(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 April 2017 (06.04.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/055833 A1

(51) International Patent Classification:
G06F 17/50 (2006.01) *H01L 27/02* (2006.01)
G06F 3/0486 (2013.01) *G06F 3/0481* (2013.01)

(21) International Application Number:
PCT/GB2016/053006

(22) International Filing Date:
28 September 2016 (28.09.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/873,918 2 October 2015 (02.10.2015) US

(71) Applicant: **ARM LTD** [GB/GB]; 110 Fulbourn Road, Cambridge CB1 9NJ (GB).

(72) Inventor: **CASSARD, Alexis**; 110 Fulbourn Road, Cambridge CB1 9NJ (GB).

(74) Agent: **TLIP LTD**; Leeds Innovation Centre, 103 Clarendon Road, Leeds Yorkshire LS2 9DF (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) Title: STANDARD CELL ARCHITECTURE LAYOUT

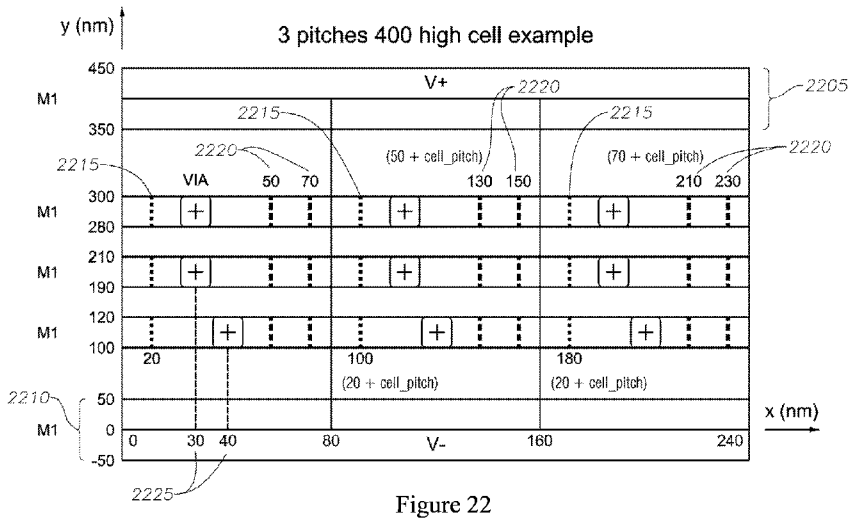


Figure 22

(57) Abstract: Various implementations described herein are directed to providing standard cell architecture layout design. A request to activate a grid is received. A request to place at least one edge of a shape at a particular location on the grid is received from an input device. The at least one edge of the shape is automatically placed in an allowed location based on pre-defined rules.

WO 2017/055833 A1

STANDARD CELL ARCHITECTURE LAYOUT

[0001] As technology has advanced, tools have been created to automatically generate integrated circuit layouts based on a pattern for simple standard cells. Usually, a first set of cells is done by hand. The process for creating this first set of cells can be time consuming. Even after the first set of cells is created, the need for a manual layout may continue to be present throughout the development of a project. A manual layout may also be needed even later for very specific cells.

[0002] Prior art tools also allow for placement of shapes on layers, however, these tools cannot be used on all shapes, are based only on a regular pitch grid (nanometer grid position), do not work with discrete values, and are based only on foundry rules. Further, prior art tools that are based on nanometer grid positions allow the designer to move items to improper positions, which will produce errors that will not be found until other design checks are applied.

[0003] Implementations of various techniques are described herein with reference to the accompanying drawings. It should be understood, however, that the accompanying drawings illustrate only the various implementations described herein and are not meant to limit the scope of various techniques described herein.

[0004] Figure 1 illustrates a virtual grid for a via in accordance with various implementations described herein.

[0005] Figure 2 illustrates moving a shape from one rail to another on a horizontal metal (M1) layer in accordance with various implementations described herein.

[0006] Figure 3 illustrates moving a shape from one position to another on a horizontal layer in accordance with various implementations described herein.

[0007] Figure 4 illustrates stretching a shape from one position to another in accordance with various implementations described herein.

[0008] Figure 5 illustrates stretching a shape from one position to another in accordance with various implementations described herein.

[0009] Figure 6 illustrates stretching a shape from one position to another in accordance with various implementations described herein.

[0010] Figure 7 illustrates a grid for a horizontal metal (M1) layer in accordance with various implementations described herein.

[0011] Figure 8 illustrates a grid for a ia layer in accordance with various implementations described herein.

[0012] Figure 9 illustrates a grid for a horizontal metal (M1) and via layer in accordance with various implementations described herein.

[0013] Figure 10 illustrates a via copy and paste operation in accordance with various implementations described herein.

[0014] Figure 11 illustrates a via copy and paste operation in accordance with various implementations described herein.

[0015] Figure 12 illustrates a via copy and paste operation in accordance with various implementations described herein.

[0016] Figure 13 illustrates a metal copy and paste operation in accordance with various implementations described herein.

[0017] Figure 14 illustrates a metal copy and paste operation in accordance with various implementations described herein.

[0018] Figure 15 illustrates a metal copy and paste operation in accordance with various implementations described herein.

[0019] Figure 16 illustrates a metal copy and paste operation in accordance with various implementations described herein.

[0020] Figure 17 illustrates a metal stretch operation in accordance with various implementations described herein.

[0021] Figure 18 illustrates a metal stretch operation in accordance with various implementations described herein.

[0022] Figure 19 illustrates a metal stretch operation in accordance with various implementations described herein.

[0023] Figure 20 illustrates a method for providing standard cell architecture layout design in accordance with various implementations described herein.

[0024] Figure 21 illustrates a method for providing standard cell architecture layout design in accordance with various implementations described herein.

[0025] Figure 22 illustrates example pre-defined coordinates for layers in a standard cell architecture.

[0026] Figure 23 illustrates computer system for providing standard cell architecture layout design in accordance with various implementations described herein.

[0027] The present disclosure provides a tool that optimizes and accelerates manual layout development. On new nodes, standard cell architectures are defined. The standard cell architectures include grid base layouts. The grid based layouts are mainly composed of rectangular shapes and define a very limited number of possible positions for the shapes. Rules are created in the layout development tool that propose architecture positions instead of a nanometer grid position. This layout development tool could be integrated with a computer-aided design (CAD) tool and the rules could be adjustable on the fly.

[0028] The present tool provides the ability to create, copy, move, or stretch a shape from one allowed position to another, removing all of the intermediate positions so that there is no need to measure each time to verify whether foundry design rules or architecture rules are satisfied. Thus, instead of creating/copying/moving/stretching a shape based on a nanometer grid (with possibly hundreds of different positions), there is instead a low number of clear positions for each layer.

[0029] The grid values include centering attributes for different layers, e.g., via, pin, Y value of horizontal metal. The grid values also include edge attributes for the various shapes, e.g., X value of horizontal metal. The values allowed for each layer are stored in an architecture file. The architecture file may be easily and quickly accessed to provide time savings to the layout designer.

[0030] In one implementation, the present tool receives a request to move, create, or stretch a shape. When the shape is moved within to the vicinity of a proper position (based on the architectural rules), the shape is automatically snapped to a proper position. The tool does not show all of the possible positions. In one implementation, the tool only shows the proper place or position where the shape should be placed.

[0031] The tool determines where to snap using pre-defined grid coordinates or rules, e.g., grid positions, for each shape. These pre-defined grid coordinates control the various options for each shape. Grid coordinates are activated for each shape editing option, e.g., create, move, copy, and stretch. The tool will not allow a designer to place a shape anyplace that does not conform to the pre-defined grid positions.

[0032] The pre-defined rules are based on the standard cell architecture. By creating pre-defined rules based on the architecture, it is not possible to move, copy, create, or stretch a shape in a position that does not conform to the pre-defined grid coordinates. For example, in a metal layer, based on the standard cell architecture, the metal layer has fixed positions that cannot change. Every position is defined for every shape that is to be applied to this layer. If a designer desires to move a via on the metal layer, this via will only snap to positions defined for that shape in the pre-defined grid coordinates.

[0033] The pre-defined grid coordinates (left, right, bottom, top, and/or center) are saved in a file, e.g., an architecture file. These pre-defined grid coordinates are based on architectural rules for the different layers of the standard cell. The possible coordinates from the given (x, y) positions defined in the pre-defined grid are applied to each shape. The pre-defined grid coordinates provide a way to describe each shape and the possible coordinates for each shape.

[0034] A standard cell architecture normally has a plurality of layers. Standard cell architectures may also have different sizes. Standard cell architectures may have different sizes. The architecture size is determined by the number of metal pieces, e.g., tracks, that can go across the height. For example, 10 tracks that are 36 nm apart determine a 360 nm size standard cell architecture. As another example, 10 tracks that are 42 nm apart determine the 420 nm size standard cell architecture. The architecture is customized based on the foundry rules. The goal is to define an architecture that is as small as possible and still meets foundry rules. The pre-defined grid coordinates are based on the fixed positions of the layers in the architecture. The pre-defined snap on positions are based on the fixed positions of the layers in the architecture and the pre-defined grid coordinates.

[0035] Most layers of a standard cell architecture have fixed positions. In one implementation, the fixed positions can be updated or modified when the architecture

changes. Once the fixed positions are modified, the shapes can be modified to meet, e.g., snap into, the new fixed positions.

[0036] When the standard cell architecture is created and is determined to be correct after performing the appropriate checks, e.g., no design rule check (DRC) errors, the architecture can be described in terms of shape coordinates. The size and position of shapes to be placed on layers within the architecture are based on the predefined grid (e.g., (x, y)) coordinates. Using the predefined grid coordinates, each shape is able to automatically be placed, i.e., snapped, into the proper position.

[0037] In one implementation, the present design tool forces a designer to only go from one allowed position to another when a layer or shape is being moved, stretched, copied, or created. This eliminates the need to measure shapes and positions of the shapes each time to verify whether foundry design rules or architecture rules have been followed. In one implementation, the architecture file is created manually or generated based on a first layout containing all possible positions for all layers. The file should be humanly readable and easy to modify to be used for another purpose, for example deriving the architecture to another one in the same process.

[0038] Various implementations of a design tool that provides the ability to layout various shapes on layers of a standard cell architecture will now be described in reference to Figures 1-22.

[0039] Figure 1 illustrates a virtual grid for a via in accordance with various implementations described herein. The virtual grid shows allowed positions 102, 104, 106, 108. When a via 110 is being moved, the design tool only proposes the allowed position.

[0040] Figure 2 illustrates moving a shape from one rail to another on a horizontal metal (M1) layer in accordance with various implementations described herein. In this implementation, metal 206 is moved from one rail 204 to an adjacent rail 202.

[0041] Figure 3 illustrates moving a shape 302 from one position to another on a horizontal layer in accordance with various implementations described herein. If the shape 302 is not the proper size for the location 304 to which the shape is being moved, the shaped is resized to fit its new position based on the closest possible edges.

[0042] Figure 4, Figure 5 and Figure 6 illustrate stretching a shape from one position to another in accordance with various implementations described herein. Figure 4 illustrates possible left positions 402, 404 and right positions 406, 408, 410 for a shape. Figure 5 illustrates possible right positions 502, 504, 506 for a shape. Figure 6 illustrates possible vertical positions 602, 604, 606 for a shape.

[0043] When shapes are moved horizontally, both the left and right side of the shape should be snapped into the pre-defined positions. When stretching to the right side, the shape should snap to the closest "right side" allowed position. There is a differentiation between the edges of a shape. Having a different set of allowed positions provides better efficiency when stretching the right side or the left side of a horizontal layer or shape. This same efficiency is provided for top and bottom positions of a vertical layer or shape.

[0044] Figure 7 illustrates an example of a grid for a horizontal metal (M1). In this implementation, there are three identical M1 rails. Each rail has an allowed top position and bottom position, allowed left positions, and allowed right positions. Elements 702, 704, 706, 708 show the left allowed positions for metal on one of the M1 rails. Elements 710, 712, 714, 716 show the right allowed positions on metal for one of the M1 rails. Elements 718, 720 show the top and bottom allowed positions on one of the M1 rails.

[0045] Figure 8 illustrates an example of a grid for via layer. In this example, positions for allowed vias are shown.

[0046] Figure 9 illustrates an example of a grid for a horizontal metal (M1) and via layer. In this example, a combination of allowed positions for metal and vias is shown. In other words, Figure 9 illustrates a combination of the elements shown on Figure 7 and Figure 8. The grid shown in Figure 9 is used to illustrate various example shapes and how these example shapes can be moved, copied, and/or stretched as shown in Figures 10-19.

[0047] Figure 10, Figure 11 and Figure 12 illustrate an example of a via copy and paste operation in accordance with various implementations described herein. In Figure 10, via shape 1004 is copied from via shape 1002 located at one via position of the bottom M1 rail. Via shape 1004 is going to be moved, e.g., pasted, to via 1006. In Figure 11, via shape 1004 is placed near via 1006 of the middle M1 rail. In Figure 12, when via shape 1004 is placed within a threshold or approximate position

(defined by the pre-defined grid rules) of via 1006, via shape 1004 snaps into place. When an existing via shape is copied and the copied via shape is moved, for example, using an input device, to an allowed position, the via shape is automatically snapped into place in the new location. The via shape is placed in the correct location when, for example, the layout designer uses the input device to drop the via shape into place.

[0048] Figure 13, Figure 14, Figure 15 and Figure 16 illustrate an example of a metal copy and paste operation in accordance with various implementations described herein. Figure 13 illustrates a metal shape placed in the top M1 rail. Figure 14 shows a copy operation of the metal 1302 in the top M1 rail. In Figure 14, metal shape 1402 is copied from via shape 1302. Metal shape 1402 is to be moved, e.g., pasted, to a location on the middle M1 rail. In Figure 15, metal shape 1402 is placed near a location of the middle M1 rail. In Figure 16, when metal shape 1402 is placed within a threshold or approximate position (defined by the pre-defined grid rules) of an allowed position on the middle M1 rail, metal shape 1402 snaps into place. When an existing metal shape is copied and the copied metal shape is moved, for example, using an input device, to a new place, the metal shape is automatically resized in both width and length to fit the closest boundaries in order to snap into place in the new location. The metal shape is placed in the correct location when, for example, the layout designer uses the input device to drop the metal shape into place.

[0049] Figure 17 illustrates an example of a metal stretch operation in accordance with various implementations described herein. During a stretch operation the designer selects an edge of the metal shape to stretch in the "x" direction. A stretch operation is similar to a paste operation in that during the process of stretching the metal shape, the edge being moved by the designer is automatically placed in the next allowable position. Figure 18 illustrates an intermediate position during the stretching of metal shape 1402. In Figure 18, the left edge of metal shape 1402 has been stretched close to, i.e., within a certain distance of, an allowable position. Figure 19 illustrates the metal shape 1402 being stretched to a pre-defined left allowable position 1902. Once the left edge of metal shape 1402 is stretched to a threshold or approximate position near left allowable position 1902, the left side of metal shape 1402 automatically snaps into place or, for example, can snap into place upon receiving an indication from an input device being used by the designer.

[0050] Figure 20 illustrates a method 2000 for providing standard cell architecture layout design in accordance with various implementations described herein. In one implementation, method 2000 may be performed by a computing system. It should be understood that while method 2000 indicates a particular order of execution of operations, in some implementations, certain portions of the operations might be executed in a different order. Further, in some implementations, additional operations or steps may be added to method 2000. Likewise, some operations or steps may be omitted.

[0051] At block 2005 a grid is activated. At block 2010, a request to place at least an edge of a shape at a particular location on the grid is received from an input device. This input is configured to place the at least one edge of the shape in an approximate location. At block 2020, the at least one edge of the shape is automatically placed in an allowed location based on pre-defined rules of the standard cell architecture. An example of the pre-defined grid coordinates or rules is described below in Figure 22.

[0052] Method 200 is configured to determine where to place i.e., snap, the at least one edge of the shape, using pre-defined grid coordinates or rules, e.g., grid positions, for each shape. These pre-defined grid coordinates control the various options for each shape. Grid coordinates are activated for each shape option, e.g., create, move, copy, and stretch. Method 200 is configured to prevent a designer to place a shape anyplace that does not conform to the pre-defined grid positions.

[0053] The pre-defined rules may be based on the standard cell architecture. By creating pre-defined rules based on the architecture, it is not possible to move, copy, create, or stretch a shape in a position that does not conform to the pre-defined grid coordinates. For example, in a metal layer, based on the standard cell architecture, the metal layer has fixed positions that cannot change. Every position is defined for every shape that is to be applied to this layer. If a designer desires to move a via on the metal layer, this via will only snap to positions defined for that shape in the pre-defined grid coordinates.

[0054] The pre-defined grid coordinates (left, right, bottom, top and/or center) are saved in an architecture file. These pre-defined grid coordinates are based on architectural rules for the different layers of the standard cell. The possible coordinates from the given (x, y) positions defined in the pre-defined grid are applied to each shape. The pre-defined grid coordinates provide a way to describe each shape and the possible coordinates for each shape. The pre-defined grid coordinates

are not mandatory for all layers. In one implementation, the pre-defined grid coordinates are applied to a subset of the layers.

[0055] The architecture is customized based on the foundry rules. The goal of layout design is to define an architecture that is as small as possible and still meets foundry rules. The pre-defined grid coordinates are based on the fixed positions of the layers in the architecture. The pre-defined snap on positions are based on the fixed positions of the layers in the architecture and the pre-defined grid coordinates.

[0056] In one implementation, user selection of the shape is received before activating the grid. This user shape selection is meant to perform a move, copy, and/or stretch on existing shapes.

[0057] In another implementation, a shape is copied and the copied shape is placed on the grid. Edges of the shape are placed, i.e., snapped, in a closest allowed position of the allowed location.

[0058] In yet another implementation, a shape is moved from a first location to the allowed location. When a shape is moved, edges of the shape are placed in a closest allowed position of the allowed location.

[0059] In still another implementation, the at least one edge of the shape is stretched. The shape may be stretched vertically and/or horizontally. The grid allows for stretching of each edge of the shape to a plurality of allowed positions.

[0060] In one implementation, the shape is a new shape that is created and placed on the grid. An active layer of the grid is selected when the new shape is created. The active layer is the layer that will be used when creating a new shape. Edges of the shape are placed in a closest allowed position of the allowed location.

[0061] In yet another implementation, the grid is displayed to a designer. Only one or more proper positions for the shape are displayed to the designer. Displaying the proper positions in this manner helps prevent errors in layout design.

[0062] In still another implementation, the allowed location based on the pre-defined rules is described in an architectural file. The grid values are included in the architectural file. The grid values include centering attributes for different layers of the architecture. The grid values also include edge attributes for the shape.

[0063] Figure 21 illustrates an implementation of a method for providing standard cell architecture layout design in accordance with various implementations described herein. In one implementation, method 2100 may be performed by a computing system. It should be understood that while method 2100 indicates a particular order of execution of operations, in some implementations, certain portions of the operations might be executed in a different order. Further, in some implementations, additional operations or steps may be added to method 2100. Likewise, some operations or steps may be omitted.

[0064] At block 2105, a user, e.g., layout designer, shape selection is received. Shape selection by the user can also include selecting an edge of a shape.

[0065] At block 2110, a layer grid is activated. In one implementation, the grid can be displayed.

[0066] At block 2115, a user requests to move the shape. The user may also request to stretch, copy, or create a shape. When a user decides to create an object, there is no shape selection at block 2105. The active layer is taken into account for grid selection. The edges of the shape are snapped to the closest allowed position if this closest allowed position is different than the edge location for any of the edges. In the case of a move or copy operation, the shape is resized to have edges snap to the closest position in the same manner as the create operation.

[0067] At block 2120, a user moves a mouse cursor to an approximate location. At block 2125, the shape is instantly snapped to an allowed location, e.g., placed within a pre-defined allowable area of an allowed location. The allowed locations are described in an architectural file.

[0068] Figure 22 illustrates example pre-defined coordinates for layers in a standard cell architecture. The architecture example shown in Figure 22 has a cell height of 400 nm.

[0069] The coordinates can be the center of the shape or the edges of the shape. These coordinates can be a relative value or an absolute value.

[0070] Elements on which the architecture can be based are cell height and architecture pitch. The cell height value is not used in this example, however, it can be useful to determine layer positions relative to the cell height. The architecture

pitch is the distance between two transistor gates in a horizontal direction. Most "x" coordinates may be relative to the pitch value.

[0071] In the present example, the distance between V+ and V-, i.e., the cell height, is 400 nm. The cell pitch is 80 nm. There are three pitches shown in this example, e.g., with boundaries at 80, 160, and 240 nanometers, respectively.

[0072] Figure 22 also illustrates five M1 rails. Each M1 rail is defined using "y" coordinate couples. There is one rail for each power net (V+ and V-) at the bottom 2210 and top of the cell 2205. The rails for each power net are 100 nm wide. The coordinates for each power net are (-50 50) and (350 450). The other three M1 rails (the rails between the top and bottom rails) are 20 nm wide and are located at the following coordinates (100 120), (190 210), and (280 300).

[0073] In this example, one predefined left edge position 2215 is defined for each pitch on the M1 rails at coordinates (100 120), (190 210), and (280 300). This left edge position is located 20 nm from the left edge of each pitch along the x-axis. For example, for each of the M1 rails, there is a left edge for each pitch, e.g., at 0 nm, 80 nm and 160 nm. As such, for the M1 rails that are not power nets, the predefined left edge position for each pitch is 20 nm, 100 nm, and 180 nm.

[0074] Two defined right edge positions 2220 are defined for each pitch on the M1 rails at coordinates (100 120), (190 210), and (280 300). Each defined right edge position is located 50 nm and 70 nm, respectively from the left edge of each pitch. As such, a first right edge of each pitch is defined along the x-axis at 50 nm, 130 nm, 210 nm, respectively. A second right edge of each pitch is defined along the x-axis at 70 nm, 150 nm, and 230 nm, respectively.

[0075] In this example, vias are the same size, e.g., 20 nm, on each side, however, this is not a requirement. Other via sizes and shapes can be utilized. There are two defined positions 2225 per pitch for vias. Possible pre-defined positions for vias in this example are placed along the x-axis at 30 nm or 40 nm measured from the left edge of the pitch to the center of the via. Thus, in this example, possible pre-defined positions for the vias for each pitch on the M1 rails at coordinates (100 120), (190 210), and (280 300) are 30 nm, 40 nm, 110 nm, 120 nm, 190 nm, and 200 nm.

[0076] Figure 23 illustrates a computing system 2300 in accordance with implementations of various techniques described herein. The computing system 2300

may include a central processing unit (CPU) 2330, a system memory 2326, a graphics processing unit (GPU) 2331 and a system bus 2328 that couples various system components including the system memory 2326 to the CPU 2330. Although only one CPU 2330 is illustrated in Figure 23, it should be understood that in some implementations the computing system 2300 may include more than one CPU 2330.

[0077] The CPU 2330 may include a microprocessor, a microcontroller, a processor, a programmable integrated circuit, or a combination thereof. The CPU 2330 can comprise an off-the-shelf processor such as a Reduced Instruction Set Computer (RISC), or a Microprocessor without Interlocked Pipeline Stages (MIPS) processor, or a combination thereof. The CPU 2330 may also include a proprietary processor.

[0078] The GPU 2331 may be a microprocessor specifically designed to manipulate and implement computer graphics. The CPU 2330 may offload work to the GPU 2331. The GPU 2331 may have its own graphics memory, and/or may have access to a portion of the system memory 2326. As with the CPU 2330, the GPU 2331 may include one or more processing units, and each processing unit may include one or more cores.

[0079] The CPU 2330 may provide output data to a GPU 2331. The GPU 2331 may generate graphical user interfaces that present the output data. The GPU 2331 may also provide objects, such as menus, in the graphical user interface. A user may provide inputs by interacting with the objects. The GPU 2331 may receive the inputs from interaction with the objects and provide the inputs to the CPU 2330. A video adapter 2332 may be provided to convert graphical data into signals for a monitor 2334. The monitor 2334 includes a screen 2305. In certain implementations, the screen 2305 may be sensitive to touching by a finger. In other implementations, the screen 2305 may be sensitive to the body heat from the finger, a stylus, or responsive to a mouse. Additionally, in certain implementations, the screen may have the capability of displaying more than one plan position indicator (PPI).

[0080] The system bus 2328 may be any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus. The system memory 2326 may include a read only memory

(ROM) 2312 and a random access memory (RAM) 2316. A basic input/output system (BIOS) 2314, containing the basic routines that help transfer information between elements within the computing system 2300, such as during start-up, may be stored in the ROM 2312.

[0081] The computing system 2300 may further include a hard disk drive interface 2336 for reading from and writing to a hard disk 2350, a memory card reader 2352 for reading from and writing to a removable memory card 2356, and an optical disk drive 2354 for reading from and writing to a removable optical disk 2358, such as a CD ROM or other optical media. The hard disk 2350, the memory card reader 2352, and the optical disk drive 2354 may be connected to the system bus 2328 by a hard disk drive interface 2336, a memory card reader interface 2338, and an optical drive interface 2340, respectively. The drives and their associated computer-readable media may provide nonvolatile storage of computer-readable instructions, data structures, program modules and other data for the computing system 2300.

[0082] Although the computing system 2300 is described herein as having a hard disk, a removable memory card 2356 and a removable optical disk 2358, it should be appreciated by those skilled in the art that the computing system 2300 may also include other types of computer-readable media that may be accessed by a computer. For example, such computer-readable media may include computer storage media and communication media. Computer storage media may include volatile and non-volatile, and removable and non-removable media implemented in any method or technology for storage of information, such as computer-readable instructions, data structures, program modules or other data. Computer storage media may further include RAM, ROM, erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), flash memory or other solid state memory technology, CD-ROM, digital versatile disks (DVD), or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computing system 2300. Communication media may embody computer readable instructions, data structures, program modules or other data in a modulated data signal, such as a carrier wave or other transport mechanism and may include any information delivery media. The term "modulated data signal" may mean a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media may include wired

media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. The computing system 2300 may also include a host adapter 2333 that connects to a storage device 2335 via a small computer system interface (SCSI) bus, a Fiber Channel bus, an eSATA bus, or using any other applicable computer bus interface.

[0083] The computing system 2300 can also be connected to a router 2364 to establish a wide area network (WAN) 2366 with one or more remote computers 2374. The router 2364 may be connected to the system bus 2328 via a network interface 2344. The remote computers 2374 can also include hard disks 2372 that store application programs 2370.

[0084] In another implementation, the computing system 2300 may also connect to the remote computers 2374 via local area network (LAN) 2376 or the WAN 2366. When using a LAN networking environment, the computing system 2300 may be connected to the LAN 2376 through the network interface or adapter 2344. The LAN 2376 may be implemented via a wired connection or a wireless connection. The LAN 2376 may be implemented using Wi-Fi™ technology, cellular technology, Bluetooth™ technology, satellite technology, or any other implementation known to those skilled in the art. The network interface 2344 may also utilize remote access technologies (e.g., Remote Access Service (RAS), Virtual Private Networking (VPN), Secure Socket Layer (SSL), Layer 6 Tunneling (L2T), or any other suitable protocol). These remote access technologies may be implemented in connection with the remote computers 2374. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computer systems may be used.

[0085] A number of program modules may be stored on the hard disk 2350, memory card 2356, optical disk 2358, ROM 2312 or RAM 2316, including an operating system 2318, one or more application programs 2320, and program data 2324. In certain implementations, the hard disk 2350 may store a database system. The database system could include, for example, recorded points. The application programs 2320 may include various mobile applications ("apps") and other applications configured to perform various methods and techniques described herein. The operating system 2318 may be any suitable operating system that may control the operation of a networked personal or server computer.

[0086] A user may enter commands and information into the computing system 2300 through input devices such as buttons 2362, which may be physical buttons, virtual buttons, or combinations thereof. Other input devices may include a microphone, a mouse, or the like (not shown). These and other input devices may be connected to the CPU 2330 through a serial port interface 2342 coupled to system bus 2328, but may be connected by other interfaces, such as a parallel port, game port or a universal serial bus (USB).

[0087] In one implementation, the one or more application programs 2320 or 2370 stored in the computer-readable media can include a plurality of instructions that when executed by a processing unit, such as a CPU 2330, cause the computing system to perform any of the techniques, or portions thereof, that are described herein.

[0088] The layout tool of the present disclosure provides fast rules coding and allows rule changes on the fly. In addition, the present tool includes a switch functionality that may be used to easily disable/enable the tool.

[0089] The present disclosure provides several advantages:

- Efficiency may be increased on manual layout design. There is no need to check the x or y value of a shape while modifying. There is no need to use a ruler. Checking the position of the layer can be done visually with no need to zoom to be more precise when drawing.
- Layout consistency may be provided. Because the layout is on a grid, a cleaner and repetitive layout is provided across a library.
- It becomes easier to derive a whole library of shapes, and the number of possible patterns is reduced.
- Errors may be reduced. Situations that can cause an error become readily apparent, for example, when there is a gap between two consecutive positions for each layer.
- Learning time may be reduced for layout designers that are new to a project.

[0090] Described herein are implementations of various technologies for providing standard cell architecture layout design. A request to activate a grid is received. A request to place at least one edge of a shape at a particular location on the grid is

received from an input device. The at least one edge of the shape is automatically placed in an allowed location based on pre-defined rules.

[0091] Described herein is also a non-transitory computer-readable medium having stored thereon a plurality of computer-executable instructions which, when executed by a computer, cause the computer to: receive a request to activate a grid, receive a request to place at least one edge of a shape at a particular location on the grid from an input device, and automatically place the at least one edge of the shape in an allowed location based on pre-defined rules.

[0092] Described herein is also a device, including a computer system having a processor and memory having stored thereon a plurality of executable instructions which, when executed by the processor, cause the processor to: receive a request to activate a grid, receive a request to place at least one edge of a shape at a particular location on the grid from an input device, and automatically place the at least one edge of the shape in an allowed location based on pre-defined rules.

[0093] In one implementation, a user selection of the shape is received before activating the grid. The shape can be copied and placed on the grid. Edges of the shape can be placed in a pre-defined position within the allowed location. The shape can be moved from a first location to the allowed location. Moving the shape from the first location to the allowed location can include placing edges of the shape in a pre-defined position within the allowed location.

[0094] In another implementation, the pre-defined rules are based on a standard cell architecture. The pre-defined rules can include pre-defined grid coordinates. Coordinates defined in the pre-defined grid coordinates can be applied to each shape. The grid allows for stretching of each edge of the shape to a plurality of allowed positions.

[0095] In yet another implementation, the shape is a new shape that is created and placed on the grid. A layer of the grid is selected when the new shape is created. Edges of the new shape can be placed in a pre-defined position within the allowed location.

[0096] In again another implementation, one or more allowed positions are displayed for the shape.

[0097] In a further implementation, the allowed location based on the pre-defined rules is described in an architecture file. Grid values can be included in the architecture file. The grid values can include centering attributes for different layers of the grid. The grid values can include edge attributes for the shape.

[0098] The discussion of the present disclosure is directed to certain specific implementations. It should be understood that the discussion of the present disclosure is provided for the purpose of enabling a person with ordinary skill in the art to make and use any subject matter defined herein by the subject matter of the claims.

[0099] It should be intended that the subject matter of the claims not be limited to the implementations and illustrations provided herein, but include modified forms of those implementations including portions of the implementations and combinations of elements of different implementations within the scope of the claims. It should be appreciated that in the development of any such implementation, as in any engineering or design project, numerous implementation-specific decisions should be made to achieve a developers' specific goals, such as compliance with system-related and business related constraints, which may vary from one implementation to another. Moreover, it should be appreciated that such a development effort maybe complex and time consuming, but would nevertheless be a routine undertaking of design, fabrication, and manufacture for those of ordinary skill having benefit of this disclosure. Nothing in this application should be considered critical or essential to the claimed subject matter unless explicitly indicated as being "critical" or "essential."

[00100] Reference has been made in detail to various implementations, examples of which are illustrated in the accompanying drawings and figures. In the following detailed description, numerous specific details are set forth to provide a thorough understanding of the present disclosure. However, the present disclosure may be practiced without these specific details. In other instances, well-known methods, procedures, components, circuits and networks have not been described in detail so as not to unnecessarily obscure aspects of the embodiments.

[00101] It should also be understood that, although the terms first, second, etc. may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first object or step could be termed a second object or step, and, similarly, a second object or step could be termed a first object or step, without

departing from the scope of the invention. The first object or step, and the second object or step, are both objects or steps, respectively, but they are not to be considered the same object or step.

[00102] The terminology used in the description of the present disclosure herein is for the purpose of describing particular implementations and is not intended to limit the present disclosure. As used in the description of the present disclosure and appended claims, the singular forms "a," "an," and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. The term "and/or" as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. The terms "includes", "including", "comprises", and/or "comprising", when used in this specification, specify a presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components and/or groups thereof.

[00103] As used herein, the term "if" may be construed to mean "when" or "upon" or "in response to determining" or "in response to detecting," depending on the context. Similarly, the phrase "if it is determined" or "if [a stated condition or event] is detected" may be construed to mean "upon determining" or "in response to determining" or "upon detecting [the stated condition or event]" or "in response to detecting [the stated condition or event]," depending on the context. As used herein, the terms "up" and "down"; "upper" and "lower"; "upwardly" and "downwardly"; "below" and "above"; and other similar terms indicating relative positions above or below a given point or element may be used in connection with some implementations of various technologies described herein.

[00104] While the foregoing is directed to implementations of various techniques described herein, other and further implementations may be devised without departing from the basic scope thereof, which may be determined by the claims that follow.

[00105] Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

What Is Claimed Is:

1. A method for providing standard cell architecture layout design, comprising:
receiving a request to activate a grid;
receiving a request to place at least one edge of a shape at a particular location on the grid from an input device; and
automatically placing the at least one edge of the shape in an allowed location based on pre-defined rules.

2. The method of claim 1, further comprising receiving user selection of the shape before activating the grid.

3. The method of claim 2, further comprising:
copying the shape; and
placing the copied shape on the grid.

4. The method of claim 3, wherein edges of the shape are placed in a pre-defined position within the allowed location.

5. The method of claim 2, further comprising moving the shape from a first location to the allowed location.

6. The method of claim 5, wherein moving the shape from the first location to the allowed location comprises placing edges of the shape in a pre-defined position within the allowed location.

7. The method of claim 1, wherein the pre-defined rules are based on standard cell architecture.

8. The method of claim 7, wherein the pre-defined rules include pre-defined grid coordinates.

9. The method of claim 8, further comprising applying coordinates defined in the pre-defined grid coordinates to each shape.

10. The method of claim 7, wherein the grid allows for stretching of each edge of the shape to a plurality of allowed positions.

11. The method of claim 1, wherein the shape is a new shape that is created and placed on the grid.

12. The method of claim 11, further comprising selecting a layer of the grid when the new shape is created.

13. The method of claim 12, further comprising placing edges of the new shape in a pre-defined position within the allowed location.

14. The method of claim 1, further comprising displaying only one or more allowed positions for the shape.

15. The method of claim 1, wherein the allowed location based on the pre-defined rules is described in an architecture file.

16. The method of claim 15, wherein grid values are included in the architecture file.

17. The method of claim 16, wherein the grid values include centering attributes for different layers of the grid.

18. The method of claim 16, wherein the grid values include edge attributes for the shape.

19. A non-transitory computer-readable medium having stored thereon a plurality of computer-executable instructions which, when executed by a computer, cause the computer to:

receive a request to activate a grid;

receive a request to place at least one edge of a shape at a particular location on the grid from an input device; and

automatically place the at least one edge of the shape in an allowed location based on pre-defined rules.

20. A device, comprising:

a computer system having a processor and memory having stored thereon a plurality of executable instructions which, when executed by the processor, cause the processor to:

receive a request to activate a grid;

receive a request to place at least one edge of a shape at a particular location on the grid from an input device; and

automatically place the at least one edge of the shape in an allowed location based on pre-defined rules.

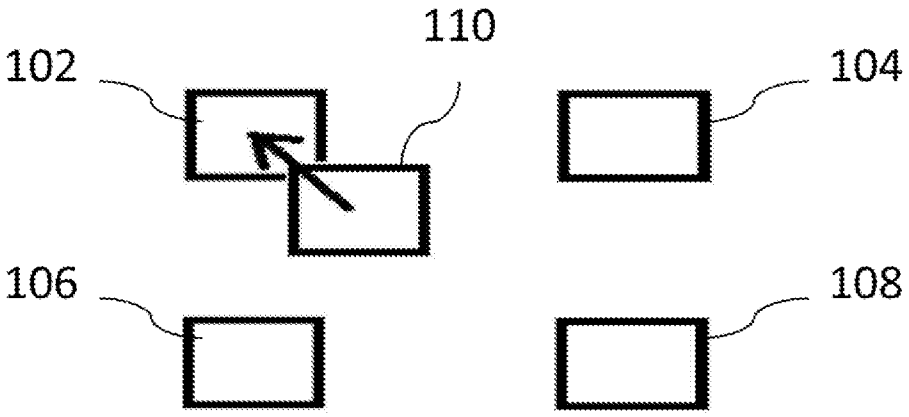


Figure 1

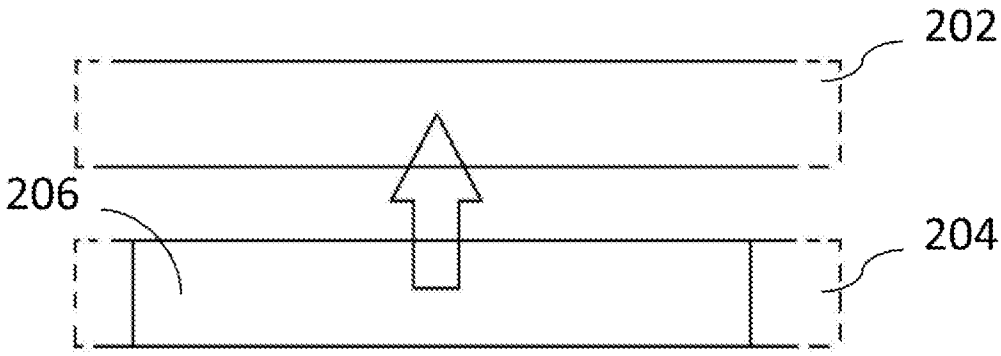


Figure 2

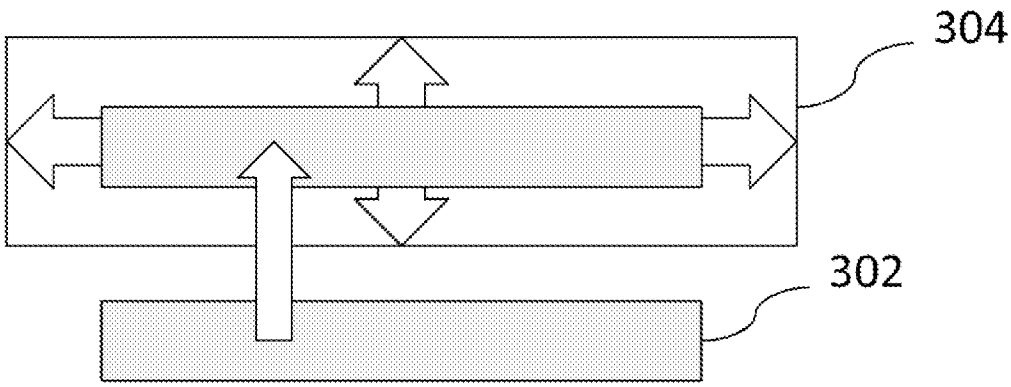


Figure 3

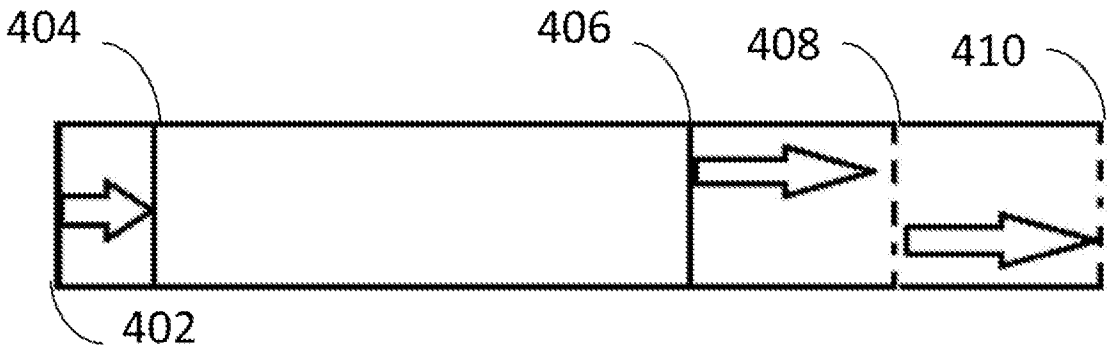


Figure 4

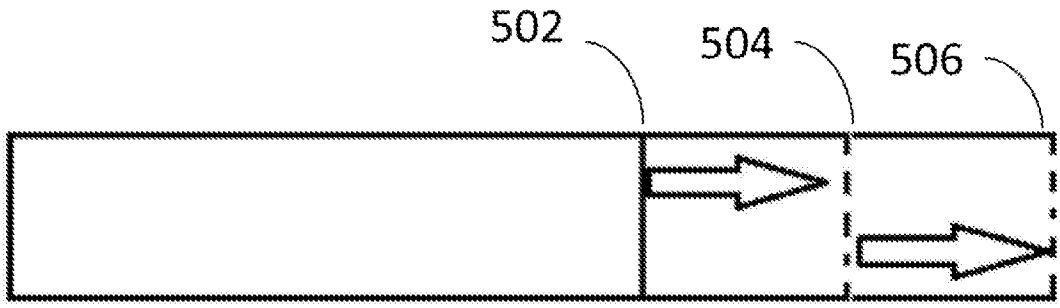


Figure 5

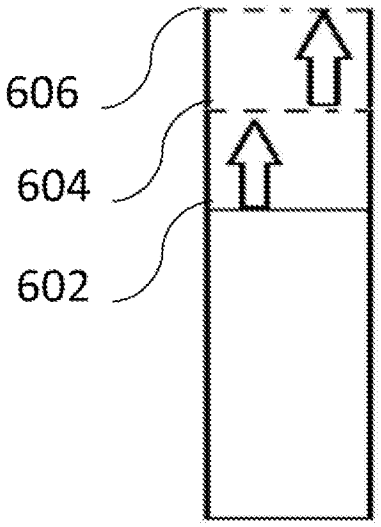


Figure 6

Figure 7

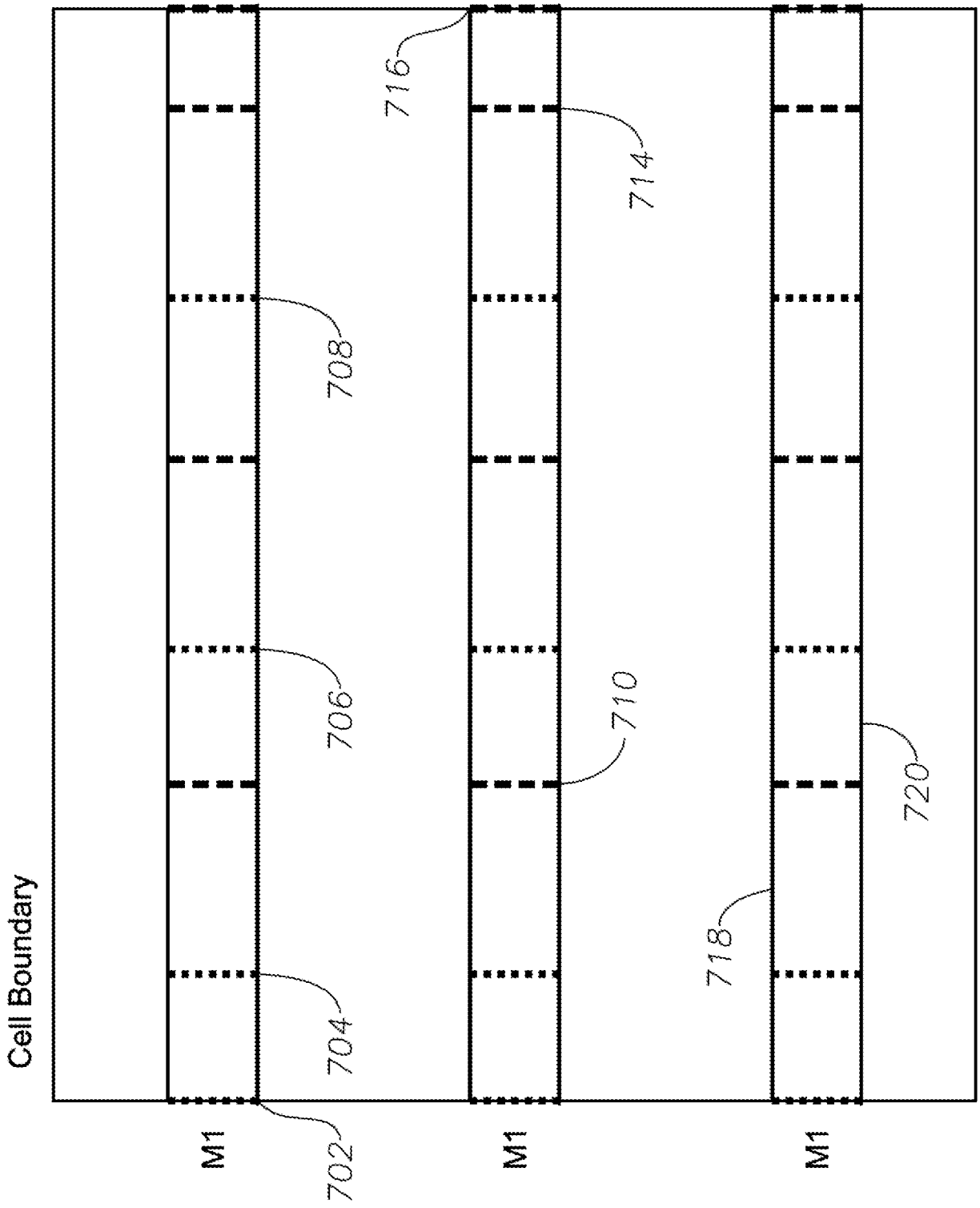


Figure 8

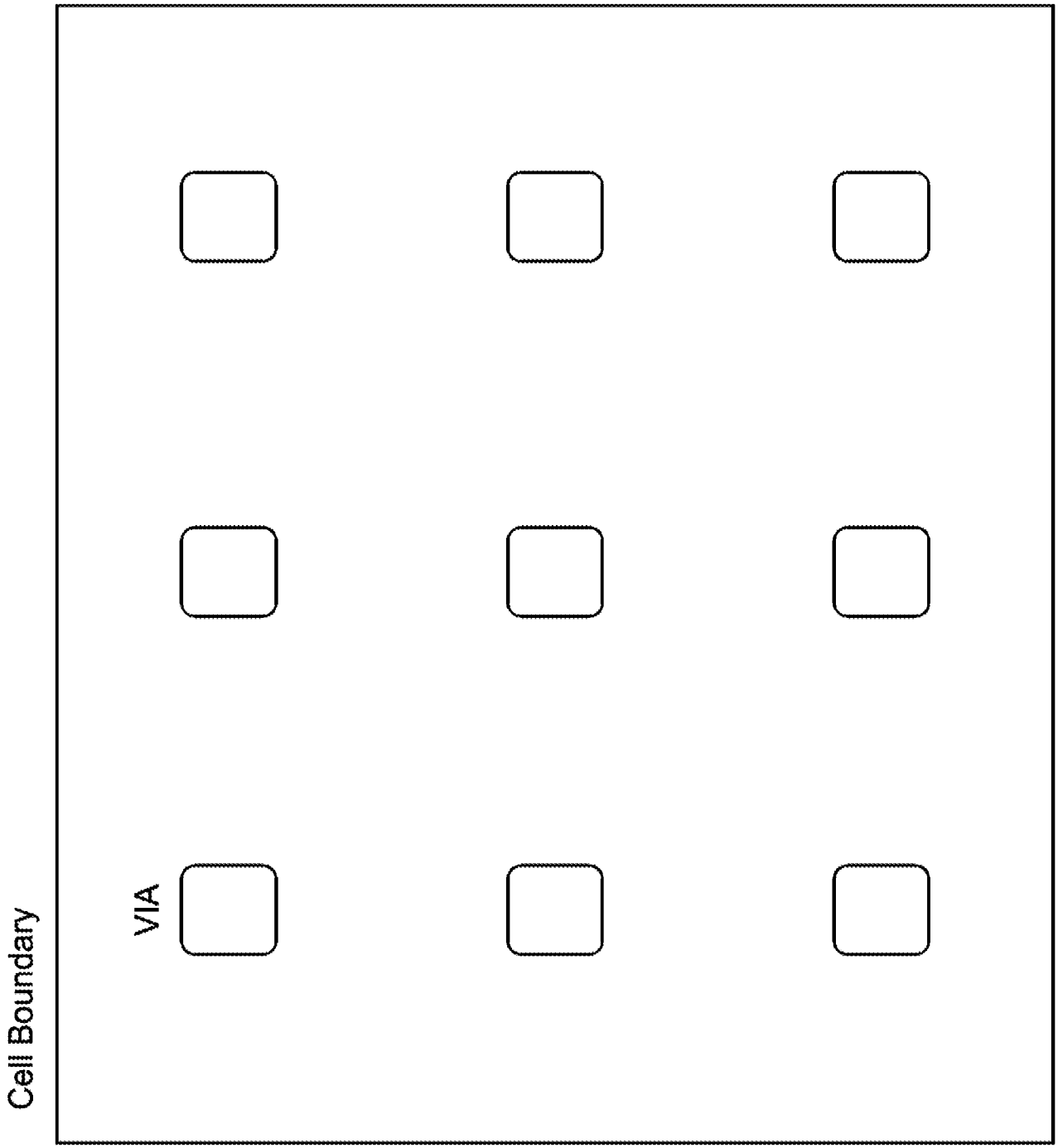


Figure 9

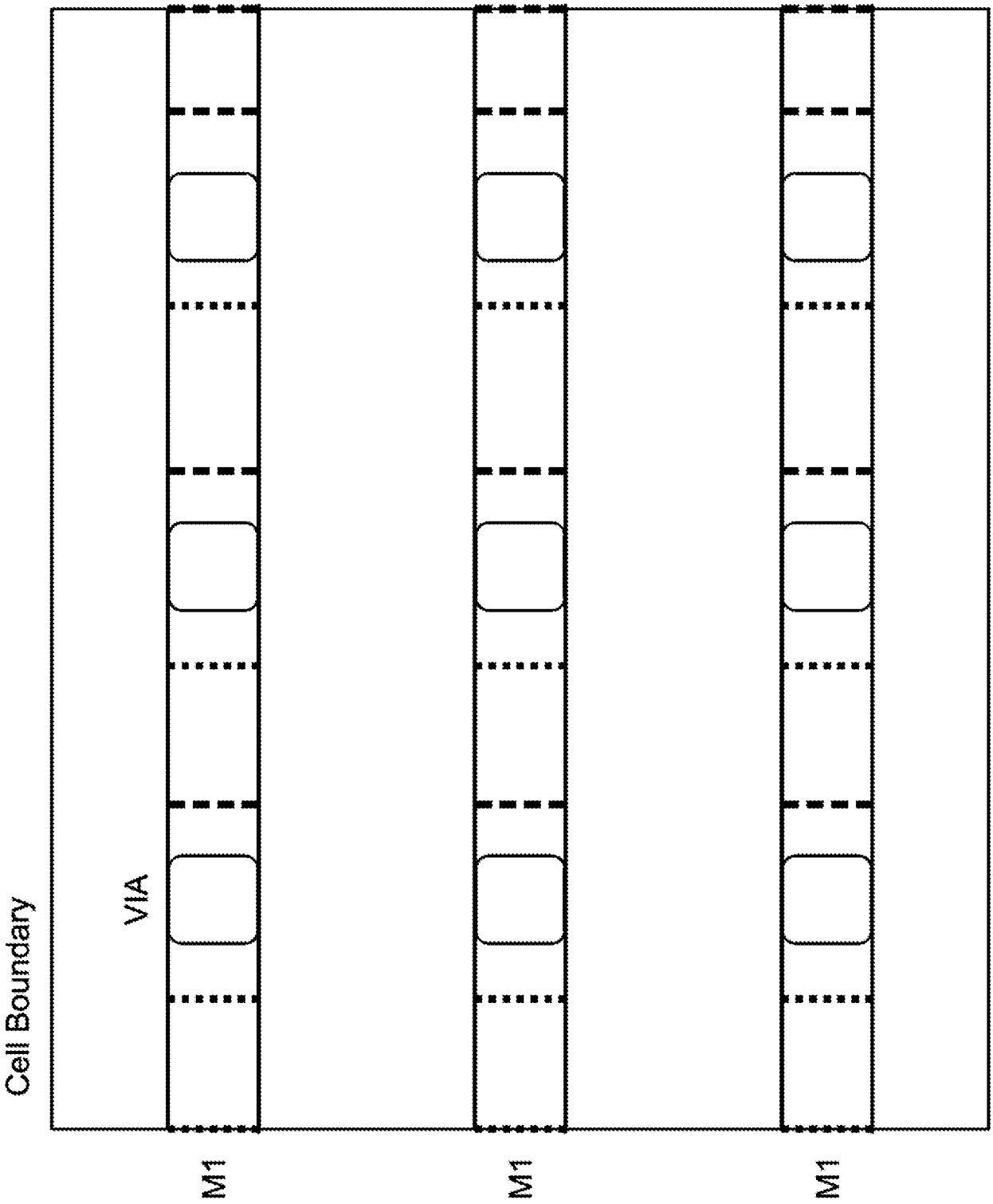


Figure 10

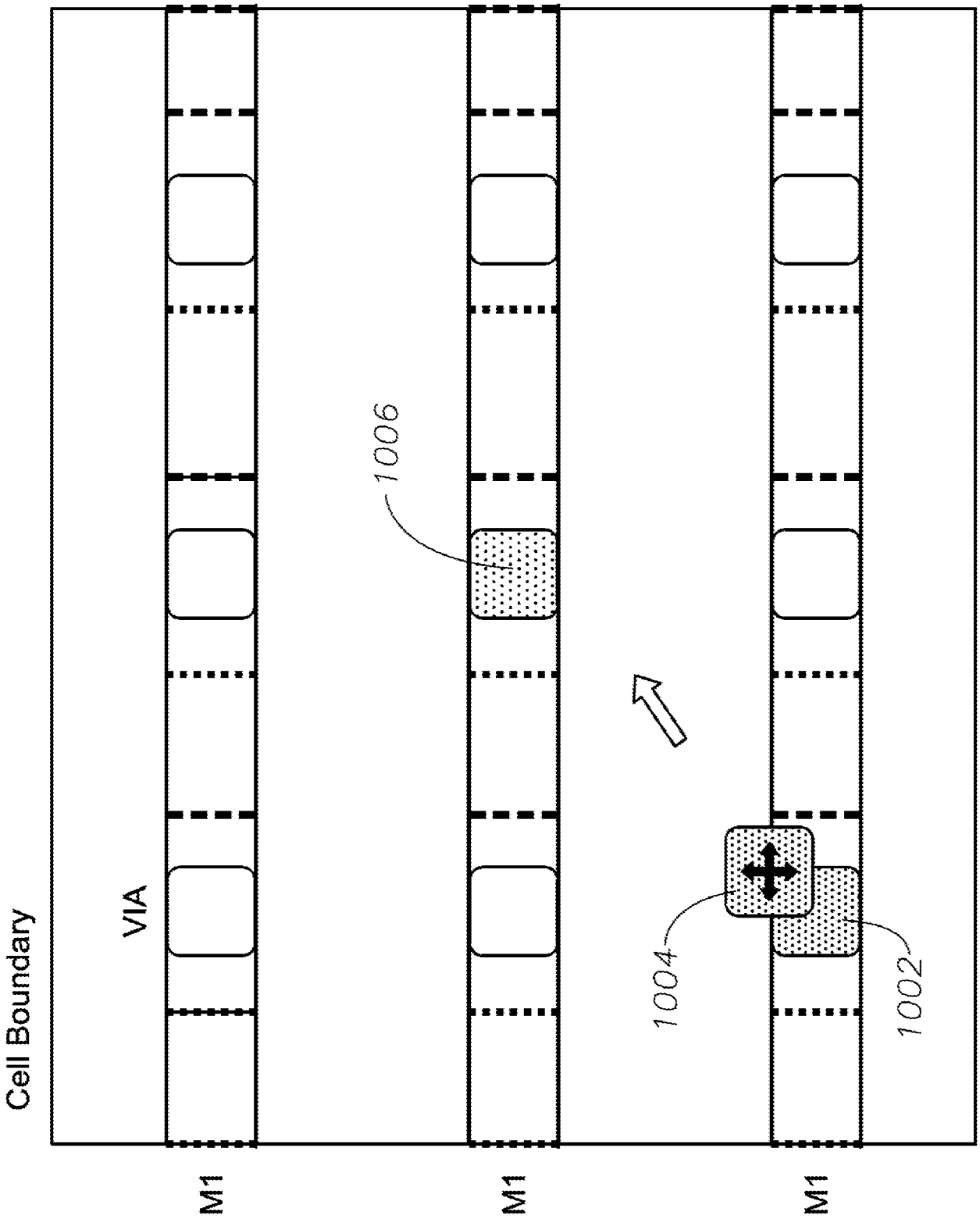


Figure 11

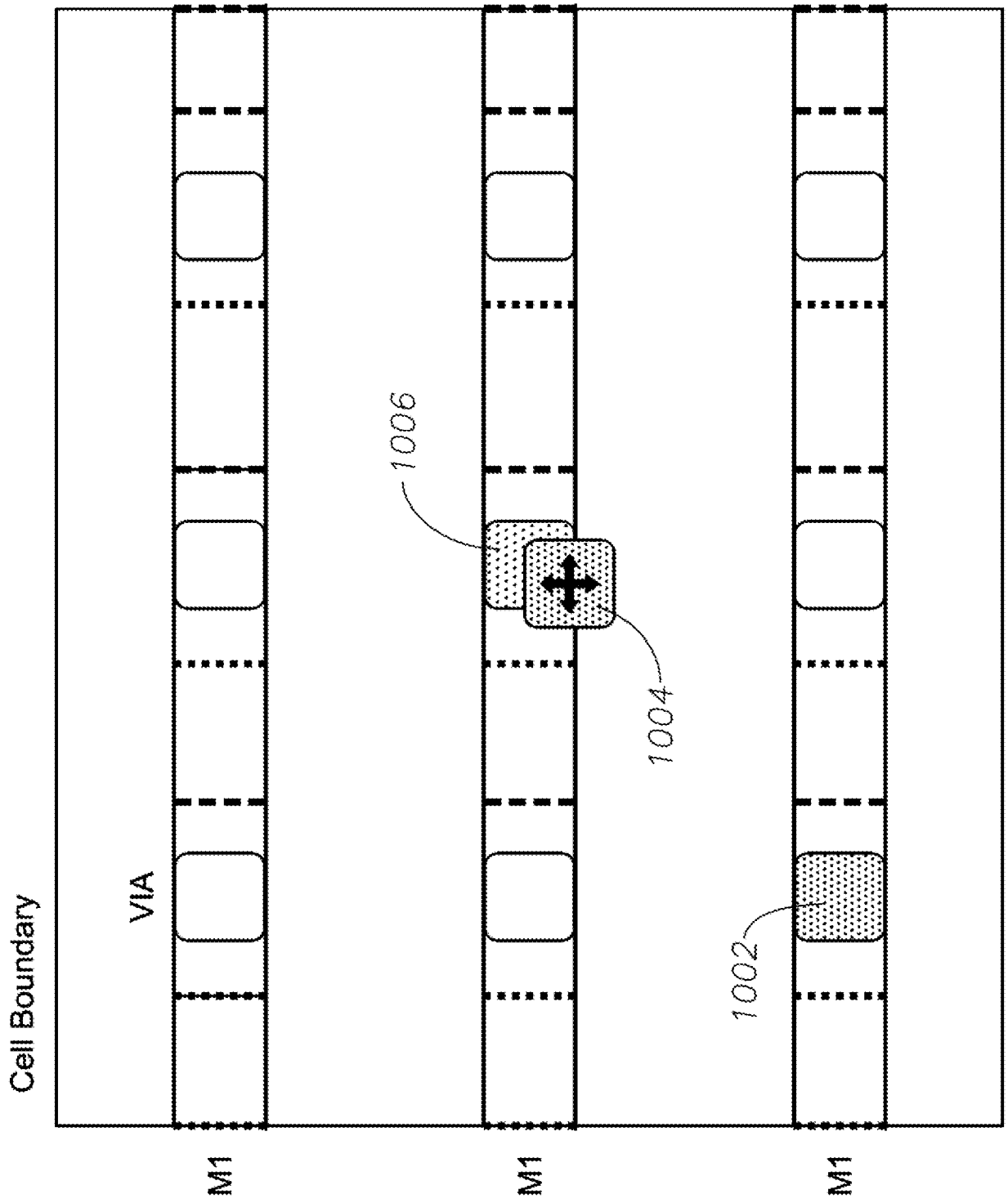


Figure 12

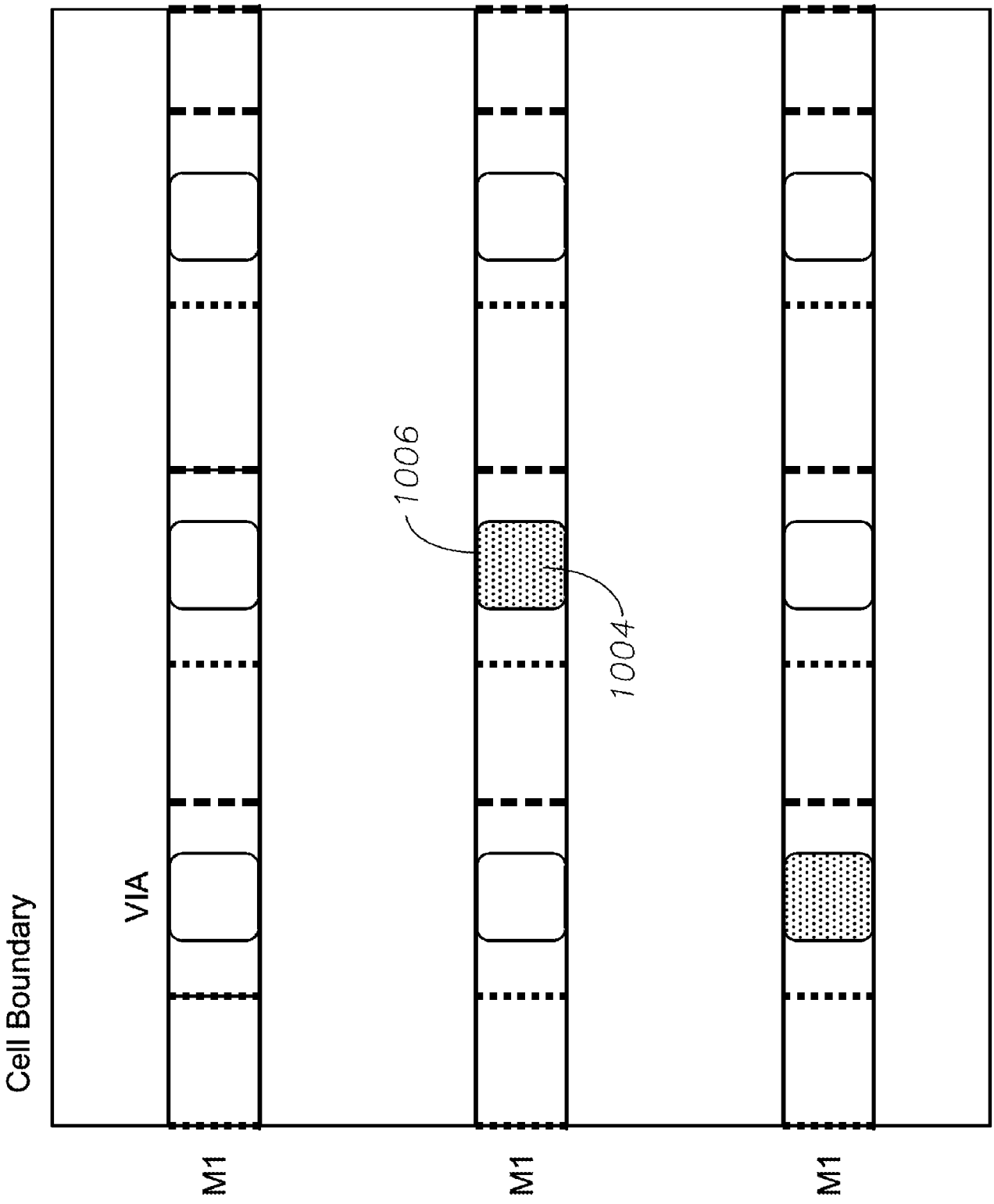


Figure 13

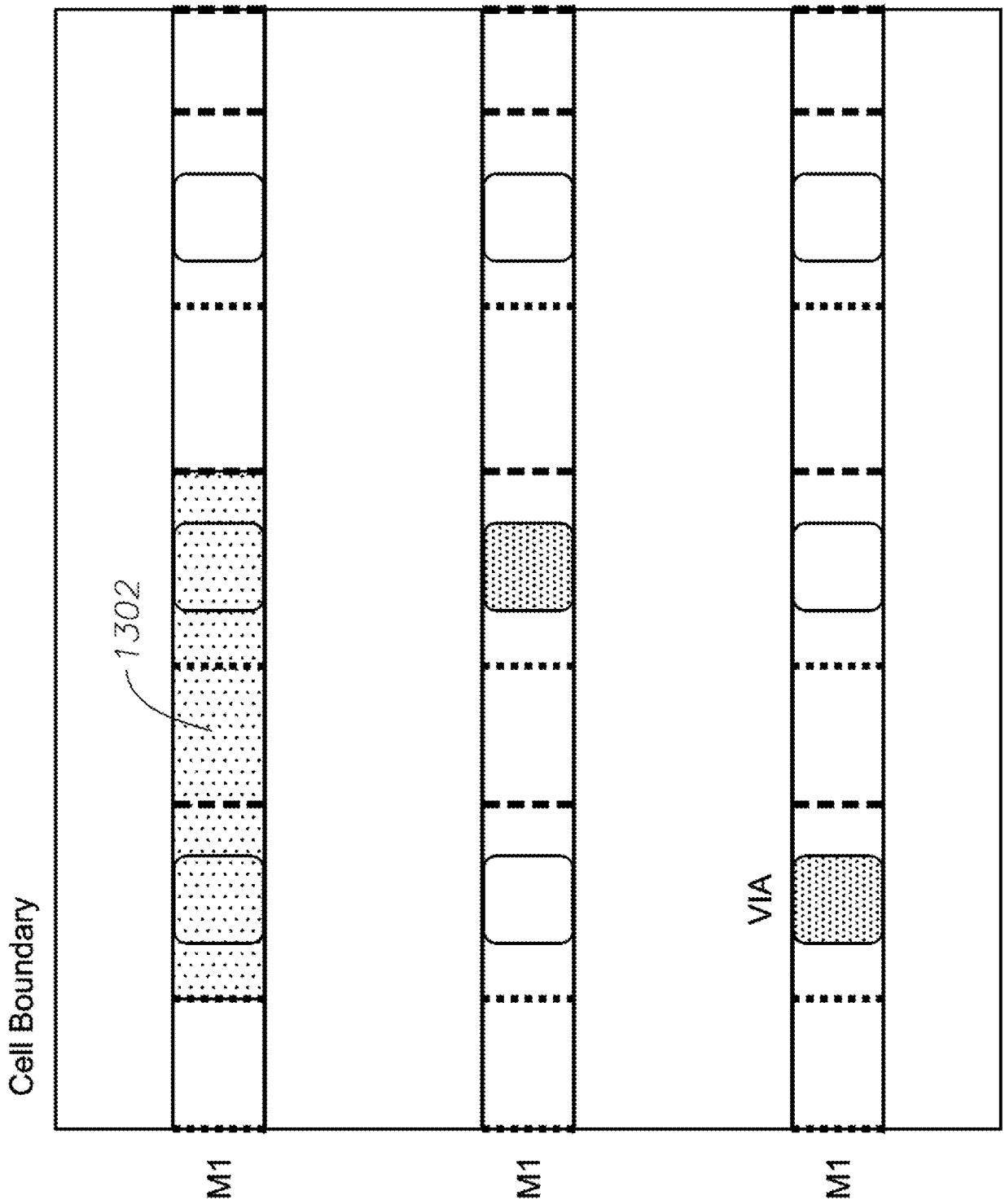


Figure 14

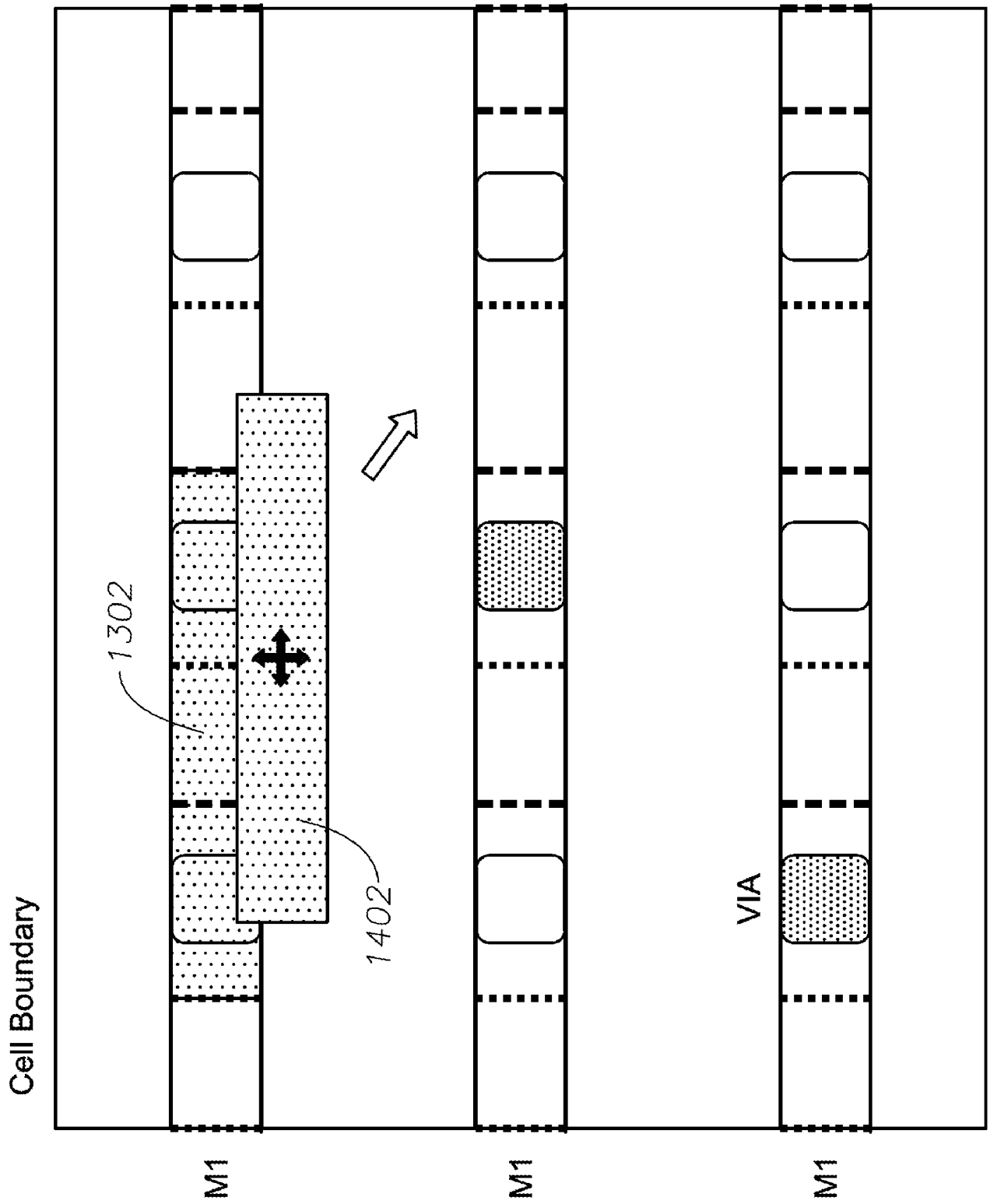


Figure 15

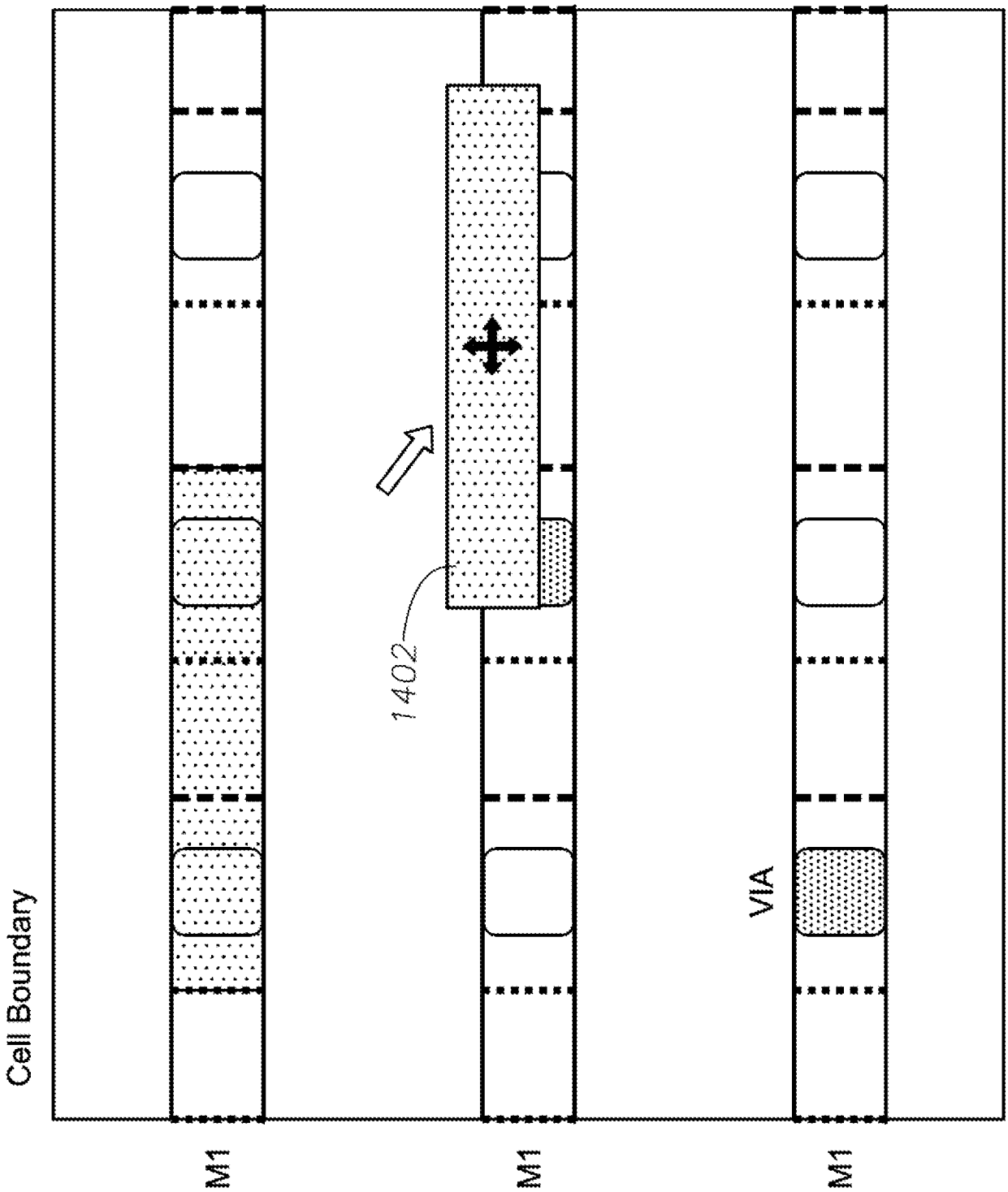


Figure 16

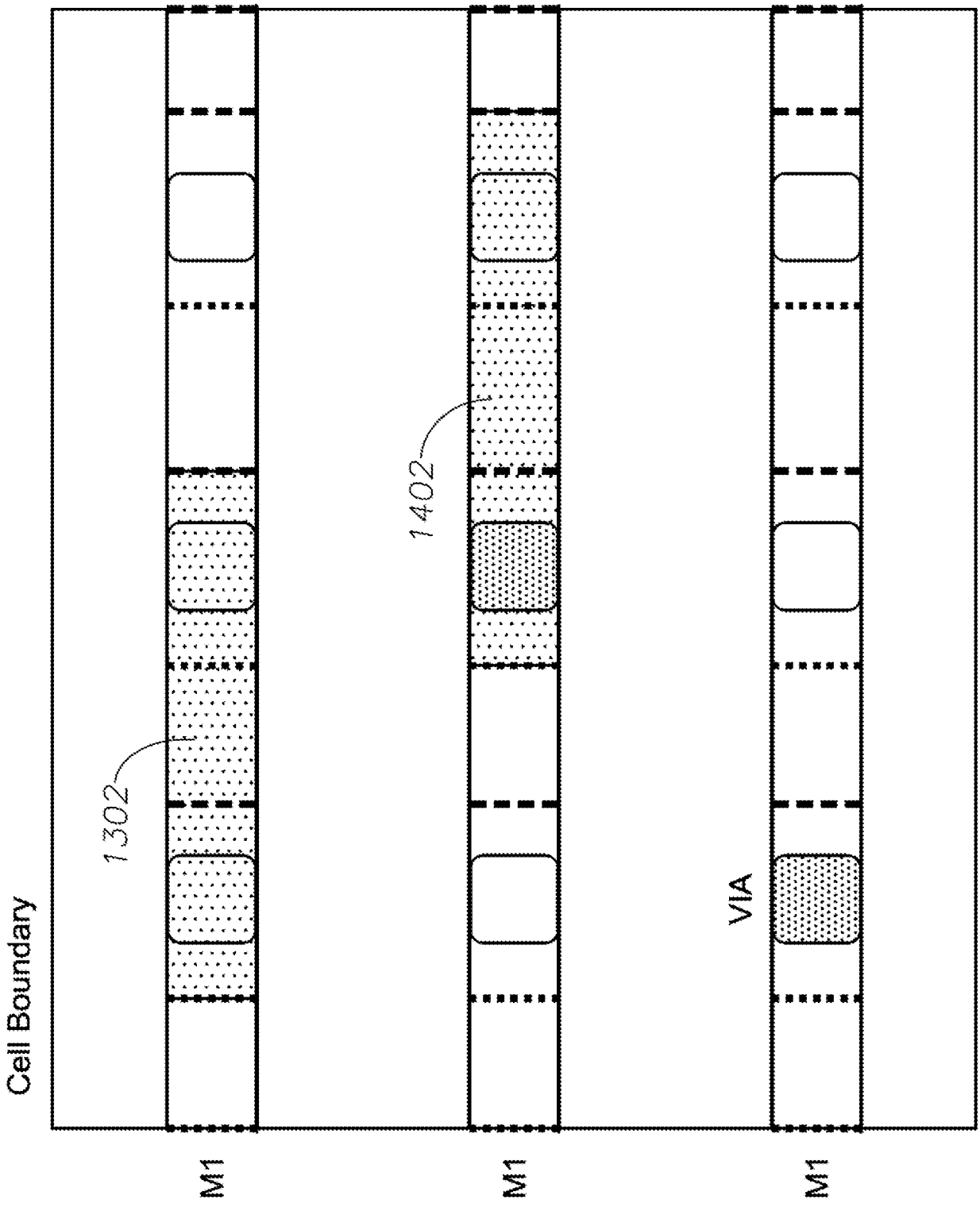


Figure 17

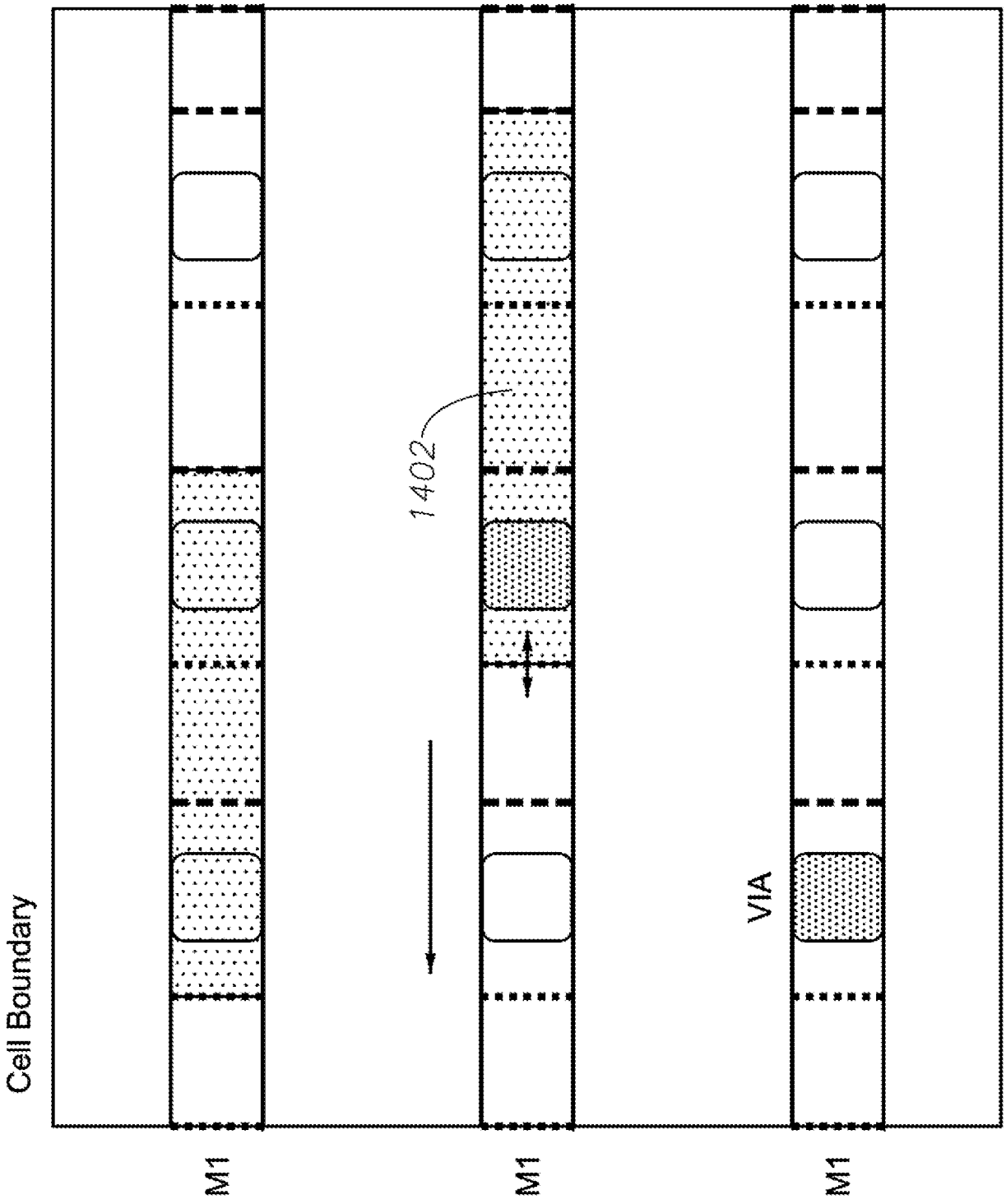


Figure 18

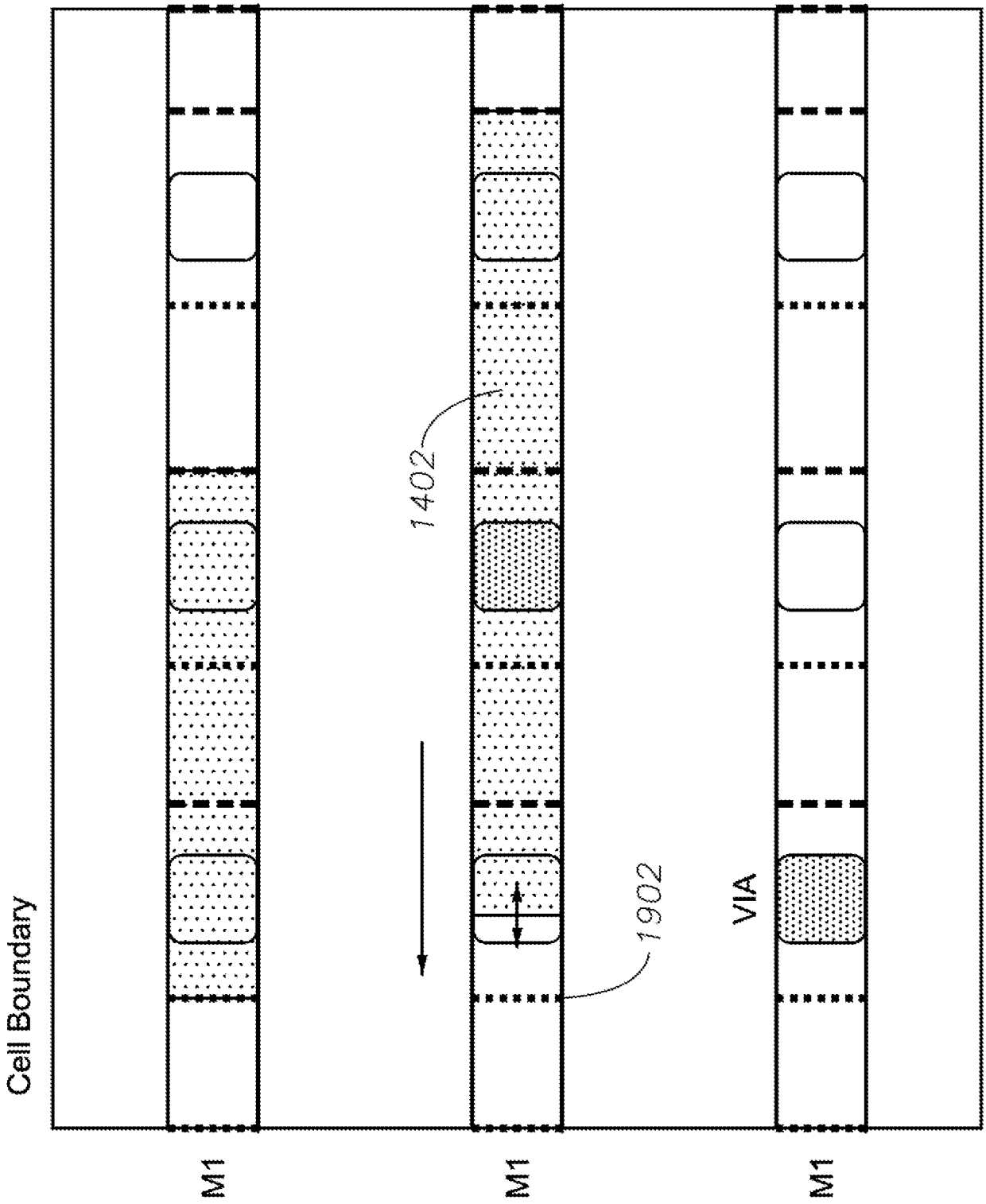
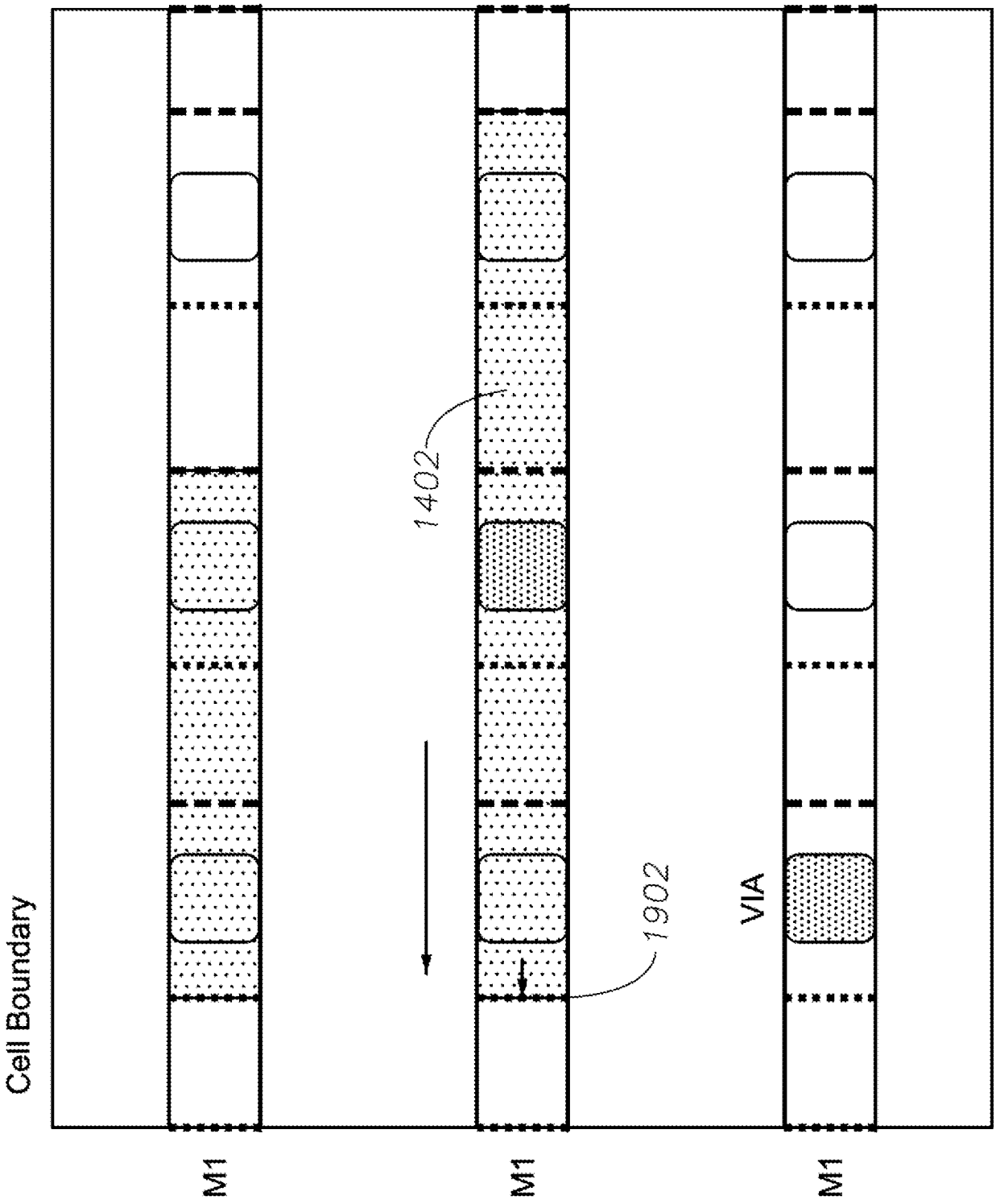


Figure 19



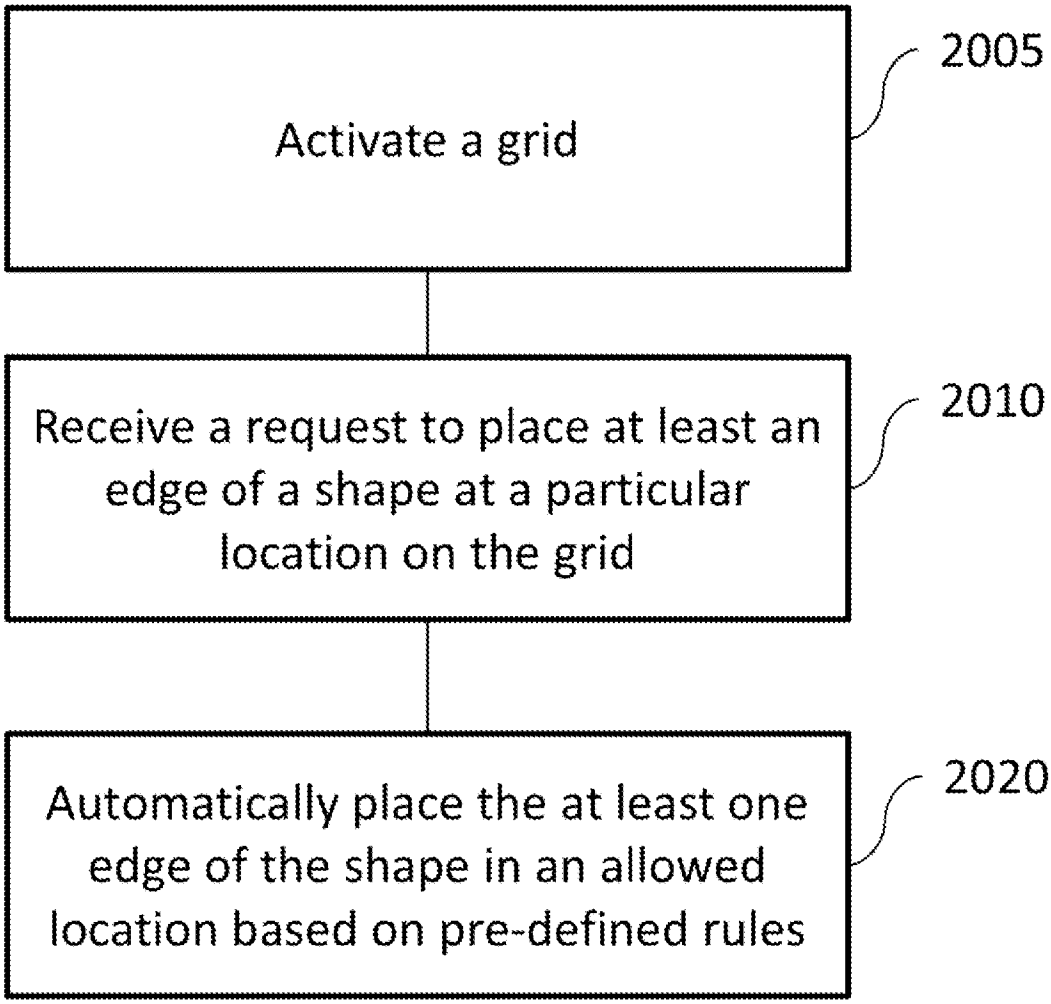

Activate a grid
2005
Receive a request to place at least an edge of a shape at a particular location on the grid
2010
Automatically place the at least one edge of the shape in an allowed location based on pre-defined rules
2020

Figure 20

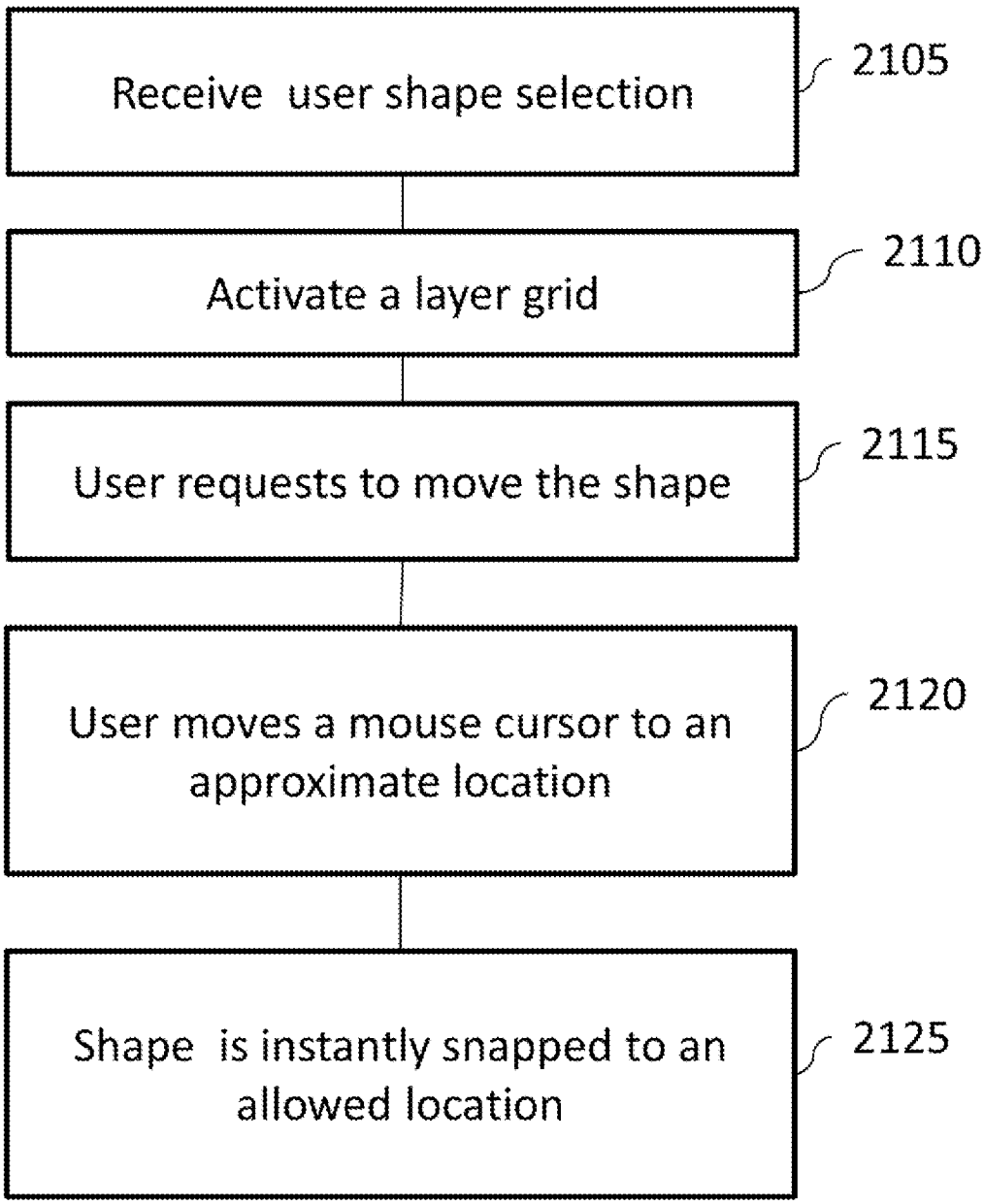

Receive user shape selection
2105
Activate a layer grid
2110
User requests to move the shape
2115
User moves a mouse cursor to an approximate location
2120
Shape is instantly snapped to an allowed location
2125

Figure 21

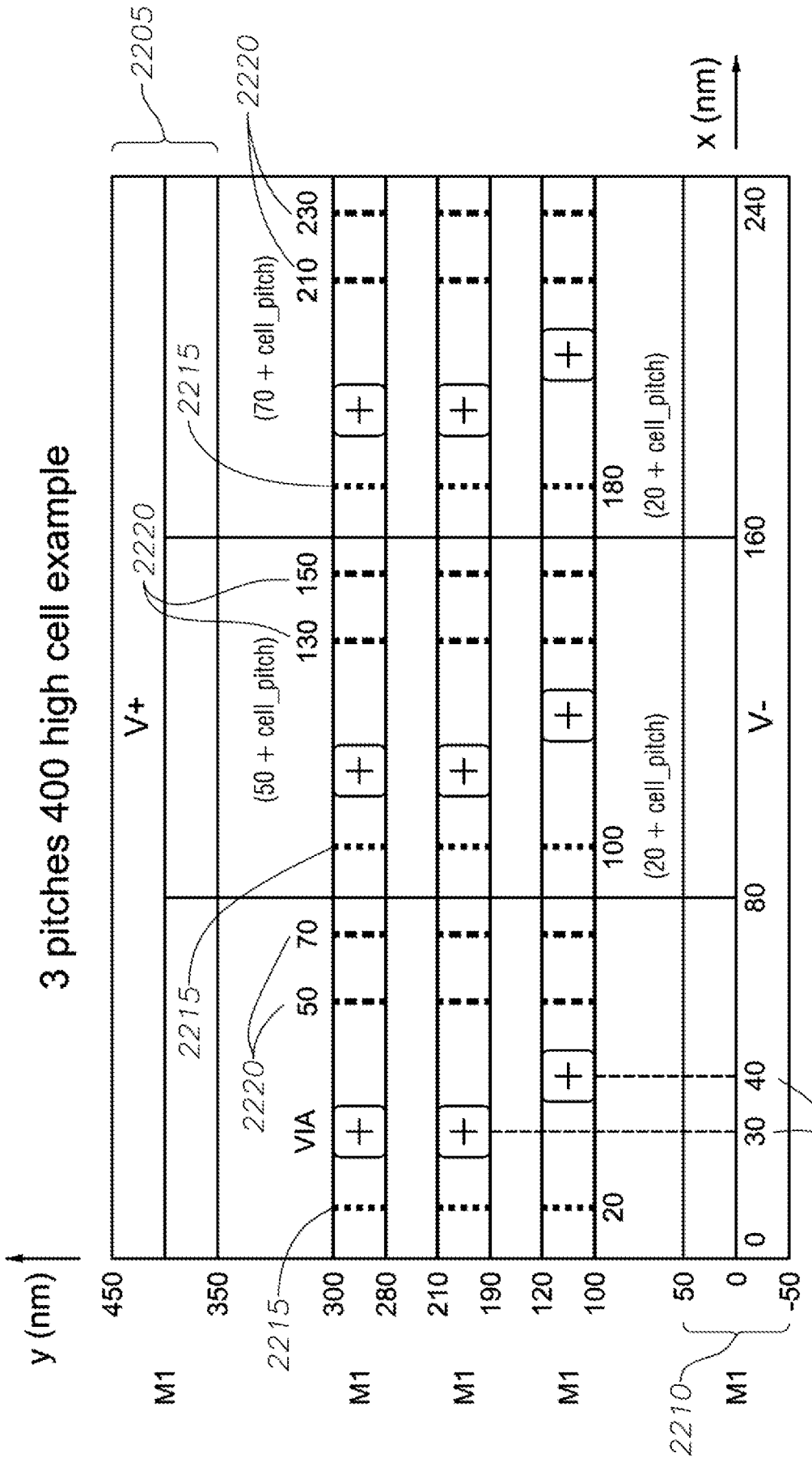

3 pitches 400 high cell example
y (nm)
x (nm)
2205
2220
2215
2210
2225
V+
V-
VIA
M1
(50 + cell_pitch)
(70 + cell_pitch)
(20 + cell_pitch)
450
350
300
280
210
190
120
100
50
0
-50
0
20
30
40
50
70
80
100
130
150
160
180
210
230
240

Figure 22

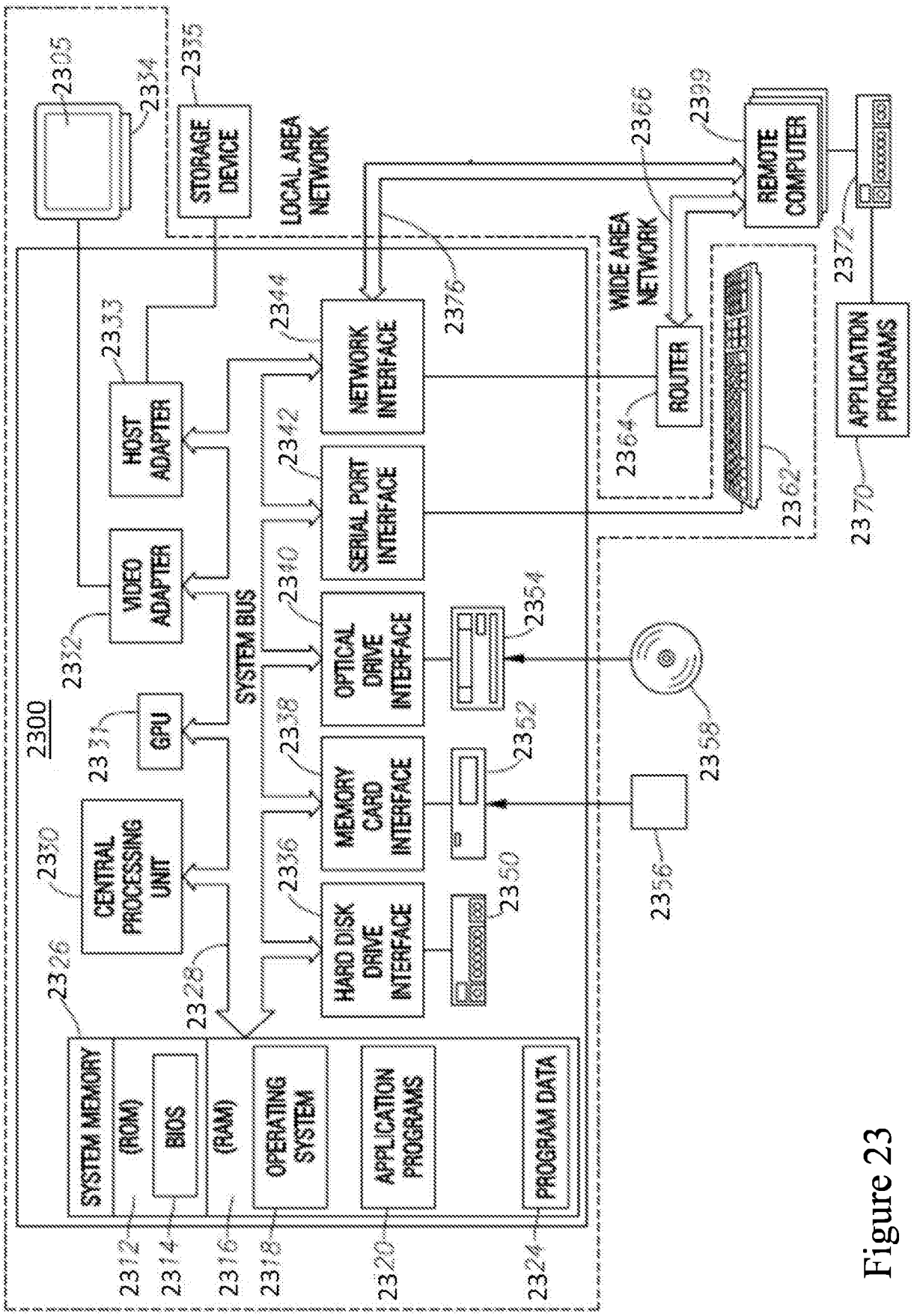

2300
2305
2334
2335
STORAGE DEVICE
LOCAL AREA NETWORK
2366
2369
REMOTE COMPUTER
WIDE AREA NETWORK
2372
APPLICATION PROGRAMS
2370
2376
2333
HOST ADAPTER
2344
NETWORK INTERFACE
2364
ROUTER
2362
2342
SERIAL PORT INTERFACE
2340
OPTICAL DRIVE INTERFACE
2354
2332
VIDEO ADAPTER
2331
GPU
SYSTEM BUS
2338
MEMORY CARD INTERFACE
2352
2358
2336
HARD DISK DRIVE INTERFACE
2350
2356
2330
CENTRAL PROCESSING UNIT
2328
2326
SYSTEM MEMORY
(ROM)
BIOS
(RAM)
OPERATING SYSTEM
APPLICATION PROGRAMS
PROGRAM DATA
2312
2314
2316
2318
2320
2324

Figure 23

INTERNATIONAL SEARCH REPORT

International application No
PCT/GB2016/053006

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/50 G06F3/0486 H01L27/02 G06F3/0481
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F H01L

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2009/271753 A1 (QUANDT JONATHAN R [US] ET AL) 29 October 2009 (2009-10-29) paragraph [0003] paragraph [0006] paragraph [0027] paragraph [0032] paragraph [0078] -----	1-20
X	US 9 053 289 B1 (LAMANT GILLES S C [US] ET AL) 9 June 2015 (2015-06-09) columns 4-7 -----	1-20
X	US 2005/138595 A1 (KHAKZADI KHOSRO [US] ET AL) 23 June 2005 (2005-06-23) paragraphs [0072] - [0075] paragraph [0080] -----	1-20

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
1 December 2016	12/12/2016

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Kamps, Stefan

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/GB2016/053006

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2009271753 A1	29-10-2009	EP 2321748 A2	18-05-2011
		JP 5599395 B2	01-10-2014
		JP 5923135 B2	24-05-2016
		JP 2012505526 A	01-03-2012
		JP 2014199934 A	23-10-2014
		JP 2016136650 A	28-07-2016
		KR 20110050638 A	16-05-2011
		KR 20160031041 A	21-03-2016
		KR 20160093095 A	05-08-2016
		MY 152456 A	30-09-2014
		SG 192532 A1	30-08-2013
		TW 201020836 A	01-06-2010
		US 2009271753 A1	29-10-2009
		US 2012273841 A1	01-11-2012
		US 2014035152 A1	06-02-2014
		US 2015143321 A1	21-05-2015
		WO 2010008948 A2	21-01-2010
US 9053289 B1	09-06-2015	NONE	
US 2005138595 A1	23-06-2005	NONE	