(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 3 822 769 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
19.05.2021 Bulletin 2021/20

(51) Int Cl.:
G06F 8/34 *(2018.01)* ***G06F 9/451*** *(2018.01)*

(21) Application number: **20199535.4**

(22) Date of filing: **01.10.2020**

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME
Designated Validation States:
KH MA MD TN

(30) Priority: **15.11.2019 CN 201911118938**

(71) Applicant: **Delta Electronics, Inc.**
Taoyuan County 33341 (TW)

(72) Inventors:
- **CHANG, Yung-Liang**
  33341 Taoyuan City (TW)
- **CHENG, Mao-Hua**
  33341 Taoyuan City (TW)
- **LIU, Kuei-Fu**
  33341 Taoyuan City (TW)

(74) Representative: **2K Patentanwälte Blasberg Kewitz & Reichel Partnerschaft mbB**
Schumannstrasse 27
60325 Frankfurt am Main (DE)

(54) METHOD FOR EDITING CONTINUAL VERTICAL LINE OF VISUAL PROGRAMMING LANGUAGE

(57) A method for editing continual vertical line of visual programming language includes following steps: entering an editing mode; selecting a source node (41) as a current node from a serial-parallel graphic (1) where the source node (41) is an endpoint of one of a plurality of normal blocks (2) in the graphic; calculating and displaying one or more candidate nodes (42) around current node; connecting the current node to any one of the candidate nodes (42) for converting the connected candidate node (42) into a connected node (43); setting the connected node (43) as the current node for continually calculating, displaying, and connecting to one or more candidate nodes (42) before exiting the editing mode; generating a continual vertical line (3) according to the source node (41) and the one or more connected nodes (43) when exiting the editing mode; and, modifying the serial-parallel relationship among the plurality of normal blocks (2) in the graphic (1) and updating the graphic (1) according to the continual vertical line (3).

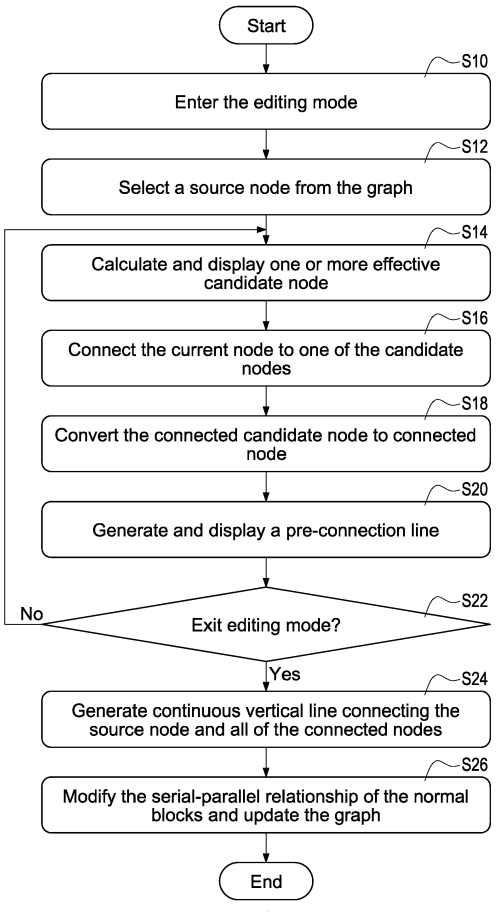


FIG.2

EP 3 822 769 A1

Description

BACKGROUND OF THE INVENTION

1. Technical field

[0001] The present invention relates to an editing method of visual programming language, especially to a method for editing continual vertical line of visual programming language.

2. Description of Prior Art

[0002] Generally, user may edit the logic connection relationship (such as serial or parallel relationship) among virtual industrial control equipment by visual programming language editor. Therefore, user can easily determine the combination of various industrial control equipment for automatic operation and set up desired system architecture.

[0003] The current visual programming language editor, such as ladder diagram editor, mainly assigns a user command to an operation for editing. User is allowed to add only one vertical line at one time when user wants to changes the connection relationship among the various device blocks (also referred to as normal block, and corresponding to various virtual equipment). Namely, user needs to send plurality of commands successively when he/she needs to construct a plurality of vertical lines. This is really troublesome.

[0004] In the operation of current editor of visual programming language, after all of the required normal blocks are added to the graph, user cannot construct a continual vertical line across multiple nodes by using only one operation to fast adjust the connection relationship of multiple normal blocks. Due to above limitation, it is hard for user to construct a complicated graph with lots of normal blocks and complex connection relationship.

SUMMARY OF THE INVENTION

[0005] One of objects of the present invention is to provide a method for editing continual vertical line of visual programming language to allow user to construct a continual vertical line across multiple nodes by using only one operation, thus fast adjust the connection relationship of multiple normal blocks and update the graph.

[0006] Accordingly, the present invention provides a method for editing continual vertical line of visual programming language and applied to an editor. The method comprise: entering an editing mode; obtaining a source node on the serial-parallel graph comprising a plurality of normal blocks and the source node being an endpoint of one of the normal blocks; calculating and displaying at least one candidate node around the current node wherein the at least one candidate node is an endpoint of at least one of the normal blocks around the current node; connecting the current node to one of the at least one candidate nodes; converting the connected candidate node into a connected node; determining whether the editor exits the editing mode; setting the connected node as the current node and continually calculating, displaying and connecting one or more candidate node before exiting the editing mode; generating a continuous vertical line connecting the source node and at least one of the connected nodes when exiting the editing mode; and modifying a serial-parallel relationship of the normal blocks based on the continuous vertical line and updating the serial-parallel graph.

[0007] By above editing method of the present invention, user only needs one-time operation to directly generate a continuous vertical line in graph, thus fast modify the connection relationship among normal blocks by using the continuous vertical line and update the graph at the same time. User may use the editor of visual programming language more efficiently.

BRIEF DESCRIPTION OF DRAWINGS

[0008]

Fig. 1A shows the first construction operation for the continual vertical line according to the first embodiment of the present invention.

Fig. 1B shows the second construction operation for the continual vertical line according to the first embodiment of the present invention.

Fig. 1C shows the third construction operation for the continual vertical line according to the first embodiment of the present invention.

Fig. 2 shows the editing flowchart according to the first embodiment of the present invention.

Fig. 3A shows the first construction operation for the continual vertical line according to the second embodiment of the present invention.

Fig. 3B shows the second construction operation for the continual vertical line according to the second embodiment of the present invention.

Fig. 3C shows the third construction operation for the continual vertical line according to the second embodiment of the present invention.

Fig. 3D shows the fourth construction operation for the continual vertical line according to the second embodiment of the present invention.

Fig. 3E shows the fifth construction operation for the continual vertical line according to the second embodiment of the present invention.

Fig. 3F shows the sixth construction operation for the continual vertical line according to the second embodiment of the present invention.

Fig. 3G shows the seventh construction operation for the continual vertical line according to the second embodiment of the present invention.

Fig. 4 is the schematic view showing the retreating of the vertical line according to the first embodiment of the present invention.

Fig. 5A is the first part of the editing flowchart according to the second embodiment of the present invention.

Fig. 5B is the second part of the editing flowchart according to the second embodiment of the present invention.

Fig. 5C is the third part of the editing flowchart according to the second embodiment of the present invention.

Fig. 6A shows the first construction operation for the continual vertical line according to the third embodiment of the present invention.

Fig. 6B shows the second construction operation for the continual vertical line according to the third embodiment of the present invention.

Fig. 6C shows the third construction operation for the continual vertical line according to the third embodiment of the present invention.

Fig. 6D shows the fourth construction operation for the continual vertical line according to the third embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0009] Reference will now be made to the drawing figures to describe the present disclosure in detail. It will be understood that the drawing figures and exemplified example of present disclosure are not limited to the details thereof.

[0010] The present invention provides a method for editing continual vertical line of visual programming language (hereinafter briefed as editing method), and the editing method is applied to editor of all kinds of visual programming languages and the editor is run on a computer.

[0011] Refer now to Figs. 1A, 1B, and 1C, which respectively show the first construction operation, the second construction operation and the third construction operation of the first embodiment of the present invention.

[0012] As shown in Fig. 1A, after user operates computer and runs the visual programming language editor on the computer, user can operate the editor and add one or more normal block 2 into the editor with the editing tools (not shown) of the editor, and construct a serial-parallel graphic (hereafter briefed as graph 1) with the normal blocks 2.

[0013] More specifically, after multiple normal blocks 2 are added to the editor, user may construct a continue block 21 with serially-connected blocks and construct a branch block 22 with the parallel-connected continue blocks 21.

[0014] As shown in Fig. 1B, according to one aspect of the present invention, user may operate human machine interface such as mouse, keyboard or touch pad of the computer to operate the editor to connect the multiple nodes 23 in graph 1, where each of the nodes 23 is corresponding to an endpoint of each normal block 2. It should be noted that, according to the present invention, user may complete the connection of the multiple nodes 23 in graph 1 by one operation (for example user just operates the mouse, keyboard or touch pad of the computer for one time).

[0015] As shown in Fig. 1C, after user completes the connection of the multiple nodes 23 in graph 1 by one operation, the editor may generate a continual vertical line 3 according to the multiple connected nodes 23 and modify the seral-parallel relationship of the multiple normal blocks 2 in the graph 1 based on the continual vertical line 3. The editor further updates the graph 1 and displays the updated graph 1.

[0016] In the present invention, user only needs one operation to construct a continuous vertical line 3 across multiple nodes 23 to complete the work, where prior art editor needs successive and multiple commands.

[0017] More particularly, as shown in Fig. 1C, after the continuous vertical line 3 is generated by the editor according to user operation, the editor may re-organize the graph 1. Therefore the normal blocks 2 on the left side of the continuous vertical line 3 are re-organized as a new branch block 22, the normal blocks 2 on the right side of the continuous vertical line 3 are also re-organized as a new branch block 22, and the editor parallel connects the two new branch blocks 22. Finally, the graph 1 is updated according to the new seral-parallel relationship of the updated normal blocks 2.

[0018] As shown in Figs. 1B and 1C, the continuous vertical line 3 comprises multiple vertical line segments. However, in the present invention, user only needs one operation to directly generate the continuous vertical line 3 instead of sending successive commands (multiple operations), thus greatly enhance the convenience of the editor.

[0019] It should be noted that even the present invention is exemplified with ladder diagram editor in Figs. 1A to 1C, however, the scope of the present invention is not limited by this specific example.

[0020] Fig. 2 shows the editing flowchart according to the first embodiment of the present invention. Fig. 2 discloses the specific editing steps of the editing method of the present invention. After the visual programming language editor executes the steps shown in Fig. 2, the editor may modify the serial-parallel relationship of the normal blocks 2 of the graph 1 similar to the examples shown in Figs. 1A to 1C and updates the graph 1 as well as redisplays the updated graph 1.

[0021] At first, the editor receives external operation from user (such as the first external operation) to enter the editing mode (step S10). In this embodiment, the user may operate the editor to add normal blocks 2 of desired number and desired types in graph 1 and then operate the editor to enter the editing mode to generate the continuous vertical line and to modify the serial-parallel relationship of the normal blocks 2 of the graph 1 according to the continuous vertical line 3 as well as to update the graph 1.

[0022] After the editor enters the editing mode, the user

may operate the computer (for example operating the mouse, the keyboard or the touch pad) to select a source node (or referred to as start node) from the graph, where the source node is the endpoint of any normal block 2 in the graph 1 (S12). After the step S12, the editor sets the user-selected source node as the current node and generates the continuous vertical line 3 from the current node. In this embodiment, the current node is the current position of the cursor (such as mouse pointer) hovering in the screen of the editor, where the cursor is controlled by mouse, keyboard or touch pad.

[0023] It should be noted that the editor may add the current position of the source node to a queue (not shown) after the source node is confirmed, thus the position of the source node of the continuous vertical line 3 can be recorded, which will be detailed later.

[0024] After step S12, the editor may calculate one or more effective candidate node around the current node and mark the candidate node on the graph 1 with specific symbol (step S14). In this embodiment, the candidate node is the end point of one or more normal block 2 around the current node. More particularly, the effective candidate node in this embodiment is the node that will not cause short circuit of any normal block when user generates a connection line connecting the current node and the effective candidate node.

[0025] For example, if ten nodes are around the current node and there are three nodes causing short circuit of any normal block 2 after they are connected with the current node, the editor will mark the seven nodes, which will not cause short circuit of any normal block, on the graph as the candidate nodes.

[0026] It should be noted that the editor may monitor the operation of user (such as monitor the current location of the cursor), then calculates and displays one or more candidate nodes around the current node only when the cursor moves from the current node and the moving distance of the cursor is larger than a threshold value, thus prevent the erroneous click (operation) of user.

[0027] Besides, when the editor monitors user operation, the editor further determines the moving direction of the cursor moving from the current node. In this embodiment, the editor only calculates and displays one or more candidate nodes above the current node when the moving direction is upward, and only calculates and displays one or more candidate nodes below the current node when the moving direction is downward. Therefore, the editor can prevent from displaying excessive candidate nodes, which will confuse the user.

[0028] After the step S 14, the editor already displays one or more effective candidate nodes, which is connectable, on the graph 1 and the editor may receive again the external operation of user (such as the second external operation) to connect the current node to one of the displayed candidate nodes (step S16).

[0029] For example, if the computer equipment currently operated by user is computer mouse, then the user may drag the mouse to move the cursor to the source node (namely the current node), press the left button of the computer mouse, and drag the cursor to the location of any one candidate node from the source node, thus connect the source node (namely the current node) to the candidate node.

[0030] For another example, if the computer equipment currently operated by user is keyboard, then the user may move the cursor to the location of the current node by arrow key and then press special function key (or shortcut key) to lock the source node. Afterward, the user may move the cursor to the location of any one candidate node from the source node by operating the arrow key, thus connect the source node to the candidate node.

[0031] However, the above description is only for explaining the embodiments of the present invention and is not limitation for the scope of the present invention.

[0032] After user connects the current node to any of the candidate nodes by operating the mouse, keyboard or touch pad, the editor converts the connected candidate node to connected node (step S16).

[0033] It should be noted that the editor adds the location of the connected node to the queue when any candidate node is converted to connected node. Therefore, the location of the nodes passed by the continuous vertical line 3 and/or the location of the end node can be recorded.

[0034] In one embodiment, the editor uses dashed line to display one or more effective candidate node and uses solid line to display the connected node. In another embodiment, the editor uses a first color (such as green) to display one or more effective candidate node and uses a second color (such as blue) to display the connected node. By above display manner to distinguish between effective candidate node and the connected node, user can easily generate the continuous vertical line 3 by using the editor.

[0035] It should be noted that the editor may selectively generate and display a pre-connection line (such as the pre-connection line 5 shown in Figs. 3C and 3D) connecting the current node and the connected node in step S20 such that the user may easily visualize the construction of the continuous vertical line 3.

[0036] During the construction of the continuous vertical line, the editor continually determines whether it receives external operation (such as the third external operation) of user command to exit the editing mode (step S22). The editor sets the last connected node as the current mode before receiving the third external operation and then repeats the execution of steps S14 to S20 to continually perform the construction of the continuous vertical line 3.

[0037] In one embodiment, user may operate mouse to perform the construction of the continuous vertical line 3. In this embodiment, the first external operation keeps pressing the button of the computer mouse (namely, the editor enters the editing mode when user keeps pressing mouse button); the second external operation drags the mouse pointer by moving the mouse with mouse button

being kept pressing (namely, the user drags the mouse to connect two nodes); the third external operation releases mouse button (namely, the editor exits editing mode when user release mouse button).

[0038] In another embodiment, the user operates the keyboard to perform the construction of the continuous vertical line 3. In this embodiment, the first external operation presses a special function key or shortcut key (namely, the editor enters the editing mode when user presses the special function key); the second external operation presses arrow keys (namely, the user moves cursor with arrow key to connect two nodes); the third external operation presses again the special function key (namely, the editor exits editing mode when user presses again the special function key).

[0039] The above examples are used for demonstrating the present invention and are not limitation of the present invention. For example, the editor may enter editing mode when user keeps pressing a specific key (such as ALT key), and exits the editing mode when the specific key is released. For another example, the editor may enter editing mode when user presses a specific button of mouse (such as left button), and exit the editing mode when the specific button is pressed again.

[0040] In other embodiment, the user may also operate the touch screen or touch pad of computer to perform the construction of the continuous vertical line 3. In this embodiment, the first external operation, the second external operation, and the third external operation may have similar operations with the previous examples, and the detailed description is omitted here for brevity.

[0041] In step S22, if the editor receives the third external operation to exit the editing mode, the editor generates the continuous vertical line 3 connecting the source node and all of the connected nodes in real time (step S24), modifies the serial-parallel relationship of the normal blocks 2 on the graph 1 at the same time, and then updates and displays the graph 1 (step S26).

[0042] It should be noted that in steps S24 to S26, the editor mainly obtains the source node and one or more connected node stored in the queue, and generates the continuous vertical line 3 based on those nodes. Besides, the editor further modifies the serial-parallel relationship of the normal blocks 2 based on the nodes stored in the queue and then updates the graph 1, which will be described in more detail later.

[0043] Refer also to Figs. 3A to 3G, which show the first construction operation to the seven construction operation for the continuous vertical line according to the second embodiment. In the embodiment shown in Figs. 3A to 3G, user uses single operation of mouse to generate a continuous vertical line, but this is not limitation of the present invention.

[0044] First as shown in Fig. 3A, user may operate the editing tool (not shown) of the editor to add a plurality of normal blocks 2 of desired number and type into the graph 1. In the embodiment shown in Fig. 3A, initially the graph 1 includes a first continue group (serial group) having normal blocks B0, B1, B2 and B3; a second continue group (serial group) having normal blocks B4, B5, B6 and B7; a third continue group (serial group) having normal blocks B8, B9, B10 and B11; and a fourth continue group (serial group) having normal blocks B12, B13, B14 and B15. Besides, the first continue group, the second continue group, the third continue group and the fourth continue group are parallel connected to each other.

[0045] As shown in Fig. 3A, each normal block 2 has endpoints LO-L11 at the serial junction. In the present invention, user may select the source node of the continuous vertical line 3 from the endpoints L0~L11, and the editor calculates one or more candidate node satisfying certain condition from the endpoints L0~L11. In other word, the editing method of the present invention can prevent the circuit error caused by user erroneously setting the continuous vertical line 3 on the normal block 2.

[0046] Afterward, as shown in Fig. 3B, user may operate mouse to control cursor in the screen to place the pointer at any endpoint, and then press mouse button (such as left button) to convert this endpoint to the source node 41. In this embodiment, user selects the endpoint L0 as the source node 41 and the editor stores the endpoint L0 in the queue Q.

[0047] After user selects a source node 41 the user keeps pressing mouse button to drag the pointer such that the pointer moves downward and the moving distance is larger than a threshold. At this time, the editor is triggered to calculate one or more effective candidate nodes 42 below the source node 41 and to display the one or more effective candidate nodes 42.

[0048] In one embodiment, the editor uses solid line to display the source node 41 and uses dashed line to display the candidate nodes 42. In another embodiment, the editor uses different colors to display the source node 41 and the candidate nodes 42 such that user may quickly distinguish between the two kinds of nodes.

[0049] Afterward, as shown in Fig. 3C, user keeps dragging the pointer to move the cursor to one candidate node 42 (such as the endpoint L4 shown in Fig. 3C), then the editor adds the candidate node 42 (namely the endpoint L4) in queue Q and converts the candidate node 42 to connected node 43. When the pointer is further moved downward and the moving distance is larger than the threshold, the editor is triggered to calculate and display one or more effective candidate node 42 below the connected node 43.

[0050] In one embodiment, the editor displays the connected node 43 with solid line. In another embodiment, the editor uses different colors to display the candidate node 42 and the connected node 43, for example green for the candidate node 42 and blue for the connected node 43. In still another embodiment, the editor uses the same color or shape to display the candidate node 42 and the connected node 43.

[0051] Afterward, as shown in Fig. 3D, the user may keep dragging the pointer to successively connect a plurality of candidate nodes 42 to convert those candidate

nodes 42 into connected nodes.

[0052] Furthermore, as shown in Figs. 3C and 3D, when user drags the pointer to connect two nodes, the editor displays a pre-connection line 5 for the two nodes on the graph 1 such that the user can quickly visualize the current connection status.

[0053] In the embodiment shown in Fig. 3D, user drags the pointer from the endpoint L0 and then passes the endpoint L4, the endpoint L6 and the endpoint L11. Therefore, the editor will set the endpoint L0 as the source node and then sequentially convert the endpoint L4, the endpoint L6 and the endpoint L11 into connected nodes 43. The editor also adds the endpoint L0, the endpoint L4, the endpoint L6 and the endpoint L11 into queue Q. When user releases mouse button to exit the editor from the editing mode, the editor may generate the continuous vertical line 3 based on the record of the endpoints in the queue Q.

[0054] As shown in Fig. 3E, when user releases mouse button, the editor determines that user ends the editing task (namely exit editing mode), then the editor fetches the first two endpoints of the queue Q, namely, the endpoints L0 and L4 and removes the first endpoint of the queue Q, namely the endpoint L0.

[0055] Afterward, the editor constructs a first cut line 31 based on the endpoints L0 and L4, sets one or more normal block 2 on the left side of the start point (namely the endpoint L0) of the first cut line31 as the first left group and sets one or more normal block 2 on the right side of the starting point of the first cut line 31 as the first right group. The editor further sets one or more normal block 2 on the left side of the end point (namely the endpoint L4) of the first cut line 31 as the second left group and sets one or more normal block 2 on the right side of the start point of the first cut line 31 as the second right group. Furthermore, the editor parallel connects the first left group with the second left group, parallel connects the first right group with the second right group, and finally serial connects the two parallel connection results through the first cut line 31.

[0056] As mentioned above, the editor removes the first endpoint (namely the endpoint L0) in the queue Q, therefore, only the endpoint L4, the endpoint L6 and the endpoint L11 remain in the queue Q after the first cut line 31 is constructed.

[0057] Afterward, as shown in Fig. 3F, the editor fetches again the first two endpoints (namely the endpoints L4 and L6) in the queue Q and removes the first endpoint (namely the endpoint L4) in the queue Q.

[0058] Afterward, the editor constructs a second cut line 32 based on the endpoints L4 and L6 and then updates the serial-parallel relationship of the normal blocks 2 with similar process mentioned above. As mentioned above, the editor removes the first endpoint (namely the endpoint L4) in the queue Q, therefore, only the endpoint L6 and the endpoint L11 remain in the queue Q after the second cut line 32 is constructed.

[0059] As shown in Fig. 3G, the editor fetches again the first two endpoints (namely the endpoints L6 and L11) in the queue Q and removes the first endpoint (namely the endpoint L6) in the queue Q after constructing the second cut line 32.

[0060] Afterward, the editor constructs a third cut line 33 based on the endpoints L6 and L11 and then updates the serial-parallel relationship of the normal blocks 2 with similar process mentioned above.

[0061] As mentioned above, the editor will remove the first endpoint in the queue Q (namely the endpoint L6) after constructing the third cut line 33. Therefore, only one endpoint L11 remains in the queue Q after constructing the third cut line 33. Because one endpoint cannot constitute a cut line, the editor will end the construction of the cut line and set the last endpoint L11 in the queue Q as the end node of the continuous vertical line 3. In this way, the editor may generate the continuous vertical line 3 by the first cut line 31, the second cut line 32 and the third cut line 33.

[0062] By the present invention, user may use single operation (such as the operation from pressing left button of mouse to releasing left button of mouse) in editing mode to generate a continuous vertical line 3 passing (connecting) a plurality of nodes, which is very convenient for user.

[0063] Fig. 4 is the schematic view showing the retreating of the vertical line. During the construction of the continuous vertical line 3 (namely the editor is still in the editing mode), the user may control the pointer backward at his disposal (namely moving the pointer toward the source node 41). When the editor determines that the pointer is moved toward the source node 41 and already leaves the connected node 43, the editor converts the connected node 43 into candidate node 42 as shown in Fig. 4 and then displays the converted candidate node 42 with other candidate node 42 on the graph 1.

[0064] As mentioned above, when a candidate node 42 is converted into a connected node, the editor will store the connected node (such as the endpoint L11) into queue Q. When user moves back the pointer to convert the connected node 43 into candidate node 42, the editor will remove the candidate node 42 (such as the endpoint L11) from the queue Q to render more flexibility to user for drawing.

[0065] Figs. 5A to 5C respectively show the first flowchart, the second flowchart and the third flowchart of the editing method. Hereinafter the detailed steps for the editing method will be described with also reference to Figs. 3A to 3G.

[0066] At first the editor receives a first external operation of user to enter the editing mode (step S30). In one embodiment, the user keeps pressing the button of mouse to execute the first external operation. In another embodiment, the user presses a special function key of the keyboard to execute the first external operation.

[0067] In the editing mode, according to user operation, the editor selects the endpoint of any normal block 2 on the graph 1 as the source node 41 (step S32). In

one embodiment, the editor may set the endpoint corresponding to the current location of the computer cursor (such as mouse pointer) as source node 41, but this is not limitation for the present invention.

[0068] After ensuring the source node 41, the editor clears the record of all nodes in queue Q (step S34). Afterward, the editor receives a second external operation of user to move the cursor from the source node 41 and calculates the moving distance of the cursor (step S36). In one embodiment, user drags the mouse to execute the second external operation. In another embodiment, user presses arrow key of the keyboard to execute the second external operation.

[0069] After step S36, the editor further determines whether the moving distance is larger than a preset threshold (step S38). The editor performs no further action if the moving distance is not larger than the preset threshold and the editor repeats above step before exiting the editing mode. The editor adds the source node 41 into queue Q (step S40) if the moving distance is larger than the preset threshold. More particularly, in step S40, the editor adds the number of the endpoint on which the source node hovers into the queue Q from back end of the queue Q and then sets the source node 41 as the current node for the cursor.

[0070] In this embodiment, the editor determines the moving direction of the cursor from the source node based on the second external operation of user, namely, determines whether the moving direction of the cursor is upward or downward (step S42). If the cursor, controlled by user, is moved downward with respect to the current node and the moving distance is larger than the preset threshold, the editor only calculates and displays one or more effective candidate node 42 below the current node (step S44). On the contrary, if the cursor is moved upward with respect to the current node and the moving distance is larger than the preset threshold, the editor only calculates and displays one or more effective candidate node 42 above the current node (step S46).

[0071] As mentioned above, the effective candidate node 42 in this present invention is the endpoint that the editor finds no short circuit for any normal block when the current node is connected to this endpoint.

[0072] The editor keeps determining whether it receives a third external operation sent by user to exit the editing mode (step S48). In one embodiment, user releases mouse button to execute the third external operation. In another embodiment, user presses again the special function key of computer keyboard to execute the third external operation. If the editor does not receive the third external operation, the editor performs the steps shown in Fig. 5B. If the editor receives the third external operation, the editor performs the steps shown in Fig. 5C to exit the editing mode.

[0073] As shown in Fig. 5B, the editor determines whether user moves the cursor by performing the second external operation and user hovers the cursor at any candidate node 42 (step S50). If user moves the cursor and the cursor hovers at any candidate node 42, the editor adds the candidate node 42 into queue Q to convert the candidate node 42 into connected node (step S52). On the contrary, if the cursor does not hover on any candidate node 42, the editor does not perform any further action.

[0074] More particularly, in step S52, if the cursor is moved and hovers at any candidate node 42 by user operation, the editor stores the number of the endpoint corresponding to the location of the candidate node 42 into the queue Q from the back end of the queue Q such that the candidate node 42 is recorded as connected node 43.

[0075] Besides, the editor further determines whether the cursor is controlled by user to move backward and leave any connected node 43 (step S54). When the editor determines that the cursor is moved backward (namely moving toward the source node 41) and leave a connected node 43, the editor converts the connected node 43 into the candidate node 42 and removes the candidate node 42 from the queue Q from the back end of the queue Q (step S56).

[0076] More particularly, in step S56, the editor removes the number of the endpoint corresponding to the location of the candidate node 42 from the queue Q.

[0077] Before exiting the editing mode, the editor returns to step S42 to continually calculate and display currently available candidate node 42, convert the candidate node 42 passed by the cursor into connected node 43 and convert (resume) the connected node 43 (not passed by the cursor) back to the candidate node 42.

[0078] As shown in Fig. 5C, if the editor exits the editing mode based on the third external operation of user, then the editor first fetches the queue Q and determines whether the number of nodes stored in the queue Q is larger than one (step S58). In the present invention, if the number of nodes stored in the queue Q is larger than one (namely at least two nodes), then the editor may construct the cut lines 31-33 as shown in Figs. 3E to 3G based on these two nodes. If the number of nodes stored in the queue Q is not larger than one (namely only one node or no node), the editor ends the construction of the continuous vertical line.

[0079] In step S58, if the number of nodes stored in the queue Q is larger than one, the editor fetches the first two nodes in the queue Q and removes the record of the first node in the queue Q (step S60). More particularly, the editor uses the "peek" command to check the first two nodes in the queue Q from the front end of the queue Q, and uses the two nodes as the start point and endpoint for the cut line. The editor uses the "pop" command to remove the first node in the queue Q from the front end of the queue Q to set the endpoint of the current cut line as the start point of the next cut line, or as the end node of the continuous vertical line.

[0080] After step S60, the editor constructs a first cut line based on the fetched two nodes, sets one or more normal block 2 on the left side of the start point of the cut

line as the first left group and sets one or more normal block 2 on the right side of the start point of the cut line as the first right group. The editor further sets one or more normal block 2 on the left side of the endpoint of the cut line as the second left group, and sets one or more normal block 2 on the right side of the endpoint of the cut line as the second right group (step S62). In other word, based on the start point and the endpoint of the cut line, the editor separates the continue blocks of the graph 1 into left group and right group in step S62.

[0081] After step S62, the editor further parallel connects the first left group and the second left group to generate the first branch (parallel) group (step S64), and parallel connects the first right group and the second right group to generate the second branch (parallel) group (step S66).

[0082] More particularly, in step S64, the editor parallel connects the first left group and the second left group to generate the new branch group based on the common start point (the leftmost part of the graph 1) and the cut line. In step S66, the editor parallel connects the first right group and the second right group to generate the new branch group based on the common endpoint (the rightmost part of the graph 1) and the cut line.

[0083] After step S66, the editor further serial connects the first branch group with the second branch group to form new continue (serial) group (step S68).

[0084] Afterward, the editor is back to step S58 to determine whether the number of the node remaining in the queue Q is larger than one after the above-mentioned cut line is generated. The editor re-executes steps S60 to S68 to generate the next cut line if the number of the nodes remaining in the queue Q is larger than one. The editor modifies the serial-parallel relationship of the normal blocks in the graph 1 based on the one or more new branch (parallel) group, and the editor then updates and displays the graph 1 (step S70).

[0085] The above example is exemplified with pointer of computer mouse. However, the editing method of the present invention can also be implemented by computer keyboard.

[0086] Refer also to Figs. 6A to 6D, which show the first construction operation to the fourth construction operation for the continuous vertical line according to the third embodiment.

[0087] As shown in Fig. 6A, after entering the editing mode, user may operate the arrow key of keyboard to move cursor of the editor and select one of the endpoints as the source node 61. After user selects the source node 61, the editor may display, after calculation, one or more effective candidate nodes 62 around the source node 61, where the source node 61 and the candidate nodes 62 may be represented with different colors or different shapes.

[0088] Afterward, as shown in Figs. 6B and 6C, user may use arrow key of keyboard to move the cursor from the source node 61 to any candidate node 62. The editor converts the candidate node 62 on which the cursor hovers into connected node 63 and generates a pre-connection line 7 between the two nodes 61 and 63. When one candidate node 62 is converted into connected node 63, the editor further calculates and displays one or more effective candidate node 62 around the connected node 63.

[0089] Finally, as shown in Fig. 6D, when user sends command through keyboard to control the editor exit the editing mode, the editor generates the continuous vertical line 8 based on the source node 61 and one or more connected node 63, and modifies the serial-parallel relationship of the normal blocks in the graph 1 based on the continuous vertical line 8. The editor updates and displays the graph 1.

[0090] By above editing method of the present invention, user only needs one-time operation by mouse, keyboard or touch pad to directly generate a continuous vertical line in graph, thus fast modify the relationship among normal blocks. User may use the editor of visual programming language more efficiently.

Claims

1. A method for editing continual vertical line of visual programming language, the method applied to an editor operated on a computer, the editor providing a serial-parallel graph (1) comprising a plurality of normal blocks (2), the method comprising:

a) receiving a first external operation to enter an editing mode;

b) obtaining a source node (41) on the serial-parallel graph (1) when entering the editing mode, and setting the source node (41) as a current node, wherein the source node (41) is an endpoint (L0~L11) of one of the normal blocks (2);

c) displaying at least one candidate node (42) around the current node, wherein the at least one candidate node (42) is an endpoint (L0~L11) of at least one of the normal blocks (2) around the current node;

d) receiving a second external operation to connect the current node to one of the at least one candidate nodes (42);

e) converting the connected candidate node (42) into a connected node (43);

f) determining whether a third external operation for exiting the editing mode is received;

g) setting the connected node (43) as the current node and repeating the step c) to the step f) before exiting the editing mode;

h) generating a continuous vertical line (3) connecting the source node (41) and at least one of the connected nodes (43) when exiting the editing mode; and

i) modifying a serial-parallel relationship of the

normal blocks based on the continuous vertical line (3) and updating the serial-parallel graph (1).

2. The method in claim 1, wherein the first external operation is performed by continuously pressing a button of a mouse of the computer, the second external operation is performed by moving the mouse for dragging a pointer, the third external operation is performed by releasing the button of the mouse.

3. The method in claim 1, wherein the first external operation is performed by pressing a special function key of a keyboard of the computer, the second external operation is performed by pressing an arrow key of the keyboard, and third external operation is performed by pressing again the special function key of the keyboard.

4. The method in claim 1, wherein in the step h), the continuous vertical line (3) is generated based on the source node (41) stored in a queue (Q) of the editor and the connected candidate nodes (42) stored in the queue (Q), wherein in the step i), the serial-parallel relationship of the normal blocks is modified based on the source node (41) stored in the queue (Q) of the editor and the connected candidate nodes (42) stored in the queue (Q).

5. The method in claim 4, wherein the step i) comprises following sub-steps:

i1) determining whether a number of the nodes stored in the queue (Q) is larger than one;
i2) fetching first two nodes in the queue (Q) and then removing a first node in the queue (Q) when the number of the nodes stored in the queue (Q) is larger than one;
i3) after the sub-step i2), generating a cut line (31, 32, 33) based on the two fetched nodes, setting one or more normal block (2) on a left side of a start point of the cut line (31, 32, 33) as a first left group, setting one or more normal block (2) on a right side of the start point of the cut line (31, 32, 33) as a first right group, setting one or more normal block (2) on a left side of an endpoint of the cut line (31, 32, 33) as a second left group, setting one or more normal block (2) on a right side of the endpoint of the cut line (31, 32, 33) as a second right group;
i4) after the sub-step i3), parallel connecting the first left group and the second left group to form a first branch group;
i5) after the sub-step i3), parallel connecting the first right group and the second right group to form a second branch group;
i6) serially connecting the first branch group and the second branch group to form a new continue group;
i7) after the sub-step i6), repeating the sub-steps i1) to i6); and
i8) modifying the serial-parallel relationship of the normal blocks based on one or more new continue group and updating the serial-parallel graph (1) if the number of the nodes stored in the queue (Q) is not larger than one.

6. The method in claim 5, wherein the sub-step i2) the first two nodes in the queue (Q) are fetched with peek command to obtain the first two nodes as the start point and the endpoint of the cut line (31, 32, 33), the first node in the queue (Q)is removed with pop command.

7. The method in claim 1, further comprising a step b1) after the step b): adding the source node (41) to a queue (Q) of the editor;
wherein in the step e), the connected candidate node (42) is added to the queue (Q) of the editor to convert the connected candidate node (42) into the connected node (43).

8. The method in claim 7, further comprising a step b0) before the step b1): clearing the queue (Q).

9. The method in claim 7, further comprising following steps after the step b):

b21) calculating a moving distance from the current node;
b22) determining whether the moving distance is larger than a threshold; and
b23) performing the step c) when the moving distance is larger than the threshold.

10. The method in claim 7, further comprising a step b3) after the step b): determining a moving direction departing from the source node;
wherein at least one of the candidate nodes (42) above the current node is displayed when the moving direction is upward;
wherein at least one of the candidate nodes (42) below the current node is displayed when the moving direction is downward.

11. The method in claim 1, wherein the candidate node (42) is an effective candidate node not causing short circuit of one or more normal block (2) in the serial-parallel graph (1).

12. The method in claim 1, wherein the candidate node (42) is displayed with dashed line, and the connected node (43) is displayed with solid line.

13. The method in claim 1, wherein the candidate node (42) is displayed with a first color, and the connected node (43) is displayed with a second color.

14. The method in claim 7, wherein in the step e), the candidate node (42) is added to the queue (Q) from a back end of the queue (Q) when the editor receives the second external operation and a cursor hovers on any candidate node (42).

15. The method in claim 7, further comprising following steps after the step e):

e1) determining whether the cursor is moved back and leaves the connected node (43); and
e2) resuming the connected node (43) to the candidate node (42) and removing the candidate node (42) from the back end of the queue (Q) when the cursor is moved back and leaves the connected node (43).

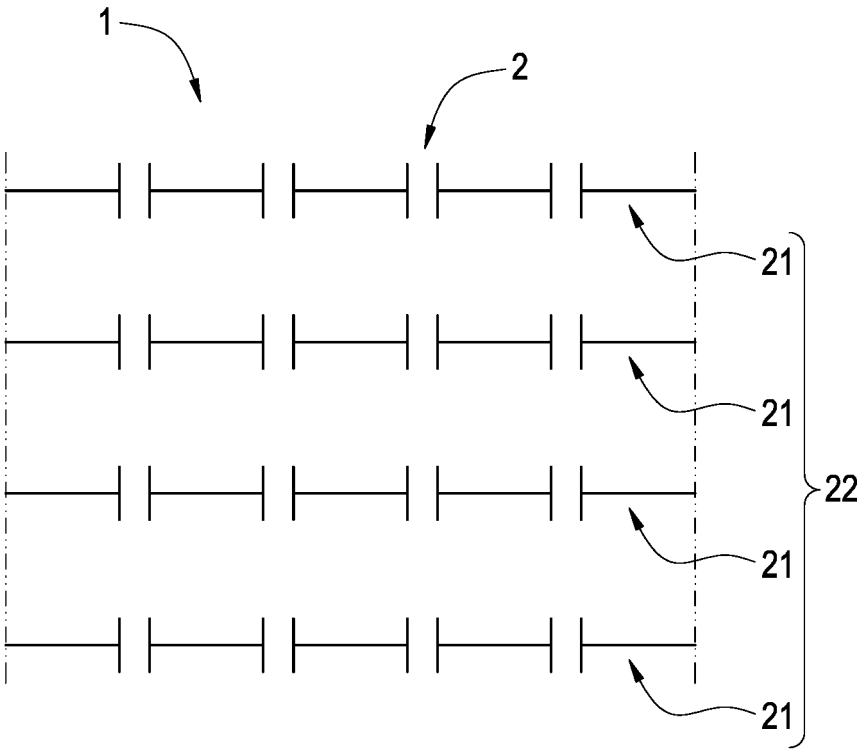


FIG.1A

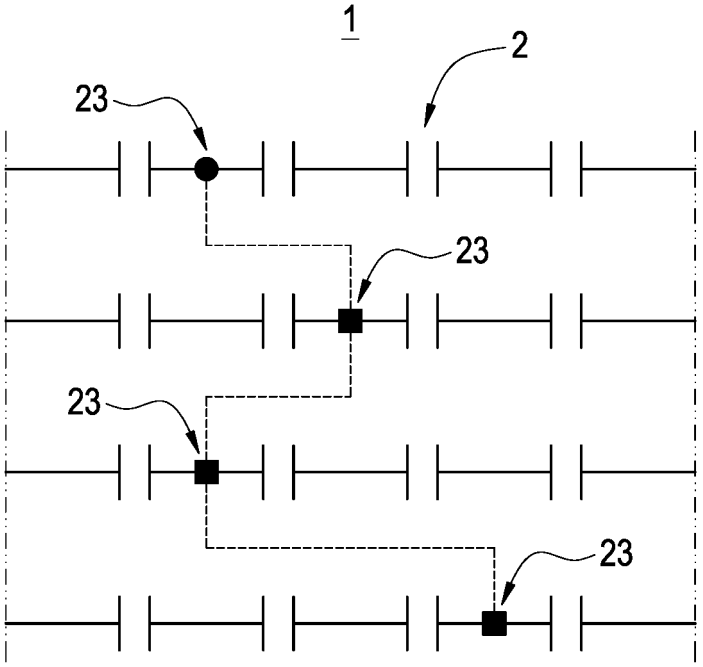


FIG.1B

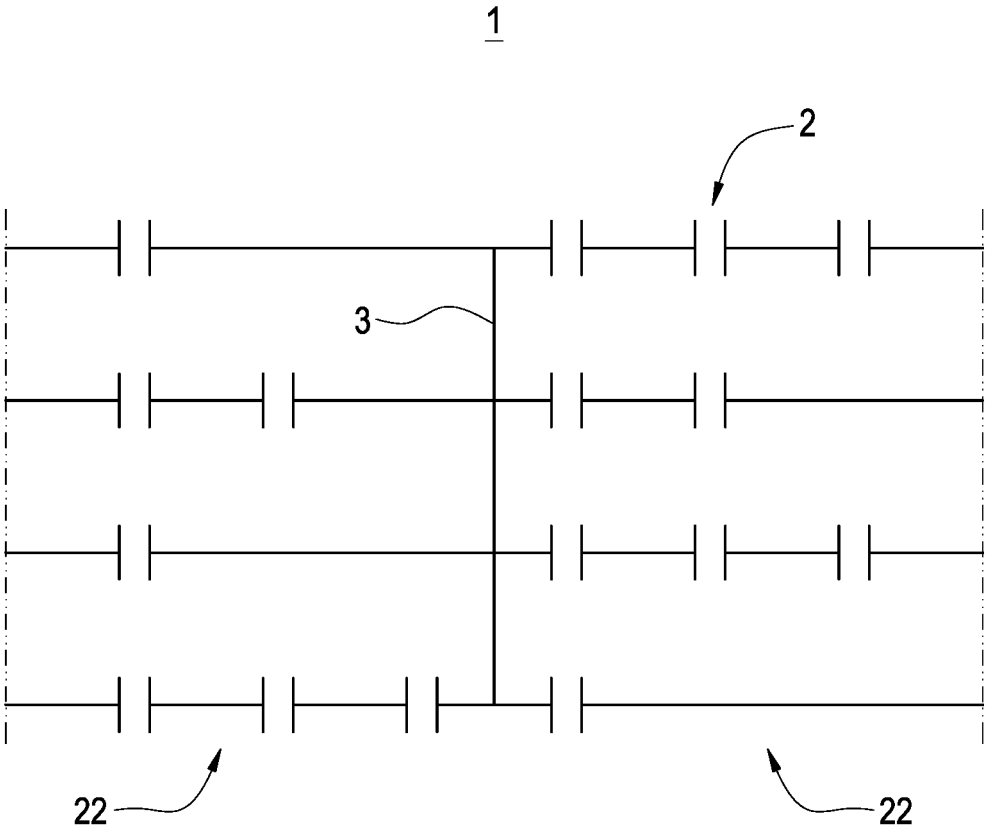


FIG.1C

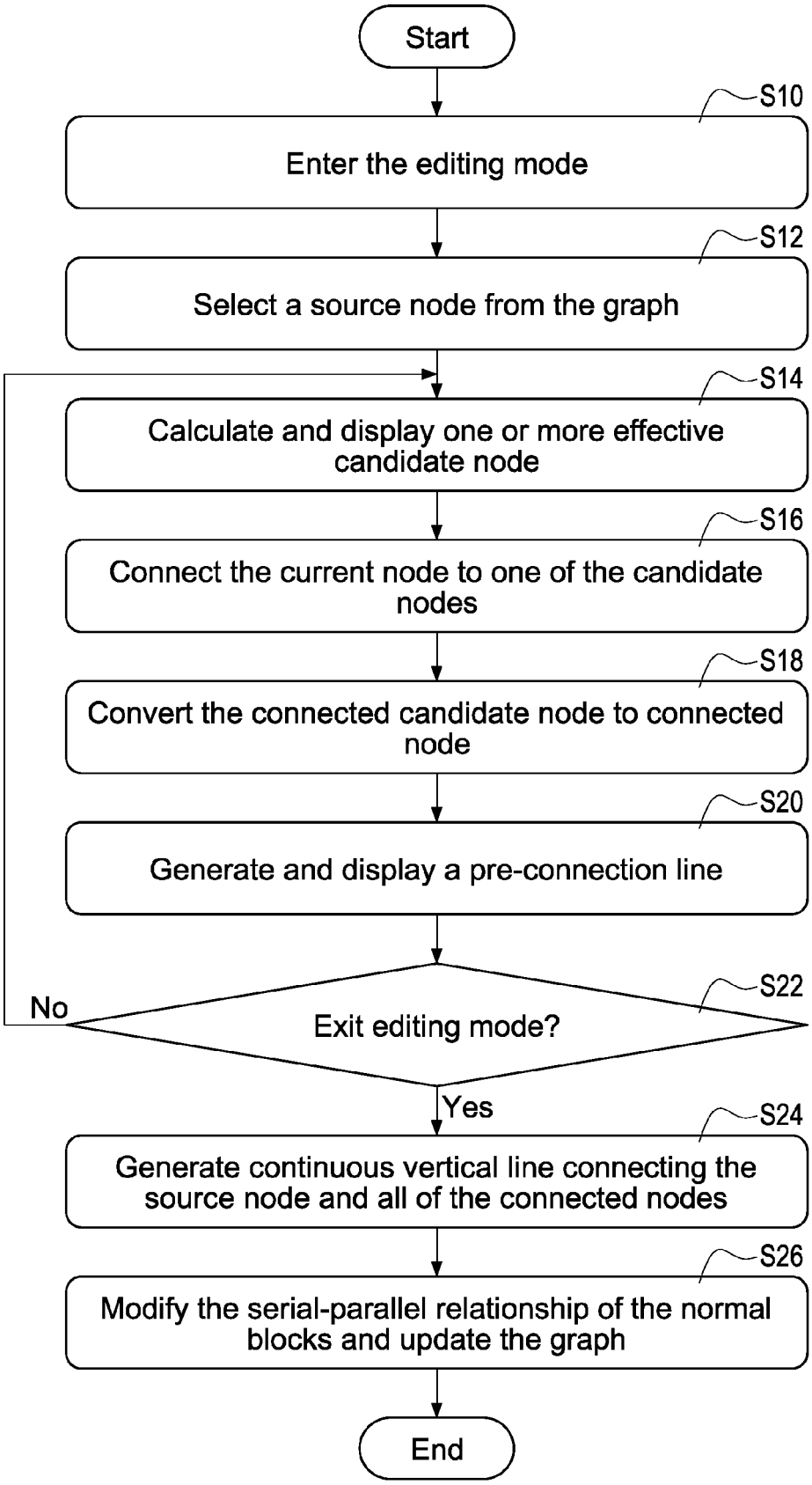

Start
S10
Enter the editing mode
S12
Select a source node from the graph
S14
Calculate and display one or more effective candidate node
S16
Connect the current node to one of the candidate nodes
S18
Convert the connected candidate node to connected node
S20
Generate and display a pre-connection line
S22
Exit editing mode?
No
Yes
S24
Generate continuous vertical line connecting the source node and all of the connected nodes
S26
Modify the serial-parallel relationship of the normal blocks and update the graph
End

FIG.2

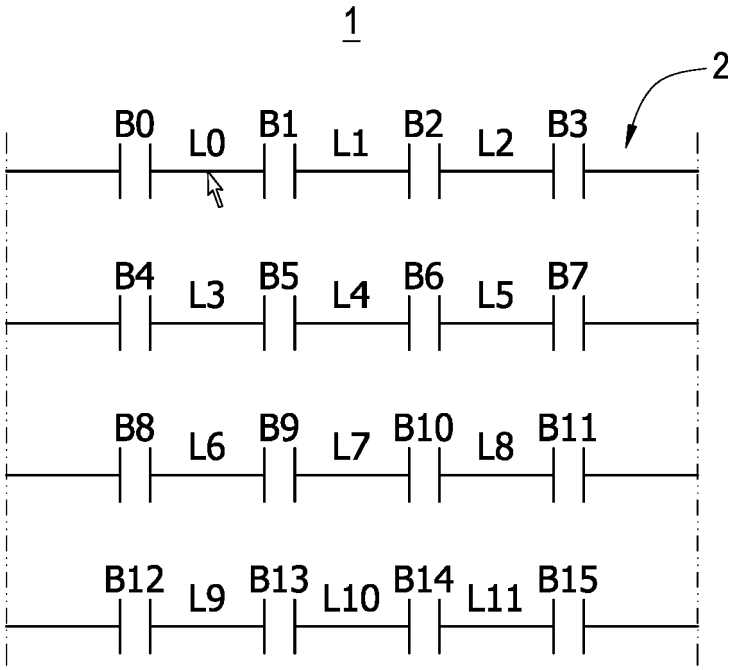

1
2
B0
L0
B1
L1
B2
L2
B3
B4
L3
B5
L4
B6
L5
B7
B8
L6
B9
L7
B10
L8
B11
B12
L9
B13
L10
B14
L11
B15
Q = {}

FIG.3A

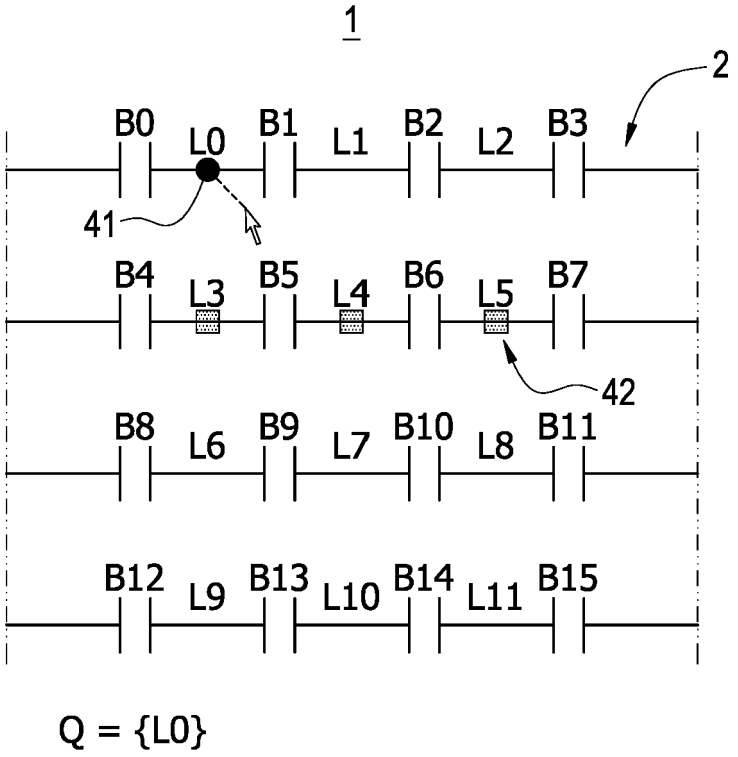
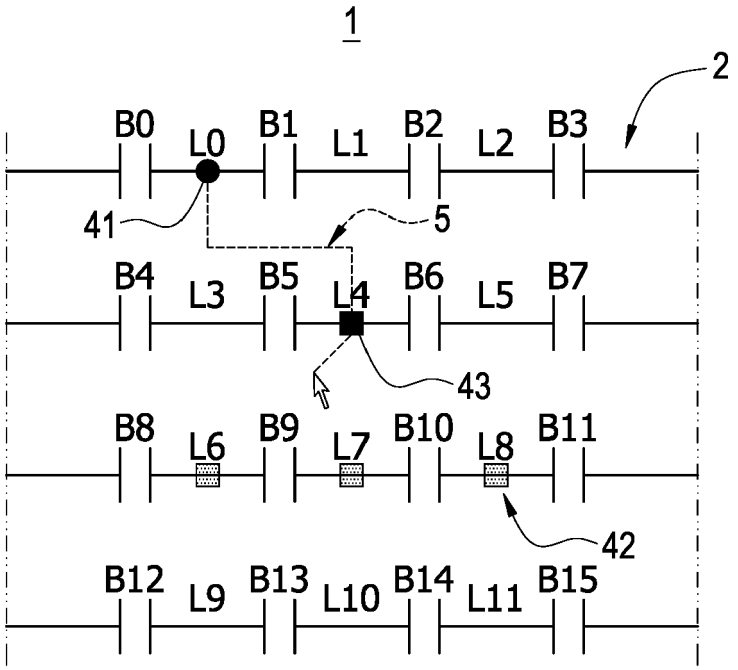
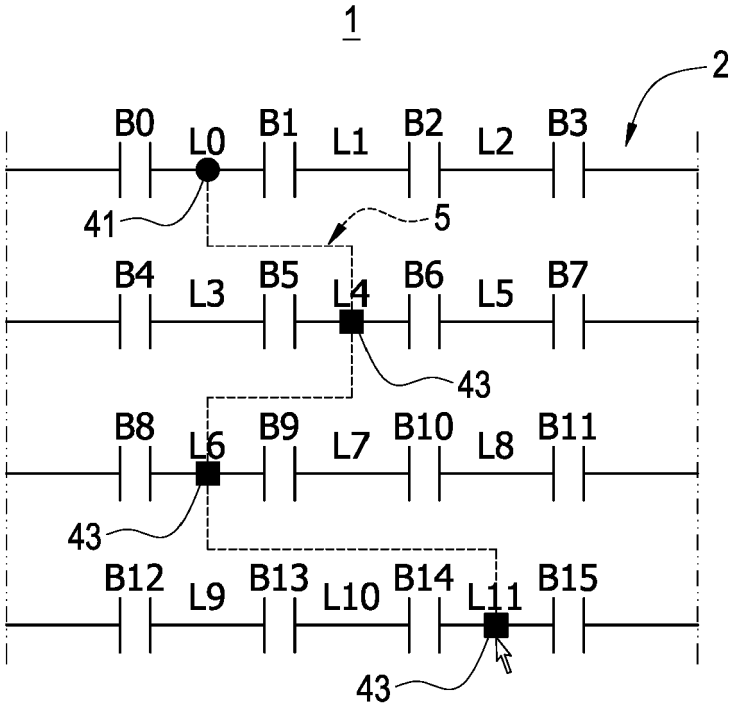

1
2
B0
L0
B1
L1
B2
L2
B3
41
B4
L3
B5
L4
B6
L5
B7
42
B8
L6
B9
L7
B10
L8
B11
B12
L9
B13
L10
B14
L11
B15
Q = {L0}

FIG.3B



Q = {L0,L4}

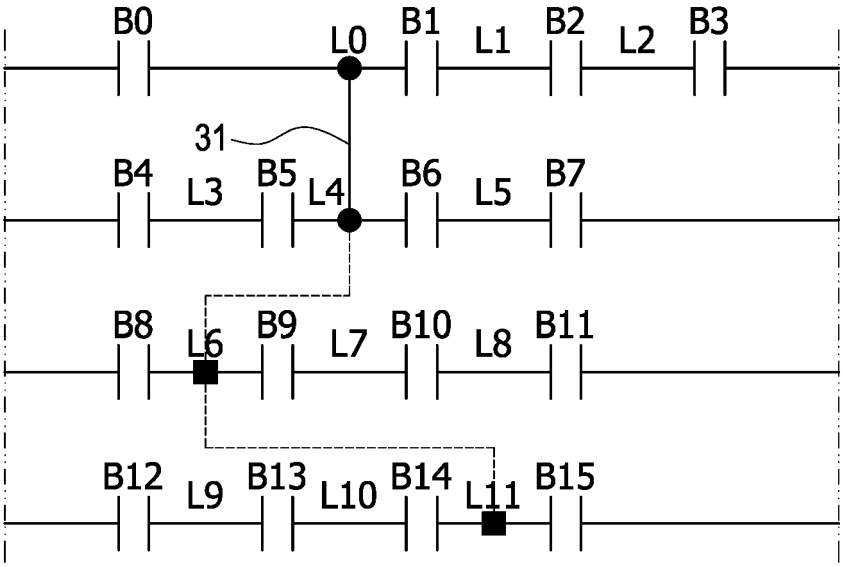
FIG.3C



Q = {L0,L4,L6,L11}

FIG.3D

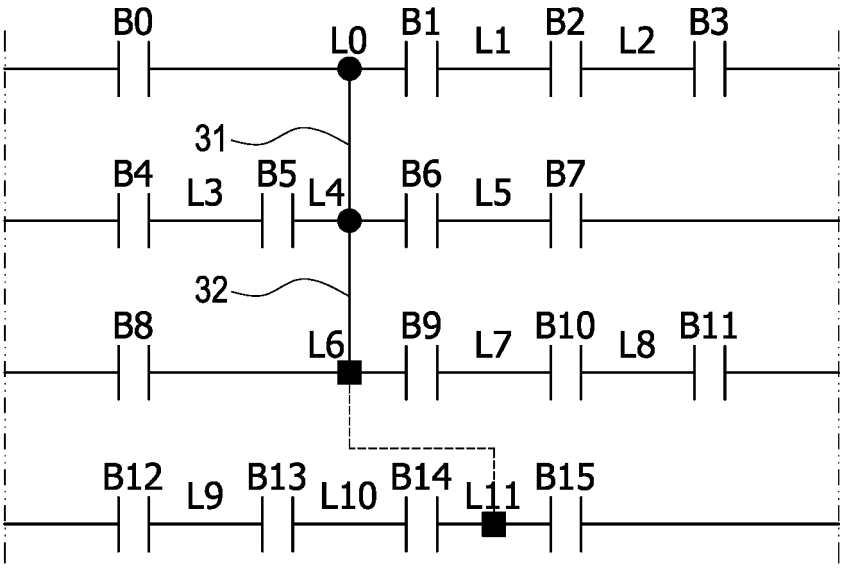
1



Q = {L4,L6,L11}

FIG.3E

1



Q = {L6,L11}

FIG.3F

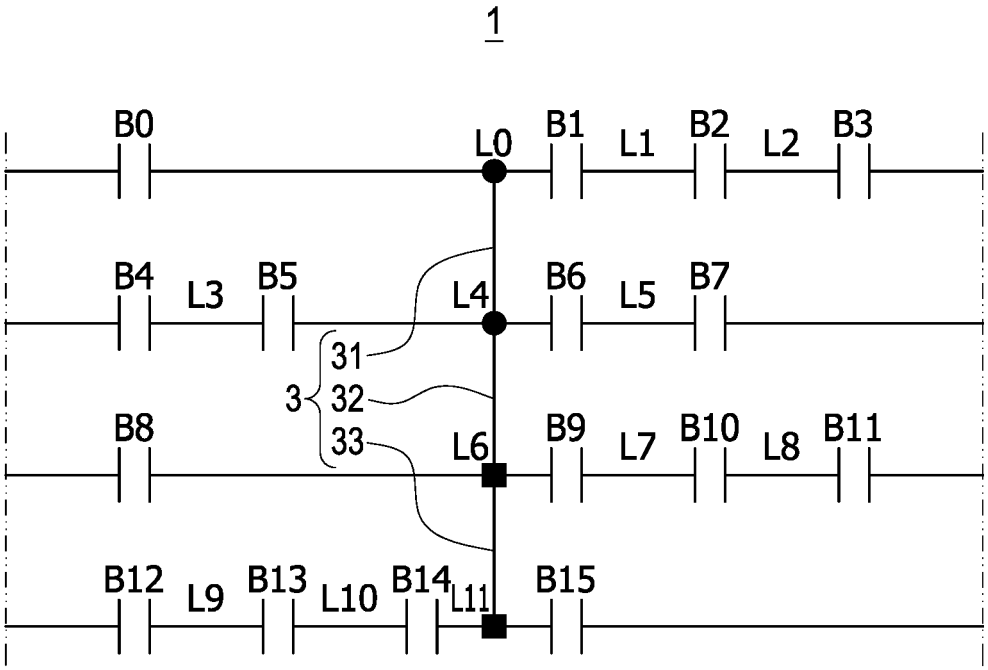

1
B0
L0
B1
L1
B2
L2
B3
B4
L3
B5
L4
B6
L5
B7
31
3
32
33
B8
L6
B9
L7
B10
L8
B11
B12
L9
B13
L10
B14
L11
B15
Q = {L11}

FIG.3G

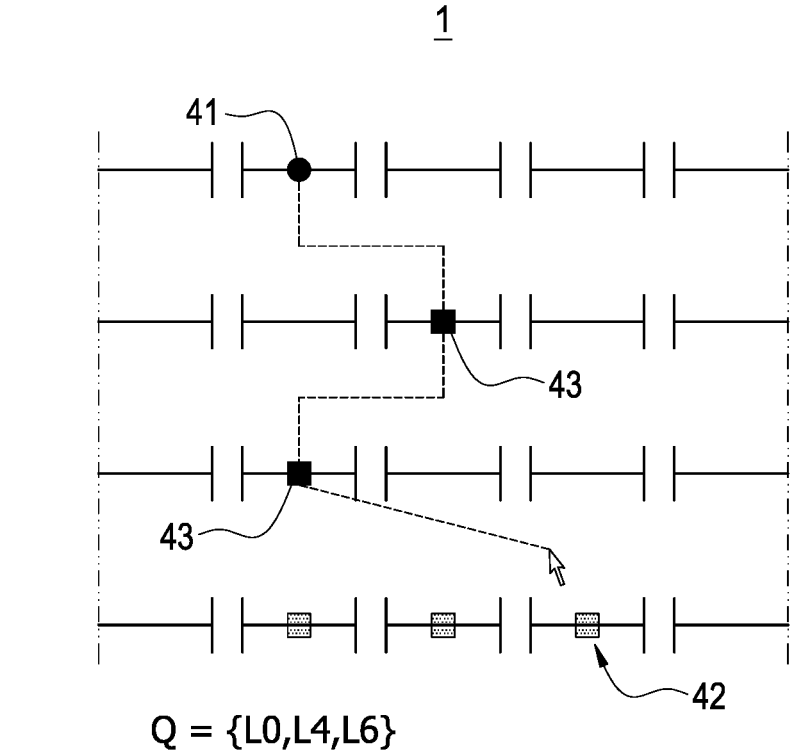

1
41
43
43
42
Q = {L0,L4,L6}

FIG.4

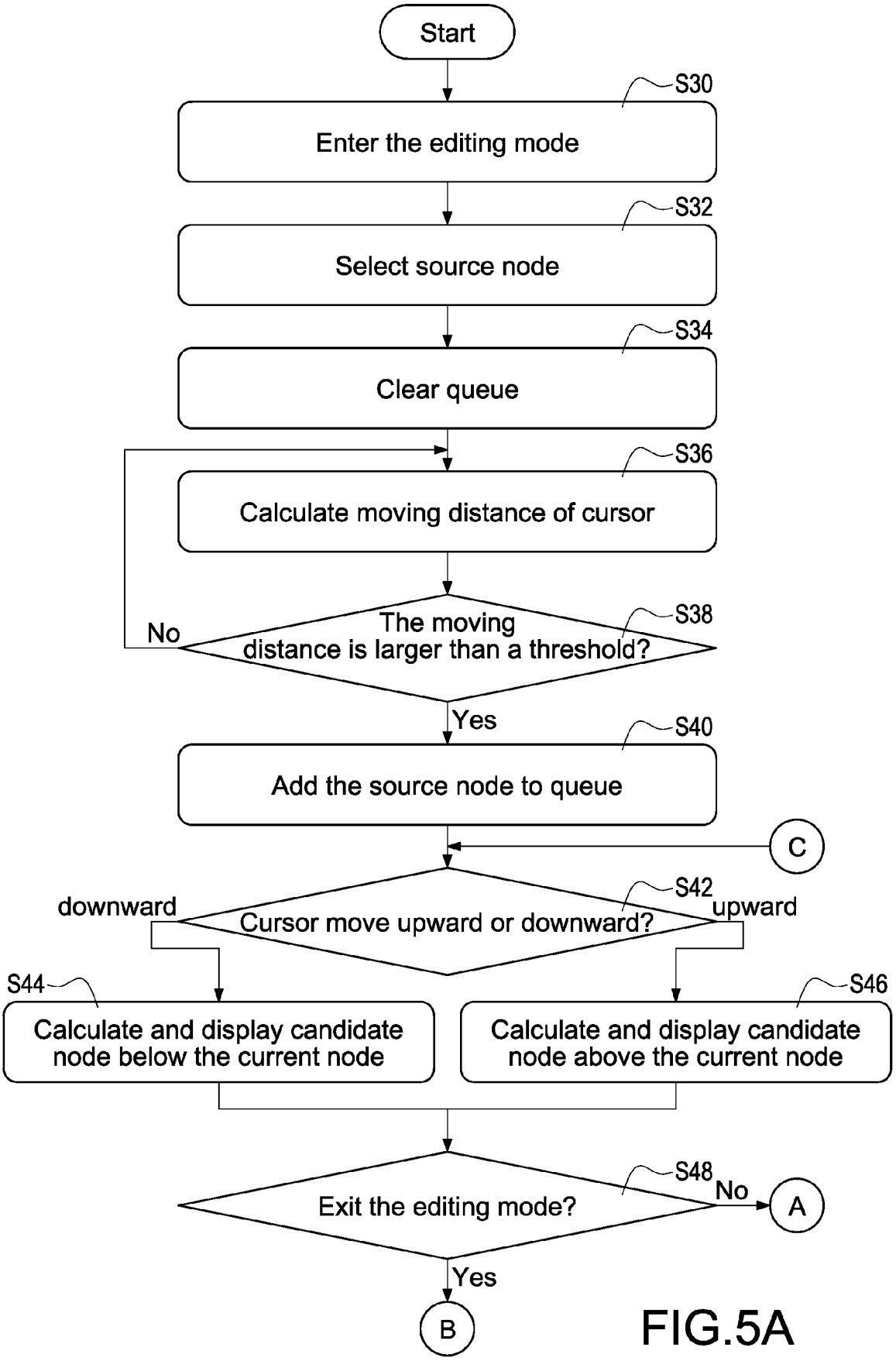

Start
S30
Enter the editing mode
S32
Select source node
S34
Clear queue
S36
Calculate moving distance of cursor
S38
The moving distance is larger than a threshold?
No
Yes
S40
Add the source node to queue
C
S42
Cursor move upward or downward?
downward
upward
S44
Calculate and display candidate node below the current node
S46
Calculate and display candidate node above the current node
S48
Exit the editing mode?
No
A
Yes
B

FIG.5A

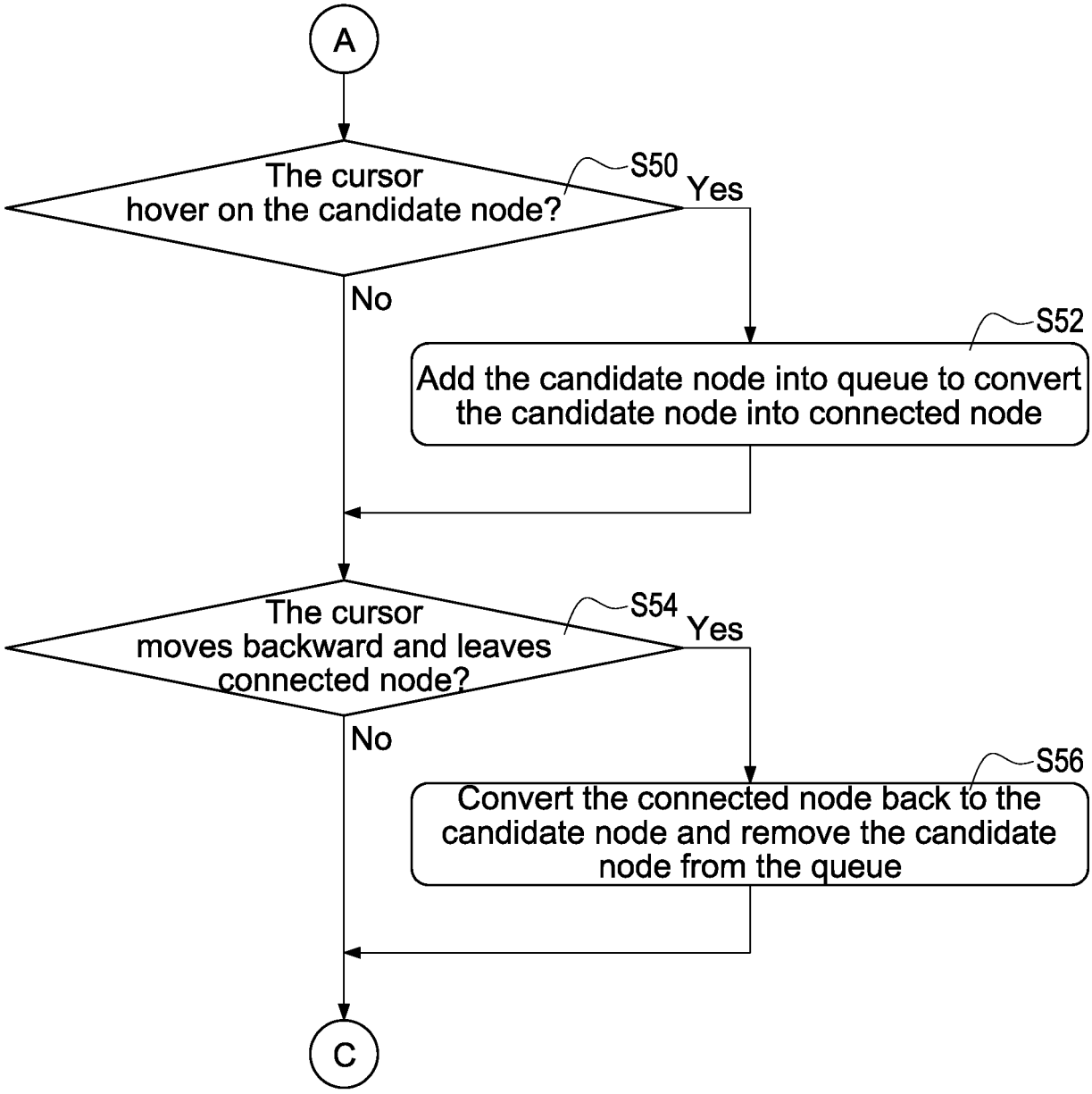

A
The cursor hover on the candidate node?
S50
Yes
No
S52
Add the candidate node into queue to convert the candidate node into connected node
The cursor moves backward and leaves connected node?
S54
Yes
No
S56
Convert the connected node back to the candidate node and remove the candidate node from the queue
C

FIG.5B

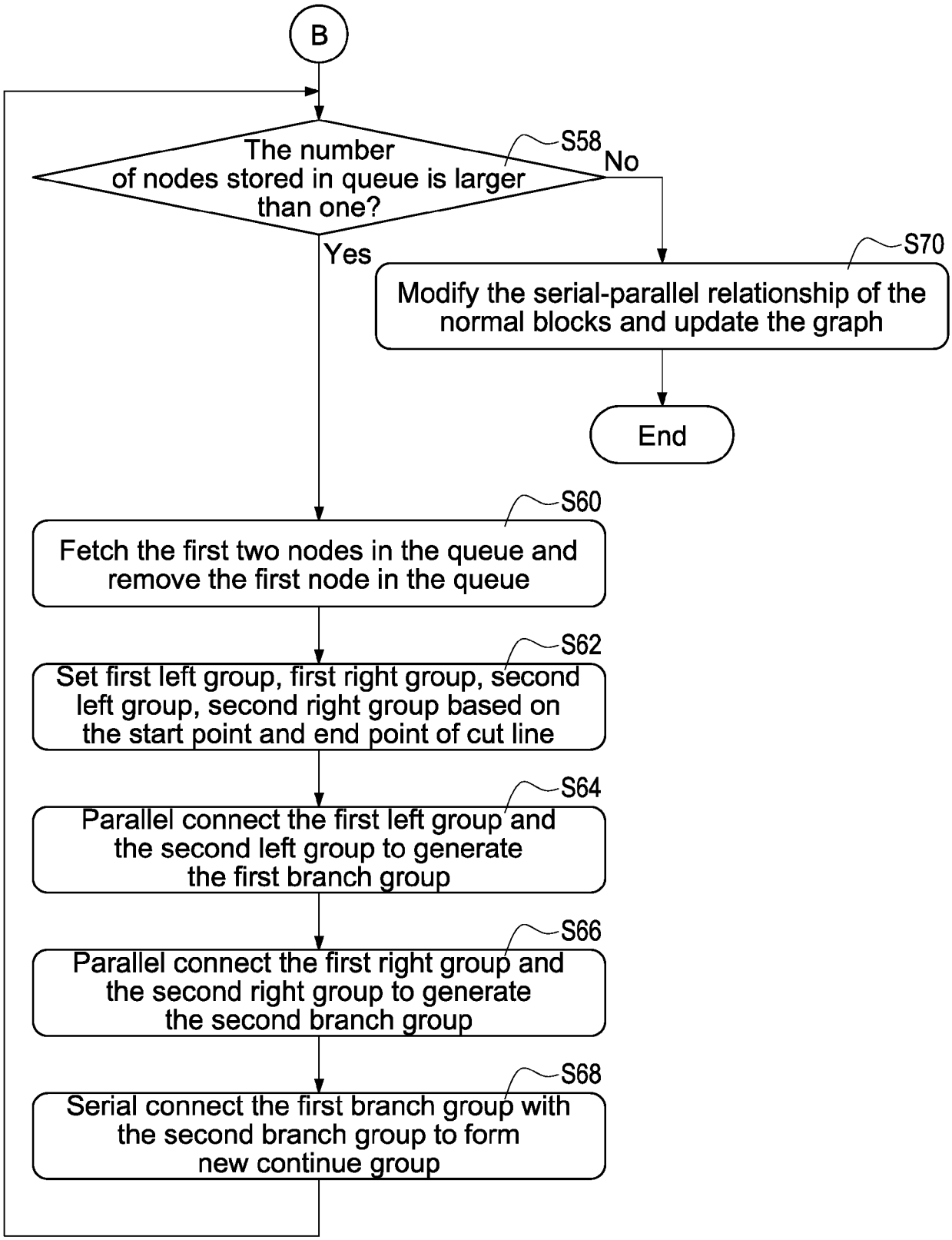

B
The number of nodes stored in queue is larger than one?
S58
No
Yes
S70
Modify the serial-parallel relationship of the normal blocks and update the graph
End
S60
Fetch the first two nodes in the queue and remove the first node in the queue
S62
Set first left group, first right group, second left group, second right group based on the start point and end point of cut line
S64
Parallel connect the first left group and the second left group to generate the first branch group
S66
Parallel connect the first right group and the second right group to generate the second branch group
S68
Serial connect the first branch group with the second branch group to form new continue group

FIG.5C

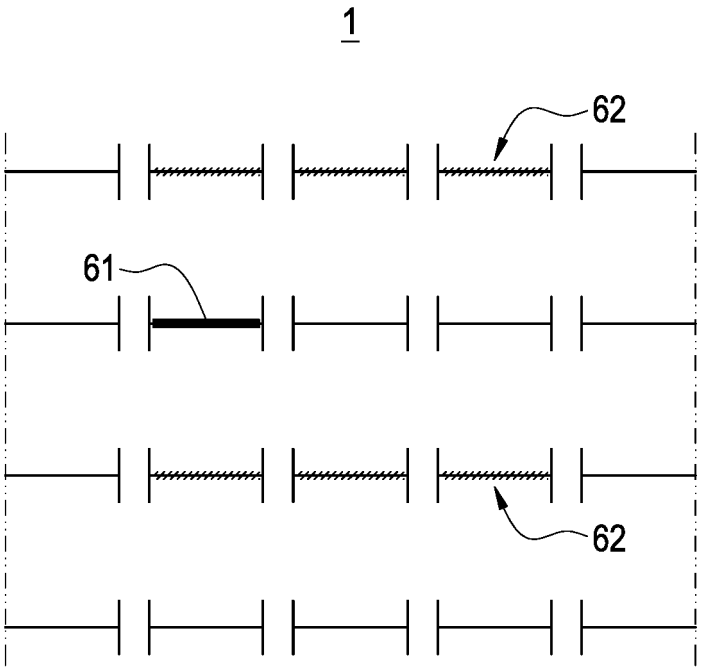


FIG.6A

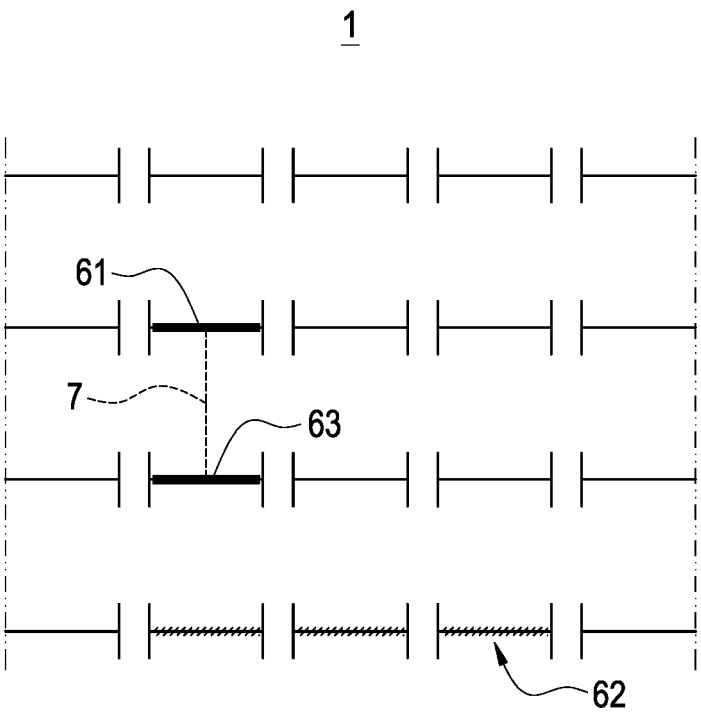


FIG.6B

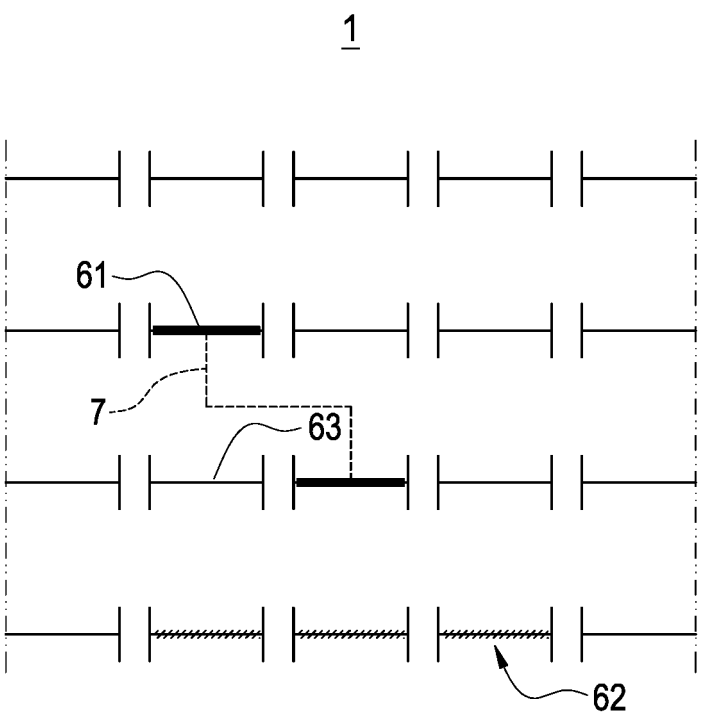


FIG.6C

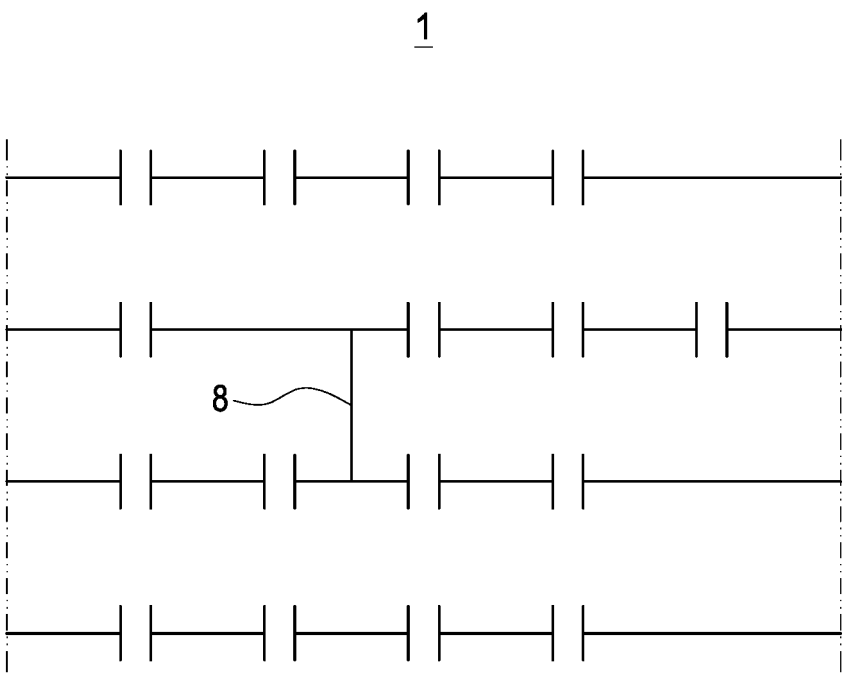


FIG.6D

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 20 19 9535

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X A	US 6 484 189 B1 (GERLACH JR JOHN D [US] ET AL) 19 November 2002 (2002-11-19) * abstract * * column 2, line 17 - column 3, line 50 * * column 6, line 20 - column 8, line 43 * * column 9, line 13 - column 10, line 25 * * column 11, line 38 - column 14, line 54 * * column 18, line 31 - line 65 * * column 20, line 50 - column 22, last line * * column 23, line 27 - column 30, line 49 * * column 32, line 56 - column 39, line 36 * * claims 1-20; figures 1-24 *	1-3,7-15 4-6	INV. G06F8/34 G06F9/451
A	US 2006/259452 A1 (GIBBONS JOHN K [US]) 16 November 2006 (2006-11-16) * abstract * * paragraph [0003] - paragraph [0006] * * paragraph [0026] - paragraph [0037] * * paragraph [0052] - paragraph [0061] * * paragraph [0063] - paragraph [0089] * * paragraph [0095] - paragraph [0108] * * claims 1-31; figures 1-20 *	1-15	TECHNICAL FIELDS SEARCHED (IPC) G06F
A	US 2013/074024 A1 (CHASE SCOTT I [US] ET AL) 21 March 2013 (2013-03-21) * abstract * * paragraph [0003] - paragraph [0027] * * paragraph [0063] - paragraph [0068] * * paragraph [0094] - paragraph [0289] * * claims 1-49; figures 1-38 *	1-15	

-/--

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	26 March 2021	Eftimescu, Nicolae

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

Europäisches Patentamt
European Patent Office
Office européen des brevets

EUROPEAN SEARCH REPORT

Application Number
EP 20 19 9535

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
A	"SIMULINK. model-Based and System-Based Design. Version 5", USING SIMULINK. SIMULINK. MMDEL-BASED AND SYSTEM-BASED DESIGN, MATHWORKS, US, 1 January 2004 (2004-01-01), page 488pages, XP007901395, * abstract * * Chapter 4 - Creating a Model * * Chapter 8 - Modeling with Simulink *	1-15	
A	US 2014/040792 A1 (KODOSKY JEFFREY L [US]) 6 February 2014 (2014-02-06) * abstract * * paragraph [0001] - paragraph [0014] * * paragraph [0027] - paragraph [0054] * * paragraph [0081] - paragraph [0088] * * paragraph [0090] - paragraph [0100] * * paragraph [0112] - paragraph [0124] * * paragraph [0127] - paragraph [0131] * * claims 1-14; figures 1-15 *	1-15	
			TECHNICAL FIELDS SEARCHED (IPC)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
Munich	26 March 2021	Eftimescu, Nicolae

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 20 19 9535

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

26-03-2021

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6484189 B1	19-11-2002	NONE	
US 2006259452 A1	16-11-2006	NONE	
US 2013074024 A1	21-03-2013	US 2013074024 A1	21-03-2013
		US 2014149955 A1	29-05-2014
US 2014040792 A1	06-02-2014	CN 104583946 A	29-04-2015
		EP 2880530 A1	10-06-2015
		US 2014040792 A1	06-02-2014
		US 2014040798 A1	06-02-2014
		US 2015169185 A1	18-06-2015
		US 2015169297 A1	18-06-2015
		WO 2014021950 A1	06-02-2014

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82