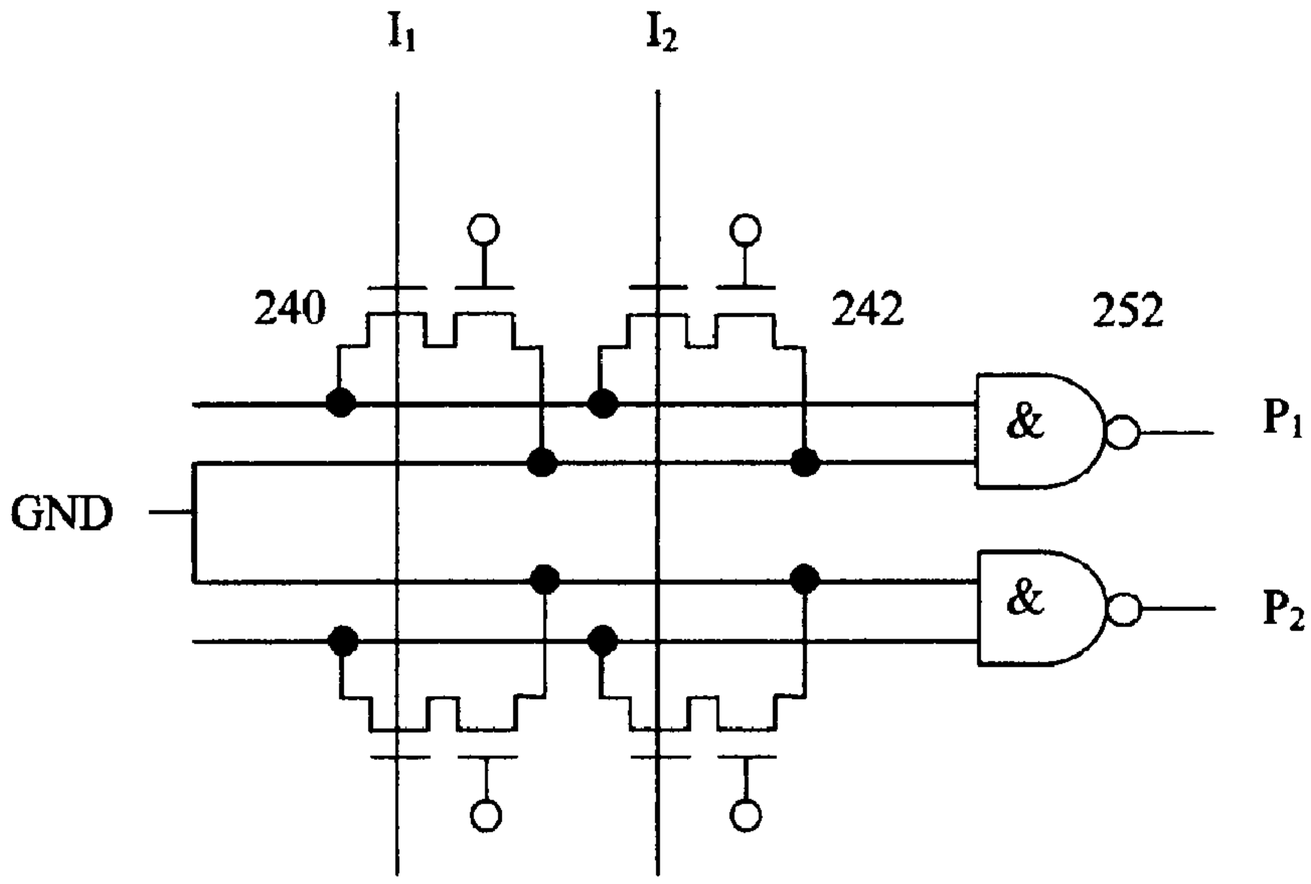


Fig. 8

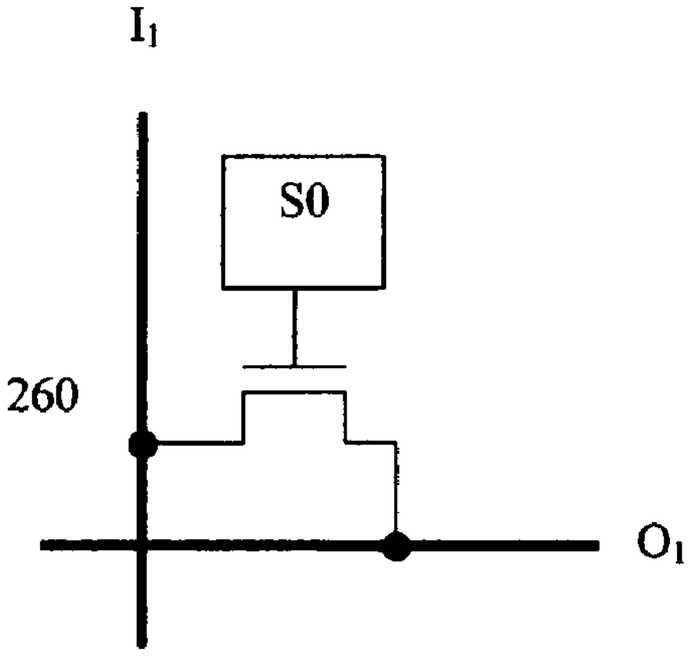


Fig. 9

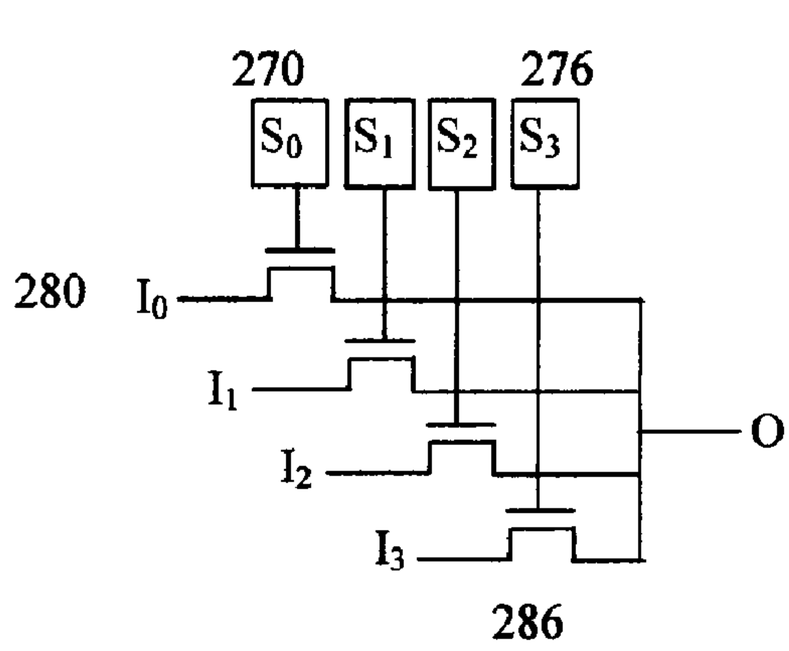


Fig. 10

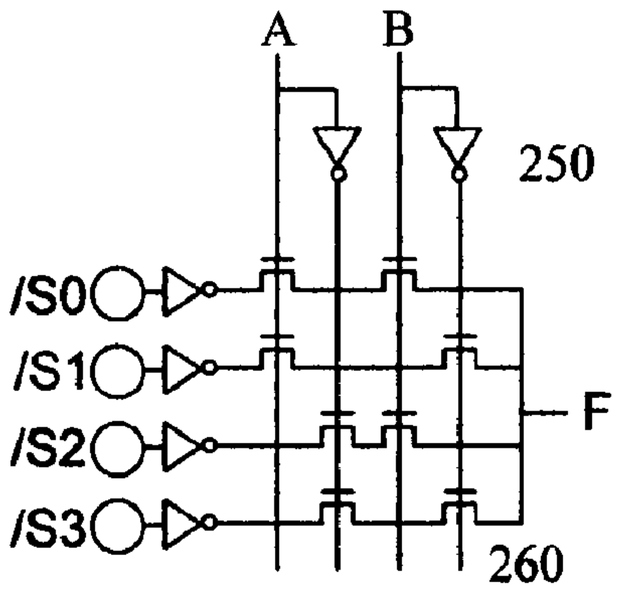

A
B
250
/S0
/S1
/S2
/S3
F
260

Fig. 11

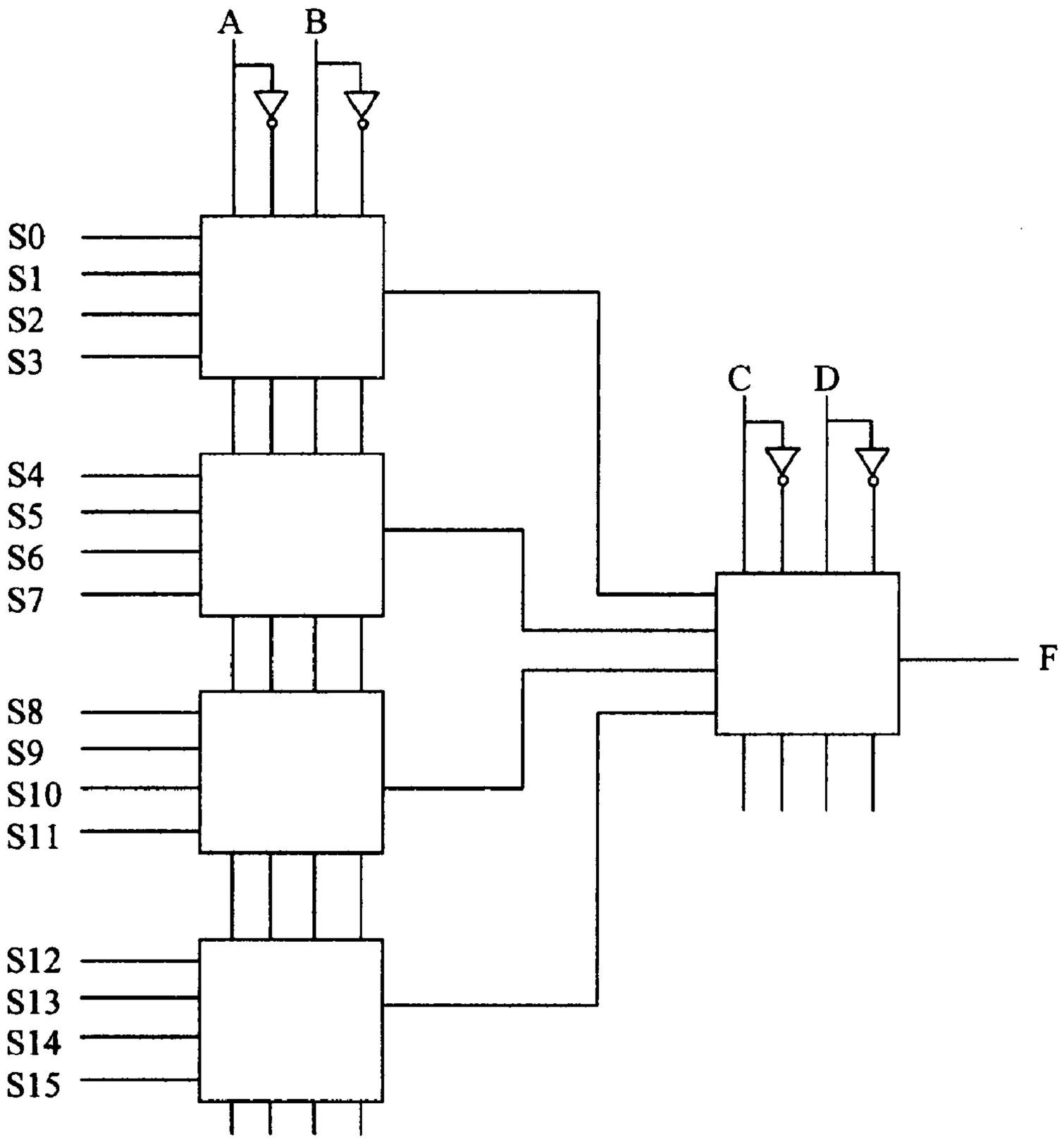

A
B
S0
S1
S2
S3
C
D
S4
S5
S6
S7
F
S8
S9
S10
S11
S12
S13
S14
S15

Fig. 12

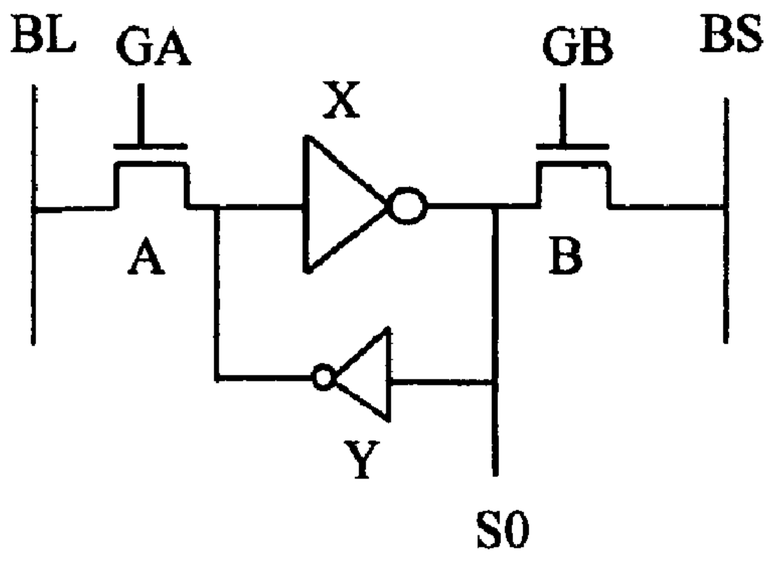


Fig. 13

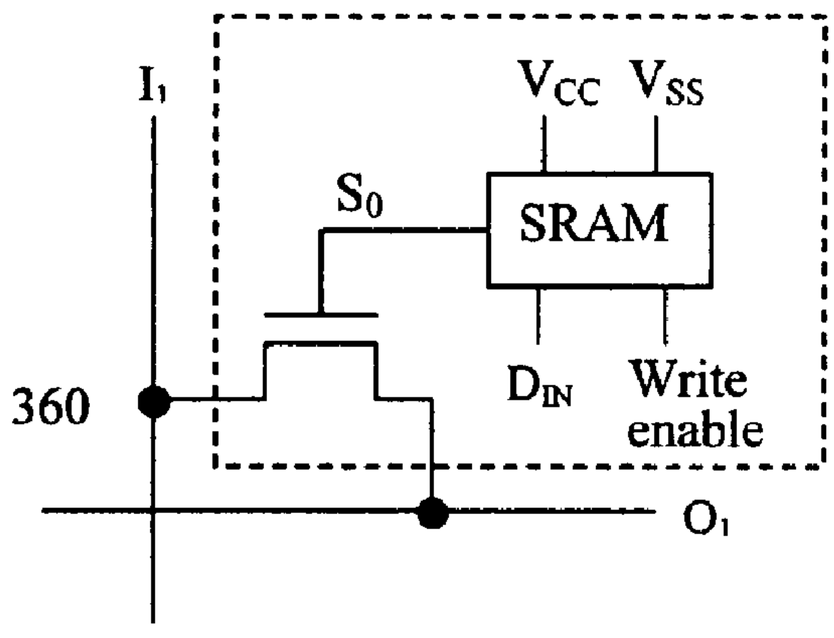


Fig. 14

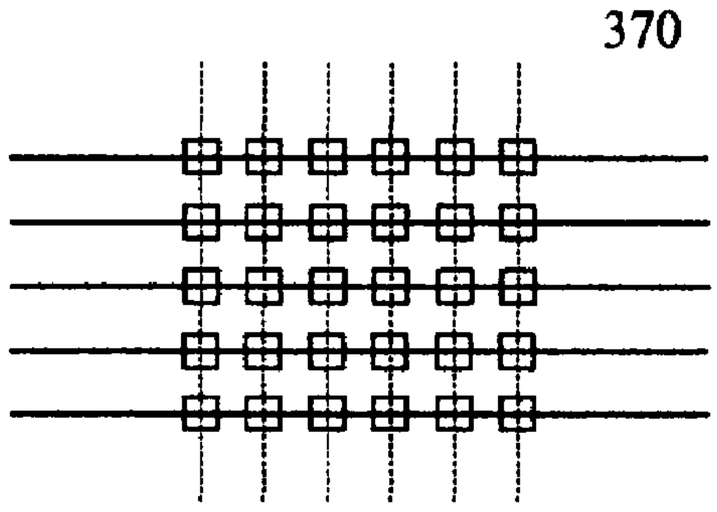


Fig. 15

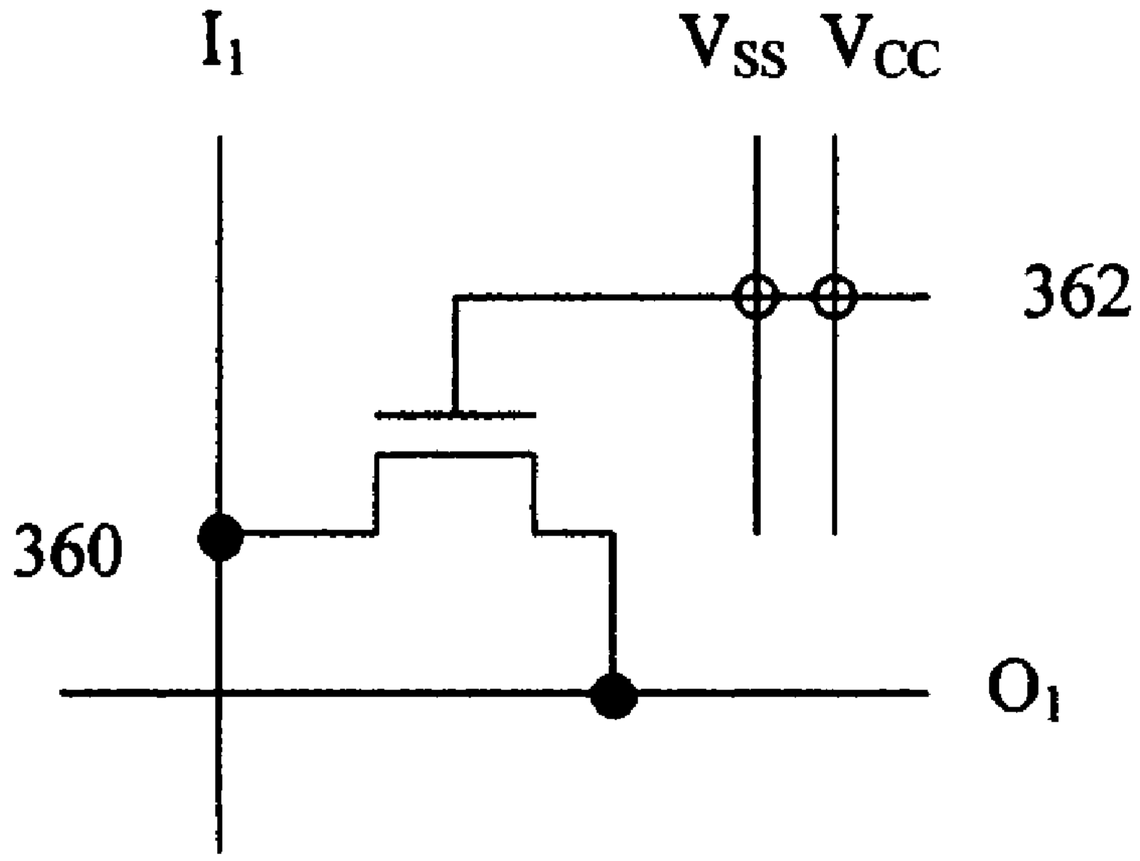


Fig. 16

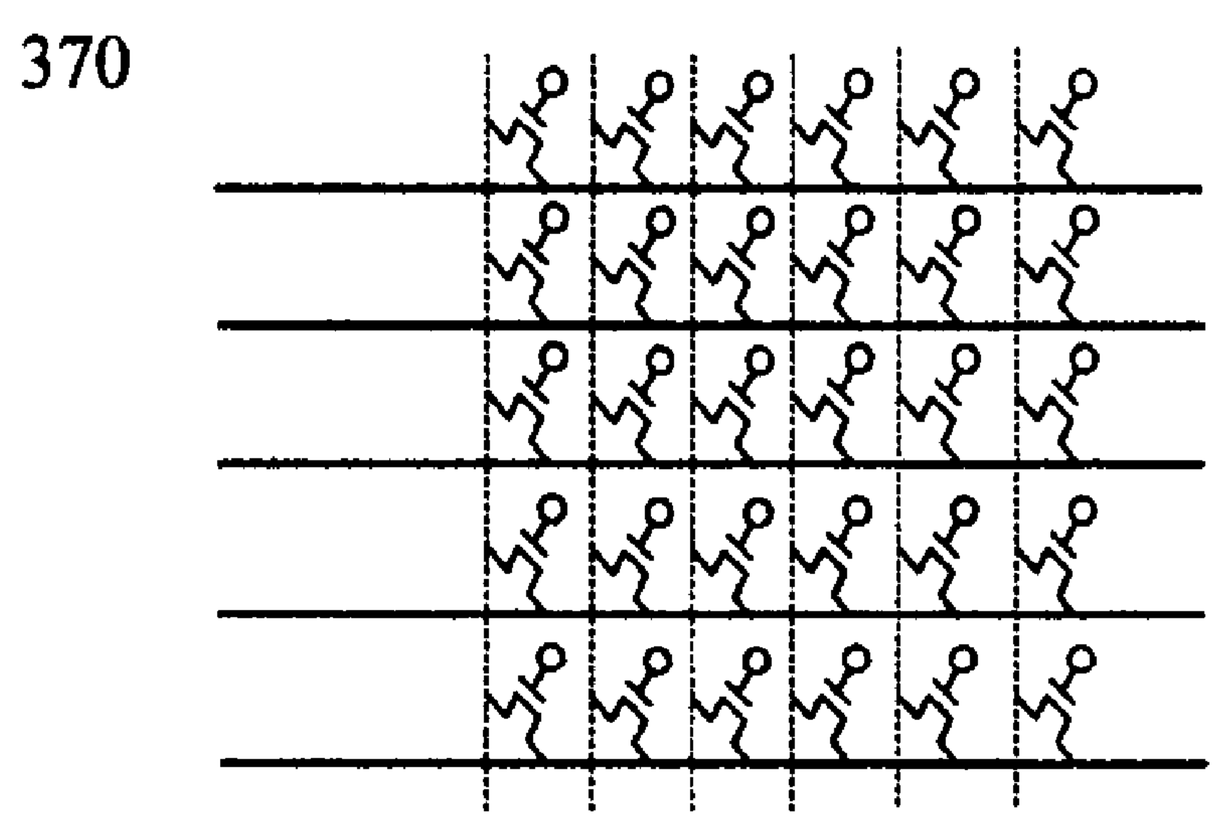


Fig. 17

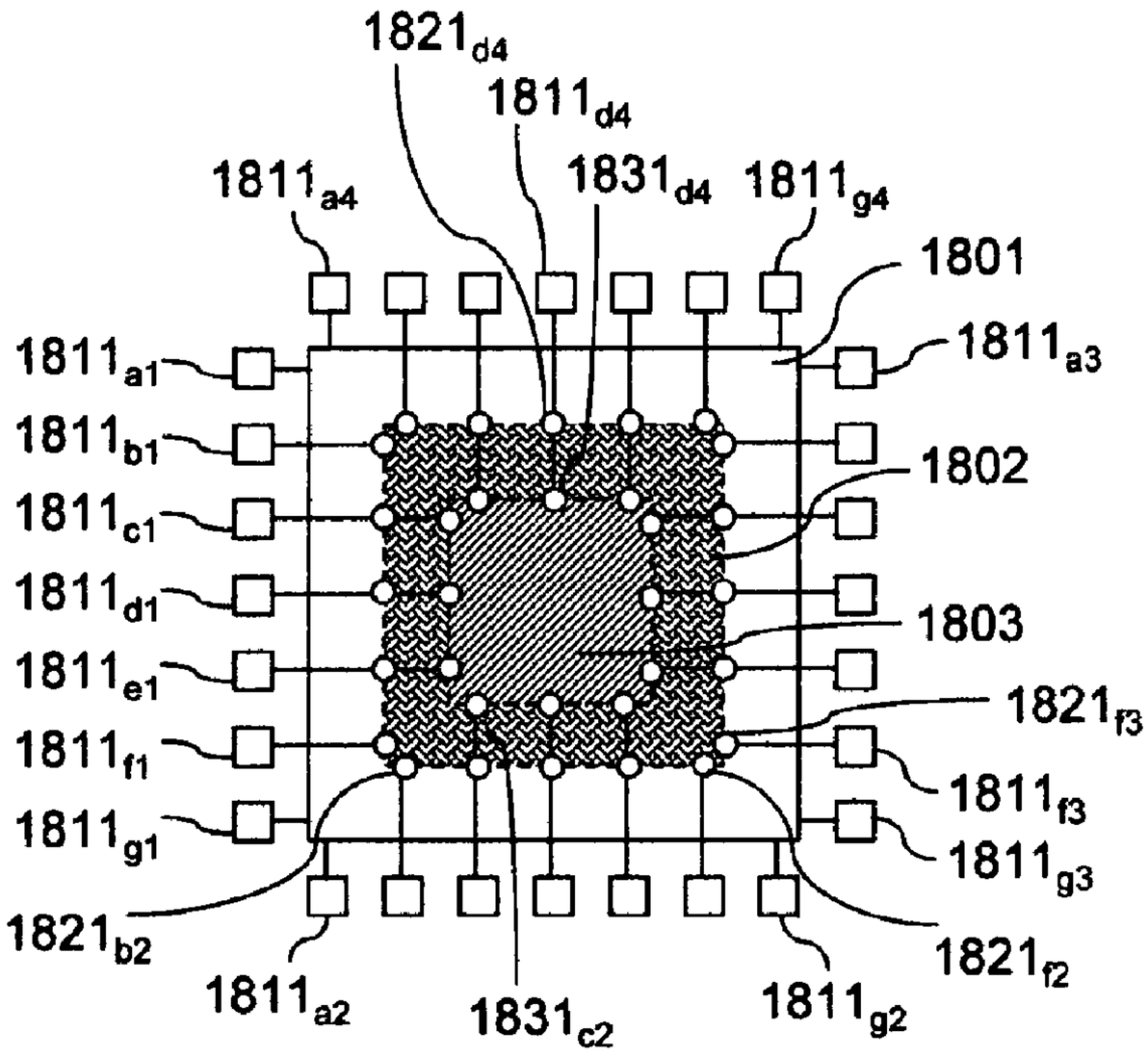


Fig-18A

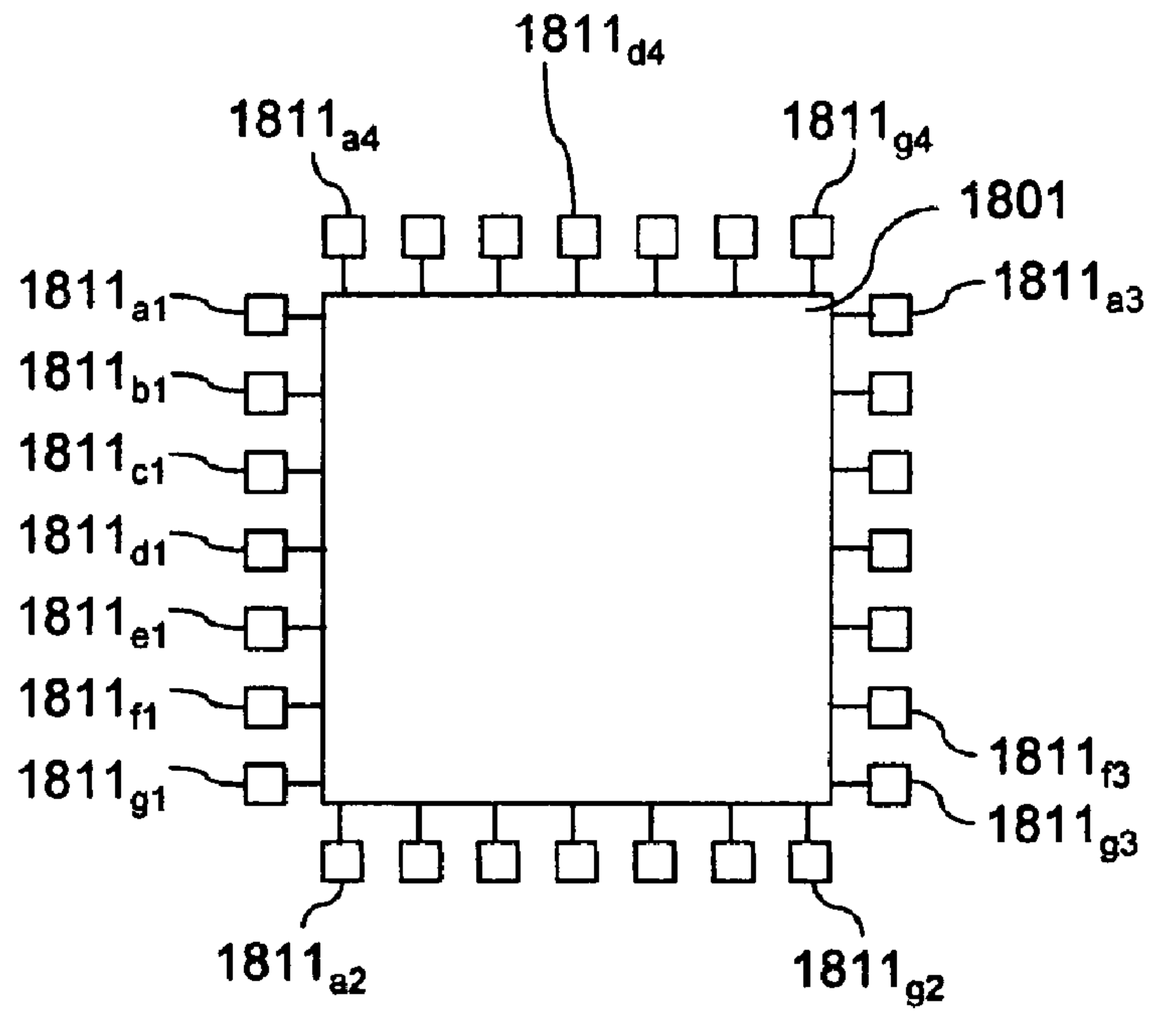


Fig-18B

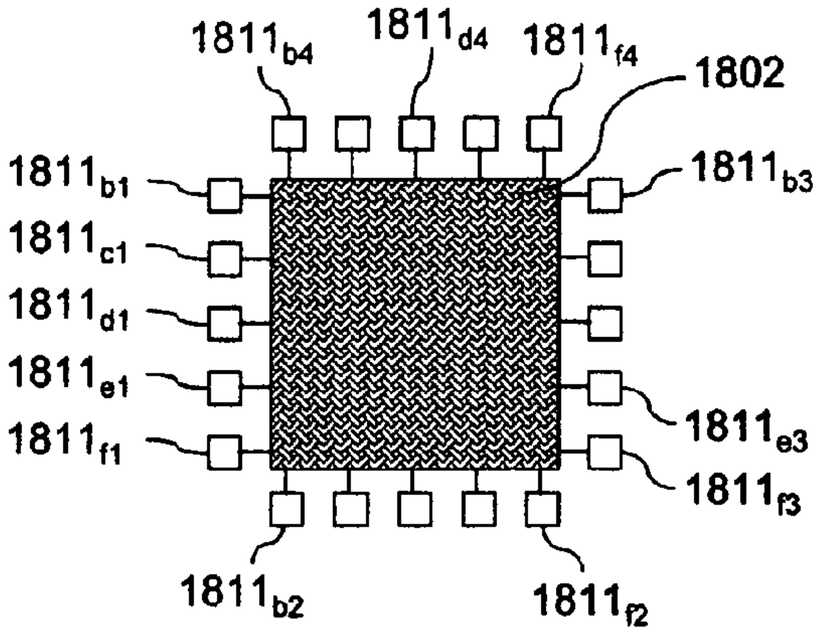


Fig-18C

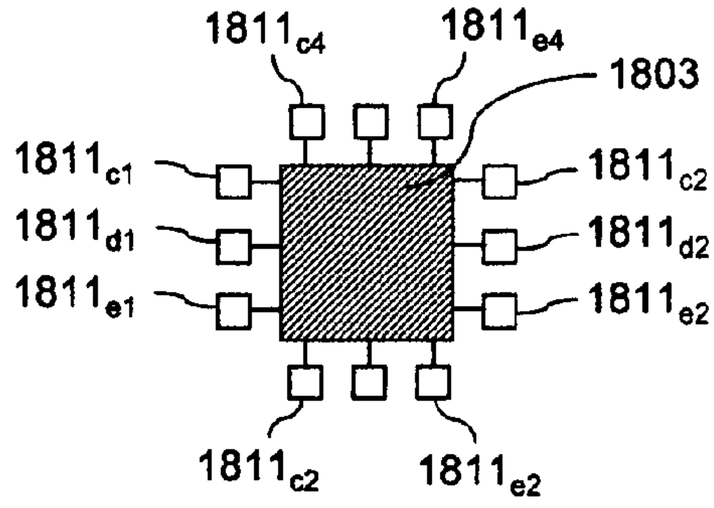


Fig-18D

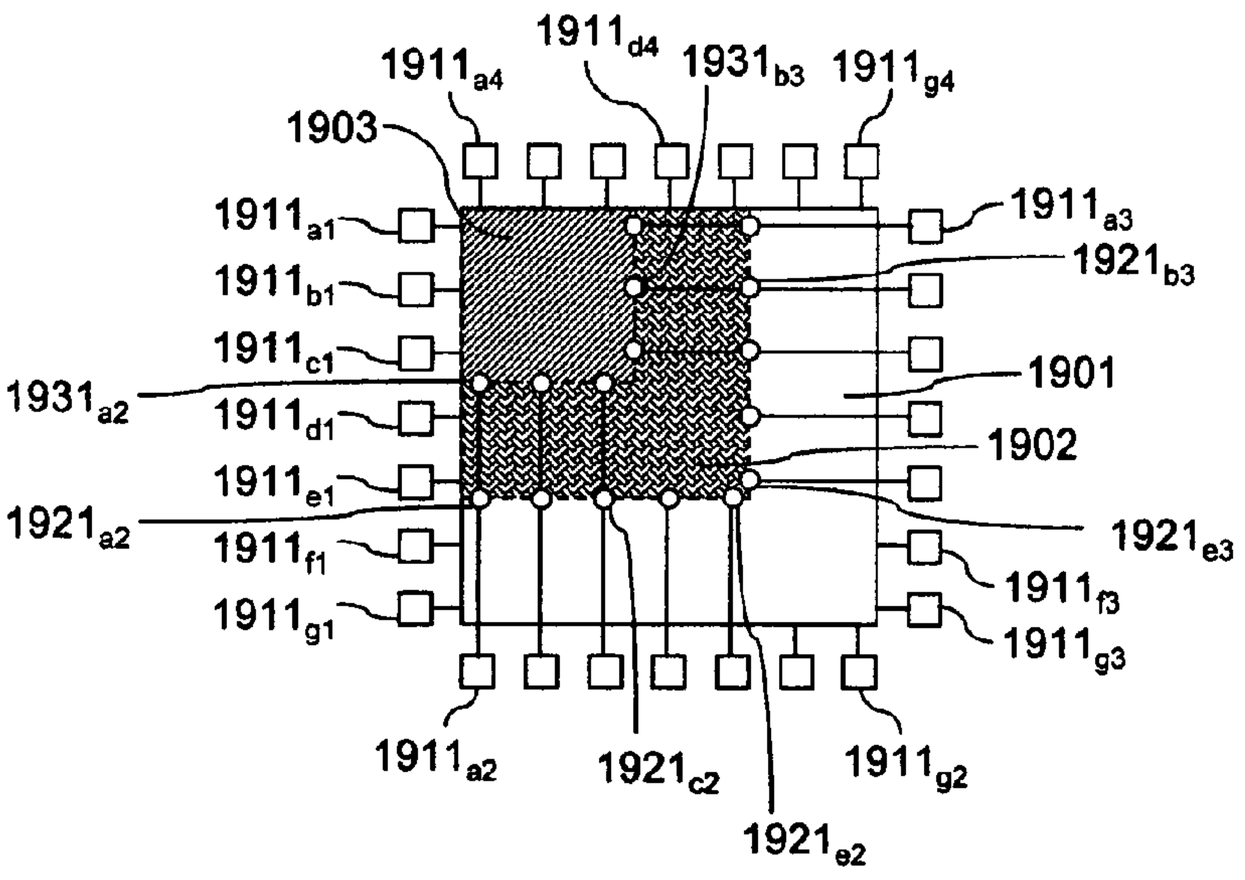


Fig-19A

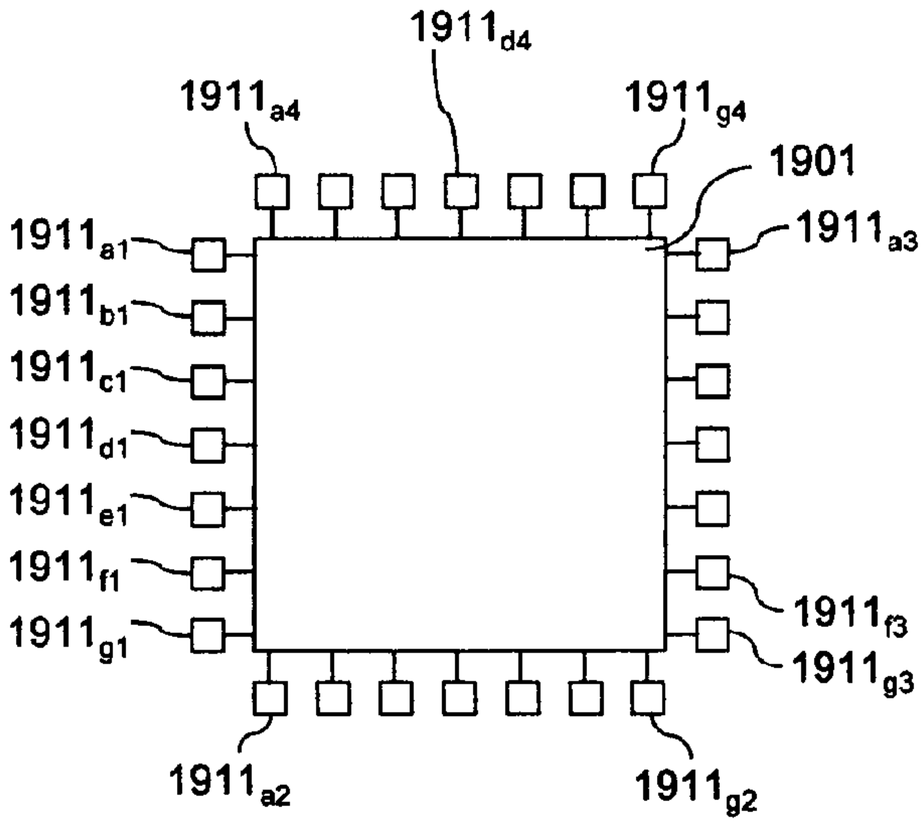


Fig-19B

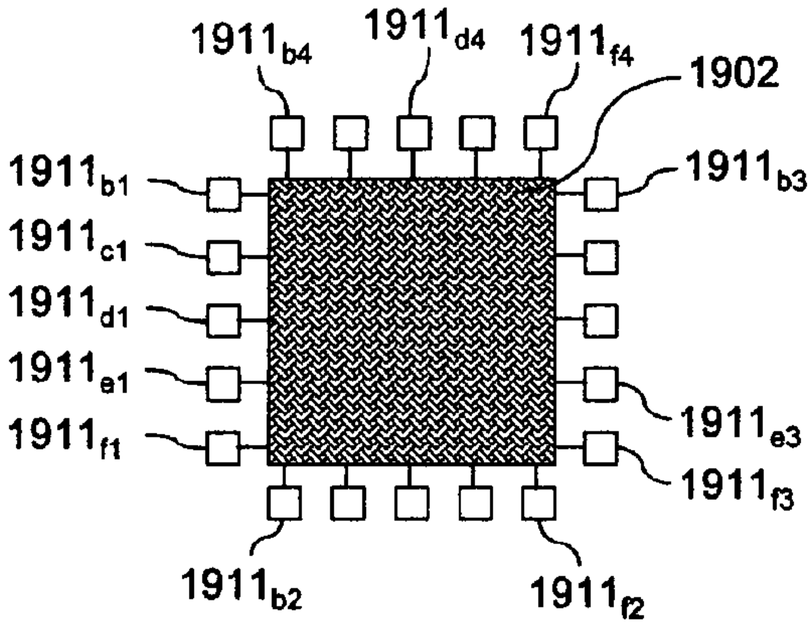


Fig-19C

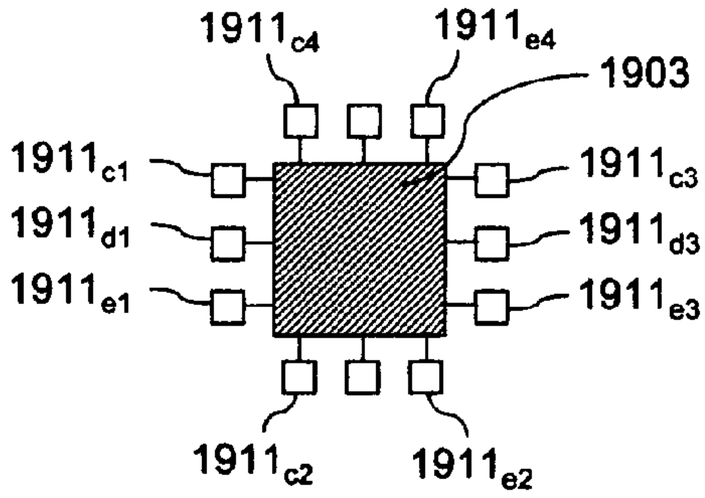


Fig-19D

MPGA PRODUCTS BASED ON A PROTOTYPE FPGA

Matter enclosed in heavy brackets [ ] appears in the original patent but forms no part of this reissue specification; matter printed in italics indicates the additions made by reissue.

This application is a continuation-in-part of application Ser. No. 11/645,313 filed Dec. 26, 2006, *now U.S. Pat. No. 7,627,848,* which is a continuation of application Ser. No. 11/384,116 filed on Mar. 20, 2006, now U.S. Pat. No. 7,356,799**[**, which is a continuation of application Ser. No. 10/825,194 (now U.S. Pat. No. 6,992,503) filed on Apr. 16, 2004, which is a continuation of application Ser. No. 10/267,511 (now U.S. Pat. No. 6,747,478) filed Oct. 8, 2002, which claims priority from Provisional App. Ser. No. 60/393,763 filed on Jul. 8, 2002 and App. Ser. No. 60/397,070 filed on Jul. 22, 2002, all of which have as inventor Mr. R. U. Madurawe and the contents of which are incorporated-by-reference**]**.

This application is related to application Ser. No. 10/267,483 and application Ser. No. 10/267,484 (now abandoned), all of which were filed on Oct. 8, 2002 and list as inventor Mr. R. U. Madurawe, the contents of which are incorporated-by-reference. This application is also related to application Ser. No. 10/691,013 (now U.S. Pat. No. 7,129,744) filed on Oct. 23, 2003, application Ser. No. 10/846,698 (now U.S. Pat. No. 7,064,018) filed on May 17, 2004, application Ser. No. 10/846,699 (now U.S. Pat. No. 7,112,994) filed on May 17, 2004, application Ser. No. 10/864,092 filed on Jun. 8, 2004, application Ser. No. 10/937,828 filed on Sep. 10, 2004, and application Ser. No. 11/102,855 filed on Apr. 11, 2005, all of which list as inventor Mr. R. U. Madurawe, and the contents of which are incorporated-by-reference.

BACKGROUND

The present invention relates to multi-dimensional integrated circuits. More specifically it relates to design conversion from a field programmable device (FPGA) to different density metal programmable application specific devices (MPGA) to reduce cost and improve performance, power and reliability.

Traditionally, integrated circuit (IC) devices such as custom, semi-custom, or application specific integrated circuit (ASIC) devices have been used in electronic products to reduce cost, enhance performance or meet space constraints. However, the design and fabrication of custom or semi-custom ICs can be time consuming and expensive. The customization involves a lengthy design cycle during the product definition phase and high Non Recurring Engineering (NRE) costs during manufacturing phase. When device geometries shrink, signal timing in an ASIC is wire-delay dominant. Wire delays are non-predictable during synthesis & placement phase, and comprise "RC" extractions of post-placement result of the best guess placement. Thus timing closure becomes a significant bottle neck in ASIC designs. Further, should errors exist in the custom or semi-custom ICs, the design/fabrication cycle has to be repeated, further aggravating the time to market and engineering cost. As a result, ASICs serve only specific applications and are custom built for high volume and low cost applications.

Another type of semi custom devices called Gate Array, Structured Array, Structured ASIC or Metal Programmable Gate Arrays (MPGA), henceforth all termed MPGAs, customizes modular blocks at a reduced NRE cost by synthesizing the design using a software model similar to the ASIC. The logic arrays are pre-fabricated; while only one or more metal layers are customize to fit the design with lower utilization over ASICs. The missing apriori wire-delays make the MPGA timing closure as difficult as in the ASIC. The missing silicon level design verification results in multiple spins and lengthy design iterations.

In recent years there has been a move away from custom or semi-custom ICs towards field programmable components whose function is determined not when the integrated circuit is fabricated, but by an end user "in the field" prior to use. Off the shelf, generic Programmable Logic Device (PLD) or Field Programmable Gate Array (FPGA) products, henceforth all termed FPGAs, greatly simplify the design cycle. These products offer user-friendly software to fit custom logic into the device through programmability, and the capability to tweak and optimize designs to optimize silicon performance. All wire delays are pre-characterized and FPGAs offer easy timing closure in a predictable manner thus solving ASIC and MPGA biggest problem. The flexibility of this programmability is expensive in terms of silicon real estate, but reduces design cycle and upfront NRE cost to the designer.

FPGAs offer the advantages of low non-recurring engineering costs, fast turnaround (designs can be placed and routed on an FPGA in typically a few minutes to a few hours), and low risk since designs can be easily amended late on in the product design cycle. It is only for high volume production runs that there is a cost benefit in using the more traditional approaches. However, the conversion from an FPGA implementation to an MPGA or ASIC implementation typically requires a complete redesign. Such redesign is undesirable in that the FPGA design effort is wasted.

Compared to an FPGA, an ASIC or MPGA has hard-wired logic connections, identified during the chip design phase, and need no configuration memory cells. They further require much less wires to connect logic. This is a large chip area and cost saving for the ASIC. Smaller ASIC die sizes lead to better performance. A full custom ASIC also has customized logic functions which take less gate counts compared to PLD and FPGA configurations of the same functions. Thus, an ASIC is significantly smaller, faster, cheaper and more reliable than an equivalent gate-count FPGA. The trade-off is between time-to-market (PLD and FPGA advantage) versus low cost and better reliability (ASIC pr MPGA advantage).

There is no convenient timing exact migration path from an FPGA used as a design verification and prototyping vehicle to the lower die size (hence lower cost) ASIC or MPGA. All of the SRAM or Anti-fuse configuration bits and programming circuitry that makes the FPGA more expensive has no value to the ASIC or MPGA. Programmable module removal from the FPGA and the ensuing layout and design customization is time consuming with severe timing variations from the original design. Input/output pad position changes also impact signal timing.

There is no convenient timing improvement or power reduction path from an FPGA used as a design verification and prototyping vehicle to the lower die size ASIC or MPGA. All of the SRAM or Anti-fuse configuration bits and programming circuitry that makes wire delays slow and power consumption high has no value to the ASIC or MPGA. Programmable module removal from the FPGA and the ensuing layout and design customization is time consuming with severe non-predictable timing variations from the original design.

There is no convenient single prototype FPGA that can be used as a design verification and prototyping vehicle, and provide an easy bit-stream compatible design conversion to a lower cost, or better performance, or lower power, or smaller density ASIC or MPGA. All of the configuration and programming overhead & pre-connected parasitic overhead within the FPGA has no value to the ASIC or MPGA. Parasitic overhead removal from the FPGA and the ensuing layout and design customization is time consuming with severe non-predictable timing variations from the original design.

What is therefore needed is a single prototype FPGA that can be used as design verification and prototyping vehicle, and further provide easy bit-stream compatible design conversion to more user desirable one or more MPGA products comprising varying densities for production.

SUMMARY

In one aspect, a three-dimensional semiconductor device with two selectable manufacturing configurations includes a first module layer having a plurality of circuit blocks; and a second module layer formed substantially above the first module layer, wherein in a first selectable configuration a plurality of memory circuits are formed to store instructions to control a portion of the circuit blocks, and wherein in a second selectable configuration a predetermined conductive pattern is formed in lieu of the memory circuit to control substantially the same portion of the circuit blocks.

Implementations of the above aspect may include one or more of the following. A third module layer can be formed substantially above the first module layer, wherein interconnect and routing signals are formed to connect the circuit modules within the first and second module layers. The second module layer in its first configuration can contain isolated through connections to connect the first module layer to the third module layer. A third module layer can be formed between the first and second module layers, wherein interconnect and routing signals are formed to connect the circuit modules within the first and second module layers. The first selectable configuration forms a programmable logic device (PLD) with one or more digital circuits formed on the first module layer; one or more programmable logic blocks formed on the first module layer and electrically coupled to the digital circuits; one or more memory blocks formed on the first module layer and electrically coupled to the digital circuits; one or more configurable memory elements formed on the second module layer and electrically coupled to the programmable logic blocks to customize the programmable content of the PLD; and one or more interconnect and routing signals formed in a third module layer, electrically coupled to first and second module layers to provide the functionality of the PLD. The second selectable configuration forms an Application Specific Integrated Circuit (ASIC) with one or more digital circuits formed on the first module layer; one or more programmable logic blocks formed on the first module layer and electrically coupled to the digital circuits; one or more memory blocks formed on the first module layer and electrically coupled to digital circuits; one or more predetermined connections formed on the second module layer and electrically coupled to the programmable logic blocks to customize the programmable content; and one or more interconnect and routing signals formed in a third module layer and electrically coupled to first and second module layers. The second module layer can be generic and user configurable to program and re-program to alter the functional response and performance of the PLD. The predetermined conductive pattern can be positioned substantially above the digital circuits. The predetermined conductive pattern can also be integrated in the first module layer or alternatively can be integrated in the third module layer. For every given memory pattern of the second module layer in the first configuration, a unique predetermined connection pattern exists in the second configuration to substantially match logic customization. One or more of the circuit blocks within the first module layer can maintain substantially identical timing characteristics under both configurations of second module layer logic control. The memory circuit can include one or more thin film devices such as thin film transistors (TFTs), resistors and capacitors. The replaceable memory can be selected from the group consisting of fuse links, antifuse capacitors, SRAM cells, DRAM cells, metal optional links, EPROM cells, EEPROM cells, Flash cells, and Ferro-electric elements. The digital circuit can include a third-party IP core. The digital circuit includes a processor capable of executing software logic instructions and other programmable logic blocks, wherein the programmable logic block is selected from one or more of a pass gate logic, multiplexer logic, truth table logic, or an AND/OR logic. The module layer one can include a substrate layer, n-well & p-well layers, field isolation regions, NMOS & PMOS gate, drain, source regions of transistors built on substrate, N+ & P+ diodes, resistors and capacitors built on substrate, gate oxide, gate poly, salicided regions, inter layer dielectric and contacts.

In another aspect, a programmable logic device includes one or more digital circuits formed on a substrate; and a non-planar circuit electrically coupled to the digital circuits, the non-planar circuit being either a memory constructed to store data to define the logic outputs of the digital circuits to fabricate a field programmable gate array (FPGA) or a conductive pattern constructed to define the logic outputs of the digital circuits to fabricate an application specific integrated circuit (ASIC), wherein the memory and the conductive pattern options have substantially matching functionality timing characteristics.

Implementations of the above aspects may include one or more of the following. The IC product is re-programmable in its initial stage with turnkey conversion to an ASIC. The IC has the end ASIC cost structure and FPGA re-programmability. The IC product offering occurs in two phases: the first stage is a generic FPGA that has re-programmability containing a programmable module, and the second stage is a timing-exact ASIC with the entire programmable module replaced by 1 to 2 customized hard-wire masks.

In another aspect, a programmable logic device includes a plurality of programmable logic cells built on a semiconductor substrate layer; and a plurality of interconnect wires constructed above said of programmable logic cells; and either configuration memory circuits or metal-wires constructed above said of programmable logic cells to program the logic functions and interconnect wire pattern; wherein a single larger FPGA can be used non-planar circuit being either a memory constructed to store data to define the logic outputs of the digital circuits to fabricate a field programmable gate array (FPGA) or a conductive pattern constructed to define the logic outputs of the digital circuits to fabricate an application specific integrated circuit (ASIC), wherein the memory and the conductive pattern options have substantially matching functionality and timing characteristics.

Implementations of the above aspects may include one or more of the following. The IC product is re-programmable in its initial stage with turnkey conversion to an ASIC. The IC has the end ASIC cost structure and FPGA re-programmability. The IC product offering occurs in two phases: the first stage is a generic FPGA that has re-programmability contain-

ing a programmable module, and the second stage is a timing-exact ASIC with the entire programmable module replaced by 1 to 2 customized hard-wire masks.

In another aspect, a smaller mask programmable gate array (MPGA) device derived from a larger field programmable gate array (FPGA), comprising: a layout of transistors and a plurality of interconnect layers identical to a region of the FPGA; and input/output pads matching a subset of the input/output pads of the FPGA; wherein, a design that is mapped to said region of the FPGA device using said subset of input/output pads by a user programmable means can be identically mapped to the MPGA by a hard-wire circuit. Such a gate array further comprises a mask programmable metal-circuit in lieu of a user programmable configuration circuit of the FPGA; and a logic output to input/output pad connection in lieu of a logic output to a register at the boundary of said region to an input/output pad connection of the FPGA.

Implementations of the above aspects may include one or more of the following. The prototype FPGA product is user re-programmable with easy bit-stream compatible turnkey design-conversion to one or more production MPGAs. The selected production MPGA has the low ASIC cost structure while the prototype FPGA has the re-programmability. The initial FPGA product is pre-designed with the capability to port designs to a varying number of smaller MPGAs. The FPGA has a core region comprising programmable circuits and an input/output (I/O) pad region. The I/O region may be in the perimeter of the die, surrounding the core. The I/O region may be on two sides of the core region. The FPGA core comprises a plurality of smaller regions: a first region, and a second region larger than the first region. A plurality of MPGAs is constructed; each MPGA core having an exact circuit layout of a said region within the FPGA core. Each MPGA core has a set of input/output pads surrounding the core, or on some sides of the core similar to the FPGA. The I/O pads in the MPGA are a subset of I/O pads of the FPGA. Logic blocks at core edges and I/O pad coupling are similarly constructed in the FPGA and MPGA. Logic blocks at the region boundary of the FPGA are not adjacent to the I/O pads, whereas in the MPGA the logic blocks are adjacent to I/O pads. A software tool and placement of registers at region boundaries is used to account for signal timing variation. For each MPGA, the subset of I/O pads is pre-assigned in the FPGA. Within the FPGA, the user has the option of selecting one of plurality of these regions available to place and route designs, including the entire FPGA. Each region offers a certain gate density of logic, user-memory and input/output pad connections to the user. Thus the user is able to place and route designs in significantly smaller portions of the FPGA, then map those designs into a matching MPGA of lower cost. As the elected region within the FPGA increases, so does the user memory, and the I/O pads assigned to that region. Each region within the FPGA may comprise one or more boundaries. Each boundary may comprise one or more registers. Some I/O pads may be directly coupled to logic blocks in the FPGA and in the MPGA. Within a region boundary of FPGA, logic outputs do not directly couple to I/O pads. Such logic outputs are made to couple to a said register at the boundary. These registers may be part of programmable logic blocks of the FPGA, or special registers placed to tie a logic output that has to couple to an I/O pad not adjacent to the region boundary. A software tool may identify a set of registers at the boundary account for I/O connections. Thus a logic output within the FPGA at the region boundary is first routed to a register at the boundary by the software tool and then to an I/O structure. The timing is pre-characterized for such a connection in the FPGA. In the corresponding MPGA no such register is provided, and the logic output is directly coupled to the adjacent I/O. For the MPGA, the timing without a register is also pre-characterized. Thus when a FPGA prototype design is converted to the MPGA, the I/O timing is adjusted by the software tool.

Advantages of the IC may include one or more of the following. A series product families can be provided with a modularized programmable element in an FPGA version followed by a turnkey custom ASIC with the same base die with 1-2 custom masks. The vertically integrated programmable module does not consume valuable silicon real estate of a base die. Furthermore, the design and layout of these product families adhere to removable module concept: ensuring the functionality and timing of the product in its FPGA and ASIC canonicals. These IC products can replace existing PLD and FPGA products and compete with existing Gate Arrays and ASIC's in cost and performance. Such products offer a more reliable and lower cost ASIC design conversion from the initial PLD and FPGA.

An easy turnkey customization of an ASIC from an original smaller PLD or FPGA would greatly enhance time to market, performance, low cost and better reliability.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** shows a cross sectional view of a first embodiment of an integrated circuit.

FIG. **2** shows a cross sectional view of a second embodiment of an integrated circuit.

FIG. **3** shows a cross sectional view of a third embodiment of an integrated circuit.

FIG. **4** shows a cross sectional view of a fourth embodiment of an integrated circuit.

FIG. **5** shows an exemplary AND-OR PLD Architecture.

FIG. **6** shows an exemplary AND-OR array gate realization of PLD.

FIG. **7** shows one EEPROM implementation of a P-Term logic array.

FIG. **8** shows P-term configuration for SRAM/hard-wired PLD architecture.

FIG. **9** shows an exemplary pass-gate logic.

FIG. **10** shows an exemplary 4-Input logic MUX.

FIG. **11** shows an exemplary 2-Input Truth Table.

FIG. **12** shows a logic tree implementation of a 4-Input Truth Table.

FIG. **13** shows an exemplary 6T SRAM.

FIG. **14** shows pass gate transistor logic controlled by SRAM.

FIG. **15** shows one embodiment of a 5×6 switch matrix.

FIG. **16** shows pass gate controlled by Vcc (power) or Vss (ground)

FIG. **17** shows the 5×6 switch matrix

FIGS. **18**A, **18**B, **18**C, **18**D show a first embodiment of a prototype FPGA that comprises three regions, each region facilitating a different MPGA conversion for production.

FIGS. **19**A, **19**B, **19**C, **19**D show a second embodiment of a prototype FPGA that comprises three regions, each region facilitating a different MPGA conversion for production.

DESCRIPTION

In the following detailed description of the invention, reference is made to the accompanying drawings which form a part hereof, and in which is shown, by way of illustration, specific embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention. Other

embodiments may be utilized and structural, logical, and electrical changes may be made without departing from the scope of the present invention.

The terms wafer and substrate used in the following description include any structure having an exposed surface with which to form the integrated circuit (IC) structure of the invention. The term substrate is understood to include semiconductor wafers. The term substrate is also used to refer to semiconductor structures during processing, and may include other layers that have been fabricated thereupon. Both wafer and substrate include doped and undoped semiconductors, epitaxial semiconductor layers supported by a base semiconductor or insulator, SOI material as well as other semiconductor structures well known to one skilled in the art. The term conductor is understood to include semiconductors, and the term insulator is defined to include any material that is less electrically conductive than the materials referred to as conductors. The following detailed description is, therefore, not to be taken in a limiting sense.

The term module layer includes a structure that is fabricated using a series of predetermined process steps. The boundary of the structure is defined by a first step, one or more intermediate steps, and a final step. The resulting structure is formed on a substrate. The term layout refers to a set of geometries arranged to define a masking layer.

The term configuration circuit includes one or more configurable elements and connections that can be programmed for controlling one or more circuit blocks in accordance with a predetermined user-desired functionality. In one embodiment, the configuration circuits include a plurality of memory circuits to store instructions to configure an FPGA. In another embodiment, the configuration circuits include a first selectable configuration where a plurality of memory circuits is formed to store instructions to control one or more circuit blocks. The configuration circuits include a second selectable configuration with a predetermined conductive pattern formed in lieu of the memory circuit to control substantially the same circuit blocks. The memory circuit includes elements such as diode, transistor, resistor, capacitor, metal link, among others. The memory circuit also includes thin film elements. In yet another embodiment, the configuration circuits include a predetermined conductive pattern, via, resistor, capacitor or other suitable circuits formed in lieu of the memory circuit to control substantially the same circuit blocks. The term "horizontal" as used in this application is defined as a plane parallel to the conventional plane or surface of a wafer or substrate, regardless of the orientation of the wafer or substrate. The term "vertical" refers to a direction perpendicular to the horizontal direction as defined above. Prepositions, such as "on", "side", "higher", "lower", "over" and "under" are defined with respect to the conventional plane or surface being on the top surface of the wafer or substrate, regardless of the orientation of the wafer or substrate.

FIG. **1** shows a cross sectional view of a first embodiment of an integrated circuit that can be selectably fabricated as either an FPGA or an ASIC. In this embodiment, a three-dimensional semiconductor device **100** is shown. The device **100** includes a first module layer **102** having a plurality of circuit blocks **104** embedded therein. The device **100** also includes a second module layer **106** formed substantially above the first module layer **102**. One or more configuration circuits **108** are formed to store instructions to control a portion of the circuit blocks **104**. In the first selectable option, circuits **108** are programmable to build FPGA products. In the second selectable option, circuits **108** are wire connections to build ASIC products. In the embodiment of FIG. **1**, wiring/routing circuits **112** are formed on a third layer **110** above the second layer **106**. Circuits **112** connect to both circuits **104** and **108** to complete the functionality of the PLD.

FIG. **2** shows a cross sectional view of a second embodiment of an integrated circuit that can be selectably fabricated as either an FPGA or an ASIC. In this embodiment, a three-dimensional semiconductor device **120** is shown. The device **120** includes a first module layer **122** having a plurality of circuit blocks **124** embedded therein. The device **120** also includes a second module layer **126** formed substantially above the first module layer **122** that includes wiring and/or routing circuitry **128**, and a third module layer **130** formed substantially above the second module layer **126** that includes configuration circuits **132**. In the first selectable option, circuits **132** are programmable to build FPGA products. In the second selectable option, circuits **132** are wire connections to build ASIC products. The wiring/routing circuitry **128** is electrically connected to the circuit blocks **124** and to configuration circuits **132** in a third module layer **130**. The configuration circuits **132** store instructions to control a portion of the circuit blocks **124**.

FIG. **3** shows a third embodiment which is substantially similar to the embodiment of FIG. **2**. In the embodiment of FIG. **3**, a fourth layer **140** having wiring/routing circuitry **142** is position above the third layer **130**. The wiring/routing circuitry **142** is electrically connected to one of the following: one or more circuit blocks **124**, one or more wiring/routing circuitry **128**, and one or more configuration circuits **132**.

FIG. **4** shows one implementation where the configuration memory element is SRAM. First, silicon transistors **150** are deposited on a substrate. A module layer of removable SRAM memory cells **152** are positioned above the silicon transistors **150**, and a module layer of interconnect wiring or routing circuit **154** is formed above the removable memory cells **152**. In the first selectable option, SRAM cells **152** are programmable to build FPGA products. In the second selectable option, cells **152** are replaced with wire connections to build ASIC products. To allow this replacement, the design adheres to a hierarchical layout structure. As shown in FIG. **4**, the SRAM cell module is sandwiched between the single crystal device layers below and the metal layers above electrically connecting to both. It also provides through connections "A" for the lower device layers to upper metal layers. The SRAM module contains no switching electrical signal routing inside the module. All such routing is in the layers above and below. Most of the programmable element configuration signals run inside the module. Upper layer connections to SRAM module "C" are minimized to Power, Ground and high drive data wires. Connections "B" between SRAM module and single crystal module only contain logic level signals and replaced later by Vcc and Vss wires to build the ASIC. Most of the replaceable programmable elements and its configuration wiring is in the "replaceable module" while all the devices and end ASIC wiring is outside the "replaceable module". In other embodiments, the replaceable module could exist between two metal layers or as the top most layer satisfying the same device and routing constraints.

Fabrication of the IC also follows a modularized device formation. Formation of transistors **150** and routing **154** is by utilizing a standard logic process flow used in the ASIC fabrication. Extra processing steps used for memory element **152** formation are inserted into the logic flow after circuit layer **150** is constructed. A full disclosure of the vertical integration of the TFT module using extra masks and extra processing is in the co-pending incorporated by reference applications discussed above.

During the customization, the base die and the data in those remaining mask layers do not change making the logistics associated with chip manufacture simple. In one embodiment, the custom wire connections can be combined with the contact in module-**1** and metal-**1** in module-**2** processing. In another embodiment, the custom wire connections can be an extra metal-**1**, via-**1** insertion compatible with logic processing. Removal of the SRAM module provides a low cost standard logic process for the final ASIC construction with the added benefit of a smaller die size. The design timing is unaffected by this migration as lateral metal routing and silicon transistors are untouched. Software verification and the original FPGA design methodology provide a guaranteed final ASIC solution to the user. A full disclosure of the ASIC migration from the original FPGA is provided in the body of this discussion.

In FIG. **4**, the third module layer is formed substantially above the first and second module layers, wherein interconnect and routing signals are formed to connect the circuit modules within the first and second module layers. Alternatively, the third module layer can be formed substantially below the first and second module layer with the interconnect and routing signals formed to connect the circuit modules within the first and second module layers. Alternatively, the third and fourth module layers positioned above and below the second module layer respectively, wherein the third and fourth module layers provide interconnect and routing signals to connect the circuit modules within the first and second module layers.

In yet another embodiment of a programmable multidimensional semiconductor device, a first module layer is fabricated having a plurality of circuit blocks formed on a first plane. The programmable multi-dimensional semiconductor device also includes a second module layer formed on a second plane. A plurality of configuration circuits is then formed to store instructions to control a portion of the circuit modules.

In another embodiment, a programmable logic device includes one or more digital circuits formed on a substrate; and a non-planar circuit electrically coupled to the digital circuits, the non-planar circuit being either a memory constructed to store data to define the logic outputs of the digital circuits to fabricate a field programmable gate array (FPGA) or a conductive pattern constructed to define the logic outputs of the digital circuits to fabricate an application specific integrated circuit (ASIC), wherein the memory and the conductive pattern options have substantially matching functionality timing characteristics.

The design/conversion of the FPGA to the ASIC is explained next. The larger and very complex FPGA designs are done with computer-aided design (CAD) tools. A design specification is converted to a logical entry format for a Design Entry CAD tool. The abstract logic functions are described using Hardware Description Language (HDL, VHDL) or Schematic Diagrams. The design entry is compiled to extract the netlist. This netlist is used to synthesize the logic to be placed in the FPGA. Design capture so far is independent of the FPGA platform. A customized Place and Route (fitter) software tool is used to select the logic gates and to make the required connections in a chosen FPGA. The design placed and routed inside the FPGA is simulated using test vectors to verify the performance and functionality. The optimized design database specifies how the FPGA programmable resources are utilized to achieve the original design objectives.

From the information contained in the design database, a configuration bitstream is generated by a tool commonly referred to as a bitstream compiler. All the logic and routing customization specific to the design is contained in this bitstream, which is a binary representation of every single configuration device in the FPGA. This is also referred to as a bitmap when the bitstream is mapped to the image of configuration elements. At the physical level, the defining binary data in the bitstream represent the ON/OFF states of the configurable switches that control logic blocks, IO blocks and interconnection in the FPGA.

At this point, the configuration bitstream either may be downloaded to the logic array thereby configuring the device or the bitstream may be saved onto disk. If the FPGA contains non volatile memory elements, a programmer is used to program the bitmap into the device. Some non volatile memory (NVM) elements such as EEPROM and Flash lend to in system programmability (ISP), allowing programming inside the design board via JTAG instructions. SRAM based FPGA allow ISP, but need a NVM content outside the device to hold the bitstream.

Even though the design has been fine tuned in software for timing and functionality, it still needs to be verified on Silicon. This is due to inaccuracies between the timing model and silicon performance. Having a pre-fabricated generic FPGA makes this verification simple and quick. The FPGA device is then programmed and tested in a system board to verify operational correctness. If the design does not work it is re-optimized to work on silicon. When the design works, it is initially fielded. Should the device prove popular, the FPGA can be converted into an ASIC by hard-coding the bitstream.

First an image file is generated for all the B contacts that exist between Module layer-**2** and Module layer-**1** in FIG. **4**. These B contacts represent configuration element control of the logic blocks. There is a one to one matching between these B contacts and the bitmap generated for the design, as every configuration element is represented in both. We can define (1,0) in the bitstream to represent SRAM output at logic (1,0) respectively. Bitstream ones represent B contacts at Vcc, while bitstream zeros represent B contacts at Vss. The bitstream can be automatically mapped to contact B file to convert those to Vcc and Vss hard connections. The contacts B are in the CAD database that generates the physical mask for wafer processing. This technique provides an error free software conversion of the bitstream to a had-wire mask. By appropriate pre-allocation of Vcc and Vss resources above the B contacts, one could conceivably generate the ASIC with only one custom mask, a considerable savings in expensive mask costs. All the C contacts in the hard mask are simply omitted as no configuration elements exist, while all the A contacts are retained.

The conversion does not result in a new placement and routing configuration that is different from the previous FPGA design. The conversion does not result in a change to the logic gates in module layer-**1** or the lateral wire routing in module layer-**3**. The vertical contact height change is negligible in the gate and wire delay components of logic propagation. Logic gate timing is not affected by control options between SRAM output or Vcc/Vss. The timing is maintained identical in this FPGA to ASIC conversion. Furthermore, this conversion can be made by the FPGA supplier, with no engineering overhead, saving valuable design resources at both end user and manufacturing sites. The final hard mask ASIC has no soft errors (no SRAM bits to flip), better reliability as fewer processing steps and fewer hard wires (one connection to replace 6-transistors) are used, and provide a secure environment against "bitstream piracy"—a technique of stealing designs by extracting the bitstream from FPGAs.

Next, details of the circuit blocks **104**, the configuration circuit **108**, and the wiring and/or routing circuit **112** in FIG. **1** are detailed.

A variety of digital or analog circuits can be used in circuit blocks **104**. These circuit blocks include programmable logic blocks to allow user customization of logic. In one embodiment, programmable logic blocks are provided to respond to input data signals. The programmable logic blocks develop corresponding complete or partial output logic signals. Registers are used to store the output logic signals and either outputting them to output terminals or returning them as inputs to additional programmable logic blocks. The registers themselves can be programmable, allowing those to be configured such as T flip-flops, JK flip-flops, or any other register. The logic blocks may contain no registers, or the registers may be programmed to be by-passed to facilitate combinational logic implementation. The programmable logic block can be selected from one of a pass gate logic, a multiplexer logic, a truth table logic, or an AND/OR logic. FIG. **5** shows an exemplary AND-OR PLD Architecture. AND and OR arrays **202** and **204** contain user configurable programmable elements. FIG. **6** shows an exemplary AND-OR array gate realization of a three input, four P-term, four output PLD. The AND and the OR array **210-212** are shown programmed to a specific pattern.

In yet other embodiments, the circuit block **104** contains a RAM/ROM logic block consisting of "logic element tree" or "P-Term logic array" blocks that perform logic functions.

FIG. **7** shows one such NAND EEPROM implementation of a P-Term in NAND-NOR logic array, while FIG. **8** shows the same P-term configuration for either SRAM, or hardwired PLD architectures. FIG. **7** shows two mirrored outputs P**1** and P**2**. For output P**1**, an AND gate **232** receives signals from pass transistors **222**, **224**, **228** and **230**. The pass transistor **222** is controlled by block **220** shown in the dashed circle, while the pass transistor **228** is controlled by block **226** shown inside the dashed circle. Similarly, the upper half of FIG. **8** includes an AND gate **252** that receives inputs from pass transistors **242**, **244**, **248** and **250**, respectively.

FIG. **9** shows an exemplary pass-gate logic **260** connecting one input to one output. The NMOS pass gate voltage level S**0** determines an ON and OFF connection. FIG. **10** shows an exemplary 4-Input logic MUX implementing an output function O where O=I**0***S**0**+I**1***S**1**+I**2***S**2**+I**3***S**3**. In the MUX, only one of S**0 270**, S**1 272**, S**2 274**, S**3 276** has a logic one. The MUX is constructed by combining four NMOS pass gate logic elements **280-286** shown in FIG. **9**.

FIG. **11** shows an exemplary 2-input truth table logic realization of an output function F where,

F=/A*/B*S**0**+/A*B*S**1**+A*/B*S**2**+A*B*S**3**(/A means not A).

The truth table logic states are represented by S**0**, S**1**, S**2** and S**3**. The realization is done through six inverters collectively designated **250** and eight pass transistors collectively designated **260**. Logic states are stored in 4 programmable registers.

FIG. **12** shows a logic tree constructed with five 2-input truth table logic blocks **320-328** to perform a full four input truth table. A four input truth table has 16 possible logic states S**0**, S**1**, . . . , S**15**. As the number of inputs grows to N, this logic tree construction requires 2^N logic states, and $2^{(N-1)}$ branches in the logic tree. For large N values, a full truth table realization is less efficient compared to a partial product term AND-OR array realization.

In another embodiment, the programmable logic block can be a programmable microprocessor block. The microprocessor can be selected from third party IP cores such as: 8051, Z80, 68000, MIPS, ARM, and PowerPC. These microprocessor architectures include superscalar, Fine Grain Multi-Threading (FGMT) and Simultaneous Multi-Threading (SMT) that support Application Specific Packet Processing (ASPP) routines. To handle Programmable Network Interface (PNI) the processor can contain hardware and software configurability. Hardware upgradeability can be greatly enhanced in microprocessors embedded in PLD's by making use of the available logic content of the PLD device. Programmable features can include varying processor speed, cache memory system and processor configuration, enhancing the degree of Instruction Level Parallelism (ILP), enhancing Thread level parallelism (TLP). Such enhancements allow the user to optimize the core processor to their specific application. Cache parameters such as access latency, memory bandwidth, interleaving and partitioning are also programmable to further optimize processor performance and minimize cache hit miss rates. Additionally, the processor block can be a Very Long Instruction Word (VLIW) processor to handle multimedia applications. The processor block can include a cache controller to implement a large capacity cache as compared with an internal cache.

While a PLD can be configured to do DSP functions, the programmable logic block can also contain a digital signal processor (DSP), which is a special purpose processor designed to optimize performance for very high speed digital signal processing encountered in wireless and fiber-optic networks. The DSP applications can include programmable content for cache partitioning, digital filters, image processing and speech recognition blocks. These real-time DSP applications contain high interrupt rates and intensive numeric computations best handled by hardware blocks. In addition, the applications tend to be intensive in memory access operations, which may require the input and output of large quantities of data. The DSP cache memory may be configured to have a "Harvard" architecture with separate, independent program and data memories so that the two memories may be accessed simultaneously. This architecture permits an instruction and an operand to be fetched from memory in a single clock cycle. A modified Harvard architecture utilizes the program memory for storing both instructions and operands to achieve full memory utilization. The program and data memories are often interconnected with the core processor by separate program and data buses. When both instructions and operands (data) are stored in a single program memory, conflicts may arise in fetching data with the next instruction. Such conflicts have been resolved in prior art for DSPs by providing an instruction cache to store conflicting instructions for subsequent program execution.

In yet another embodiment, programmable logic block can contain software programmability. These software functions are executed in DSP, ARM, or MIPS type inserted IP cores, or an external host CPU. Accelerators connected by a configurable SRAM switching matrix enhance the computation power of the processors. The microprocessor has local permanent SRAM memory to swap, read, and write data. The switch matrix is pre-designed to offer both hard-wire and programmable options in the final ASIC. In this situation, the circuit block **104** can be a functional block that performs well-defined, commonly-needed function, such as special D/A or A/D converter, standard bus interface, or such block that implements special algorithms such as MPEG decode. The special algorithms implemented can be hardware versions of software. For example, algorithms relating to digital radio or cellular telephone such as WCDMA signal processing can be implemented by the functional block. Other func-

tional blocks include PCI, mini-PCI, USB, UART blocks that can be configured by specifying the SRAM logic blocks.

In yet another embodiment, the circuit block **104** can be memory such as a register file, cache memory, static memory, or dynamic memory. A register file is an array of latches that operate at high speed. This register length counter may be programmable by the user. A cache memory has a high access throughput, short access latency and a smaller capacity as compared with main memory. The cache memory may be programmable to partition between the different requirements of the system design. One such need is the division between L**1** and L**2** cache requirements for networking applications. The memory can also be static random access memory or (SRAM) device with an array of single port, or multi-port addressable memory cells. Each cell includes a four transistor flip-flop and access transistors that are coupled to input/output nodes of the flip-flop. Data is written to the memory cell by applying a high or low logic level to one of the input/output nodes of the flip-flop through one of the access transistors. When the logic level is removed from the access transistor, the flip-flop retains this logic level at the input/output node. Data is read out from the flip-flop by turning on the access transistor. The memory can also be dynamic random access memory (DRAM). Generally, a DRAM cell consists of one transistor and a capacitor. A word line turns on/off the transistor at the time of reading/writing data stored in the capacitor, and the bit line is a data input/output path. DRAM data is destroyed during read, and refresh circuitry is used to continually refresh the data. Due to the low component count per bit, a high density memory device is achieved.

In another embodiment, the circuit block **104** can be an intellectual property ("IP") core which is reusable for licensing from other companies or which is taken from the same/previous design. In core-based design, individual cores may be developed and verified independently as stand-alone modules, particularly when IP core is licensed from external design source. These functions are provided to the user as IP blocks as special hardware blocks or pre-configured programmable logic blocks. The IP blocks connect via a programmable switching matrix to each other and other programmable logic. The hardware logic block insertion to any position in a logic sequence is done through the configurable logic matrix. These hardware logic blocks offer a significant gate count reduction on high gate count frequently used logic functions, and the user does not require generic "logic element" customization. In both cases, the user saves simulation time, minimize logic gate count, improve performance, reduce power consumption and reduce product cost with pre-defined IP blocks. The switch matrix is replaced by hard-wires in the final ASIC.

The circuit blocks **104** can also be an array of programmable analog blocks. In one embodiment, the analog blocks include programmable PLL, DLL, ADC and DAC. In another embodiment, each block contains an operational amplifier, multiple programmable capacitors, and switching arrangements for connecting the capacitors in such as a way as to perform the desired function. Switched capacitor filters can also be used to achieve an accurate filter specification through a ratio of capacitors and an accurate control of the frequency of a sampling clock. Multiple PLL's can be programmed to run at different frequencies on the same chip to facilitate SoC applications requiring more than one clock frequency.

The circuit blocks **104** also contain data fetch and data write circuitry required to configure the configuration circuits **108**. This operation may be executed by a host CPU residing in the system, or the PLD device itself. During power up, these circuits initialize and read the configuration data from an outside source, either in serial mode or in parallel mode. The data is stored in a predefined word length locally and written to the configurability allocation. The programmed configuration data is verified against the locally stored data and a programming error flag is generated if there is a mismatch. These circuits are redundant in the conversion of the PLD to an ASIC. However, these circuits are used in both FPGA and ASIC for test purposes, and has no cost penalty. A pin-out option has a "disable" feature to disconnect them for the customer use in the FPGA and ASIC.

Configuration circuits **108** provide active circuit control over digital circuits **104**. One embodiment of the configuration circuit includes an array of memory elements. The user configuration of this memory amounts to a specific bitmap of the programmable memory in a software representation.

Suitable memory elements include volatile or non volatile memory elements. In non-volatile memory (NVM) based products, configurable data is held in one of metal link fuse, anti-fuse, EPROM, Flash, EEPROM memory element, or ferro-electric elements. The first two are one time programmable (OTP), while the last four can be programmed multiple times. As EPROM's require UV light to erase data, only Flash & EEPROM's lend to in-system programmability (ISP). In volatile products, the configurable data storage can be SRAM cells or DRAM cells. With DRAM cells, the data requires constant refresh to prevent losses from leakages. Additionally, one or more redundant memory cells controlling the same circuit block can be used to enhance device yield.

The components of the memory element array can be a resistor, capacitor, transistor or a diode. In another embodiment of the configuration circuit, a memory element can be formed using thin film deposition. The memory element can be a thin film resistor, thin film capacitor, thin film transistor (TFT) or a thin film diode or a group of thin film devices connected to form an SRAM cell.

This discussion is mostly on SRAM elements and can easily extend to include all other programmable elements. In all cases, the design needs to adhere to rules that allow programmable module elimination, with no changes to the base die, a concept not used in PLD, FPGA, Gate Array and ASIC products today.

An exemplary 6T SRAM cell, shown in FIG. **13**, needs no high voltage capability, nor added process complexity. The cell of FIG. **13** has two back-to-back inverters **350-352** whose access is controlled by pass transistors **354-356**. In addition, R-load & Thin Film Transistor (TFT) load PMOS based SRAM cells can be used for PLDs and FPGAs. To achieve zero stand-by power by eliminating sensing circuitry, and reduce memory element count for low input functions, these SRAM cells are embedded in truth table logic (also called Look-Up-Table) based architectures.

Pass gate transistor **360** logic controlled by SRAM is shown in FIG. **14**. In this embodiment, the memory cell (such as the cell of FIG. **13**) drives the pass transistor **360** to e affect an outcome. A 5×6-switch point matrix 370 controlled by 30-SRAM cells coupled to 30-NMOS pass gates is shown in FIG. **15**. FIG. **16** shows the NMOS pass gate **360** logic controlled by the SRAM in FIG. **14** converted to hard-wire logic. A contact **362**, connected to Vcc (logic 1) or Vss (logic 0) depending on the SRAM logic content, replace the SRAM cell. The SRAM logic mapping to hard wire connections are automatic and done by a software program that is verifiable against the bit-map.

Similarly, FIG. **17** shows the 5×6-switch point matrix **370** hard-wired by replacing the SRAM bits that control NMOS gates with hard-wires to Vcc or Vss. In FIG. **17**, the bubble may represent either SRAM or hard-wire Vcc or Vss control

on NMOS pass gates. In the case of Fuse or Antifuse arrays, contact or no contact between the two metal lines in FIG. **15** directly replaces the programmable element and there is no NMOS pass-gate needed.

The P-Term logic builds the core of PLD's and complex PLD's (CPLD's) that use AND-OR blocks **202-204** (or equivalent NAND-NOR type logic functions) as shown in the block diagram of FIG. **5** and one expansion is shown in FIG. **6** with and gates **210** and or gates **212**. Gate implementation of two inputs (I**1**, I**2**) and two P-terms (P**1**, P**2**) NAND function can be single poly EEPROM bits as shown in FIG. **10**. The dotted circle contains the charge trapping floating gate, the programming select transistor, tunneling diode, a control gate capacitor and programming access nodes. The SRAM cell replaces that entire circle in this invention as detailed next. The SRAM NAND-NOR array (also AND-OR array) replacement has not been realized in prior art as SRAM cells require Nwell & Pwell regions that consume large silicon area to prevent latch-up. The SRAM in TFT do not have well related constraints as NMOS and PMOS bodies are isolated from each other. Keeping the two pass gates in silicon layers and moving SRAM to TFT layers allow P-Term logic implementation with SRAM cells and subsequent replacement with hard-wires. In TFT SRAM conversion to final ASIC, the bubble on NMOS gate becomes a hard-wire connection to Vcc or Vss.

The length of input and output wires, and the drive on NMOS pass gates and logic gate delays determine the overall PLD delay timing, independent of the SRAM cell parameters. By moving SRAM cell to TFT upper layers, the chip X, Y dimensions are reduced over 20% to 50% compared to traditional SRAM FPGA's, providing a faster logic evaluation time. In addition, removal of SRAM cell later does not alter lateral wire length, wire loading and NMOS pass gate characteristic. The vertical dimension change in eliminating the memory module is negligible compared to the lateral dimension of the ASIC, and has no impact on timing. This allows maintaining identical timing between the FPGA and ASIC implementations with and without the SRAM cells. The final ASIC with smaller die size and no SRAM elements have superior reliability, similar to an ASIC, leading to lower board level burn-in and field failures compared to PLD's and FPGA's in use today.

Next, the wiring and/or routing circuit **112** is discussed. The wiring and/or routing circuit connects each logic block to each other logic block. The wiring/routing circuit allows a high degree of routing flexibility per silicon area consumed and uniformly fast propagation of signals, including high-fanout signals, throughout the device. The wiring module may contain one or many levels of metal interconnects.

One embodiment of a switch matrix is a 6×5 programmable switch-matrix with 30 SRAM bits (or 30 Anti-fuses, or 30 fuses), shown in FIG. **15**. The box in FIG. **14** contains the SRAM cell shown inside dotted box of FIG. **14**, where the pass gate makes the connection between the two wires, and the SRAM bit holds the configuration data. In this configuration, the wire connection in circuit **112** occurs via a pass transistor located in circuit **104** controlled by an SRAM cell in circuit **108**. During power-up, a permanent non-volatile memory block located in the system, loads the correct configuration data into SRAM cells. In Fuse or Anti-fuse applications, the box simply represents the programmable element in circuit **108** between the two wires in circuit **112**. During the ASIC conversion this link is replaced with an open or short between the wires.

Another embodiment provides short interconnect segments that could be joined to each other and to input and output terminals of the logic blocks at programmable interconnection points. In another embodiment, direct connections to adjacent logic blocks can be used to increase speed. For global signals that traverse long distances, longer lines are used. Segmented interconnect structures with routing lines of varied lengths can be used. In yet other embodiments, a hierarchical interconnect structure provides lines of short lengths connectable at boundaries to lines of longer lengths extending between the boundaries, and larger boundaries with lines of even longer length extending between those boundaries. The routing circuit can connect adjacent logic blocks in two different hierarchical blocks differently than adjacent logic blocks in the same hierarchical block. Alternatively, a tile-based interconnect structure can be used where lines of varying lengths in which each tile in a rectangular array may be identical to each other tile. In yet another implementation, the interconnect lines can be separated from the logic block inputs by way of a routing matrix, which gives each interconnect line more flexible access to the logic block inputs. In another embodiment, the interconnect routing is driven by programmable buffers. Long wire lengths can be sub-divided into smaller length segments with smaller buffers to achieve a net reduction in the overall wire delay, and to obtain predictable timing in the logic routing of the PLD.

FIG. **18**A shows a first embodiment of an FPGA constructed as a regular 2D FPGA or a modular 3D FPGA. It may have configuration circuits in the same module layers as the transistors as in conventional FPGAs. It may have configuration circuits in a second module layer positioned above a first module layer that comprises logic circuits as presented in FIG. **1** thru FIG. **4**. The FPGA has a core region **1801**. The core region comprises programmable logic blocks and programmable interconnects. It further comprises user memory such as single-port or dual-port memory. It further comprises IP blocks such as microprocessor cores, DSP cores, analog cores and other circuits typically found in ICs. It further comprises registers, storage devices, clocks, PLLs, DLLs and control circuits. The core region **1801** interfaces with input/output pad regions **1811**. These pad structures may be arranged around the perimeter as shown in FIG. **18**A. In that, I/O structures $\mathbf{1811}_{A1}$-$\mathbf{1811}_{G1}$ are left perimeter structures, $\mathbf{1811}_{A2}$-$\mathbf{1811}_{G2}$ are bottom perimeter structures, $\mathbf{1811}_{A3}$-$\mathbf{1811}_{G3}$ are right perimeter structures, and $\mathbf{1811}_{A4}$-$\mathbf{1811}_{G4}$ are top perimeter structures. These pad structures may be arranged in any other method that is found in ICs, such as two sides of the core or distributed in an array through the core. Configuration circuits provide access for a user to program the functionality and routing of the FPGA to achieve a desirable functionality and performance. Configuration circuits further provide programmable interface of I/O pads to inputs of logic blocks and outputs of logic blocks in the FPGA core.

The FPGA in FIG. **18**A further comprises smaller regions within the core such as **1802** and **1803**. Region **1802** comprises the entire area inside the shaded region including region **1803**. Therefore the resource content of region **1803** is the least, **1802** is greater than in **1803**, and **1801** has the most. Each region comprises a subset of resources found in the core region **1801**. Thus a smaller design that requires fewer resources may be placed inside region **1802**, or even **1803**. A software tool is able to determine the resource content, and choose an appropriate region to place and route the design. The region **1802**, which is smaller than region **1801**, interfaces with a subset of I/O structures **1811**; the subset I/O structures also labeled **1811** in FIG. **18**A. For region **1802**, I/O structures $\mathbf{1811}_{B1}$-$\mathbf{1811}_{F1}$ are left perimeter structures, $\mathbf{1811}_{B2}$-$\mathbf{1811}_{F2}$ are bottom perimeter structures, $\mathbf{1811}_{B3}$-$\mathbf{1811}_{F3}$ are right perimeter structures, and $\mathbf{1811}_{B4}$-$\mathbf{1811}_{F4}$ are

top perimeter structures. Power and ground pad structures are arranged in such a way, they are common to all regions. A pad structure is assumed to include buffer circuits, bond pad structures, ESD structures, registers, control circuits and other circuits found in ICs. In FIG. **18**A, unlike for region **1801** wherein the I/O structures were adjacent to the perimeter of the region, for region **1802**, the IO structures are distanced by circuits contained in region **1801** from the perimeter of **1802**. Registers **1821** are provided in the FPGA hardware such that a software tool is able to automatically recognize the region boundaries not adjacent to I/O structures; insert a register and couple either logic input or logic output to the register output or input respectively first; then couple the register input or output to corresponding I/O structure as shown. Register $\mathbf{1821}_{B2}$ couples to I/O structure $\mathbf{1811}_{B2}$. Similarly, the region **1803**, which is smaller than region **1802**, interfaces with a subset of I/O structures **1811**; the subset I/O structures also labeled **1811** in FIG. **18**A. For region **1803**, I/O structures $\mathbf{1811}_{C1}$-$\mathbf{1811}_{E1}$ are left perimeter structures, $\mathbf{1811}_{C2}$-$\mathbf{1811}_{E2}$ are bottom perimeter structures, $\mathbf{1811}_{C3}$-$\mathbf{1811}_{E3}$ are right perimeter structures, and $\mathbf{1811}_{C4}$-$\mathbf{1811}_{E4}$ are top perimeter structures. Power and ground pad structures are arranged in such a way, they are common to all regions. For region **1803**, the 10 structures are distanced by circuits contained in region **1801** and in **1802** from the perimeter of **1803**. Registers **1831** are provided in the FPGA hardware such that a software tool is able to automatically recognize the region boundaries not adjacent to I/O structures; insert a register and couple either logic input or logic output to the register output or input respectively first; then couple the register input or output to corresponding I/O structure as shown. Register $\mathbf{1831}_{C2}$ couples to I/O structure $\mathbf{1811}_{C2}$. It is understood that the FPGA in FIG. **18**A comprises more than the few pads shown for illustrative purposes, and region **1803** may comprise hundreds of I/O structures.

FIGS. **18**B, **18**C and **18**D show three MPGA devices constructed with the identical resources found in regions **1801**, **1802** and **1803** respectively. A first module layer comprising transistors of region **1802** is substantially identically duplicated in the core region of FIG. **18**C. A second module layer comprising a plurality of interconnects, positioned above the first module layer is also substantially identically duplicated in FIG. **18**C. The subset of I/O regions common to region **1802** in FIG. **18**A is also duplicated in FIG. **18**C. In one embodiment, registers **1821** are specially placed in FIG. **18**A, and those are not duplicated in FIG. **18**C; instead logic input or output is directly coupled to the corresponding I/O structure. The software tool is able to identify the timing difference in the case of having a register in the FPGA to having no-register in the MPGA and use the appropriate delay numbers to calculate signal delays in the MPGA. As seen to one of ordinary skill, when the transistor layouts and metal interconnects layouts are substantially identical between regions **1802** in FIG. **18**A, and region **1802** in FIG. **18**C; the logic placement is identical and logical net connects is identical and the timing delays are also identical within the region **1802** in both devices. Only input/output delays are different, but the difference is pre-characterized and known such that the software tool is able to provide an accurate design conversion from the FPGA to the cheaper and economical smaller MPGA. One of ordinary skill may appreciate that a design placed and routed in the region **1803** within FPGA in FIG. **18**A, can be identically placed and routed in MPGA shown in FIG. **18**D; and a design placed and routed in the region **1801** within FPGA in FIG. **18**A, can be identically placed and routed in MPGA shown in FIG. **18**B. It can also be seen the subset of I/O pad structures for the region **1803** in FIG. **18**A is matched to I/O structures in MPGA of FIG. **18**D without the need to duplicate registers **1831** of FIG. **18**A in the MPGA of FIG. **18**D.

The arrangement of regions **1801**-**1803** is not limited to concentric regions, or to only two regions. It is conceivable that many regions may exist within the MPGA, each region uniquely mapping to an MPGA of equal resources and I/O density. A second embodiment of an FPGA is shown in FIG. **19**A, which further provides the ability to port identically mapped designs to MPGAs shown in FIGS. **19**B, **19**C and **19**D. The methodology to map designs is identical to that described for FIG. **18**, and not repeated. The difference between FIG. **18**A and FIG. **19**A is in the manner in which regions **1901**, **1902** and **1903** are defined. All three regions now contain two edges of the die, thus sharing common pads $\mathbf{1911}_{A1}$-$\mathbf{1911}_{C1}$ and $\mathbf{1911}_{A4}$-$\mathbf{1911}_{C4}$. These I/O structures do not require special registers to interface with circuit blocks within regions **1901**-**1903** regardless of the region chosen for the placement. However, regions **1902** and **1903** comprise at least two region boundaries that do not coincide with the I/O boundary. Such regional boundaries are provided with registers (registers **1921** for region **1902**, and registers **1931** for region **1903**) similar to that in FIG. **18**A for the software tool to pick when coupling to I/O structures $\mathbf{1911}_{A2}$-$\mathbf{1911}_{C2}$ and $\mathbf{1911}_{A3}$-$\mathbf{1911}_{C3}$ is required.

An integrated circuit design platform, comprising: a field programmable gate array (FPGA) prototype device in FIG. **18**A comprised of: a circuit layout (in region **1801**) comprising a plurality of field programmable logic blocks and a plurality of layers of field programmable interconnects; and a set of input/output pad structures **1811**; and a first region **1802** within the circuit layout, said region having registers **1821** at one or more boundaries of the region, a said register capable of coupling to a said input/output pad structure **1811**; and a first metal programmable gate array (first MPGA FIG. **18**C) production device comprised of: an substantially identical layout of programmable logic blocks in region **1902** as in the first region **1802** of the FPGA FIG. **18**A; and a substantially identical layout of one or more layers of programmable interconnects as in the first region **1802** of the FPGA; wherein, a design mapped into the first region **1802** of the FPGA FIG. **18**A is identically mapped to the first MPGA of FIG. **19**C.

The design platform, wherein the FPGA in FIG. **18**A comprises a configuration circuit to field program the programmable logic blocks and programmable interconnects; and the logic blocks are formed on a first module layer and the configuration circuit is formed on a second module layer positioned substantially above the first module layer. Thus when the configuration circuits are converted from user programmable circuits to mask-programmable circuits in the MPGA, the underlying logic placement and the interconnects are substantially kept identical between the two devices, enabling a very easy and simple design conversion from the expensive FPGA to cheaper and better MPGA.

An integrated circuit design platform, comprising: a prototype field programmable (FPGA) device in FIG. **19**A comprising a layout of electronic circuits (in region **1901**) and input/output pads **1911**; and a production mask programmable (MPGA) device in FIG. **19**C comprising: a layout of electronic circuits (in region **1902**) substantially identical to a region **1802** within the prototype FPGA in FIG. **19**A; and a subset of input/output pads **1911** as within the prototype FPGA **1811**; wherein, a design placed and routed within the region **1802** of the prototype FPGA using the subset of input/output pads **1811** as in the production MPGA **1911** is identically placed and routed in the production MPGA in FIG. **19**C.

A smaller mask programmable gate array (MPGA) device in FIG. **19**D derived from a larger field programmable gate

array (FPGA) device in FIG. **19**A, comprising: a layout of transistors and a plurality of interconnect layers (in region **1903** of FIG. **19**D) substantially identical to a region (**1903** in FIG. **19**A) of the FPGA; and input/output pads (**1911** in FIG. **19**D) matching a subset of the input/output pads of the FPGA (**1911** in FIG. **19**A); wherein, a design that is mapped to said region of the FPGA (**1901** in FIG. **19**A) device using said subset of input/output pads (**1911** in FIG. **19**A) by a user programmable means can be identically mapped to the MPGA (in FIG. **19**D) by a hard-wire circuit.

A method of mapping a design to a smaller region of an FPGA (**1802** in FIG. **18**A), comprising: inserting a register **1821** at a boundary of a smaller region **1802** of the FPGA during logic placement; and coupling a logic block (within region **1802**) to said register **1821**, and coupling said register **1821** to an input/output pad **1811** located at the edge of the FPGA die. The method further comprised of identically mapping the same design to an MPGA (in FIG. **18**C) comprising substantially identical transistor layout of said region **1802** of the FPGA, comprising: coupling said logic block to an input/output pad of the MPGA die (**1811** in FIG. **18**C) without the intermediate register.

In the conversion platform comprising the FPGA in FIG. **18**A and MPGA in FIG. **18**C, said region **1802** of the FPGA and said MPGA further comprises an exact layout of one or more pass-gate devices to couple a said programmable logic block to a said interconnect wire, wherein: in the FPGA, the pass-gate device couples the logic block to the interconnect wire, said pass-gate device controlled by an output of a RAM bit, said RAM bit comprising: a logic one to couple the logic block to the interconnect wire; and a logic zero to decouple the logic block from the interconnect wire; and in the MPGA, the pass-gate device couples the logic block to the interconnect wire, said pass-gate device controlled by an output of a ROM bit, said ROM bit comprising: a metal connection to power bus to couple the logic block to the interconnect wire; and a metal connection to ground bus to decouple the logic block from the interconnect wire. When the RAM bit in a 3D configuration within the FPGA is replaced by a hard-wired ROM bit in the MPGA, the circuit behavior and interconnect pattern is maintained between the two devices.

In the conversion platform comprising the FPGA in FIG. **18**A and MPGA in FIG. **18**C, said region **1802** of the FPGA and said MPGA further comprises an exact layout of one or more pass-gate devices to couple said one or more logic blocks to a said interconnect wire, wherein: in the FPGA, a pass-gate device couples a logic block to an interconnect wire, said pass-gate device controlled by an output of a RAM bit, said RAM bit comprising: a logic one to couple the logic block to the interconnect wire; and a logic zero to decouple the logic block from the interconnect wire; and in the MPGA, said pass-gate device is decoupled from said interconnect wire when the RAM bit comprises a logic zero; and in the MPGA, said pass-gate device is replaced by a metal jumper when the RAM bit comprises a logic one. By this conversion, an interconnect wire in the FPGA comprises a high capacitance due to the pass-gate device junctions coupled to the interconnect wire, and wherein said interconnect wire in the MPGA comprises less capacitance due to the pass-gate junctions decoupled from the interconnect wire; and an interconnect wire coupled to a logic block encounters a high resistance from the on pass-gate device in the FPGA, and wherein said interconnect wire coupled to the logic block in the MPGA encounters less resistance due to the metal-jumper. It is easily appreciated that in an FPGA any given wire segment is coupled to a plurality of logic blocks, both inputs and outputs of logic blocks, which makes FPGA slower and consume more power. In the conversion to an MPGA, much of the unwanted logic blocks can be decoupled from the wire segment, thereby making the MPGA faster or consume less power. It can be further appreciated that the position of the wire segment has not altered between the FPGA and the MPGA, thus the wire delays are easily pre-characterized and the conversion is made bit-stream compatible.

Clearly the new modular concept described in this and enclosed-by-reference applications disclose devices where the configuration circuit module layer is positioned above a circuit module layer. Thus between the FPGA (that comprises user RAM configuration module layer) and the derivative MPGA (that comprises a hard-wire configuration circuit) the circuit transistors in first module layer comprises an identical layout. The interconnect wires between the two devices also comprises a nearly identical layout: in a first embodiment when RAM bit is replaced by a ROM bit, the interconnect is identical; and in a second embodiment when nodes are disconnected from wire segments and pass-gates are replaced by metal jumpers, the interconnect is nearly identical. In both cases, the interconnect segments have not positionally changed.

Next, a brief description of the manufacturing process is discussed. During manufacturing, one or more digital circuits can be formed on a substrate. Next, the process selectively fabricates either a memory circuit or a conductive pattern substantially above the digital circuits to control portion of digital circuits. Finally, the process fabricates an interconnect and routing layer substantially above the digital circuits and memory circuits to connect digital circuits and one of the memory circuit or the conductive pattern.

The process can be modified to fabricate a generic field programmable gate array (FPGA) with the constructed memory circuit or an application specific integrated circuit (ASIC) with the constructed conductive pattern. Multiple ASICs can be fabricated with different variations of conductive patterns. The memory circuit and the conductive pattern have one or more substantially matching circuit characteristics. In this case, timing characteristics substantially unchanged by the circuit control option. The process thus fabricates a programmable logic device by constructing digital circuits on a substrate; and constructing a non-planar circuit on the substrate after constructing the digital circuits, the non-planar circuit being either a memory deposited to store data to configure the digital circuits to form a field programmable gate array (FPGA) or a conductive pattern deposited to hard-wire the digital circuits to form an application specific integrated circuit (ASIC), wherein the deposited memory and the conductive pattern have substantially matching timing characteristics. In another embodiment, the hard-wire ASIC option may be incorporated into the digital circuit layer **100**. In another embodiment, the hard-wire ASIC option is incorporated into the routing layer **110**.

Although an illustrative embodiment of the present invention, and various modifications thereof, have been described in detail herein with reference to the accompanying drawings, it is to be understood that the invention is not limited to this precise embodiment and the described modifications, and that various changes and further modifications may be effected therein by one skilled in the art without departing from the scope or spirit of the invention as defined in the appended claims.

What is claimed is:

1. An integrated circuit design platform, comprising:

a field programmable gate array (FPGA) prototype device comprised of:

a circuits layout comprising a plurality of field programmable logic blocks and a plurality of layers of field programmable interconnects; and

a set of input/output pad structures; and

a first region within the circuits layout, said region having registers at one or more boundaries of the region, said registers capable of coupling to said input/output pad structures; and

a metal programmable gate array (MPGA) production device fabricated separately from the FPGA prototype device, the MPGA production device comprised of:

a substantially identical circuit layout as in the first region of the FPGA; and

a substantially identical layout of one or more layers of programmable interconnects as in the first region of the FPGA;

wherein, a design mapped to the first region of the FPGA is identically mapped to the MPGA.

2. The platform of claim **1**, wherein the FPGA comprises a configuration circuit to field program the programmable logic blocks and programmable interconnects.

3. The platform of claim **2**, wherein in the FPGA, the transistors for logic circuits are formed on a first module layer and the configuration circuit is formed on a second module layer positioned substantially above or below the first module layer.

4. The platform of claim **2**, wherein the configuration circuit comprises one or more of: resistor, capacitor, SRAM cell, DRAM cell, Flash cell, EPROM cell, EEPROM cell, Carbon nano-tube, resistance modulating element, ferro-electric element, electro-chemical cell, electro-mechanical element, optical element and electro-magnetic cell.

5. The platform of claim **1**, wherein the MPGA comprises a customized metal circuit to mask program the programmable logic blocks and programmable interconnects.

6. The platform of claim **5**, wherein the customized metal circuit comprises one or more of: wire connection, wire disconnect, via connection, resistor element, shorted capacitor, capacitor, power bus connection, ground bus connection, transistor short, logic zero output connection, and logic one output connection.

7. The platform of claim **1**, wherein said first region of the FPGA and said **[**first**]** MPGA further **[**comprises**]** *comprise* a substantially identical layout of one or more pass-gate devices to couple said programmable logic block to **[**said**]** *an* interconnect wire, wherein:

in the FPGA, the pass-gate device couples the logic block to the interconnect wire, said pass-gate device controlled by an output of a RAM bit, said RAM bit comprising:

a logic one to couple the logic block to the interconnect wire; and

a logic zero to decouple the logic block from the interconnect wire; and

in the MPGA, the pass-gate device couples the logic block to the interconnect wire, said pass-gate device controlled by an output of a ROM bit, said ROM bit comprising:

a metal connection to *a* power bus to couple the logic block to the interconnect wire; and

a metal connection to *a* ground bus to decouple the logic block from the interconnect wire.

8. The platform of claim **1**, wherein said first region of the FPGA and said **[**first**]** MPGA further **[**comprises**]** *comprise* a substantially identical layout of one or more pass-gate devices to couple said one or more logic blocks to **[**said**]** *an* interconnect wire, wherein:

in the FPGA, a pass-gate device couples a logic block to an interconnect wire, said pass-gate device controlled by an output of a RAM bit, said RAM bit comprising:

a logic one to couple the logic block to the interconnect wire; and

a logic zero to decouple the logic block from the interconnect wire; and

in the MPGA, said pass-gate device is decoupled from said interconnect wire when the RAM bit comprises a logic zero;

wherein an interconnect wire in the FPGA comprises a high capacitance due to the pass-gate device junctions coupled to the interconnect wire, and wherein said interconnect wire in the MPGA comprises less capacitance due to the pass-gate junctions decoupled from the interconnect wire.

9. The platform of claim **1**, wherein said first region of the FPGA and said **[**first**]** MPGA further **[**comprises**]** *comprise* a substantially identical layout of one or more pass-gate devices to couple said one or more logic blocks to **[**a said**]** *an* interconnect wire, wherein:

in the FPGA, a pass-gate device couples a logic block to **[**an**]** *said* interconnect wire, said pass-gate device controlled by an output of a RAM bit, said RAM bit comprising:

a logic one to couple the logic block to the interconnect wire; and

a logic zero to decouple the logic block from the interconnect wire; and

in the MPGA, said pass-gate device is replaced by a metal jumper when the RAM bit comprises a logic one;

wherein an interconnect wire coupled to a logic block encounters a high resistance from the on pass-gate device in the FPGA, and wherein said interconnect wire coupled to the logic block in the MPGA encounters less resistance due to the metal-jumper.

10. The platform of claim **1**, further comprising:

said field programmable gate array (FPGA) prototype device comprised of:

a second region within the circuits layout, said second region having registers at one or more boundaries of the region, said **[**register**]** *registers* capable of coupling to said input/output pad structure; and

a second metal programmable gate array (second MPGA) production device comprised of:

a substantially identical circuits layout as in the second region of the FPGA; and

a substantially identical layout of one or more layers of programmable interconnects as in the second region of the FPGA;

wherein, a design mapped to the second region of the FPGA is identically mapped to the second MPGA.

11. An integrated circuit design platform, comprising:

a prototype field programmable (FPGA) device comprising a layout of electronic circuits and input/output pads; and

a production mask programmable (MPGA) device fabricated separately from the FPGA prototype device, the MPGA device comprising:

a layout of electronic circuits substantially identical to a region within the prototype FPGA device; and

a subset of input/output pads as within the prototype FPGA device;

wherein a design placed and routed within the region of the prototype FPGA device using the subset of input/output pads as in the production MPGA device, is identically placed and routed in the production MPGA device.

12. The platform of claim 11, wherein the region of the FPGA and the MPGA has substantially identical layouts of transistors and substantially identical layouts of one or more layers of interconnects.

13. The platform of claim 11, wherein *the* MPGA has at least one customized interconnect layer to map field programmable data in the prototype FPGA to mask programmable data.

14. The platform of claim 11, wherein the FPGA comprises a configuration circuit further comprising one or more of: resistor, capacitor, SRAM cell, DRAM cell, Flash cell, EPROM cell, EEPROM cell, Carbon nano-tube, resistance modulating element, ferro-electric element, electro-chemical cell, electro-mechanical element, optical element and electromagnetic cell.

15. The platform of claim 11, wherein the MPGA comprises a customized metal circuit further comprising one or more of: wire connection, wire disconnect, via connection, resistor element, shorted capacitor, capacitor, power bus connection, ground bus connection, transistor short, logic zero output connection, and logic one output connection.

16. The platform of claim 11, wherein the FPGA comprises a RAM bit, and wherein the MPGA comprises a hard-wired ROM bit.

17. The platform of claim 11, wherein the MPGA comprises one or more of:

a metal link to couple a node to a power supply voltage; and

a metal link to couple a node to a ground supply voltage; and

a metal jumper to short two nodes; and

a metal disconnect to isolate two nodes.

18. A small mask programmable gate array (MPGA) device derived from a large field programmable gate array (FPGA) device fabricated separately from the FPGA prototype device, the MPGA device, comprising:

a layout of transistors and a plurality of interconnect layers substantially identical to a smaller region of the FPGA device; and

input/output pads matching a subset of the input/output pads of the FPGA device;

wherein, a design that is mapped to said small region of the FPGA device using said subset of input/output pads by a user programmable means is identically mapped to the MPGA device by a hard-wire circuit during a subsequent fabrication of the MPGA device.

19. The device of claim 18, further comprising:

a mask programmable metal-circuit in lieu of a user programmable configuration circuit of the FPGA; and

a logic block to input/output pad connection in lieu of a logic block to a register at the boundary of said smaller region to an input/output pad connection of the FPGA.

20. The device of claim 18, wherein:

a first set of input/output [pad] *pads* to a first set of logic blocks within the MPGA is identically mapped from that of the FPGA; and

a second set of input/output [pad] *pads* to a second set of logic blocks within the MPGA is mapped from the corresponding logic blocks to registers at region boundary to corresponding input/output pads of the FPGA.

21. A method of producing a metal programmable gate array (MPGA), said method comprising:

accessing a design for a circuits layout for a field programmable gate array (FPGA), said circuits layout in said design comprising a plurality of field programmable logic blocks configured in a core region, said core region in said design comprising a plurality of sub-regions that are smaller than said core region, said design for said FPGA further comprising a first region within said circuits layout and a register at a boundary of said first region, a plurality of layers of field programmable interconnects, and a set of input/output (I/O) pad structures; and

fabricating an MPGA device based on said design, said design configured so that said fabricating of said MPGA device is separate from fabrication of an FPGA according to said design, said MPGA device comprising a circuit layout comprising a subset of said plurality of said sub-regions that are in said design, said circuit layout of said MPGA device comprising a substantially identical layout as in said first region of said design but excluding said register at said boundary of said first region, wherein said MPGA device further comprises a substantially identical layout of one or more of said layers of programmable interconnects as in said first region of said design;

wherein said first region in said design is identically mapped to said MPGA device.

22. The method of claim 21, wherein said design for said FPGA comprises a plurality of input/output (I/O) pads arranged around a perimeter of said core region, wherein said MPGA device interfaces with a subset of said plurality of I/O pads that are in said design.

23. The method of claim 21, wherein said design comprises power and ground pad structures that are arranged in positions that are common to all of said sub-regions.

24. The method of claim 21, wherein said MPGA device further comprises:

a first module layer comprising field programmable logic blocks; and

a second module layer adjacent to said first module layer and comprising routing circuitry coupled to said field programmable logic blocks of said MPGA device.

25. The method of claim 21, further comprising:

identifying a timing difference between an input/output (I/O) delay for said design and an I/O delay determined for said MPGA device; and

compensating for said timing difference in said MPGA device.

26. The method of claim 21, wherein said plurality of sub-regions comprises a first region and a second region positioned concentrically about the perimeter of said first region.

27. The method of claim 21, wherein sub-regions of said plurality of sub-regions share a first boundary and a second boundary.

28. The method of claim 21, wherein said MPGA device further comprises a customized metal circuit to mask program programmable logic blocks and programmable interconnects in said MPGA device.

29. The method of claim 28, wherein said customized metal circuit is selected from the group consisting of: wire connection, wire disconnect, via connection, resistor element, shorted capacitor, capacitor, power bus connection, ground bus connection, transistor short, logic zero output connection, and logic one output connection.

30. The method of claim 21, wherein said MPGA device and said first region of said design further comprise a substantially identical layout of one or more pass-gate devices configured to couple a field programmable logic block to an interconnect wire, wherein in said MPGA device, said pass-gate device is operable for coupling said logic block to said interconnect wire, said pass-gate device is controlled by an output of a ROM bit, and said ROM bit comprises:

a metal connection to a power bus to couple said field programmable logic block to said interconnect wire; and

a metal connection to a ground bus to decouple said field programmable logic block from said interconnect wire.

31. The method of claim 21, wherein said MPGA device and said first region of said design further comprise a substantially identical layout of one or more pass-gate devices configured to couple a field programmable logic block to an interconnect wire, wherein in said MPGA device, said pass-gate device is decoupled from said interconnect wire when a RAM bit comprises a logic zero.

* * * * *