



[12] 发明专利申请公开说明书

[21] 申请号 01815229.5

[43] 公开日 2004 年 10 月 6 日

[11] 公开号 CN 1535466A

[22] 申请日 2001.7.25 [21] 申请号 01815229.5

[30] 优先权

[32] 2000. 7. 25 [33] US [31] 60/220,790

[32] 2000. 8. 8 [33] US [31] 60/223,644

[32] 2000. 8. 8 [33] US [31] 60/224,080

[32] 2001. 3. 5 [33] US [31] 60/273,383

[86] 国际申请 PCT/US2001/023333 2001.7.25

[87] 国际公布 WO2002/009117 英 2002.1.31

[85] 进入国家阶段日期 2003.3.6

[71] 申请人 WM·马什赖斯大学

地址 美国得克萨斯

[72] 发明人 詹姆斯·M·图尔 比尔·范·灿特

克里斯托弗·赫斯本德

萨默·赫斯本德

[74] 专利代理机构 中原信达知识产权代理有限责
任公司

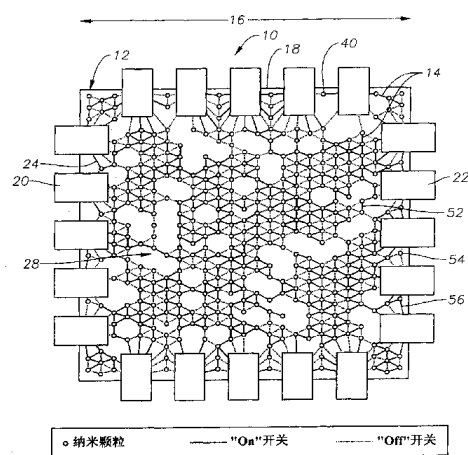
代理人 丁业平 王维玉

权利要求书 7 页 说明书 227 页 附图 4 页

[54] 发明名称 可编程分子装置

[57] 摘要

提供了一种包含随机纳米网络的可编程分子装置(10), 其中该随机纳米网络包含多个分子电路部件。优选的分子电路部件包含呈现负微分电阻的分子二极管。一种对该分子装置进行编程的方法, 可以包含配置该分子部件。配置分子部件可以包含将电压施加至连接到纳米网络的输入和输出引线两端。所述电压可以根据一个用于将该装置编程为用作例如逻辑单元或者存储器单元的自适应算法加以确定。分子计算机可以包含多个由金属线路(14)互连的可编程分子装置(10)。



1. 一个可编程分子装置，包含：

至少一个输入引线；

5 至少一个输出引线；以及

一个跨越所述输入引线和所述输出引线的纳米网络，其中所述纳米网络包含多个分子电路部件。

10 2. 如权利要求 1 所述的可编程分子装置，其特征在于：所述纳米网络是自组装的。

3. 如权利要求 1 所述的可编程分子装置，其特征在于：所述纳米网络是随机的。

15 4. 如权利要求 1 所述的可编程分子装置，其特征在于：所述装置可通过一种用于配置所述分子电路部件的自适应算法进行编程。

20 5. 如权利要求 4 所述的可编程分子装置，其特征在于：所述自适应算法选自遗传算法、模拟退火算法、跟踪优胜算法、时差学习算法、增强学习算法、以及它们的组合。

6. 如权利要求 4 所述的可编程分子装置，其特征在于：可通过向所述输入引线和所述输出引线施加电压来配置所述分子电路部件。

25 7. 如权利要求 1 所述的可编程分子装置，其特征在于：所述装置可编程为用作一个逻辑单元。

30 8. 如权利要求 7 所述的可编程分子装置，其特征在于：所述逻辑单元选自包含由所述至少一个输入引线和所述至少一个输出引线支持的真值表的组。

9. 如权利要求 8 所述的可编程分子装置，其特征在于：所述逻辑单元可编程为用作选自 AND、OR、XOR、NAND、NOT、加法器、半加法器、反相半加器、多路复用器、解码器、以及它们的组合的装置。

10. 如权利要求 1 所述的可编程分子装置，其特征在于：所述装置可编程为用作一个存储器单元。

11. 如权利要求 1 所述的可编程分子装置，其特征在于：所述装置是可重新编程的。

12. 如权利要求 1 所述的可编程分子装置，其特征在于：所述分子电路部件选自分子开关、分子二极管、分子线路、分子整流器、电阻、晶体管、分子存储器以及它们的组合。

13. 如权利要求 12 所述的可编程分子装置，其特征在于：所述分子电路部件包括分子开关。

14. 如权利要求 13 所述的可编程分子装置，其特征在于：所述装置可通过一种用于设置所述分子开关的算法进行编程。

15. 如权利要求 14 所述的可编程分子装置，其特征在于：可通过向所述输入引线 and 所述输出引线施加电压来设置所述开关。

16. 如权利要求 1 所述的可编程分子装置，其特征在于：所述纳米网络进一步包含纳米尺度部件。

17. 如权利要求 16 所述的可编程分子装置，其特征在于：所述的纳米尺度部件选自纳米管、纳米颗粒、纳米棒、以及它们的组合。

18. 如权利要求 17 所述的可编程分子装置，其特征在于：所述纳米尺度电路部件包括纳米颗粒，而且所述分子电路部件包括分子开关且所述分子开关互连所述纳米颗粒。

5

19. 如权利要求 18 所述的可编程分子装置，其特征在于：所述纳米颗粒是随机排列的。

10

20. 如权利要求 18 所述的可编程分子装置，其特征在于：所述分子开关随机地互连所述纳米颗粒。

21. 一种制造电子部件的方法，包含：

(a) 提供一个自组装的纳米单元；以及

(b) 编程该纳米单元以用作电子部件。

15

22. 如权利要求 21 所述的方法，其特征在于：纳米单元包含：

至少一个输入引线；

至少一个输出引线；以及

20

一个跨越该输入引线和该输出引线的纳米网络，其中该纳米网络包含多个分子电路部件。

25

23. 如权利要求 22 所述的方法，其特征在于：所述分子电路部件选自分子开关、分子二极管、分子线路、分子整流器、分子电阻、分子晶体管、分子存储器以及它们的组合。

24. 如权利要求 23 所述的方法，其特征在于：该分子电路部件包含分子谐振隧道二极管。

30

25. 如权利要求 24 所述的方法，其特征在于：该分子电路部件呈现负微分电阻。

26. 如权利要求 22 所述的方法，其特征在于：该纳米网络进一步包含选自纳米管、纳米颗粒、纳米棒、以及它们的组合的纳米尺度部件。

5

27. 如权利要求 22 所述的方法，其特征在于：所述纳米网络是随机的。

28. 如权利要求 21 所述的方法，其特征在于：步骤（b）包含：
（b1）配置该分子电路部件。

10

29. 如权利要求 28 所述的方法，其特征在于：步骤（b1）包含：
（b1.i）通过把电压加到输入引线和输出引线两端来调整至少一个分子电路部件的受导电率影响的属性。

15

30. 如权利要求 29 所述的方法，其特征在于：受导电率影响的属性选自电荷、构造状态、电子态、以及它们的组合。

31. 如权利要求 28 所述的方法，其特征在于：步骤（b）进一步包含：
（b2）测试该纳米单元的性能。

20

32. 如权利要求 31 所述的方法，其特征在于：步骤（b）进一步包含：

（b3）应用一种自适应算法以重新配置该分子电路部件。

25

33. 如权利要求 32 所述的方法，其特征在于：该自适应算法选自遗传算法、模拟退火算法、跟踪优胜算法、时差学习算法、增强学习算法、以及它们的组合。

30

34. 如权利要求 32 所述的方法，进一步包含：

(b4) 重复步骤 (b2) 和 (b3) 直到纳米单元起电子部件作用为止。

5 35. 如权利要求 22 所述的方法，其特征在于：该电子部件包含一个逻辑单元。

36. 如权利要求 35 所述的方法，其特征在于：所述逻辑单元选自包含由该输入引线和该输出引线支持的真值表的组。

10

37. 如权利要求 36 所述的方法，其特征在于：该逻辑单元选自 AND、OR、XOR、NOR、NAND、NOT、加法器、半加法器、反相半加器、多路复用器、解码器、以及它们的组合。

15 38. 如权利要求 22 所述的方法，其特征在于：该电子部件包含一个存储器单元。

39. 如权利要求 22 所述的方法，其特征在于：步骤 (a) 包含：

(a1) 允许多个纳米尺度部件自组装成为一个随机阵列；

20 (a2) 允许多个分子电路部件自组装成为在该纳米尺度部件之间互连的随机分子；以及

(a3) 利用分子接线夹把该分子电路部件接合到纳米尺度部件。

25 40. 如权利要求 39 所述的方法，其特征在于：分子接线夹选自硫、氧、硒、磷、异腈、吡啶、羧化物、以及硫醇分子部分。

41. 如权利要求 39 所述的方法，其特征在于：该纳米尺度部件选自纳米管、纳米颗粒、纳米棒、以及它们的组合。

30 42. 如权利要求 39 所述的方法，其特征在于：该分子电路部件

选自分子开关、分子二极管、分子线路、分子整流器、分子电阻、分子晶体管、以及它们的组合。

43. 一个分子计算机，包含：
- 5 多个可编程纳米单元，每个纳米单元包含：
- 多个纳米颗粒；以及
- 多个分子二极管；
- 其中所述分子二极管互连所述纳米颗粒；以及
- 多个金属线路；
- 10 其中所述金属线路互连所述纳米单元。
44. 如权利要求 43 所述的分子计算机，其特征在于：所述纳米单元是自组装的。
- 15 45. 如权利要求 43 所述的分子计算机，其特征在于：所述纳米颗粒是随机排列的。
46. 如权利要求 43 所述的分子计算机，其特征在于：所述分子二极管随机地互连所述纳米颗粒。
- 20 47. 如权利要求 43 所述的分子计算机，其特征在于：每个所述纳米单元包含最多大约 2 微米的一个线性尺寸。
48. 如权利要求 43 所述的分子计算机，其特征在于：所述纳米单元的至少一个可编程以起一个逻辑单元的作用。
- 25 49. 如权利要求 48 所述的分子计算机，其特征在于：所述逻辑单元选自从包含由所述线路互连支持的真值表的组。
- 30 50. 如权利要求 49 所述的方法，其特征在于：所述纳米单元中

的至少一个可编程为用作选自 AND、OR、XOR、NOR、NAND、NOT、加法器、半加法器、反相半加法器、多路复用器、解码器、以及它们的组合的装置。

5 51. 如权利要求 43 所述的分子计算机，其特征在于：所述纳米单元中的至少一个可编程以起一个存储器单元的作用。

10 52. 如权利要求 43 所述的分子计算机，其特征在于：所述纳米单元可通过一种用于配置所述纳米单元的分子二极管的算法进行编程。

53. 如权利要求 43 所述的分子计算机，其特征在于：所述纳米单元进一步包含：

第一和第二引线；以及

15 其中所述二极管可通过向所述第一和第二引线施加电压来配置。

54. 如权利要求 43 所述的分子计算机，其特征在于：所述分子二极管中的至少一个呈现负微分电阻。

20 55. 一种制造计算机的方法，包含：

(a) 提供多个已训练的自组装纳米单元；

(b) 把所述已训练的纳米单元互连到多个未经训练的纳米单元；

(c) 允许已训练的纳米单元训练所述未经训练的纳米单元。

25 56. 依据权利要求 53 所述的方法，进一步包含：

(d) 多级重复步骤 (b) 和 (c)。

可编程分子装置

5 有关联邦政府资助研究或者开发的声明

这项工作通过海军研究所得到了来自 DARPA 的基金支持，拨款号为 R13160。

对光盘附录的说明和在 37 C.F.R § 1.52(e)(5)下的声明

10 依据 37 C.F.R § 1.52(e)(5)，以双重拷贝（“Copy 1”和“Copy 2”）形式附有一个与 MS - Windows 和 MS - DOS 兼容的 IBM - PC 格式的光盘一只读存储器（CD - ROM），并且被包含在此作为参考。Copy 1 和 Copy 2 是相同的，并且在 1 个主目录和 2 个子目录中包含有 269 个文件，如根据 MS - DOS 命令“dir e : / s”的下列输出标识的那样，其中输出对每个文件包含有[月 / 日 / 年 时间、字节、文件名.扩展名]
15 标准格式的一行，以向本领域普通技术人员标识每个文件的创建日期、大小、名称、和类型：

Volume in drive E is PatentFiles

20

Directory of E:\

25

01/01/01	12:00a	<DIR>	.
01/01/01	12:00a	<DIR>	..
07/24/01	01:59p	<DIR>	Dynamic Nanocell Simulator
07/24/01	09:34a		250,368 Exhaustive Truth Table Tests.ppt
07/24/01	01:58p	<DIR>	Spice Nanocell Simulator
07/24/01	01:30p		44,553,216 Trained Nanocell.doc
		6 File(s)	44,803,584 bytes

30

Directory of E:\Dynamic Nanocell Simulator

35

07/24/01	09:34a	<DIR>	.
07/24/01	09:34a	<DIR>	..
07/24/01	01:59p	<DIR>	molec
07/23/01	09:43a		2,341 SimCellV1.jcp
07/10/01	12:09p		390 SimCellV1.jcw
07/11/01	12:42p		239 SimCellV12.jcw
07/24/01	01:59p	<DIR>	src

```

07/18/01  07:53a          1,022 src_simcellv1.txt
              8 File(s)  3,992 bytes

Directory of E:\Dynamic Nanocell Simulator\molec

5
07/24/01  01:58p          <DIR>      .
07/24/01  01:58p          <DIR>      ..
07/24/01  01:59p          <DIR>      cell
07/24/01  01:59p          <DIR>      control
10
07/24/01  01:59p          <DIR>      ga
07/24/01  01:59p          <DIR>      table
07/24/01  01:59p          <DIR>      ui
              7 File(s)  0 bytes

15
Directory of E:\Dynamic Nanocell Simulator\molec\cell

04/16/01  01:07p          <DIR>      .
04/16/01  01:07p          <DIR>      ..
07/18/01  07:54a          490 Link.class
20
07/18/01  07:54a          9,255 Nanocell.class
07/18/01  07:54a          1,206 Node.class
07/18/01  07:54a          1,503 Util.class
              6 File(s) 12,454 bytes

25
Directory of E:\Dynamic Nanocell Simulator\molec\control

07/18/01  07:54a          <DIR>      .
07/18/01  07:54a          <DIR>      ..
07/18/01  07:54a          2,372 nanoControl$1.class
30
07/18/01  07:54a          2,060 nanoControl$10.class
07/18/01  07:54a          1,942 nanoControl$11.class
07/18/01  07:54a          745 nanoControl$12.class
07/18/01  07:54a          839 nanoControl$13.class
07/18/01  07:54a          745 nanoControl$14.class
35
07/18/01  07:54a          1,426 nanoControl$15.class
07/18/01  07:54a          835 nanoControl$16.class
07/18/01  07:54a          751 nanoControl$17.class
07/18/01  07:54a          2,083 nanoControl$18.class
07/18/01  07:54a          751 nanoControl$19.class
40
07/18/01  07:54a          1,251 nanoControl$2.class
07/18/01  07:54a          1,560 nanoControl$20.class
07/18/01  07:54a          1,383 nanoControl$21.class
07/18/01  07:54a          1,054 nanoControl$22.class
07/18/01  07:54a          749 nanoControl$23.class
45
07/18/01  07:54a          846 nanoControl$24.class
07/18/01  07:54a          749 nanoControl$25.class
07/18/01  07:54a          2,672 nanoControl$26.class
07/18/01  07:54a          1,251 nanoControl$3.class
07/18/01  07:54a          1,213 nanoControl$4.class
50
07/18/01  07:54a          1,233 nanoControl$5.class

```

```

07/18/01  07:54a      1,278 nanoControl$6.class
07/18/01  07:54a      744 nanoControl$7.class
07/18/01  07:54a     1,290 nanoControl$8.class
07/18/01  07:54a      744 nanoControl$9.class
5  07/18/01  07:54a     5,091 nanoControl.class

```

29 File(s) 37,657 bytes

Directory of E:\Dynamic Nanocell Simulator\molec\ga

```

10 07/18/01  07:54a      <DIR>      .
   07/18/01  07:54a      <DIR>      ..
   07/18/01  07:54a      158 Evaluatable.class
   07/18/01  07:54a     4,057 GA.class
   07/18/01  07:54a     2,176 Individual.class
15 07/18/01  07:54a     3,132 Population.class
   07/18/01  07:54a     1,020 Util.class

```

7 File(s) 10,543 bytes

Directory of E:\Dynamic Nanocell Simulator\molec\table

```

20 07/18/01  07:54a      <DIR>      .
   07/18/01  07:54a      <DIR>      ..
   07/18/01  07:54a     2,128 TruthTable.class

```

3 File(s) 2,128 bytes

25

Directory of E:\Dynamic Nanocell Simulator\molec\ui

```

   07/18/01  07:54a      <DIR>      .
   07/18/01  07:54a      <DIR>      ..
30 07/18/01  07:54a     6,747 GCell.class
   07/24/01  01:59p      <DIR>      inputWindows
   07/18/01  07:54a     487 mainMenu$1.class
   07/18/01  07:54a     5,985 mainMenu.class
   07/18/01  07:54a     427 NodeCheckBoxMenuItem.class

```

35 7 File(s) 13,646 bytes

Directory of E:\Dynamic Nanocell Simulator\molec\ui\inputWindows

```

   07/24/01  01:59p      <DIR>      .
40 07/24/01  01:59p      <DIR>      ..
   07/18/01  07:54a     2,517 AInputWindow.class
   07/18/01  07:54a     656 GAPropsWin.class
   07/18/01  07:54a     1,068 makeNano.class
   07/18/01  07:54a     2,775 PropsGui.class
45 07/18/01  07:54a     701 SimPropsWin.class
   07/18/01  07:54a     3,006 TableGui.class

```

8 File(s) 10,723 bytes

Directory of E:\Dynamic Nanocell Simulator\src

50

```

07/24/01 01:59p      <DIR>      .
07/24/01 01:59p      <DIR>      ..
07/24/01 01:59p      <DIR>      cell
07/24/01 01:59p      <DIR>      control
5  07/24/01 01:59p      <DIR>      ga
07/24/01 01:59p      <DIR>      table
07/24/01 01:59p      <DIR>      ui
                                7 File(s)      0 bytes

10      Directory of E:\Dynamic Nanocell Simulator\src\cell

07/24/01 01:59p      <DIR>      .
07/24/01 01:59p      <DIR>      ..
05/11/01 02:33p      516 Link.java
15 07/17/01 01:32p      30,722 Nanocell.java
07/09/01 11:18a      1,230 Node.java
04/04/01 09:37a      1,312 Util.java
                                6 File(s)      33,780 bytes

20      Directory of E:\Dynamic Nanocell Simulator\src\control

04/04/01 09:37a      <DIR>      .
04/04/01 09:37a      <DIR>      ..
07/18/01 03:45p      20,315 nanoControl.java
25                                3 File(s)      20,315 bytes

      Directory of E:\Dynamic Nanocell Simulator\src\ga

07/18/01 03:45p      <DIR>      .
30 07/18/01 03:45p      <DIR>      ..
04/17/01 06:59p      87 Evaluatable.java
07/14/01 06:40a      6,716 GA.java
07/14/01 06:16a      3,268 Individual.java
06/11/01 10:32a      4,467 Population.java
35 06/16/01 06:00a      854 Util.java
                                7 File(s)      15,392 bytes

      Directory of E:\Dynamic Nanocell Simulator\src\table

40 07/14/01 06:40a      <DIR>      .
07/14/01 06:40a      <DIR>      ..
04/17/01 06:56p      2,074 TruthTable.java
                                3 File(s)      2,074 bytes

45      Directory of E:\Dynamic Nanocell Simulator\src\ui

04/17/01 06:56p      <DIR>      .
04/17/01 06:56p      <DIR>      ..
07/14/01 05:39a      9,636 GCell.java
50 07/24/01 01:59p      <DIR>      images

```

	07/24/01	01:59p	<DIR>	inputWindows
	05/23/01	01:49p		9,295 mainMenu.java
	04/06/01	09:15a		327 NodeCheckBoxMenuItem.java
			7 File(s)	19,258 bytes
5				
	Directory of E:\Dynamic Nanocell Simulator\src\ui\images			
	07/18/01	07:54a	<DIR>	.
	07/18/01	07:54a	<DIR>	..
10	02/05/01	05:07p		291 cell_prop.gif
	02/05/01	04:50p		1,087 chip.gif
	06/02/00	11:11a		885 clone.gif
	06/02/00	11:11a		81 hand.gif
	06/02/00	11:11a		1,476 looping.gif
15	02/05/01	02:45p		235 middle.gif
	02/05/01	05:02p		843 molecule.gif
	06/02/00	11:11a		139 new.gif
	06/02/00	11:11a		146 open.gif
	06/02/00	11:11a		184 save.gif
20	06/02/00	11:11a		852 start.gif
	06/02/00	11:11a		858 stop.gif
	02/05/01	04:51p		1,422 truth.gif
	02/05/01	05:05p		211 zap.gif
			16 File(s)	8,710 bytes
25				
	Directory of E:\Dynamic Nanocell Simulator\src\ui\inputWindows			
	02/05/01	05:05p	<DIR>	.
	02/05/01	05:05p	<DIR>	..
30	06/05/01	01:31p		2,592 AInputWindow.java
	07/13/01	10:01p		522 GAPropsWin.java
	07/03/01	02:31p		691 makeNano.java
	07/03/01	02:35p		2,007 PropsGui.java
	07/13/01	03:36p		526 SimPropsWin.java
35	02/28/01	01:04p		2,744 TableGui.java
			8 File(s)	9,082 bytes
	Directory of E:\Spice Nanocell Simulator			
40	07/24/01	01:59p	<DIR>	.
	07/24/01	01:59p	<DIR>	..
	03/26/01	04:24p		92 1bit.tt
	03/26/01	04:25p		412 2bit.tt
	07/03/01	01:42p		86 2Nand.pin
45	07/03/01	01:24p		153 2Nand.tt
	07/03/01	02:56p		86 4Nand.pin
	07/03/01	03:56p		86 4Nand.pin.pin
	07/03/01	03:55p		410 4Nand.tt
	07/23/01	03:01p		4,562 AllScores
50	03/26/01	04:25p		36 And.tt

	06/08/01	02:25p	2,272 BigRemoteTest
	02/12/01	10:27a	2,036 Connector.cpp
	05/30/01	12:46p	1,030 Connector.h
	01/25/01	01:35p	1,157 Conprop.cpp
5	01/25/01	01:35p	1,245 Conprop.h
	03/28/01	05:10p	86 D1Bit.pin
	03/26/01	04:29p	93 D1bit.tt
	03/28/01	10:12a	86 D1Bit_1.pin
	03/27/01	03:20p	63 D1Bit_1.tt
10	03/28/01	10:29a	88 D1Bit_2.pin
	03/28/01	10:22a	63 D1Bit_2.tt
	07/24/01	01:58p	<DIR> Debug
	01/25/01	01:35p	1,249 Device.cpp
	01/29/01	02:25p	2,604 Device.h
15	03/14/01	02:16p	0 DNand.pin
	03/14/01	02:06p	42 DNand.tt
	03/16/01	01:04p	85 DXor.pin
	03/14/01	02:12p	36 DXor.tt
	01/25/01	01:35p	1,117 FindDlg.cpp
20	01/25/01	01:35p	1,213 FindDlg.h
	07/18/01	12:07p	91 HalfAdder.pin
	07/18/01	12:06p	59 HalfAdder.tt
	01/25/01	01:35p	8,644 icaps.cpp
	02/06/01	06:45p	2,169 icaps.h
25	07/20/01	04:52p	1 intusoft.err
	03/12/01	11:08a	137 Inverter.pin
	03/26/01	04:26p	33 Inverter.tt
	05/10/01	11:38a	89 Inverter2.pin
	03/24/01	04:09p	89 InverterLess.pin
30	06/15/01	12:42p	13,570 MainFrm.cpp
	06/15/01	12:30p	2,602 MainFrm.h
	06/15/01	12:40p	78,956 mc6.aps
	07/24/01	01:25p	16,911 mc6.clw
	02/06/01	04:28p	6,218 mc6.cpp
35	03/28/01	12:24p	7,062 mc6.dsp
	02/12/01	10:33a	711 mc6.dsw
	02/03/01	01:58p	1,432 mc6.h
	07/24/01	01:25p	484,352 mc6.ncb
	06/15/01	12:30p	1,001 mc6.odl
40	07/24/01	01:25p	61,952 mc6.opt
	07/23/01	02:10p	2,748 mc6.plg
	06/15/01	12:40p	49,154 mc6.rc
	01/18/01	04:32p	660 mc6.reg
	07/20/01	01:14p	26,643 mc6Doc.cpp
45	06/15/01	12:42p	4,078 mc6Doc.h
	02/07/01	04:21p	3,033 mc6View.cpp
	02/06/01	07:44p	1,841 mc6View.h
	01/25/01	01:35p	949 Molecule.cpp
	01/25/01	01:35p	668 Molecule.h
50	07/20/01	04:51p	1,329 Moletronic.cir

	07/20/01	04:52p	1,382 Moletronic.ckt
	07/20/01	04:52p	632 moletronic.err
	07/20/01	04:52p	3,869 moletronic.out
	07/20/01	04:52p	1,330 Moletronic.prm
5	07/20/01	04:52p	1,329 Moletronic.tmp
	03/27/01	11:04a	41,480 Moletronic2
	03/28/01	12:24p	142 mssccprj.scc
	06/08/01	02:25p	81 NAdder.pin
	05/07/01	02:38p	87 NAdder.tt
10	05/11/01	03:42p	85 Nand.pin
	03/26/01	04:26p	37 Nand.tt
	03/26/01	02:18p	85 NandLess.pin
	07/19/01	03:33p	76,429 Nanocell.cpp
	07/23/01	02:10p	5,784 Nanocell.h
15	01/25/01	01:35p	2,717 Nanocolor.cpp
	01/25/01	01:35p	1,530 Nanocolor.h
	04/16/01	01:08p	1,912 Nanoprop.cpp
	04/16/01	01:08p	1,452 Nanoprop.h
	03/26/01	04:16p	15,595 NewPinStatus.cpp
20	03/12/01	11:49a	3,125 NewPinStatus.h
	03/12/01	03:27p	43,329 NewTT.cpp
	03/12/01	02:59p	7,659 NewTT.h
	04/20/01	05:23p	81 NHalfAdder.pin
	06/15/01	04:26p	81 NHalfAdder.pin.pin
25	04/20/01	01:32p	59 NHalfAdder.tt
	04/23/01	11:54a	81 NHalfAdderBig.pin
	03/26/01	04:26p	35 Or.tt
	07/18/01	12:50p	1,090 papertest
	07/18/01	02:35p	331 papertest2
30	04/17/01	02:40p	10,311 Pin.cpp
	03/28/01	09:59a	920 Pin.h
	03/26/01	04:16p	4,562 Pinprop.cpp
	06/08/01	10:55a	1,567 Pinprop.h
	03/26/01	04:17p	2,861 PinStatus.cpp
35	03/15/01	02:59p	1,886 PinStatus.h
	01/18/01	04:32p	4,650 readme.txt
	06/25/01	12:54p	410 RemoteTest
	06/21/01	04:04p	2,220 RemoteTestBig
	07/24/01	01:58p	<DIR> res
40	06/15/01	12:40p	13,766 resource.h
	01/21/01	02:56p	515 Results.tmp
	07/20/01	04:50p	133 Scores
	07/23/01	03:01p	616 ScoresSmall
	05/24/01	11:49a	3,503 Simprop.cpp
45	05/24/01	11:44a	1,947 Simprop.h
	01/25/01	01:35p	4,971 spice4.cpp
	01/25/01	01:35p	1,412 spice4.h
	05/28/01	10:58a	1,419 SpiceSocket.cpp
	06/18/01	02:41p	1,358 SpiceSocket.h
50	01/25/01	01:35p	205 StdAfx.cpp

```

03/12/01 12:57p      1,639 StdAfx.h
07/20/01 12:47p      8,074 Switch.cpp
07/19/01 03:03p      1,241 Switch.h
03/14/01 02:10p      1,106 Truthtable.cpp
5  03/12/01 03:24p      1,494 Truthtable.h
03/28/01 01:46p      912 vssver.scc
04/03/01 12:08p      81 Xor.pin
03/26/01 04:27p      36 Xor.tt
07/19/01 05:57p      385 xortest
10      120 File(s)      1,083,699 bytes

```

Directory of E:\Spice Nanocell Simulator\Debug

```

15 07/24/01 01:59p      <DIR>      .
07/24/01 01:59p      <DIR>      ..
07/05/01 11:27a      0 AllScores
07/23/01 02:10p      5,161,984 mc6.bsc
      4 File(s)      5,161,984 bytes

```

20 Directory of E:\Spice Nanocell Simulator\res

```

07/05/01 11:27a      <DIR>      .
07/05/01 11:27a      <DIR>      ..
01/17/01 08:59p      1,078 mc6.ico
25 01/13/01 04:31p      395 mc6.rc2
01/13/01 04:31p      1,078 mc6Doc.ico
03/09/01 02:57p      778 Toolbar.bmp
04/16/01 01:07p      96 vssver.scc
      7 File(s)      3,425 bytes

```

```

30      Total Files Listed:
      269 File(s)      51,252,446 bytes
      0 bytes free

```

35 发明领域

本发明一般地说涉及可编程电子装置，更具体地说涉及基于分子电路部件的可编程纳米级装置。

发明背景

40 一个计算机的基本功能包含信息处理和存储。在冯诺依曼（串行）体系结构中，那些算术、逻辑、和存储器操作由能够可逆地在通常被称为“0”和“1”的两种状态之间切换的装置执行。执行这些不同功能的半导体装置必须能够使用最低量的电能以超高速在两个状态

之间进行切换，以便允许计算机执行基本操作。晶体管在计算机中执行基本切换功能。

5 虽然高能效、最新电子装置的设计和生產越来越依赖于在集成电路内制作更高密度电路元件的能力，但基于半导体的计算机技术和体系结构已经发展到近于这种结构的量子力学极限。很快，尺寸和价格将限制高性能计算机未来增长的发展。调节高性能计算机的这些属性的一个主要部件是存储器，尤其是存储器电路密度。因为这些仪器的巨大数据存储要求，所以需要一种新的、紧凑的、廉价的、极高容量的、高速存储器电路结构。与缩小电子装置规模有关的问题的更详细讨论见于美国专利 6,259,277、6,219,833、5,589,692 和 5,475,341，它们中每一个都在此引入作为参考。

15 分子尺度电子装置是这样一個学科领域，其提出使用单个分子或者一组分子用作将来计算装置中的关键部件。尤其是，可受控设置电荷壁垒的分子能够用作开关。除了显著地减小尺寸之外，分子装置的响应时间能够在飞秒的范围之内，而当前最快速的装置在纳秒的范围中工作。尤其是如果其它电路元件没有限制工作性能的话，则因此在速度方面可以得到 10^5 到 10^6 的增加。

20 优化传统基本单元（通常是晶体管）的尺寸和它们的速度（受它们的自然时间响应所限制）是相冲突的设计目标。因此必须进行几个平衡。在计算技术中最重要的折中是硬件软件对偶性，其物化为编程逻辑（存储器-或者软件-占优势的）对布线逻辑（CPU -、或者硬件主导）的要求。编程逻辑的部件比布线逻辑系统小而且能处理更大的问题；然而布线逻辑要比编程逻辑快。在一个极端，能够有一个具有少量逻辑门的位加法器（能够求和的最小逻辑单元），其需要一个大的存储器来获得结果，而在另一个极端，能够有一个具有所有特定功能都布线到系统中的 CPU，其将能够处理整个问题，并具有仅仅一个小的存储器用于输入输出数据。当前技术非常倾向于编程逻辑，例如，

25

30

具有一个大存储器和一个快速但是简单 CPU 的计算机。

5 实现分子尺度电子器件所正在面临的一个挑战是用于寻找将分子
部件排列成为具有逻辑功能结构的方法。因此，已经有在使分子部件
用作在构造逻辑装置中的基本开关单元的体系结构中的研究。任何逻辑
10 门都可以由一整套的一个或多个基本门来构造。一个以上的这些基
本门可以串联或者并联、或者两者都有地进行排列，以形成其它逻辑
功能。因此，显示基本门的功能就变得特别重要。NAND 门是一个本
身形成一个完全集的基本门。NOR 门是另一个本身形成一个完全集的
15 基本门。其它完全集包含 AND 门和 XOR 门的组合、OR 门和 XOR 门
的组合、AND 门和 NOT（也称为反相器）门的组合、以及 OR 门和
NOT 门的组合。

在一种方法中，基本的逻辑功能已经提出使用由键合在一起的较
15 小分子构成单分子。每个较小的分子可被设计成模仿一个传统电路元
件的功能。在由 James C. Ellenbogen 和 J. Christopher Love 所著、2000
年三月的 *Proceedings of the IEEE*，第 386 - 426 页的图 12、13、和 14
中显示了这样的抽象分子。这篇文章在此引入作为参考。该文献的
图 12、13、和 14 中显示的分子被建议分别用作 AND 门、OR 门、和
20 半加法器。这个方法的一个缺点在于合成这种提议的分子有困难。此
外，这些分子段的动态构造变化易于在这些分子段之间产生短路。

在另一种方法中，传统电路元件和开关的混合阵列呈现出基本逻辑
功能，其中该混合阵列在引线之间包含有单分子层的数百万个分子
25 二极管。定位于两个传统金属板比如电容器平板之间的单分子层分子
二极管装置已呈现出开关功能。单分子层是具有一个分子厚度的分子
层。在该单分子层中，具有带有官能团的相反端的分子被并排定位，
该官能团可以键合到金属并且渐被称为分子接线夹。该功能端和金属
平板键合。在由 David P. Nackashi 和 Paul D. Franzon 所著、*Proc, SPIE*
30 2001，卷 4236、页 80 - 88 中，标题为 “Moletronics : A circuit design

perspective”的文章的图 5 中画出了包括基于单分子层的开关装置的示例电路，其具有 NAND 和 NOR 功能。这篇文章在此全文引入作为参考。此外，在代理人卷号为 OCR 1049、于 2000 年 4 月 18 日提出的、名为“Molecular Scale Electronic Devices”的美国专利申请中也描述了包括定向单分子层的电路，其在此引入作为参考。

在每一种上述方法中，分子尺度装置都是布线逻辑的实施方式。这和在当前朝着编程逻辑方向的技术趋势相反。此外，布线逻辑趋向于比编程逻辑更不能容忍缺陷。因为分子尺度装置的工业规模制造是成本有效和有效率的，所以该装置必须能容忍可能在化学组装该装置的过程中出现的缺陷。

分子尺度电子装置提供了使我们的电流变小的可能计算电源。因此一种用于以有效、坚固、和可再生的方式从分子部件中创建编程逻辑的技术是所期望的。

发明概述

在一个优选实施例中，本发明特征在于一种使用分子部件的编程逻辑。替代地，本发明提供了一种使用分子部件的编程存储器。在一个形成微小可编程单元的纳米单元中排列这些分子部件。一个纳米单元优选包含多至数万亿个分子，其中数千个分子处于适当取向用于开关。这在期望由单个分子逻辑实现的微型化和期望坚固、可编程的功能之间提供了一个在尺度方面的平衡。本发明中的纳米单元具有这样的优点，即通过直接的湿化学技术组合的单个纳米单元可能首先被编程以作为一个逻辑单元执行，然后可选择性地重编程用作另一个逻辑单元。此外，该纳米单元适合于装入到标准计算机中来代替传统的逻辑单元，同时在一个比当前在传统硅基逻辑单元中可实现的更小尺度上提供了类似的功能。

30 通过从允许自组装到一个结构中的分子部件中构造该纳米单元来

5 实现当前纳米单元的多用性、坚固性、和容易生产。除非由架子引导，否则分子部件会组合成为一个随机结构，比如一个随机网络。因为该网络优选在从约 1 nm 到约 2 μ m 的尺度上变化，所以它在此被称为一个纳米网络。该随机结构具有这样的好处，即如果在一个特定位置缺乏一个特定分子部件，则这对于该纳米单元的功能几乎没有影响。也就是说，无论该分子部件的精确结构如何，该纳米单元也是可编程的。该纳米单元可由一种被称为自适应算法的迭代方法编程，其中该算法根据分子部件的结构加以调整。

10 因此，本发明包含这些特征和优点的组合，使其能够克服现有技术装置的各个问题。依据本发明优选实施例的以下详细说明，以及参见附图，本领域的技术人员将很容易地理解如上所述的各个特征，以及其它特色。

15 附图简要说明

对本发明优选实施例的更详细说明将参照附图给出，其中：

图 1 是依据本发明一个实施例的一个纳米单元的示意图；

图 2A 和 2B 是依据本发明一个实施例的引线结构的一个示意图；

图 3 是依据本发明一个实施例的分子部件的示意图；

20 图 4A 和 4B 显示了在图 3 中描述的分子的 I (V) 响应图；

图 5 是依据本发明一个示例实施例的一个分子计算机的示意图；

图 6 是含有吡啶基团作为“接线夹”的分子装置的示意图；

图 7 是依据本发明一个例示性实施例的模拟纳米单元的示意图，其中“on”高导通分子显示为黑线，以及“off”低导通分子显示为白线；

25 图 8 是图 7 中的模拟纳米单元的示意图，其被编程以起一个反相门的作用；

图 9 是图 7 中的模拟纳米单元的示意图，其被重新编程以起一个 NAND 门的作用；以及

30 图 10 是图 7 中的模拟纳米单元的示意图，其被重新编程以起反

相半加法器门的作用。

优选实施例的详细说明

纳米单元

5 首先参见图 1，一个分子电子装置 10 包含一个纳米单元 12。纳米单元 12 包含至少一个以及优选多个分子电路部件 14。纳米单元 12 优选具有一个最多大约 $2\mu\text{m}$ 、更优选在大约 1nm 和大约 $2\mu\text{m}$ 之间的线性尺寸 16。线性尺寸 16 可以是纳米单元 12 的边 18 的长度。边 18 封闭分子电路部件 14，也就是说限定包含分子电路部件 14 的边界。

10 纳米单元 12 可以包含任意数量的侧边，并且可以是一维到三维。纳米单元 12 在图 1 中被画成一个方形结构。应理解可以考虑替换的结构，比如圆形、矩形、以及任何其它适当的结构。

 仍然参见图 1，纳米单元 12 优选进一步包含至少一个输入引线 20

15 和至少一个输出引线 22。输入引线和输出引线的数量无关紧要。引线数目优选仅仅受用于形成引线 20、22 的技术比如传统的光刻技术，以及纳米单元 20 的尺寸所约束。引线 20、22 在图 1 中显示在纳米单元 12 的边缘。应理解可以考虑引线的其它配置。例如从纳米单元 27 边缘延伸的输入引线 23 和输出引线 25 可以交叉，比如在图 2A 中显示的那样。做为选择，输入引线 29 和输出引线 33 可以从限定纳米单元 39 边缘的同心圆周 37 中向外延伸，如图 2B 所示的那样。

20

 纳米网络 20 优选覆盖每个输入引线 20 和每个输出引线 22。引线 20、22 可以是金属的而且被设计成连接到传统的平版印刷互连装置，比如金属线路。边缘分子电路部件 24 通过分子接线夹 26 连接到

25 引线 20、22。分子接线夹包含基于比如硫、氧、硒、磷、异腈、吡啶、以及羧化物的部分而与金属键合的粘性端基团。一个特别优选的基于硫的分子接线夹是硫醇基。应理解：分子电路部件 14 可以包含两个、三个、四个、五个、六个或更多末端，比如在 Tour, J. M.; Kozaki, M.; 和 Seminario, J. M. 所著、J. Am. Chem. Soc. 120, 页 8486 - 8493 (1998)

30

的“Molecular Scale Electronics : A Synthetic / Computational Approach to Digital Computing”中公开的那样，其在此引入作为参考，以及在美国专利 6,259,277 中描述的那样，其也在此引入作为参考。每个末端优选一个包含分子接线夹的端部。

5

仍然参见图 1，纳米单元 12 优选一个具有这样一个网络结构的纳米网络，其中该分子电路部件形成该网络的至少一部分或者多个部分。纳米网络 28 优选一个随机纳米网络。尤其是，纳米网络 28 优选具有以下随机单元中的至少一个。纳米网络 28 的 X 射线晶体结构可以不包含指示分子电路部件的一个周期或者半周期结构的显著峰值，更优选在大约 1 nm 和 2 μ m 之间的长度尺度上。做为选择，该纳米网络 28 的 X 射线晶体结构可以包含至少一个指示缺少大约 1 nm 和 2 μ m 之间特征长度尺度的峰值。还做为选择，纳米网络 28 可以具有一个呈现缩放行为、多缩放行为、分形特征等的结构。还做为选择，纳米网络 28 可以具有一个包含分子电路部件 14 相对于沿着一个已知随机分布的任意轴定向的结构，该已知的随机分布比如是在该网络中的纳米颗粒之间几个分子的泊松分布。仍然还做为选择，纳米网络 28 可以具有一个包含遵循一个已知随机分布比如是非晶或者非定形固体特征的分子电路部件 14 的质量中心位置的结构。应理解，在此使用的术语“随机”可以包含任何其它传统的定义而且可以和术语“不规则”和“无规律”可交换地使用。此外，应理解：对于一定的长度尺度，会发生随机性。尤其是，此处的术语随机网络包含具有较少的长程次序的网络。长程表示相对于组成网络的部件的长度尺度，距离是长的。分子电子装置 10 中分子电路部件 14 的随机排列具有这样的优点，即装置 10 可以是容错的。

仍然参见图 1，在一个优选实施例中，纳米网络 28 是自组装的。如在本领域中所知的那样，自组装网络是一个响应于激励比如反应条件的变化而从其组成部件中创建它本身的网络。一个自组装的纳米网络优选具有非预先决定的结构。此外，在这个实施例中的一个自组装

纳米网络优选在相邻纳米颗粒之间仅仅具有短程的次序，而且对于较长的长度尺度，优选是无序的。

5 适于在本发明中使用的纳米网络包含但不限于如下面所述构造的纳米网络。金属纳米颗粒沉积在一个氧化物网格上。该氧化物网格可以是一个半导体衬底，其中去除材料以限定一个提供该纳米网络边界的孔隙。一个覆盖各纳米颗粒的分子自组装单分子层可以用来控制在纳米颗粒之间的间距。经由先前所述方法，分子开关被插入到围绕各纳米颗粒的惰性自组装单分子层的壁垒中，并且由此将相邻的纳米
10 颗粒互连。此方法已经在 Dunbar T. D.; Cygan, M. T.; Bumm, L. A.; McCarty, G. S.; Burgin, T. P.; Reinerth, W. A.; Jones, II, L. Jackiw, J. J.; Tour, J. M.; Weiss, P. S.; Allara, D. L.所著、J. Phys. Chem. B.的 2000, 104 的页 4880 - 4893 中公开了，其在此引入作为参考。

15 仍然参见图 1，考虑可训练的而且包含任何适当的传统分子电路部件的纳米网络 28。因此，可以从分子线路、分子整流器、分子二极管、分子开关、分子电阻、分子晶体管等等以及它们的组合中选择分子电路部件 14。分子线路、整流器、二极管、开关、电阻、或者晶体管是任何能够用在电路中分别类似于传统的线路、整流器、二极管、
20 开关、电阻、或者晶体管的分子。示例的分子线路包含低聚（亚苯基亚乙炔基）等。示例分子整流器包含 hexadecylquinolinium tricyanoquinodimethanide、等等。

25 仍然参见图 1，分子电路元件 14 优选包含共轭的分子段。这些共轭分子段优选用在起分子接线夹作用的末端处的基团代替。用作分子电路元件的共轭分子段的示例共轭分子，以及用作分子接线夹功能的示例共轭分子，在 Tour, J. M.所著、Accounts of Chemical Research, 卷 33、号 11、第 791 - 804（2000）的“Molecular Electronics. Synthesis and Testing of Components”中；以及 Seminario, J. M.所著 J. Am. Chem.
30 Soc.的卷 120, 8486 - 8493（1998）页的“Molecular Scale Electronics:

- A Synthetic/Computational Approach to Digital Computing”中；Dirk, S. M.,等人所著 Tetrahedron 57, 页 5109 - 5121 (2001) 的 “Accoutrements of a molecular computer: switches, memory components and alligator clips” 中都进行了描述，这些文章每个都在此引入作为参考。此外，
- 5 分子电路元件 14 可以包含在美国专利申请_____、代理人卷号为 OCR 1049、于 2000 年 4 月 18 日提出、名称为 “Molecular Scale Electronic Devices” 中的公开的分子、导电有机材料、或者导电通路中的任何一个，该申请在此引入作为参考。
- 10 分子电路元件 14 优选一个呈现负微分电阻的分子。传统的谐振隧道二极管也呈现负微分电阻。然而，传统的谐振隧道二极管是基于砷化镓的。负微分电阻在设计逻辑中是一个特别有用的属性，因为它允许非。
- 15 现在参见图 3，分子电路元件 14 可以是一个分子二极管 30。示例的分子二极管包含一个单硝基取代的低聚亚苯基 32，尤其是 4,4'-二亚苯基亚乙炔基-2'-硝基-1-苯硫酚和二硝基取代的低聚亚苯基 34，尤其是 2',5'-二硝基-4,4'-二亚苯基亚乙炔基-1-苯硫酚。
- 20 替代的分子二极管包含分子 32 和 34 的二硫醇取代的类似物、尤其是分别为 4,4'-二亚苯基亚乙炔基-2'-硝基-1,4"-苯二硫酚和 2',5'-二硝基-4,4'-二亚苯基亚乙炔基-1,4"-苯二硫酚。这些分子中的每一个都在每个端部包含一个硫醇基。这种结构优选用于在各端部与金接合的分子电路元件 14。在此所述术语分子开关也包含这些分子，当它们在电
- 25 环境中时，允许它们起一个开关的作用。可以通过增加或者改变取代基、通过使另一个分子和分子二极管键合、或者通过把该分子二极管比如通过分子接线夹连接到一个电路元件来创建该电环境。
- 30 纳米单元 12 还可以包含纳米尺度的部件 40。纳米尺度部件 40 优选作为纳米网络 28 的部分而排列。纳米尺度部件可以具有电子连

接器的功能，以帮助把分子部件 14 构造成一个导电网络。此外，纳米尺度部件可以具有电子电路部件比如电导、电容、电阻、阻抗等的功能。示例的纳米尺度部件包含纳米管、纳米颗粒、纳米棒，以及它们的组合。纳米颗粒可以是金属的、半导体的、介电的等等。示例

5 纳米颗粒和纳米管在 Reed, M.A.和 Tour, J.M.所著、Scientific American 282 页 86 - 93 (2000) 中进行了描述，其在此引入作为参考。示例纳米棒在 Martin, B.R.等人所著、Adv. Mater. 11, 1021 - 1025 (1999) 页的文章 “Orthogonal self - assemble on colliodal gold - platinum nanorods” 中进行了描述，其在此引入作为参考。

10

应理解，在图 1 中由一条线表示一个分子电路部件 14 的地方，可以用多个分子电路部件 14 来代替。例如，多个分子电路部件 14 可以连接一对纳米尺度部件 40 中的每一个，从而横跨纳米尺度部件。

15

仍然参见图 1，在一个示例排列中，一个纳米单元 10 包含分子开关 52 和纳米颗粒 54。纳米颗粒 54 优选是金属的、更优选为金。分子开关 52 优选为具有硫醇分子接线夹在每个端部的开关，更优选为 2',5'-二硝基-4,4'-二亚苯基亚乙炔基-1,4"-苯二硫酚。边缘分子开关 56 连接到输入引线 20 和输出引线 22。分子开关 52 互连纳米颗粒 54。

20 此处使用的互连具有使可以电连接的意思。在这种意义上，换一个角度说，纳米颗粒 54 互连分子开关 52。此外，由一个分子开关 52 提供的电连接不需要是永久的，而且能够通过配置分子开关 54 来进行中断。

25

纳米网络 28 优选由分子开关 52 和纳米颗粒 54 形成。尤其是，纳米颗粒 54 优选几乎没有次序地进行排列。此外，分子开关 52 互连纳米颗粒 54。不是所有纳米颗粒 54 都连接其它纳米颗粒 54，而是某些纳米颗粒 54 连接到一个以上或两个以上的其它纳米颗粒，而且连接可以是随机分布的。

30

应理解：可以通过改变纳米颗粒 54 中的金属，分子电路部件 14 的共轭全链，用于分子电路部件 14 接线夹的分子部分，引线 20、22 的几何形状，及其它用于调整阻抗的适当属性中的任何一个或者它们的组合，来优化纳米单元 12 的阻抗属性。

5

此外应理解：分子电路部件 14 可以是多态分子，比如三个、四个、五个、或者六个状态的分子。例如， C_{60} 具有通过逐渐地获取六个电子达到的六个独立状态。因此，分子电路部件 14 不局限于二进制“0”和“1”，或者“开（on）”和“关（off）”，而且例如可以

10

现在参见图 5，多个可编程的电子装置 62，优选纳米单元 64，可以由标准平版印刷产生的金属线路互连以形成一个分子计算机 66。纳米单元 64 优选相对于图 1 如上所述构造，更优选例如如图 4 所示构造。可以考虑用于由线路 65 互连的任何传统体系结构。

15

可编程性

再次参见图 1，分子电子装置优选是可编程的。尤其是，分子电子装置 10 优选可用一种自适应算法编程。如在此使用的那样，自适应算法是一种能够使用一个迭代过程“演变”的算法，其中该算法查询并且调整一个系统以便把该系统向一个期望的状态移动。更具体地说，自适应算法是一类包含一组规则用于把系统的实际结果和一个目标结果进行比较，并且根据实际结果和目标结果之差的函数调整该系统的输入的算法。依据该系统的行为，下一个实际结果与调整的输入

20

25

分子装置 10 优选可通过用于构造分子电路部件 14 的自适应算法进行编程。

30

5 在一个优选实施例中，分子电路部件 14 优选可通过在引线 20、22 两端应用一个电压来进行配置。例如，分子电路部件 14 可以包含一个导电率影响的属性可通过把电压加到引线 20、22 两端来进行调整的分子。被调整的、受导电率影响的属性优选选自电荷、构造状态、电子态等，以及它们的组合。

应理解：分子电路部件 14 可以通过其它方法配置。

10 基于低聚亚苯基的分子线路和开关是导电率受电荷、电子态、以及构造状态影响的分子示例。可以相信，把电压加到这些分子的两端能够影响电子态之间的跃迁。电压可以导致分子捕获电子；因此增加它的电荷。进一步，当充电时，该分子跃迁到一个电子激发态。苯基环相对于彼此旋转以便电子轨道比如 π -轨道对准，从而形成一个在该分子长度延伸的分子轨道。当存在一个外加电压时，可以相信通过
15 分子轨道和分子导体可建立电子连接。开关功能的分子机制的说明包含在 Donhauser, Z.J.等人所著、Science 292、2303 - 2307 (2001) 页中，其内容在此引入作为参考。

20 在一个优选排列中，用于使引线连接分子的材料电特性和该分子的电子跃迁能量特性相匹配。尤其是，优选连接一个共轭分子电路元件的金属的费米能量在能量方面接近于分子电路元件中的最低空置分子轨道 (LUMO) 的能量。这具有优化金属和分子之间连接的阻抗特性的优点。

25 分子开关的操作不同于分子线路。通过施加然后撤除一个电压，开关的导电率能够切换到一个稳定相对长时间的状态。参见图 3，至少 24 小时的稳定时间已经利用分子 34 获得。此外，可以预期，提高包含一个分子开关的系统的密封；使用具有多硝基的、基于低聚亚苯基的类似分子；或者使用新类型的分子，将允许更长的稳定时间，比
30 如几天或者几个月。一个优选的分子开关可通过施加一个开关电压配

置，而且通过施加一个小于该开关电压的工作电压在一个高或者低导电率状态中工作。

现在参见图 3，通过分子 34 的操作来举例说明分子开关的操作。
5 当一个超过 2.0V 的开关电压施加于分子上时，分子 34 切换到高导电性状态而且当一个低于 - 2.0V 的相应电压施加于分子 34 上时，其将切换到一个低导电率状态。对于高导电率状态，开关电压优选是在大约 0.2 和 3.0V 之间，而且对于低导电率状态在 - 0.2 和 - 3.0V 之间。
10 在图 4 中，高导电率状态与通过黑点描绘的 I (V) 曲线相关联，而且低导电率状态与由白点描绘的较低 I (V) 曲线相关联。通过这两个曲线之间的区别来确定在该高低导电率状态之间的区别程度。当在大约 - 2 V 和 2V 之间的一个工作电压施加于分子 36 上时，它们依据它们最近切换到的高或者低导电率状态而导通。如果施加了超过该负微分电阻 (NDR) 限制的电压的话，则在该高电导率状态中的分子也呈现低
15 导电率。通过在由黑点描绘的峰值和谷值之间的比值确定由于 NDR 的影响，处于高电导率状态的分子的高低导电率之间的区别程度。操作电压的绝对值优选为在大约 0.2V 和大约 2.0V 之间。

再次参见图 1，纳米单元 12 优选可由一种用于设置分子开关 54
20 的算法编程。分子开关 54 优选可通过把电压加到引线 20、22 的两端来设置。优选地，一种用于编程纳米单元 10 的自适应算法能够得出可施加于引线 20、22 的电压组合，其中引线 20、22 将构造远程分子开关，也就是说不直接连接到引线 20、22 的分子开关。

25 应理解：自适应算法的类型不是关键的。可以使用任何能够训练比如纳米网络 28 的网络的适当传统自适应算法。示例的自适应算法包含遗传算法、模拟退火算法、增强学习算法、时差算法、跟踪优胜算法、等等。在 Goldberg, D.E. 所著、Genetic algorithms in Search, Optimization, and Machine Learning (Addison Wesley, Reading, MA,
30 1989) 的第 1 - 15 页和第 221 - 229 页中描述了自适应算法的原理，其

在此引入作为参考。

5 自适应算法具有从错误中恢复的优点。此外，自适应算法的使用也提供了容错优点。因此，分子电子装置 10 适合于通过能够在一个具有成本有效可靠性的工业规模上实现的自动装配方法来制造。该自适应算法可以在一个辅助计算机中编码。

10 本发明的一个优点是分子电子装置 10 的可编程性，意思指在初次组装时，该装置不必用作具体的逻辑装置。因此，分子电子装置 10、纳米单元 12、和纳米网络 28 不需要具有一个预定的结构。纳米单元 12，尤其是纳米网络 28 可以被自组装成为一个可以是随机未定结构。自适应算法可以用来把装置 10 编程为起一个期望装置的作用。

15 在一个优选实施例中，装置 10 可编程为用作选自 AND、OR、XOR、NOR、NOT、和 NAND 门等的逻辑单元。因此，在这个实施例中，当装置 10 已经被编程了时，它是一个具有选自 AND、OR、XOR、NOR、NOT、NAND 等的逻辑单元的编程逻辑装置。

20 在另一个优选实施例中，装置 10 可编程为用作选自加法器、半加法器、多路复用器、解码器等的逻辑单元。因此，在这个实施例中，当装置 10 已经被编程了时，它是一个具有选自加法器、半加法器、多路复用器、解码器等的逻辑单元的编程逻辑装置。在还有另一个优选实施例中，装置 10 可编程为用作一个存储器单元。

25 应理解：装置 10 优选可以用作任何具有由输入 / 输出管脚支持的真值表的门。

30 装置 10 优选是可重新编程的。尤其是，起初被编程为用作上述逻辑或存储器单元之一的装置 10，能够被重新编程为用作上述逻辑或存储装置中的另一个。因此，装置 10 具有多用性的优点。

上述可编程性优选通过采用上述纳米单元结构的优选拓扑结构并结合如下所述的优选编程方法来实现。

5 编程方法

构造一个电子元件的优选方法包含提供一个自组装的纳米单元以及把该纳米单元编程为起电子部件的作用。该纳米单元优选是一个依据如上所述任何实施例的纳米单元。

10 编程该纳米单元优选包含配置该分子电路部件。配置该分子电路部件优选为包含通过把电压加到输入引线和输出引线两端来调整至少一个分子电路部件的受导电率影响的属性。该受导电率影响的属性可以选自上述任何受导电率影响的属性。

15 编程该纳米单元优选进一步包含测试该纳米单元的性能。例如，可以通过把该纳米单元的输入 / 输出操作电压关系和一个目标真值表比如一个期望的逻辑真值表进行比较来测试性能。

20 编程该纳米单元优选更进一步包含重复配置该分子电路部件和测试该纳米单元性能的步骤，直到该纳米单元起期望的电子元件作用为止。例如，可以重复该步骤直到该输入 / 输出操作电压关系在一个期望的预定误差内与上述目标真值表相符。一旦被编程，该电子部件起任何上述逻辑或存储器单元或者其它类似装置的作用。

25 提供一个自组装的纳米单元优选包含允许多个纳米尺度的部件自组装成为一个无规则排列，允许多个分子电路部件自组装成为在该纳米尺度部件之间的互连网络，以及利用分子接线夹把该分子电路部件连接到纳米尺度部件。该无规则排列可以是一个具有近程有序和长程无序的排列。该分子接线夹可以包含可作为分子接线夹使用的任何上述分子部分。一个优选的分子部分是硫醇基。该纳米尺度部件可以是，
30

例如，任何上述的纳米尺度部件。该分子电路部件可以是，例如，任何上述的分子电路部件。

如上所述，用于编程或者训练一个纳米单元的任何实施例能够用于由多个纳米单元组装一台计算机。制造计算机的方法优选包含提供多个训练过的、自组装的纳米单元，把该训练过的纳米单元与多个未经训练的纳米单元互连、并且允许训练过的纳米单元训练该未经训练的纳米单元。上述方法的一个优点是训练过的纳米单元用于该未经训练纳米单元的引导训练。因此，该方法可以包含多级重复互连纳米单元并且使用最新训练过的纳米单元来训练未经训练的纳米单元。以这种方式，可以从多个纳米单元中快速并且有效地构造一台分子计算机。

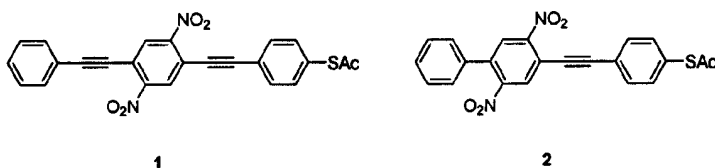
实施例

15 实施例 1

共轭分子的合成

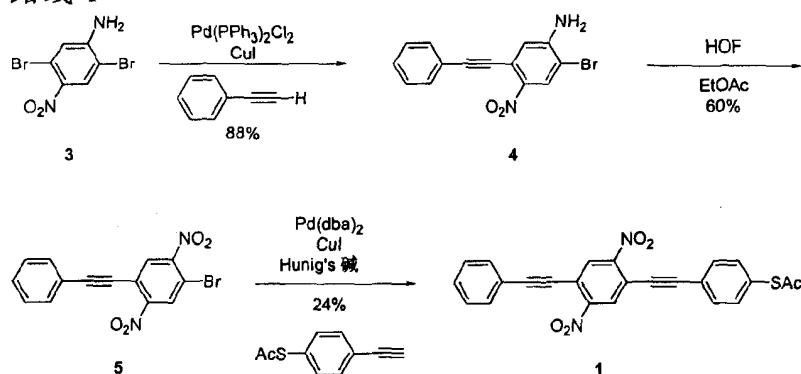
开关和存储器部件

20 为了通过添加更多的硝基来提高电子的存储时间，我们选用 1 和 2 两个合成目标。用酸或碱处理时 SAc 易于裂解为自由的硫醇（SH）。化合物 1 的合成如路线 1 所示。



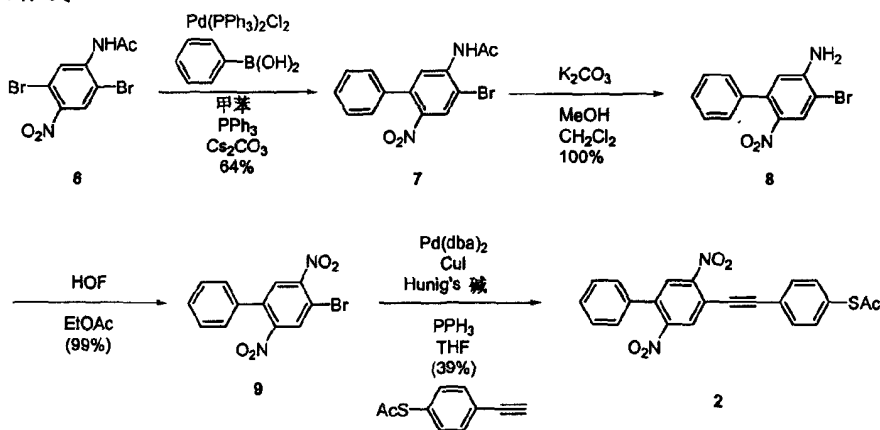
合成目标 1 始于 2,5-二溴-4-硝基苯胺 (3) 与苯基乙炔通过 Sonogashira 偶联而得到 4, 4 再通过 HOF 氧化来形成 5。最终的偶联以 24%的产率, 生成所需化合物 1。此偶联中的低产率可能暗示着, 易脱保护的硫醇或稳定的钯环 (palladacycle) 中间体在偶联过程中形成。

路线 1



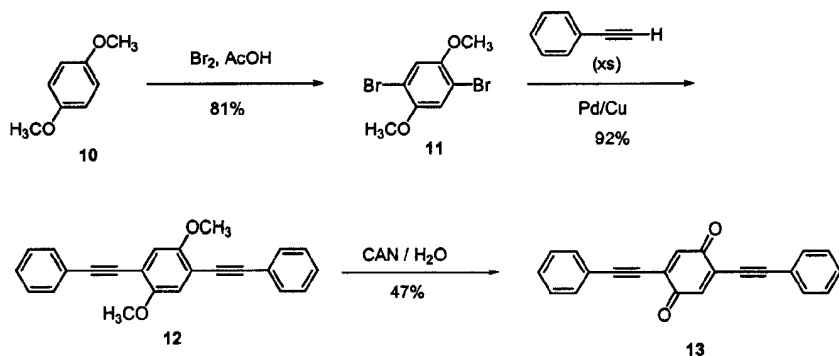
- 5 为了导电，共轭分子中所有苯环应优先彼此在同一平面上。如果苯基取代了最终的苯基乙炔基，该体系不能得到平面性。在测定转动势垒（即：传导势垒）效应的工作中，化合物 2 的合成始于 2,5-二溴-4-硝基乙酰苯胺（6）与苯基硼酸通过 Suzuki 偶联来形成化合物 7。去除乙酰基得到苯胺（8）官能团，随后经过 HOF 氧化以近乎定量产率得到 9。最终通过 Sonogashira 偶联合成出 2。
- 10

路线 2



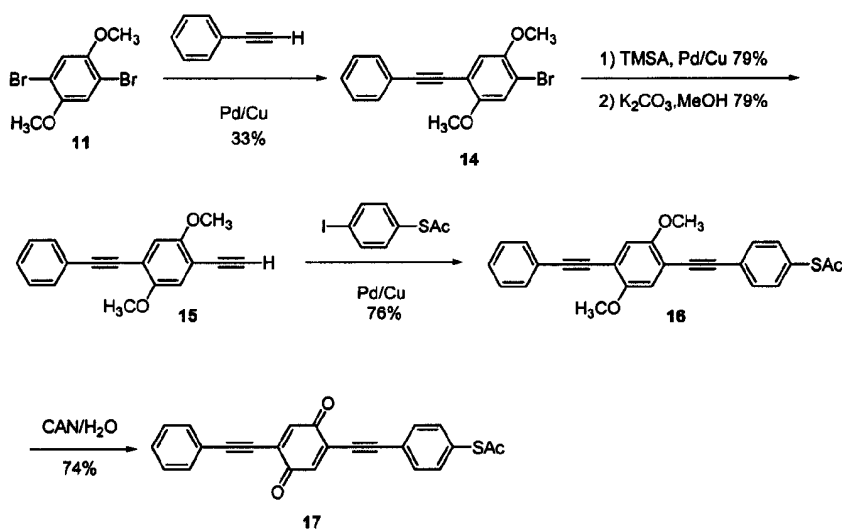
15

路线 3



合成 13 是为了研究含醌分子体系的电化学性质。路线 3 显示从 1,4-二甲氧基苯 (10) 合成 13。使用溴和冰醋酸可以在 10 转化到 11 的过程中得到较好的产率。接着化合物 11 和过量苯基乙炔交叉偶联得到化合物 12，随后 12 被氧化到醌就得到所需化合物 13。必须采取这条合成路线，因为醌一般不能用在钯催化的偶联中，因为已经知道醌将钯由 0 价氧化到 2 价，从而终止催化循环。硝酸高铈铵 (CAN) 是一种温和的、中性的氧化介质，已知可以从二甲氧基苯生成醌，对于此过程是一个合理的选择¹⁸。这个氧化过程得到的醌化合物产率为 47%。在这些体系中，此氧化反应的最佳条件还未得到。

路线 4

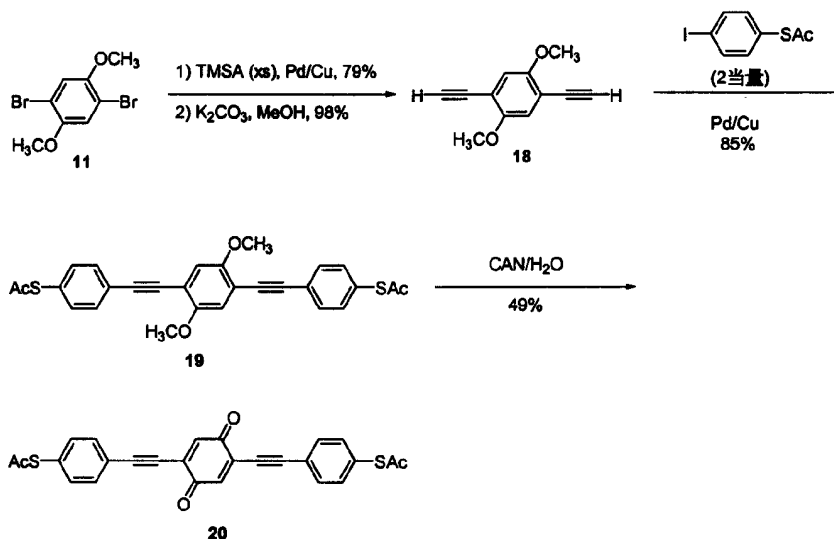


路线 4 显示使用一个硫代乙酸酯基团作为保护的接线夹合成含醌

分子体系。11 和苯基乙炔的交叉偶联得到 14，在 Sonogashira 偶联条件下，由于 11 的两个芳基溴的同等反应能力，其中等而统计可估的产率是 33%。15 是通过三甲基甲硅烷基乙炔和 14 交叉偶联，紧接着炔脱保护而得到的。进一步与 4-碘苯硫代乙酸酯通过钯催化的交叉偶联得到化合物 16。最终化合物 17 通过 CAN 氧化而以 74%产率获得。但是，这个产率是孤立的。其他尝试都得到低很多的结果（~20%）。对于 CAN 氧化条件的最优化还有很多工作要做。

路线 5

10



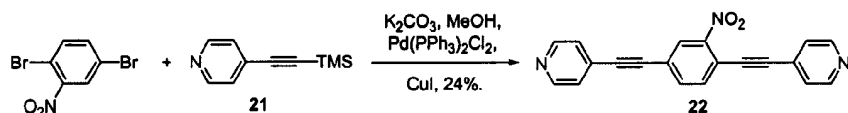
路线 5 表示的是两端都带有接线夹的含醌分子体系（5）的合成。这种化合物可以用来交联金属纳米颗粒，作为未来分子电子装置中的分路连接。11 与过量的三甲基甲硅烷基乙炔发生交叉偶联反应，随后进行脱保护而纯净地得到二炔 18。接着与二当量的 4-碘苯硫代乙酸酯发生交叉偶联得到化合物 19。最后，19 通过 CAN 氧化步骤，以中等产率生成 20。

20

接线夹

为引入分子电子装置中的几种含有吡啶接线夹的化合物的合成是从化合物 21 开始的。22 的合成是吡啶 21 和 2,5-二溴硝基苯通过如反

应式 1 所示的偶联而完成的。产率低可能是由于 TMS 基脱下后形成了稳定的乙炔化铜。如果不采取原位脱保护,这种吡啶炔已证明是不稳定的。



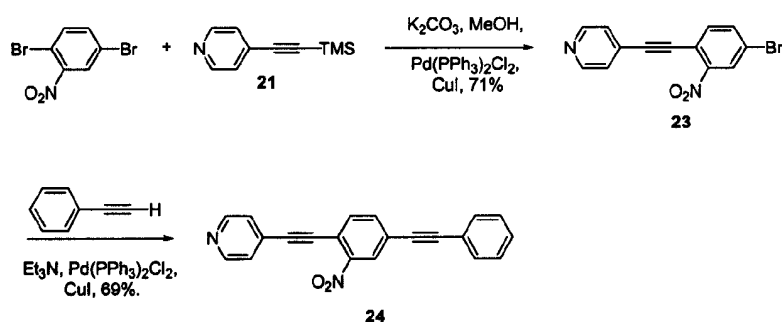
(1)

5

24 依据路线 6 合成。合成始于将 1 当量的 21 选择性的偶联到 2,5-二溴硝基苯中硝基的邻位而得到 23。偶联 23 到苯基乙炔上生成 24 而完成合成。

10

路线 6

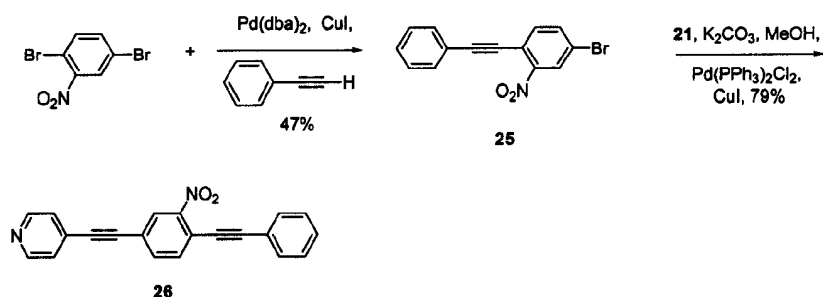


15

化合物 26 的合成始于研究硝基关于化学吸附吡啶接线夹的影响。为此,化合物 24 采用与如路线 7 所示的 23 合成类似的方式合成。将一当量的苯基乙炔选择性的偶联到 2,5-二溴硝基苯上生成 25,接着偶联到 21 上得到很高产率的 26 而完成此合成。

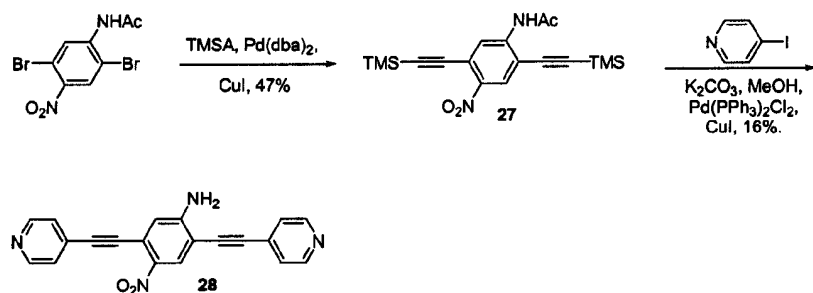
路线 7

20



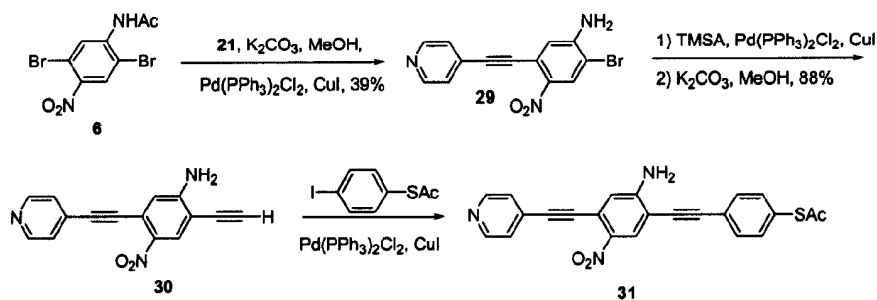
连接分子 28 的合成依据路线 8。合成始于将 2,5-二溴-4-硝基乙酰苯胺与过量的三甲基甲硅烷基乙炔偶联而给出 27，接着在原位脱保护，并与 4-碘代吡啶偶联得到 28，产率较低。偶联反应的低产率可能是由于硝基和炔单元的环化过程。

路线 8

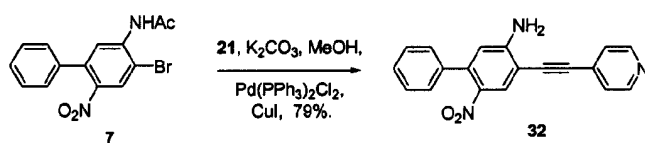


化合物 31 的合成是为了形成一个 SAM，经由被保护的苯硫酚链端基，使分子的吡啶末端引入装置中时与苯基相比作为更好的顶点与金属接触。31 的合成是通过偶联 2,5-二溴-4-硝基乙酰苯胺与 21，低产率得到化合物 29。29 接着与三甲基甲硅烷基乙炔偶联，随后用碳酸钾脱保护而生成 30。最终，30 与 4-碘苯硫代乙酸酯偶联，得到高产率（75%）的分子装置 31。

路线 9

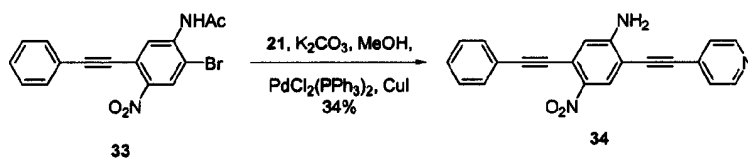


合成 32 是为了研究如前 2 所述的转动势垒的影响。32 的合成始于前面合成的 7 与 21 偶联，而得到较高产率，如反应式 2 所示。



(2)

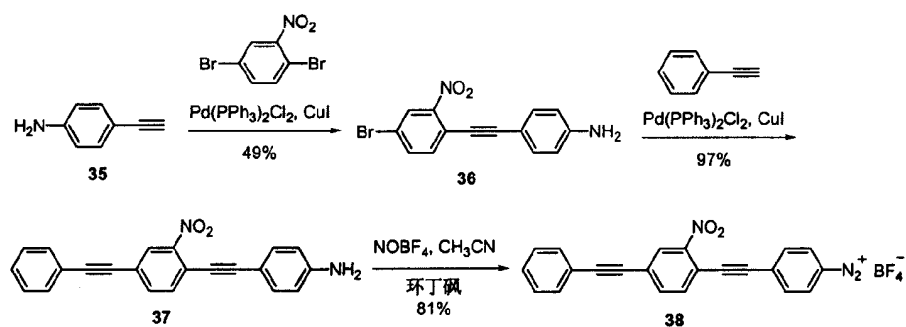
依据反应式 3，使用前面提到的 33 来合成化合物 34。34 与先前在装置实施例中呈现负微分电阻（NDR）以硫醇为端基的硝基苯胺类似。



(3)

除了含吡啶的体系外，还合成了三种可能的以重氮盐为末端的存储器 and 开关部件。38 与以硫代乙酰基为末端的 NDR 和存储器部件¹以及吡啶基为末端的 24 相类似。38 的合成始于将 35⁷ 偶联到 2,5-二硝基苯上，以中等的产率生成 36，接着 36 偶联到苯基乙炔上形成化合物 37。37 的重氮化最终生成高产率的分子 38。

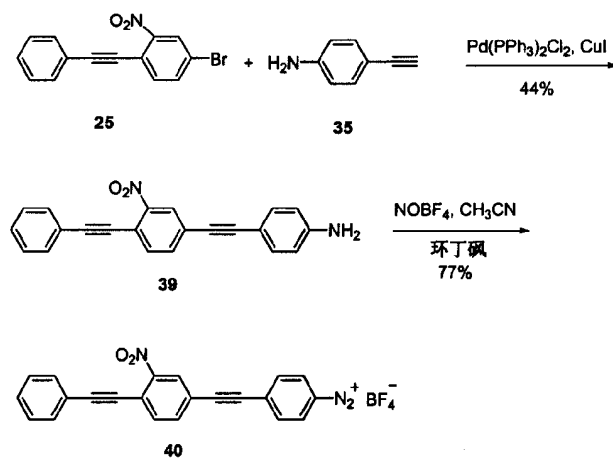
路线 10



- 5 40 与 26 的结构相似，除了吡啶基已被芳基重氮盐取代。40 的合成如路线 11 所示。偶联苯胺 35 到硝基化合物 25 上生成中等产率的重氮前体 39。苯胺 42 的重氮化生成所需产物 37。

路线 11

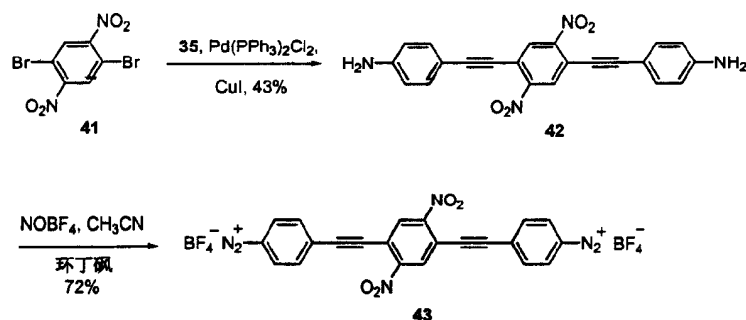
10



纳米颗粒连接分子 43 依据路线 12 合成。从二硝基 41 开始，与苯胺 35 偶联得到二硝基二苯胺 42，随后重氮化生成高产率的 43。

15

路线 12



实验

一般步骤

- 5 所有反应除非注明都在干燥氮气气氛中进行。试剂级乙醚和四氢呋喃（THF）在氮气下从二苯酮羰游基钠中蒸馏。试剂级二氯甲烷（CH₂Cl₂）在氮气下从氢化钙（CaH₂）中蒸馏。三乙胺和 N,N-二异丙胺（Hünig's 碱）在氮气气氛下从氢化钙中蒸馏。将大量的己烷蒸馏备用。重力柱层析和闪式柱层析使用来自 EM Science 的 230-400 目的
- 10 硅胶。薄层层析（TLC）在 0.25mm 厚度上采用 Merck 40 F₂₅₄ 进行。

一般钯/铜偶联反应步骤

- 在烘箱干燥的带有玻璃螺帽的试管中加入所有固体，包括芳基卤（溴或碘），炔，碘化铜，三苯基膦和钯催化剂。在真空中除去空气，
- 15 用干燥氮气（3×）替代。液态的四氢呋喃，与 Hünig's 碱或三乙胺加于其中，反应在搅拌状态下在油浴中加热。反应混合物一冷却就采用重力过滤法过滤，以去除固体，并用二氯甲烷稀释。反应混合物用氯化铵（NH₄Cl）（3×）水溶液提取出来。有机层用硫酸镁干燥并过滤。溶剂在真空中除去。

20

三甲基甲硅烷基保护的炔的脱保护一般步骤

- 在装有搅棒的圆底烧瓶中加入被保护的炔，碳酸钾（5 当量于被保护的炔），甲醇和二氯甲烷。加热反应，一完成就用二氯甲烷稀释反应混合物，并用盐水（3×）洗涤。有机层用硫酸镁干燥。溶剂在
- 25 真空中除去。

HOF 氧化的一般步骤

5 在 125 毫升聚乙烯瓶子中加入 H_2O (2 毫升)和 CH_3CN (60 毫升), 冷却到 -20°C 。 F_2 (在 He 中占 20%)以 50sccm 速率经溶液发泡 2 小时。 得到的 HOF/ CH_3CN 溶液充 He 气 15 分钟。 将要氧化的物质加入丙酮 或乙酸乙酯(10 毫升)中, 在 -20°C 下混合 5 分钟, 再加入饱和碳酸 氢钠溶液进行中和。 接着分离出有机相, 用硫酸镁干燥, 溶剂在真空 中除去。

10 苯胺与四氟硼酸亚硝鎓盐在乙腈-环丁砜体系中的重氮化反应的一般步 骤

NOBF₄ 在充满氮气的干燥器皿里称重, 置于装有磁力搅棒的圆底 烧瓶中, 封上隔膜。 乙腈和环丁砜以 5 比 1 的体积比注射进反应器, 所得悬浮液在干冰/丙酮浴中冷却到 -40°C 。 苯胺溶液的制备是通过在 氮气层下将热的环丁砜($45-50^\circ\text{C}$)加到胺中, 超声处理 1 分钟, 随后 15 加入乙腈(体积比 10-20%)。 接着苯胺溶液在 10 分钟以内加到亚硝 鎓盐悬浮液中。 反应混合物在 -40°C 保持 30 分钟, 接着加热到室温。 这时, 通过加入乙醚或二氯甲烷使重氮盐沉淀出来, 过滤收集, 用乙 醚或二氯甲烷洗涤并干燥。 使用乙醚和/或二氯甲烷从 DMSO 中再次 沉淀而使产物得到再次纯化。

20

4-乙炔基苯基-2,4-二硝基溴苯(5)

25 2-溴-4-硝基-5-乙炔基苯基苯胺(490 毫克, 1.48 毫摩尔)在乙酸 乙酯(10 毫升)中依据 HOF 氧化的一般步骤进行氧化, 得到黄色固 体 320 毫克(产率 60%)。 IR (KBr) 3442.7, 3101.4, 2216.8, 1610.6, 1540.9, 1461.3, 1384.8, 1358.7, 1337.1, 1264.4, 906.2, 849.6, 824.4, 760.2, 689.8 cm^{-1} 。 ^1H NMR (400 MHz, CDCl_3) δ 8.41 (s, 1 H), 8.09 (s, 1 H), 7.60-7.58 (m, 2 H), 7.41-7.39 (m, 3 H)。 ^{13}C NMR (100 MHz, CDCl_3) δ 152.1, 150.4, 132.7, 131.7, 131.0, 130.7, 129.1, 121.5, 119.8, 113.9, 102.0。 HRMS: 计 算值为 345.9589, 实测值为 345.9585。

30

2',5'-二硝基-4,4'-二乙炔基苯基-1-硫代乙酰苯 (1)

4 (300 毫克, 0.86 毫摩尔)、4-乙炔基(硫代乙酰基)苯 (183 毫克, 1.04 毫摩尔)、双(二亚苄基丙酮)钼 (12 毫克, 0.02 毫摩尔)、碘化亚铜 (4 毫克, 0.02 毫摩尔)、三苯基膦 (13 毫克, 0.05 毫摩尔)、Hünig's 碱 (0.60 毫升) 和四氢呋喃 (20 毫升) 依据一般偶联步骤进行反应。反应混合物依据上述步骤操作, 加热到 60℃ 过夜。粗产物用闪式柱层析法 (氧化硅, 3: 1 的二氯甲烷: 己烷) 来纯化, 得到亮黄色固体 90 毫克 (产率 24%)。IR (KBr) 2220.2, 1705.2, 1545.5, 1499.81, 1396.8, 1337.5, 1286.1, 1252.1, 1108.6, 1087.2, 953.2, 926.0, 868.3, 827.2, 756.7, 684.1, 618.3 cm^{-1} 。 ^1H NMR (400 MHz, CDCl_3) δ 8.34 (d, $J = 0.4$, 1 H), 8.35 (d, $J = 0.4$, 1 H), 7.63-7.59 (m, 4 H), 7.46-7.40 (m, 5 H), 2.49 (s, 3 H)。 ^{13}C NMR (100 MHz, CDCl_3) δ 193.2, 151.1, 134.7, 133.1, 132.7, 131.0, 130.7, 129.1, 122.8, 121.7, 119.4, 118.6, 102.4, 100.9, 84.8, 83.5, 30.8。HRMS: 计算值为 442.0623, 实测值为 442.0634。

2-溴-4-硝基-5-苯基乙酰苯胺 (7)

6 (676 毫克, 2 毫摩尔)、三苯基膦 (52 毫克, 0.2 毫摩尔)、苯基硼酸 (293 毫克, 2.4 毫摩尔)、二氯化双(三苯基膦)钼 (70 毫克, 0.1 毫摩尔) 和碳酸铯 (977 毫克, 3 毫摩尔) 置于 100 毫升圆底烧瓶中, 除去空气, 用氮气替代。加入甲苯 (30 毫升), 反应加热到 60℃ 保持 2 天。用乙醚稀释反应物, 用氯化铵水溶液 (2×) 洗涤, 用硫酸镁干燥, 溶剂在真空中除去。粗产物用闪式柱层析法 (CH_2Cl_2) 纯化, 得到白色固体 430 毫克 (产率 64%)。IR (KBr) 3373.6, 3322.4, 3086.5, 1774.0, 1681.7, 1568.9, 1528.8, 1445.8, 1389.4, 1358.6, 1245.8, 1179.1, 1112.5, 1056.1, 1030.4, 999.6, 872.0, 850.9, 768.9, 697.1 cm^{-1} 。 ^1H NMR (400 MHz, CDCl_3) δ 8.54 (s, 1H), 8.15 (s, 1H), 7.80 (br s, 1H), 7.40-7.38 (m, 3H), 7.29-7.27 (m, 2H) 2.26 (s, 3H)。 ^{13}C NMR (100 MHz, CDCl_3) δ 168.44, 143.77, 139.34, 137.74, 136.81, 128.67, 128.51, 128.47, 127.85, 123.31, 110.59, 25.05。HRMS: $\text{C}_{14}\text{H}_{11}\text{BrN}_2\text{O}_3$ 计算值为 333.9953, 实测值为 333.9952。

2-溴-4-硝基-5-苯基苯胺 (8)

7 (500 毫克, 1.49 毫摩尔)、碳酸钾 (1.031 克, 7.46 毫摩尔)、
甲醇 (30 毫升) 和二氯甲烷 (30 毫升) 加到 100 毫升圆底烧瓶里,
5 在氮气气氛下室温搅拌 2 小时。过滤碳酸钾, 并用二氯甲烷洗涤, 得
到 437 毫克目标化合物 (产率 100%)。IR (KBr) 3463.7, 3349.2, 3221.3,
1623.9, 1584.6, 1555.4, 1495.5, 1443.6, 1406.9, 1305.6, 1259.4, 1123.9,
1051.7, 896.7, 846.5, 760.1, 701.3, 632.1, 563.8 cm^{-1} 。 ^1H NMR (400 MHz,
10 CDCl_3) δ 8.21 (s, 1H), 7.39-7.36 (m, 3H), 7.23-7.21 (m (重叠), 2H), 6.61
(s, 1H)。 ^{13}C NMR (100 MHz, CDCl_3) δ 148.5, 139.4, 138.5, 130.7, 128.8,
128.4, 128.2, 128.1, 117.2, 106.3。HRMS: 计算值为 291.9848, 实测值
为 291.9846。

2,5-二硝基-4-苯基溴苯 (9)

8 (373 毫克, 1.28 毫摩尔) 溶于乙酸乙酯 (10 毫升) 中, 依据
15 HOF 氧化的一般步骤进行氧化, 得到 407 毫克橙色的固体 (产率 99%)。
IR (KBr) 3446.7, 3090.4, 1542.8, 1461.1, 1443.1, 1347.3, 1257.7, 1114.6,
1076.2, 1051.8, 1021.0, 904.5, 842.5, 768.8, 743.7, 699.9, 551.0, 485.16
 cm^{-1} 。 ^1H NMR (400 MHz, CDCl_3) δ 8.16 (s, 1 H), 7.89 (s, 1 H), 7.47-7.45
20 (m, 3 H), 7.31-7.29 (m, 2 H)。 ^{13}C NMR (100 MHz, CDCl_3) δ 151.5, 150.6,
137.2, 134.4, 130.8, 130.1, 129.7, 128.9, 128.1, 114.1。HRMS: 计算值为
321.9589, 实测值为 321.9592。

2',4'-二硝基-5'-苯基-4-乙炔基苯基-1-硫代乙酰苯 (2)

25 9 (147 毫克, 0.46 毫摩尔)、4-乙炔基(硫代乙酰基)苯 (106 毫
克, 0.60 毫摩尔)、双(二亚苄基丙酮)钯 (26 毫克, 0.05 毫摩尔)、
碘化亚铜 (9 毫克, 0.05 毫摩尔)、三苯基膦 (12 毫克, 0.05 毫摩尔)、
Hünig's 碱 (0.16 毫升) 和四氢呋喃 (20 毫升) 依据偶联反应一般步
骤进行偶联。对反应混合物搅拌, 并加热到 45°C 过夜。粗产物用柱层
30 析法 (氧化硅, 3: 1 的二氯甲烷: 己烷) 提纯, 得到 75 毫克橙色固

体(产率 39%)。IR (KBr) 2922.7, 2214.3, 1702.7, 1542.8, 1488.1, 1357.1, 1271.1, 1115.1, 1088.6, 956.0, 908.6, 829.9, 770.5, 707.0, 623.4 cm^{-1} 。 ^1H NMR (400 MHz, CDCl_3) δ 8.16 (s, 1 H), 8.10 (s, 1 H), 7.63 (d, $J = 8.4$, 2 H), 7.48-7.44 (m, 5 H), 7.36-7.33 (m, 2 H), 2.44 (s, 3 H)。 ^{13}C NMR (100 MHz, CDCl_3) δ 193.3, 151.2, 150.6, 136.8, 134.8, 134.7, 133.1, 130.8, 130.2, 130.1, 129.6, 128.6, 128.1, 122.9, 119.0, 99.8, 84.5, 30.8。HRMS: 计算值为 418.0623, 实测值为 418.0619。

2,5-二溴-1,4-二甲氧基苯 (11)

在 100 毫升圆底烧瓶中, 1,4-二甲氧基苯 (10.0 克, 72.4 毫摩尔) 溶于冰醋酸 (20 毫升)。溴 (7.42 毫升, 145.0 毫摩尔) 的冰醋酸 (7.5 毫升) 溶液在 40 分钟内在室温逐滴的加到第一个溶液中。反应混合物搅拌 2 小时。粗产物用冰水和冰甲醇洗涤以得到细的白色晶体。浓缩母液并冷却以得到更多白色晶体 (15.9 克, 产率 74%)。Mp 136-138 $^{\circ}\text{C}$ (文献²¹ mp 144-145 $^{\circ}\text{C}$)。IR (KBr) 3091.9, 3022.1, 2968.8, 2944.4, 2842.8, 1694.9, 1494.2, 1475.6, 1436.5, 1358.2, 1275.0, 1211.8, 1185.0, 1065.4, 1021.9, 860.5, 760.4, 441.8 cm^{-1} 。 ^1H NMR (400 MHz, CDCl_3) δ 7.13 (s, 2 H), 3.87 (s, 6 H)。 ^{13}C NMR (100 MHz, CDCl_3) δ 150.93, 117.53, 110.90, 57.43。

2,5-二(乙炔基苯基)-1,4-二甲氧基苯 (12)

11 (8.745 克, 29.55 毫摩尔)、二氯化双(三苯基膦)钯 (0.415 克, 0.591 毫摩尔)、碘化亚铜 (0.225 克, 1.182 毫摩尔)、三苯基膦 (0.310 克, 1.182 毫摩尔)、四氢呋喃 (35 毫升)、Hünig's 碱 (20.5 毫升, 118 毫摩尔) 和苯基乙炔 (7.8 毫升, 70.92 毫摩尔) 按照以下偶联的一般步骤进行反应。溶液在 65 $^{\circ}\text{C}$ 油浴中加热 3 天。从苯中重结晶得到所需产品 9.22 克 (产率 92%)。mp 175-177 $^{\circ}\text{C}$ (文献¹⁶ 176-177 $^{\circ}\text{C}$)。 ^1H NMR (400 MHz, CDCl_3) δ 7.57 (m, 4 H), 7.34 (m, 6H), 7.03 (s, 2H), 3.89 (s, 6 H)。 ^{13}C NMR (100 MHz, CDCl_3) δ 154.10, 131.89, 128.60, 128.50, 123.39, 115.86, 113.57, 95.23, 85.86, 56.66。

2,5-二(乙炔基苯基)苯醌 (13)

12 (0.300 克, 0.886 毫摩尔) 和四氢呋喃 (6 毫升) 加到 25 毫升装有搅棒的圆底烧瓶中。硝酸高铈铵 (1.46 克, 2.658 毫摩尔) 3 毫升水溶液慢慢加到烧瓶中, 搅拌 15 分钟。加水, 有机物质用二氯甲烷提取出来。闪式柱层析法 (硅胶, 用 1:1 的己烷/二氯甲烷作为洗脱液) 得到所需产物 (0.129 克, 47%)。IR (KBr) 3047.5, 2203.0, 1716.2, 1655.3, 1568.3, 1215.4, 1100.6, 902.1, 757.6, 686.4 cm^{-1} . ^1H NMR (400 MHz, CDCl_3) δ 7.58 (dd, $J = 7.9, 1.5$ Hz, 4 H), 7.38 (m, 6 H), 6.99 (s, 2 H). ^{13}C NMR (100 MHz, CDCl_3) δ 182.87, 136.55, 133.34, 132.83, 130.57, 128.97, 121.83, 105.26, 82.90. HRMS: $\text{C}_{12}\text{H}_{12}\text{O}_2$ 计算值为 308.0837, 实测值为 308.0834。

2-溴-5-乙炔基苯基-1,4-二甲氧基苯 (14)

11 (2.96 克, 10.0 毫摩尔)、双(二亚苄基丙酮)钯 (0.115 克, 0.20 毫摩尔)、碘化亚铜 (0.038 克, 0.20 毫摩尔)、三苯基膦 (0.131 克, 0.50 毫摩尔)、四氢呋喃 (15 毫升)、Hünig's 碱 (6.97 毫升, 40.0 毫摩尔) 和苯基乙炔 (1.21 毫升, 11.0 毫摩尔) 按照以下偶联的一般步骤进行反应。试管在 50°C 油浴中加热 18 小时。柱层析法 (硅胶, 用 19:1 的己烷/乙醚作为洗脱液) 得到所需产物, 不是很纯 (NMR 检验有约 15% 的杂质), 中等产率 (1.02 克, 产率 32%)。就以这种不纯的形态加入到下一步反应中去。 ^1H NMR (400 MHz, CDCl_3) δ 7.54 (m, 2 H), 7.33 (m, 3 H), 7.09 (s, 1 H), 7.02 (s, 1 H), 3.86 (s, 6 H)。

1,4-二甲氧基-2-乙炔基苯基-5-(三甲基甲硅烷基乙炔基)苯

14 (1.0 克, 3.15 毫摩尔)、双(二亚苄基丙酮)钯 (0.036 克, 0.063 毫摩尔)、碘化亚铜 (0.012 克, 0.063 毫摩尔)、三苯基膦 (0.042 克, 0.16 毫摩尔)、四氢呋喃 (20 毫升)、Hünig's 碱 (2.2 毫升, 12.6 毫摩尔) 和三甲基甲硅烷基乙炔 (0.89 毫升, 6.3 毫摩尔) 按照以下偶联的一般步骤进行反应。试管加盖在 60°C 油浴中加热 1 天。闪式柱

层析法（硅胶，用 24:1 的己烷/乙酸乙酯作为洗脱液）得到稍有不纯的产品（0.83 克，产率 79%）。¹H NMR (400 MHz, CDCl₃) δ 7.55 (m, 2 H), 7.32 (m, 3 H), 6.98 (s, 1 H), 6.95 (s, 1 H), 3.84 (s, 3 H), 3.83 (s, 3 H), 0.27 (s, 9 H)。

5

1,4-二甲氧基-2-乙炔基-5-(乙炔基苯基)苯 (15)

1,4-二甲氧基-2-乙炔基苯基-5-(三甲基甲硅烷基乙炔基)苯 (0.830 克, 2.48 毫摩尔)、碳酸钾 (1.71 克, 12.4 毫摩尔)、甲醇 (50 毫升) 和二氯甲烷 (50 毫升) 按照以下脱保护的一般步骤进行反应，
10 得到所需产品 (0.513 克, 产率 79%)。¹H NMR (400 MHz, CDCl₃) δ 7.55 (m, 2 H), 7.33 (m, 3 H), 7.00 (s, 1 H), 6.98 (s, 1 H), 3.87 (s, 3 H), 3.86 (s, 3 H), 3.39 (s, 1 H)。

4,4'-二(乙炔基苯基)-2',5'-二甲氧基-1-苯硫代乙酸酯 (16)

15 15 (0.513 克, 1.96 毫摩尔)、双(二亚苄基丙酮)钼(0) (0.058 克, 0.10 毫摩尔)、碘化亚铜 (0.019 克, 0.10 毫摩尔)、三苯基膦 (0.066 克, 0.25 毫摩尔)、四氢呋喃 (20 毫升)、Hünig's 碱 (1.37 毫升, 7.84 毫摩尔) 和 4-(硫代乙酰基)碘代苯 (0.608 克, 2.16 毫摩尔) 按照以下偶联的一般步骤进行反应。试管加盖在 55°C 油浴中加热 3 天。闪
20 式柱层析法(硅胶, 用二氯甲烷作为洗脱液)得到稍有不纯的产品 (0.621 克, 产率 76%)。¹H NMR (400 MHz, CDCl₃) δ 7.57 (m, 4 H), 7.38 (d, J = 8.1 Hz, 2 H), 7.33 (m, 3 H), 7.03 (s, 1 H), 7.02 (s, 1 H), 3.874 (s, 3 H), 3.870 (s, 3 H), 2.40 (s, 3 H)。

25 2-乙炔基苯基-5-((4'-硫代乙酰基)乙炔基苯基)苯醌 (17)

16 (0.050 克, 0.12 毫摩尔)、乙腈 (5 毫升) 和四氢呋喃 (5 毫升) 加到 25 毫升装有搅棒的圆底烧瓶中。一次加入硝酸高铈铵 (0.13 克, 0.24 毫摩尔) 在水 (1 毫升) 中的溶液。室温下搅拌 30 分钟，再
30 加入相当的硝酸高铈铵 (0.13 克, 0.24 毫摩尔) 溶液。再过 20 分钟，加入水 (30 毫升) 结束反应，沉淀出橙色固体。闪式柱层析法 (硅胶，

用二氯甲烷作为洗脱液)得到所需产物(0.034 克, 产率 74%)。IR (KBr) 3053.0, 2924.3, 2852.6, 2205.4, 1703.4, 1652.7, 1568.8, 1483.7, 1442.2, 1354.8, 1221.3, 1105.4, 1089.4, 949.6, 920.1, 830.9, 758.2, 688.2, 620.6 cm^{-1} . ^1H NMR (400 MHz, CDCl_3) δ 7.58 (m, 4 H), 7.42 (m, 2 H), 7.38 (m, 3 H), 6.98 (s, 1 H), 6.97 (s, 1 H), 2.42 (s, 3 H). ^{13}C NMR (100 MHz, CDCl_3) δ 193.22, 182.74, 182.67, 136.88, 136.51, 134.63, 133.34, 133.24, 132.99, 132.84, 130.94, 130.63, 128.99, 122.81, 121.80, 105.38, 103.99, 84.17, 82.92, 30.80。HRMS: $\text{C}_{24}\text{H}_{14}\text{O}_3\text{S}$ 计算值为 382.0664, 实测值为 382.0663。

1,4-二甲氧基-2,5-双(三甲基甲硅烷基乙炔基)苯

11 (1.75 克, 5.91 毫摩尔)、二氯化双(三苯基膦)钯 (0.207 克, 0.296 毫摩尔)、碘化亚铜 (0.113 克, 0.591 毫摩尔)、三苯基膦 (0.155 克, 0.591 毫摩尔)、四氢呋喃 (20 毫升)、Hünig's 碱 (4.1 毫升, 23.64 毫摩尔) 和三甲基甲硅烷基乙炔 (2.51 毫升, 17.73 毫摩尔) 按照以下偶联的一般步骤进行反应。试管加盖在 55°C 油浴中加热 2 天。闪式柱层析法 (硅胶, 用 1: 1 己烷/二氯甲烷作为洗脱液) 得到所要产物 (1.54 克, 产率 79%)。IR (KBr) 2957.0, 2898.2, 2851.2, 2829.0, 2149.1, 1496.8, 1464.1, 1449.1, 1388.2, 1283.7, 1249.0, 1223.6, 1203.1, 1172.4, 1039.6, 883.2, 841.3, 757.4, 714.9, 696.2, 626.4 cm^{-1} . ^1H NMR (400 MHz, CDCl_3) δ 6.89 (s, 2 H), 3.81 (s, 6 H), 0.25 (s, 18 H) ^{13}C NMR (100 MHz, CDCl_3) δ 154.56, 116.59, 113.81, 101.22, 100.84, 56.83, 0.40。HRMS: $\text{C}_{18}\text{H}_{26}\text{O}_2\text{Si}_2$ 计算值为 330.1471, 实测值为 330.1468。

1,4-二甲氧基-2,5-二乙炔基苯 (18)

1,4-二甲氧基-2,5-双(三甲基甲硅烷基乙炔基)苯 (1.50 克, 4.54 毫摩尔)、碳酸钾 (6.27 克, 45.4 毫摩尔)、甲醇 (50 毫升) 和二氯甲烷 (50 毫升) 按照以下脱保护的一般步骤进行反应, 得到所需产品 (0.829 克, 产率 98%)。 ^1H NMR (400 MHz, CDCl_3) δ 6.96 (s, 2 H), 3.84 (s, 6 H), 3.37 (s, 2 H)。

2,5-双(4'-(硫代乙酰基)乙炔基苯基)-1,4-二甲氧基苯 (19)

18 (0.810 克, 4.35 毫摩尔)、双(二亚苄基丙酮)钼 (0.253 克, 0.44 毫摩尔)、碘化亚铜 (0.084 克, 0.44 毫摩尔)、三苯基膦 (0.115 克, 0.44 毫摩尔)、四氢呋喃 (30 毫升)、Hünig's 碱 (4.5 毫升, 26.1 毫摩尔) 和 4-(硫代乙酰) 碘代苯²² (2.54 克, 9.14 毫摩尔) 按照以下偶联的一般步骤进行反应。搅拌溶液在 60℃ 油浴中加热 16 小时。从二氯甲烷/己烷中结晶得到所要产物 (1.81 克, 85%)。IR (KBr) 3129.1, 3057.4, 3006.2, 2975.5, 2940.0, 2847.4, 2207.2, 1697.7, 1506.8, 1483.1, 1463.1, 1396.2, 1279.2, 1223.5, 1122.2, 1034.2, 949.5, 898.8, 825.5, 765.6, 616.8 cm⁻¹. ¹H NMR (400 MHz, CDCl₃) δ 7.57 (dt, J = 8.5, 1.9 Hz, 4 H), 7.39 (dt, J = 8.5, 2.0 Hz, 4 H), 7.01 (s, 2 H), 3.89 (s, 6 H), 2.42 (s, 6 H). ¹³C NMR (100 MHz, CDCl₃) δ 193.85, 154.43, 134.58, 132.65, 128.64, 124.84, 116.08, 113.75, 94.76, 87.73, 56.91, 30.70。HRMS: C₂₈H₂₂O₄S₂ 计算值为 486.0960, 实测值为 486.0956。

2,5-双(4'-(硫代乙酰基)乙炔基苯基)苯醌 (20)

19 (0.050 克, 0.103 毫摩尔)、乙腈 (5 毫升) 和四氢呋喃 (3 毫升) 加到 25 毫升装有搅棒的圆底烧瓶中, 把硝酸高铈铵 (0.339 克, 0.618 毫摩尔) 2 毫升水溶液分两部分以 30 分钟的间隔加到烧瓶中。室温下搅拌 3 小时, 通过加水中止反应, 得到橙色的固体沉淀。闪式柱层析法 (硅胶, 用二氯甲烷作为洗脱液) 得到所要的产品 (0.023 克, 产率 49%)。IR (KBr) 2922.2, 2847.4, 2203.4, 1694.9, 1660.1, 1569.9, 1351.8, 1212.3, 1119.7, 1084.6, 1013.2, 960.3, 826.8, 620.6 cm⁻¹. ¹H NMR (400 MHz, CDCl₃) δ 7.60 (dt, J = 8.3, 1.6 Hz, 4 H), 7.42 (dt, J = 8.3, 1.6 Hz, 4 H), 7.00 (s, 2 H), 2.43 (s, 6 H). ¹³C NMR (100 MHz, CDCl₃) δ 193.23, 182.61, 136.86, 134.64, 133.25, 133.07, 130.97, 122.78, 104.14, 84.08, 30.79。HRMS: C₂₆H₁₆O₄S₂ 计算值为 456.0500, 实测值为 456.0490。

2,4-双(4'-乙炔基吡啶基)-1-硝基苯 (22)

将 2,5-二溴硝基苯 (0.28 克, 0.997 毫摩尔)、二氯化双(三苯基磷)钯 (0.07 克, 0.098 毫摩尔)、碘化亚铜 (0.019 克, 0.098 毫摩尔)、三苯基磷 (0.106 克, 0.40 毫摩尔) 以及碳酸钾 (1.1 克, 7.96 毫摩尔) 溶于四氢呋喃 (4 毫升) 中配成溶液。通过套管将 21 (0.377 克, 2.15 毫摩尔) 在四氢呋喃 (4 毫升) 和甲醇 (2 毫升) 中的溶液加入上述溶液中。将此混合物在 64℃ 下加热 20 小时, 然后用旋转蒸发法去掉溶剂, 黑色剩余物用碳酸钾水溶液洗涤, 并用乙醚萃取。合并的有机层用硫酸钠干燥并过滤, 溶剂在真空中蒸除。用闪式柱层析法 (硅胶, 己烷/乙酸乙酯 70/30, 50/50, 20/80, 0/100) 纯化, 得到目标化合物黄色固体 60 毫克 (产率 24%)。Mp: 178-180℃。IR (KBr) 3414.0, 3036.7, 1616.0, 1589.4, 1538.1, 1519.9, 1407.9, 1345.7, 1271.1, 1214.1, 828.3 cm^{-1} 。 ^1H NMR (400 MHz, DMSO- d_6) δ 8.69 (br s, 4 H), 8.44 (d, $J=1.4$ Hz, 1 H), 8.04 (1/2 ABqd, $J=8.0, 1.4$ Hz, 1 H), 7.99 (1/2 ABq, $J=8.0$ Hz, 1 H), 7.60 (d, $J=5.8$ Hz, 2 H), 7.57 (d, $J=5.8$ Hz, 2 H)。 ^{13}C NMR (100 MHz, DMSO- d_6) δ 150.21, 150.13, 149.42, 136.27, 135.36, 129.16, 129.11, 127.96, 125.50, 125.39, 123.25, 116.55, 94.98, 90.63, 90.59, 88.13。HRMS: $\text{C}_{20}\text{H}_{11}\text{N}_3\text{O}_2$ 计算值为 325.0851, 实测值为 325.0847。

20 1-溴代-4-(4'-乙炔基吡啶)-3-硝基苯 (23)

将 2,5-二溴硝基苯 (0.43 克, 1.53 毫摩尔)、二氯化双(三苯基磷)钯 (II) (0.052 克, 0.074 毫摩尔)、碘化亚铜 (0.015 克, 0.078 毫摩尔)、三苯基磷 (0.079 克, 0.30 毫摩尔) 以及碳酸钾 (0.83 克, 6.0 毫摩尔) 溶于四氢呋喃 (2 毫升) 中配成溶液。通过套管将 21 (0.342 克, 1.95 毫摩尔) 在四氢呋喃 (4 毫升) 和甲醇 (1.5 毫升) 中的溶液加到上述溶液中。将此混合物在 23℃ 下加热 2 小时, 然后用旋转蒸发法去掉溶剂, 剩余物用水稀释, 并用乙醚萃取。合并的有机层用硫酸钠干燥并过滤, 溶剂在真空中蒸除。用闪式柱层析法 (硅胶, 己烷/乙酸乙酯 90/10, 70/30, 50/50) 纯化, 得到目标化合物灰白色固体 330 毫克 (产率 71%)。熔点: 166-171 °C。IR (KBr) 3424.4, 3093.3, 1592.3,

1521.4, 1409.3, 1341.4, 1272.6 cm^{-1} . ^1H NMR (400 MHz, CDCl_3) δ 8.68 (br s, 2 H), 8.29 (d, $J=1.9$ Hz, 1 H), 7.79 (dd, $J=8.3$, 2.0 Hz, 1 H), 7.62 (d, $J=8.3$ Hz, 1 H), 7.44 (d, $J=4.7$ Hz, 2 H). ^{13}C NMR (100 MHz, CDCl_3) δ 149.96, 136.22, 135.69, 130.14, 128.08, 126.67, 125.65, 123.19, 116.48, 94.80, 87.81. HRMS: $\text{C}_{13}\text{H}_7\text{BrN}_2\text{O}_2$ 计算值为 303.9672, 实测值为 303.9682。

5-乙炔基苯基-2-(4'-乙炔基吡啶基)-1-硝基苯 (24)

将 23 (88.8 毫克, 0.293 毫摩尔)、二氯化双(三苯基膦)钯 (II) (0.011 克, 0.016 毫摩尔)、碘化亚铜 (0.004 克, 0.021 毫摩尔) 和三苯基膦 (0.008 克, 0.029 毫摩尔) 溶于四氢呋喃 (4 毫升) 中配成溶液。向上述溶液加入 Et_3N (0.25 毫升, 1.76 毫摩尔) 和苯基乙炔 (0.1 毫升, 9.1 毫摩尔)。将此混合物在 56°C 下搅拌 36 小时, 在真空中将溶剂蒸除。剩余物用水稀释, 并用乙醚萃取。合并的有机层用硫酸镁干燥并过滤, 溶剂在真空中蒸除。用闪式柱层析法 (硅胶, 乙酸乙酯/己烷 20/80) 纯化, 得到目标化合物黄色固体 65 毫克 (产率 69%)。熔点: $130\text{--}132^\circ\text{C}$ 。IR (KBr) 3445.3, 3046.3, 2203.5, 1548.5, 1529.1, 1399.9, 1341.6 cm^{-1} . ^1H NMR (400 MHz, CDCl_3) δ 8.67 (br d, $J=4.9$ Hz, 2 H), 8.27 (d, $J=1.5$ Hz, 1 H), 7.76 (1/2 ABqd, $J=8.0$, 1.6 Hz, 1 H), 7.72 (1/2 ABqd, $J=8.0$, 0.5 Hz, 1 H), 7.56 (m, 2 H), 7.45 (dd, $J=5.9$, 1.7 Hz, 2 H), 7.40 (m, 3 H). ^{13}C NMR (100 MHz, CDCl_3) δ 149.58, 135.39, 134.65, 131.81, 129.34, 128.54, 127.67, 125.32, 121.85, 116.66, 95.30, 94.30, 88.52, 86.63. HRMS: $\text{C}_{21}\text{H}_{12}\text{N}_2\text{O}_2$ 计算值为 324.0899, 实测值为 324.0895。

1-溴代-4-乙炔基苯基-3-硝基苯 (25)

将 2,5-二溴硝基苯 (0.937 克, 3.34 毫摩尔)、双(二亚苄基丙酮)钯 (0.095 克, 0.166 毫摩尔)、碘化亚铜 (0.032 克, 0.168 毫摩尔)、三苯基膦 (0.173 克, 0.66 毫摩尔) 溶于四氢呋喃 (4 毫升) 中配成溶液。向上述溶液加入 Et_3N (1 毫升, 7.2 毫摩尔) 和苯基乙炔 (0.5 毫

升, 4.56 毫摩尔)。将此混合物在 23℃下搅拌 48 小时。用饱和氯化铵溶液洗涤, 并用乙醚萃取。合并的有机层用硫酸钠干燥并过滤, 溶剂在真空中蒸除。用闪式柱层析法(硅胶, CH₂Cl₂/己烷 1/8)纯化, 得到目标化合物黄色固体 0.48 克(产率 47%)。熔点: 58-74 °C。IR (KBr) 3421.9, 3085.5, 2213.4, 1595.7, 1545.9, 1521.3, 1336.5, 1269.2 cm⁻¹。 ¹H NMR (400 MHz, CDCl₃) δ 8.23 (d, J=1.9 Hz, 1 H), 7.72 (dd, J=8.3 Hz, 2.0, 1 H), 7.59 (m, 3 H), 7.40 (m, 3 H)。 ¹³C NMR (100 MHz, CDCl₃) δ 149.71, 135.91, 135.45, 131.99, 129.44, 128.46, 127.78, 122.03, 121.75, 117.69, 98.43, 84.00。 HRMS: C₁₄H₈NO₂Br 计算值为 302.9720, 实测值为 302.9725。

2-乙炔基苯基-5-(4'-乙炔基吡啶基)-1-硝基苯 (26)

将 25 (0.306 克, 1.01 毫摩尔)、碳酸钾 (0.713 克, 5.16 毫摩尔)、二氯化双(三苯基膦)钯 (0.035 克, 0.05 毫摩尔)、碘化亚铜 (0.009 克, 0.047 毫摩尔)、三苯基膦 (0.052 克, 0.198 毫摩尔) 溶于四氢呋喃(2 毫升)中配成溶液。通过套管向上述溶液中加入 21(0.217 克, 1.24 毫摩尔)在四氢呋喃(2 毫升)和甲醇(1 毫升)中的溶液。将此混合物在 60℃下加热 18 小时, 然后用旋转蒸发法去掉溶剂, 棕色的剩余物用水稀释, 并用乙醚萃取。合并的有机层用硫酸钠干燥并过滤, 溶剂在真空中蒸除。用闪式柱层析法(硅胶, 乙酸乙酯/己烷 20/80, 40/60)纯化, 得到目标化合物黄色固体 260 毫克(产率 79%)。熔点: 144-146 °C。IR (KBr) 3442.3, 3053.0, 2209.4, 1631.3, 1584.8, 1524.7, 1404.3, 1344.7, 1269.0, 826.4, 755.2, 686.6 cm⁻¹。 ¹H NMR (400 MHz, CDCl₃) δ 8.67 (dd, J=4.4, 1.6 Hz, 2 H), 8.27 (br s, 1 H), 7.74 (m, 2 H), 7.63 (d, J=1.8 Hz, 1 H), 7.60 (m, 1 H), 7.42 (m, 5 H)。 ¹³C NMR (100 MHz, CDCl₃) δ 149.99, 135.41, 134.65, 132.14, 130.19, 129.61, 128.54, 127.95, 125.50, 122.68, 122.06, 119.15, 99.67, 90.83, 90.27, 84.62。 HRMS: C₂₁H₁₂N₂O₂ 计算值为 324.0899, 实测值为 324.0897。

2,5-双(三甲基甲硅烷基乙炔基)-4-硝酸乙酰苯胺 (27)

将 6 (0.78 克, 2.3 毫摩尔)、双(二亚苄基丙酮)钯 (0.068 克, 0.118 毫摩尔)、碘化亚铜 (0.023 克, 0.012 毫摩尔)、三苯基膦 (0.123 克, 0.47 毫摩尔) 溶于四氢呋喃 (8 毫升) 中配成溶液。向上述溶液加入 Et₃N (1 毫升, 7.2 毫摩尔) 和三甲基甲硅烷基乙炔 (1 毫升, 7.0 毫摩尔)。将此混合物在 67℃ 下加热 48 小时, 然后用旋转蒸发法去掉溶剂, 棕色剩余物用水稀释, 并用乙醚萃取。合并的有机层用硫酸钠干燥并过滤, 溶剂在真空中蒸除。用闪式柱层析法 (硅胶, CH₂Cl₂/己烷 35/65) 纯化, 得到目标化合物灰白色固体 410 毫克 (产率 47%)。熔点: 162-164℃。IR (KBr) 3372.9, 2962.9, 2146.0, 1727.2, 1611.2, 1544.9, 1501.5, 1457.1, 1404.3, 1338.2, 1250.6, 1222.3, 881.9 cm⁻¹。 ¹H NMR (400 MHz, CDCl₃) δ 8.75 (s, 1 H), 8.15 (s, 1 H), 8.10 (br s, 1 H), 2.27 (s, 3 H), 0.33 (s, 9 H), 0.28 (s, 9 H)。 ¹³C NMR (100 MHz, CDCl₃) δ 168.21, 144.19, 142.41, 128.11, 123.82, 120.18, 111.52, 106.66, 106.16, 99.50, 97.44, 24.90, -0.31, -0.46。 HRMS: C₁₈H₂₄N₂O₃Si₂ 计算值为 372.1326, 实测值为 372.1326。

2,5-双(4'-乙炔基吡啶基)-4-硝基苯胺 (28)

将 27 (0.056 克, 0.15 毫摩尔)、4-碘代吡啶 (0.08 克, 0.39 毫摩尔)、碳酸钾 (0.17 克, 1.2 毫摩尔)、二氯化双(三苯基膦)钯 (II) (0.01 克, 0.015 毫摩尔)、碘化亚铜 (0.004 克, 0.021 毫摩尔)、三苯基膦 (0.016 克, 0.061 毫摩尔) 溶于四氢呋喃 (4 毫升) 中配成溶液, 向上述溶液加入甲醇 (1 毫升)。将此混合物在 60℃ 下加热 50 小时, 然后用旋转蒸发法去掉溶剂, 棕色剩余物用水稀释, 并用乙酸乙酯萃取。合并的有机层用硫酸钠干燥并过滤, 溶剂在真空中蒸除。用闪式柱层析法 (硅胶, 乙酸乙酯) 纯化, 得到目标化合物黄色固体 8 毫克 (产率 16%)。熔点: 154-160 °C。IR (KBr) 3730.2, 3438.6, 2204.8, 1592.4, 1541.1, 1409.8, 1308.5, 1249.9, 818.8 cm⁻¹。 ¹H NMR (400 MHz, CDCl₃) δ 8.67 (dd, J=4.4, 1.7 Hz, 2 H), 8.65 (dd, J= 4.5, 1.7 Hz, 2 H), 8.34 (s, 1 H), 7.44 (dd, J=4.5, 1.7 Hz, 2 H), 7.40 (dd, J=4.4, 1.6 Hz, 2 H), 6.99 (s, 1 H), 5.03 (br s, 2 H)。 ¹³C NMR (100 MHz, CDCl₃) δ 151.26, 150.03,

149.90, 139.56, 130.71, 130.52, 130.00, 125.65, 125.33, 120.33, 118.52, 106.57, 94.67, 94.19, 89.55, 87.27. HRMS: $C_{20}H_{12}N_4O_2$ 计算值为 340.0960, 实测值为 340.0958.

5 2-氨基-4-(4'-乙炔基吡啶基)-5-硝基溴代苯 (29)

将 6 (0.877 克, 8.84 毫摩尔)、碳酸钾 (1.08 克, 7.81 毫摩尔)、二氯化双(三苯基膦)钯 (0.054 克, 0.077 毫摩尔)、碘化亚铜 (0.025 克, 0.13 毫摩尔)、三苯基膦 (0.068 克, 0.26 毫摩尔) 溶于四氢呋喃 (4 毫升) 中配成溶液。通过套管将 21 (0.404 克, 2.30 毫摩尔) 在四氢呋喃 (8 毫升) 和甲醇 (3 毫升) 中的溶液加入上述溶液中。将此混合物在 23℃ 下加热 1 天, 在真空中将溶剂蒸除。剩余物用水稀释, 并用乙酸乙酯萃取。有机层用硫酸镁干燥并过滤, 溶剂在真空中蒸除。用闪式柱层析法 (硅胶, 乙酸乙酯/己烷 40/60 50/50) 纯化, 得到目标化合物黄色固体 290 毫克 (产率 39%)。熔点: 226- 228 °C。

15 IR (KBr) 3385.4, 3297.7, 3171.3, 1646.8, 1591.7, 1556.9, 1471.3, 1297.8 cm^{-1} 。 ^1H NMR (400 MHz, DMSO- d_6) δ 8.66 (br d, $J=3.8$ Hz, 2 H), 8.32 (d, $J=1.3$ Hz, 1 H), 7.53 (br d, $J=4.5$ Hz, 2 H), 7.06 (d, $J=1.3$ Hz, 1 H), 6.94 (br s, 2 H)。 ^{13}C NMR (100 MHz, DMSO- d_6) δ 151.33, 150.12, 136.44, 130.70, 129.64, 125.32, 118.13, 117.73, 106.02, 91.85, 89.72。 HRMS: $C_{13}H_8BrN_3O_2$ 计算值为 316.9800, 实测值为 316.9801。

20

4-氨基-2-(4'-乙炔基吡啶基)-1-硝基-5-(三甲基甲硅烷基乙炔基)苯

将 29 (0.310 克, 0.975 毫摩尔)、二氯化双(三苯基膦)钯 (0.035 克, 0.05 毫摩尔)、碘化亚铜 (0.011 克, 0.05 毫摩尔)、三苯基膦 (0.026 克, 0.10 毫摩尔) 溶于四氢呋喃 (10 毫升) 中配成溶液。向上述溶液加入 Et_3N (0.9 毫升, 6.5 毫摩尔) 和三甲基甲硅烷基乙炔 (0.2 毫升, 1.4 毫摩尔)。将此混合物在 60℃ 下加热 2 天, 在真空中将溶剂蒸除。剩余物用水稀释, 并用乙酸乙酯萃取。合并的有机层用硫酸镁干燥并过滤, 溶剂在真空中蒸除。用闪式柱层析法 (硅胶, 乙醚)

25

30 纯化, 得到目标化合物黄色固体 160 毫克 (产率 49%)。熔点: 145-150

°C。IR (KBr) 3451.9, 3379.1, 2960.5, 2149.5, 1620.4, 1597.9, 1545.5, 1512.2, 1317.0 cm^{-1} 。 ^1H NMR (400 MHz, CDCl_3) δ 8.65 (dd, $J=4.6, 1.5$ Hz, 2 H), 8.25 (s, 1 H), 7.44 (dd, $J=4.3, 1.5$ Hz, 2 H), 6.93 (s, 1 H), 4.90 (s, 2 H), 0.30 (s, 9 H)。 ^{13}C NMR (100 MHz, CDCl_3) δ 151.44, 149.90, 139.35, 130.65, 130.43, 125.65, 119.56, 118.06, 107.93, 104.28, 98.37, 93.70, 89.79, -0.15。HRMS: $\text{C}_{18}\text{H}_{17}\text{N}_3\text{O}_2\text{Si}$ 计算值为 335.1090, 实测值为 335.1089。

4-氨基-5-乙炔基-2-(4'-乙炔基吡啶基)-1-硝基苯 (30)

将 4-氨基-2-(4'-乙炔基吡啶基)-1-硝基-5-(三甲基甲硅烷基乙炔基)苯 (160 毫克, 0.477 毫摩尔) 溶于甲醇(15 毫升)和 CH_2Cl_2 (15 毫升), 并向此溶液加入碳酸钾 (0.66 克, 4.77 毫摩尔)。将此溶液在 23°C 下搅拌 2 小时, 反应混合物用水稀释, 并用乙酸乙酯萃取。合并的有机层用硫酸镁干燥并过滤, 溶剂在真空中蒸除。得到目标化合物黄色固体 0.11 克 (产率 88%)。由于产物太不稳定, 所以无法得到其所有性质的数据。 ^1H NMR (400 MHz, $\text{DMSO}-d_6$) δ 8.67 (dd, $J=4.5, 1.6$ Hz, 2 H), 8.12 (s, 1 H), 7.53 (dd, $J=4.5, 1.6$ Hz, 2 H), 7.03 (s, 1 H), 6.97 (br s, 2 H), 4.70 (s, 1 H)。

4-氨基-2-(4'-乙炔基吡啶基)-5-(4'-硫代乙酰基苯基乙炔基)-1-硝基苯 (31)

将 30 (0.110 克, 0.418 毫摩尔)、4-硫代乙酰基碘代苯¹⁰ (0.124 克, 0.446 毫摩尔)、二氯化双(三苯基膦)钯(II) (0.015 克, 0.021 毫摩尔)、碘化亚铜 (0.004 克, 0.021 毫摩尔)、三苯基膦 (0.014 克, 0.053 毫摩尔) 溶于四氢呋喃 (13 毫升) 中配成溶液。向上述溶液加入 Et_3N (0.4 毫升, 2.9 毫摩尔)。将此混合物在 50°C 下搅拌 2 天。反应用薄层层析(乙酸乙酯/己烷 75/25)检测。再加入二氯化双(三苯基膦)钯 (0.014 克, 0.020 毫摩尔)、碘化亚铜 (0.035 克, 0.018 毫摩尔)、三苯基膦 (0.085 克, 0.0324 毫摩尔), 反应在 60°C 下搅拌 1 天, 在真空中将溶剂蒸除。剩余物用水稀释, 并用乙酸乙酯萃取。合并的有

机层用硫酸镁干燥并过滤，溶剂在真空中蒸除。用闪式柱层析法（硅胶，乙酸乙酯/己烷 66/33）纯化，得到目标化合物黄色固体 130 毫克（产率 75%）。熔点：185-188 °C。IR (KBr) 3438.2, 3195.9, 2922.4, 1695.4, 1627.7, 1596.5, 1545.1, 1514.8, 1477.2, 1402.8, 1316.4, 1249.9 cm⁻¹。¹H NMR (400 MHz, DMSO-d) δ 8.68 (br d, J=4.0 Hz, 2 H), 8.23 (s, 1 H), 7.79 (d, J=8.1 Hz, 2 H), 7.54 (d, J=5.0 Hz, 2 H), 7.49 (d, J= 8.0 Hz, 2 H), 7.13 (br s, 2 H), 7.06 (s, 1 H), 2.46 (s, 3 H)。¹³C NMR (100 MHz, DMSO-d) δ 192.98, 153.79, 150.13, 136.28, 134.31, 132.32, 130.69, 129.67, 128.66, 125.34, 123.05, 118.70, 118.26, 105.43, 95.72, 92.51, 90.12, 85.54, 30.32。HRMS: C₂₃H₁₅N₃O₃S 计算值为 413.0834, 实测值为 413.0940。

2-(4'-乙炔基吡啶基)-4-硝基-5-苯基苯胺 (32)

将 7 (80.5 毫克, 0.241 毫摩尔)、碳酸钾 (0.151 克, 1.09 毫摩尔)、二氯化双(三苯基膦)钯(II) (0.009 克, 0.014 毫摩尔)、碘化亚铜 (0.003 克, 0.014 毫摩尔)、三苯基膦 (0.014 克, 0.053 毫摩尔) 溶于四氢呋喃 (2 毫升) 中配成溶液。通过套管将 1 (0.053 克, 0.3 毫摩尔) 在四氢呋喃 (2 毫升) 和甲醇 (1 毫升) 中的溶液加入上述溶液中。将此混合物在 70°C 下加热 3 天, 然后用旋转蒸发法去掉溶剂, 棕色剩余物用水稀释, 并用乙醚萃取。合并的有机层用硫酸钠干燥并过滤, 溶剂在真空中蒸除。用闪式柱层析法（硅胶, 乙酸乙酯/己烷 30/70）纯化, 得到目标化合物黄色固体 60 毫克（产率 79%）。熔点：187-190°C。IR (KBr) 3410.2, 3323.4, 3212.1, 2215.1, 1627.6, 1592.4, 1548.4, 1511.7, 1410.5, 1331.9 cm⁻¹。¹H NMR (400 MHz, CDCl₃) δ 8.64 (br d, J=4.8, 2 H), 8.16 (s, 1 H), 7.39 (m, 5 H), 7.27 (m, 2 H), 6.62 (s, 1 H), 5.03 (br s, 2 H)。¹³C NMR (100 MHz, CDCl₃) δ 151.23, 149.82, 140.65, 138.82, 138.19, 130.49, 128.36, 128.06, 127.52, 125.34, 116.41, 104.85, 93.24, 87.89。HRMS: C₁₉H₁₃N₃O₂ 计算值为 315.1008, 实测值为 315.1011。

1-溴代-4-(4'-乙炔基)吡啶-3-硝基苯 (34)

将 33¹ (0.84 克, 2.34 毫摩尔)、二氯化双(三苯基膦)钯 (0.083 克, 0.117 毫摩尔)、碘化亚铜 (0.022 克, 0.117 毫摩尔)、碳酸钾 (1.94 克, 14.04 毫摩尔) 溶于四氢呋喃 (4 毫升) 中配成溶液。通过
5 套管将 21 (0.451 克, 2.57 毫摩尔) 在四氢呋喃 (12 毫升) 中的溶液
和甲醇 (4 毫升) 加入上述溶液中。将此混合物在 55℃ 下加热 14 小
时, 然后用旋转蒸发法去掉溶剂, 剩余物用水稀释, 接着用盐水洗涤,
并用乙酸乙酯萃取。合并的有机层用硫酸镁干燥并过滤, 溶剂在真空
中蒸除。用闪式柱层析法 (硅胶, 乙酸乙酯) 纯化, 得到目标化合物
10 黄色固体 271 毫克 (产率 34%)。熔点: 224-229 °C。IR (KBr) 3451.7,
3351.1, 3202.6, 2206.4, 1622.9, 1588.4, 1539.0, 1474.4, 1306.7, 1249.8
cm⁻¹. ¹H NMR (400 MHz, DMSO-d) δ 8.64 (d, J= 5.7 Hz, 2 H), 8.25 (s, 1
H), 7.67 (dd, J=4.5, 1.5 Hz, 2 H), 7.59 (m, 2 H), 7.47 (m, 3 H), 7.15 (br s, 1
H), 7.03 (s, 1 H)。¹³C NMR (100 MHz, DMSO-d) δ 153.97, 149.83, 136.31,
131.67, 131.17, 130.01, 129.69, 128.99, 125.45, 121.78, 120.40, 118.06,
15 103.92, 96.13, 93.41, 88.37, 86.25。HRMS; C₂₁H₁₃N₃O₂ 计算值为 339.1008,
实测值为 339.1004。

1-溴代-3-硝基-4-(4-氨基苯基乙炔基)苯 (36)

将 1,4-二溴-2-硝基苯 (5.62 克, 20.0 毫摩尔)、二氯化双(三苯
20 基膦)钯 (0.140 克, 0.20 毫摩尔)、碘化亚铜 (0.038 克, 0.20 毫摩尔)、
三乙胺 (10.0 毫升)、四氢呋喃 (10 毫升) 以及 35 (1.170 克, 10.0
毫摩尔) 进行一般偶联步骤。将此混合物在室温下搅拌 4 小时, 在真
空中除去溶剂, 剩余物在氧化硅柱上 (用二氯甲烷作洗脱液) 进行层
析, 得到红色固体为所要产物及其异构体的混合物。然后在二氯甲烷/
25 己烷中进行二重 (two-fold) 重结晶将所要产物从混合物中分离出来,
得到细的红色针状固体 (1.561 克, 产率 49%)。熔点: 147-149°C。IR
(KBr) 3457, 3367, 2194, 1623, 1593, 1513, 1550, 1334, 1273, 1136, 834,
817, 528 cm⁻¹. ¹H NMR (400 MHz, CDCl₃) δ 8.21 (d, J=2.0 Hz), 7.67 (dd,
J=8.4, 2.0 Hz), 7.51 (d, J=8.4 Hz), 7.96 (m, AA'XX'模式的 AA'部分,
30 J=8.2, 2.7, 1.9, 0.4 Hz, 2 H), 7.93 (m, AA'XX'模式的 XX'部分, J=8.2, 2.7,

1.9, 0.4 Hz, 2 H), 3.39 (s, 2 H). ^{13}C NMR (100 MHz, CDCl_3) δ 149.27, 147.85, 135.82, 135.12, 133.71, 127.73, 120.62, 118.59, 114.63, 111.09, 100.24, 82.86。HRMS: $\text{C}_{14}\text{H}_9\text{N}_2\text{BrO}_2$ 计算值为 315.9848, 实测值为 315.9845。

5

4-(2-硝基-4-苯基乙炔基苯基乙炔基)苯胺 (37)

将 36 (0.697 克, 2.20 毫摩尔)、二氯化双(三苯基膦)钯 (0.062 克, 0.088 毫摩尔)、碘化亚铜 (0.0084 克, 0.044 毫摩尔)、三乙胺 (10.0 毫升) 以及乙炔苯 (0.306 克, 3.00 毫升) 进行一般偶联步骤。将此混合物在 80℃ 下搅拌 2 小时, 在真空中除去溶剂, 剩余物在氧化硅柱上用二氯甲烷进行层析, 得到红色针状所要产物 (0.72 克, 产率 97%)。熔点: 166-168℃。IR (KBr) 3454, 3381, 3360, 2177, 2197, 1594, 1623, 1539, 1520, 1299, 1342, 1133, 829, 758, 690, 527 cm^{-1} 。 ^1H NMR (400 MHz, CDCl_3) δ 8.20 (dd, $J=1.6, 0.3$ Hz), 7.66 (dd, $J=8.2, 1.6$ Hz), 7.61 (d, $J=8.1$ Hz), 7.52-7.57 (m, 2 H), 7.36-7.43 (m, 5 H), 3.94 (s, 2 H)。 ^{13}C NMR (100 MHz, CDCl_3) δ 148.93, 147.81, 135.12, 134.04, 133.76, 131.74, 129.04, 128.49, 127.59, 122.97, 122.18, 118.95, 114.64, 111.29, 100.75, 93.03, 87.05, 83.71。HRMS: $\text{C}_{22}\text{H}_{14}\text{N}_2\text{O}_2$ 计算值为 338.1055, 实测值为 338.1058。

20

四氟硼酸 4-(2-硝基-4-苯基乙炔基苯基乙炔基)苯重氮盐 (38)

按照一般重氮化步骤将 37 (0.0845 克, 0.250 毫摩尔) 在乙腈 (2 毫升) / 环丁砜 (2 毫升) 中用 NOBF_4 (0.0322 克, 0.275 毫摩尔) 处理。产物用乙醚 (12 毫升) 处理, 沉淀出暗橙色鳞片状产物。再用乙醚洗涤此盐, 然后从 DMSO (0.5 毫升) 和 CH_2Cl_2 (20 毫升) 中重结晶, 得到有光泽的暗橙色片状产物 (0.0885 克, 产率 81%)。IR (KBr) 3103, 2279, 2209, 1576, 1345, 1540, 1084, 841, 764 cm^{-1} 。 ^1H NMR (400 MHz, $\text{CDCl}_3/\text{DMSO}-d_6$, 观察的线宽约为 1.9 Hz) δ 8.78 (d, $J=8.9$ Hz, 2 H), 8.30 (s, 1 H), 8.03 (d, $J=8.9$ Hz, 2 H), 7.85-7.92 (m, 2 H), 7.57-7.60 (m, 2 H), 7.42-7.44 (m, 3H)。 ^{13}C NMR (100 MHz, $\text{CDCl}_3/\text{DMSO}-d_6$) δ 149.00,

30

135.46, 134.85, 134.15, 133.31, 132.84, 1.34, 129.13, 128.21, 127.15, 125.66, 121.06, 114.81, 114.25, 94.57, 94.42, 94.11, 86.29。

4-(3-硝基-4-苯基乙炔基苯基乙炔基)苯胺 (39)

5 将 25 (1.208 克, 4.0 毫摩尔)、二氯化双(三苯基膦)钯 (0.070 克, 0.10 毫摩尔)、碘化亚铜 (0.019 克, 0.10 毫摩尔)、三乙胺 (6.0 毫升)、四氢呋喃 (6.0 毫升) 以及 35 (0.479 克, 4.10 毫升) 进行一般偶联步骤。将此混合物在室温下搅拌 15 小时, 在真空中除去溶剂, 剩余物在短的氧化硅柱上用二氯甲烷/己烷 (1:1) 进行层析, 得到所
10 要产物橙色固体 (0.560 克, 产率 44%)。熔点: 175-177°C。IR (KBr) 3303, 2985, 1696, 1587, 1522, 1406, 1314, 1243, 1153, 1060, 839, 757, 692 cm^{-1} 。 ^1H NMR (400 MHz, CDCl_3) δ 8.16 (t, $J=1.0$ Hz, 1H), 7.64 (d, $J=1.0$ Hz, 2H), 7.58-7.61 (m, 2H), 7.34-7.40 (m, 3H), 7.35 (m, AA'XX'模式的 AA'部分, $J=8.0, 2.5, 2.0, 0.4$ Hz, 2 H), 6.65 (m, AA'XX'模式的 XX'部分, $J=8.0, 2.5, 2.0, 0.4$ Hz, 2 H), 3.91 (s, 2H)。 ^{13}C NMR (100 MHz, CDCl_3) 149.4, 147.5, 134.9, 134.3, 133.3, 132.0, 129.3, 128.5, 127.1, 124.9, 122.3, 117.1, 114.7, 11.1, 98.4, 94.9, 85.3, 85.0。 HRMS; $\text{C}_{22}\text{H}_{14}\text{N}_2\text{O}_2$ 计算值为 338.1055, 实测值为 338.1059。

20 四氟硼酸 4-(3-硝基-4-苯基乙炔基苯基乙炔基)苯重氮盐 (40)

按照一般重氮化步骤将 39 (0.0676 克, 0.200 毫摩尔) 在乙腈 (2 毫升) / 环丁砜 (2 毫升) 中用 NOBF_4 (0.025 克, 0.210 毫摩尔) 处理。产物用乙醚 (20 毫升) 处理, 沉淀出细的橙红色晶体。再用乙醚洗涤此盐, 然后从 DMSO (0.6 毫升) 和 CH_2Cl_2 (10 毫升) 中再次沉淀,
25 得到有强烈光泽的红色片状产物 (0.0676 克, 产率 77%)。IR (KBr) 3101, 2279, 2209, 1576, 1540, 1346, 1083, 1034, 840, 764 cm^{-1} 。 ^1H NMR (400 MHz, $\text{CDCl}_3/\text{DMSO}-d_6$) δ 7.94 (m, AA'XX'模式的 AA'部分, $J=8.7, 2.4, 1.7, 0.4$ Hz, 2 H), 7.82 (dd, $J=1.7, 0.4$ Hz, 1 H), 7.49 (m, AA'XX'模式的 XX'部分, $J=8.7, 2.4, 1.7, 0.4$ Hz, 2 H), 7.62 (dd, $J=8.1, 1.7$ Hz, 1 H), 7.56
30 (dd, $J=8.1, 0.4$ Hz, 1 H), 7.07 (m, AA'XX'Y 模式的 AA'部分, $J=7.8, 7.6,$

1.8, 1.3, 1.3, 0.6 Hz, 2 H), 6.94 (tt, $J = 7.6, 1.3$ Hz, 1 H), 6.91 (m, AA'XX'Y 模式的 YY' 部分, $J = 7.8, 7.6, 1.8, 1.3, 1.3, 0.6$ Hz, 2 H). ^{13}C NMR (100 MHz, $\text{CDCl}_3/\text{DMSO}-d_6$) δ 137.24, 136.97, 136.23, 135.40, 133.72, 133.00, 131.08, 129.96, 129.48, 122.81, 122.75, 120.68, 114.12, 100.47, 98.81, 91.04, 85.57。

4-(2,5-二硝基-4-(4-氨基苯基乙炔基)苯基乙炔基)苯胺 (42)

1,4-二溴-2,5-二硝基苯¹² (0.977 克, 3.0 毫摩尔)、二氯化双(三苯基磷)钯 (0.042 克, 0.06 毫摩尔)、碘化亚铜 (0.011 克, 0.06 毫摩尔)、三乙胺 (5.0 毫升)、四氢呋喃 (5.0 毫升) 以及 4-乙炔基苯胺 (0.468 克, 4.00 毫升) 进行一般偶联步骤。将此混合物在室温下搅拌 12 小时, 在真空中除去溶剂, 剩余物用二氯甲烷 (10 毫升) 超声处理, 并过滤。过滤所得固体再用二氯甲烷 (10 毫升) 洗涤五次, 真空中干燥, 得到二胺 42 的黑紫色晶体 (0.432 克, 产率 36%)。熔点 $>270^\circ\text{C}$ 。IR (KBr) 3494, 3387, 2184, 1600, 1400, 1523, 1537, 1308, 1337, 1251, 1136 cm^{-1} 。 ^1H NMR (400 MHz, $\text{DMSO}-d_6$) δ 8.37 (s, 2 H), 7.27-7.29 (m, 2 H), 6.59-6.61 (m, 2 H), 5.93 (br s, 4 H)。 ^{13}C NMR (100 MHz, $\text{DMSO}-d_6$) δ 151.18, 149.89, 133.67, 129.43, 116.95, 113.66, 106.10, 103.45, 82.23。 HRMS: $\text{C}_{22}\text{H}_{14}\text{N}_4\text{O}_4$ 计算值为 398.1015, 实测值为 398.1018。

四氟硼酸 4-(2,5-二硝基-4-(4-重氮磷苯基乙炔基)苯基乙炔基)苯重氮盐 (43)

按照重氮化的一般步骤, 用 NOBF_4 (0.128 克, 1.10 毫摩尔) 在乙腈 (5.0 毫升) / 环丁砜 (5.0 毫升) 中处理 42 (0.199 克, 0.500 毫摩尔)。产物用乙醚 (20 毫升) 沉淀。再用乙醚洗涤所得的盐, 从 DMSO 和二氯甲烷中再次沉淀得到光敏黄色的晶体 (0.215 克, 产率 72%)。IR (KBr) 3107, 2291, 1579, 1546, 1342, 1078, 830 cm^{-1} 。 ^1H NMR (400 MHz, $\text{CDCl}_3/\text{DMSO}-d_6$) δ 8.85 (s, 2H), 8.79 (d, $J = 9$ Hz, 2 H), 8.20 (d, $J = 9$ Hz, 2 H)。 ^{13}C NMR (100 MHz, $\text{CDCl}_3/\text{DMSO}-d_6$) δ 150.60, 133.93, 133.83, 133.14, 132.40, 131.75, 117.62, 116.32, 96.91, 91.51。

已合成了许多包含有以醌和硝基为中心的可逆还原性官能团的
(亚苯基亚乙炔基)低聚体。这些分子从标准的保护的硫代基团到新的
的重氮盐和吡啶基键合都有方法可以附着到金属的表面。

5

实施例 2

含吡啶单元的分子电子装置

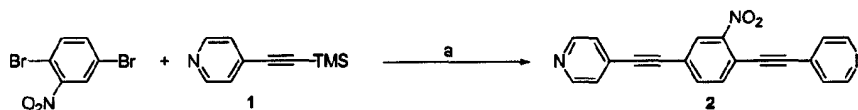
图 6 显示的是两组已被合成的潜在分子装置。第一组在内部苯环
上有硝基官能团,其被设计用来保存电子使分子可以作为存储器元
件。

10

第二组具有硝基和氨基,即便在较低的温度下也有相似的功用。

潜在分子装置 2 和 4 设计为含有两个吡啶端基,以作为金连接的
交联剂。

15



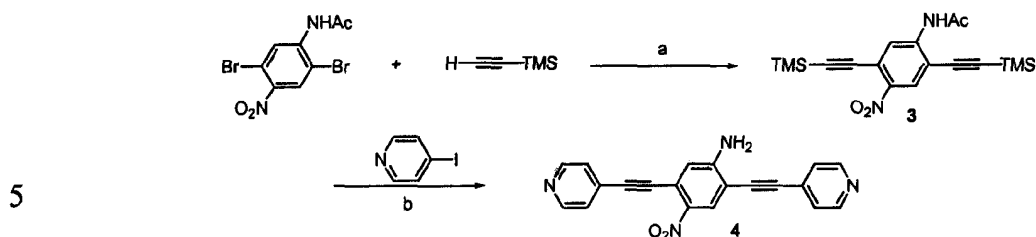
20

路线 2 (a) K_2CO_3 , MeOH , $\text{Pd}(\text{PPh}_3)_2\text{Cl}_2$, PPh_3 , CuI , THF , 64°C , 20 h,
24%

路线 2 概述了由 2,5-二溴硝基苯合成 2 的过程。可通过 4-碘代吡
啶⁷和三甲基甲硅烷基乙炔发生 Sonogashira 偶联⁶,很容易地制得 1
(99%)。碳酸钾用来作为原位去除 TMS 保护和偶联的碱,几个小
时后分解出自由的炔。室温下进行此反应得到大量的双(乙炔基吡啶),
且在二溴芳基的一个位点发生偶联。

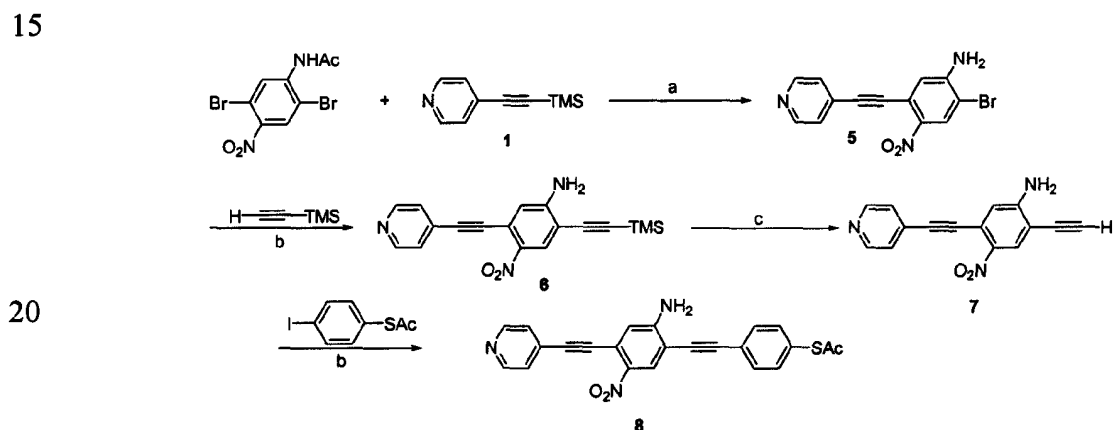
25

30



路线 2 (a) Et_3N , $\text{Pd}(\text{dba})_2$, PPh_3 , CuI , THF, 60°C , 48 h, 47%. (b) K_2CO_3 , MeOH, $\text{Pd}(\text{PPh}_3)_2\text{Cl}_2$, PPh_3 , CuI , THF, 60°C , 50 h, 16%

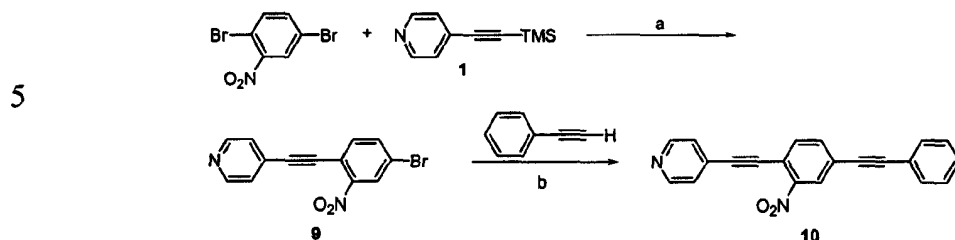
10 化合物 4 与 2 相似，但是以硝基苯胺为中心取代硝基苯为中心。不象潜在分子装置 2，4 的合成（路线 2）始于 2,5-二溴-4-硝基乙酰苯胺⁹与三甲基甲硅烷基乙炔的偶联反应给出 3，接着 3 与 4-碘代吡啶偶联，产率较低。这个低产率的偶联反应可能是由于硝基与炔单元的环化作用。



路线 3 (a) K_2CO_3 , MeOH, $\text{Pd}(\text{PPh}_3)_2\text{Cl}_2$, PPh_3 , CuI , THF, rt, 24 h, 39%. (b) Et_3N , $\text{Pd}(\text{PPh}_3)_2\text{Cl}_2$, PPh_3 , CuI , THF, 60°C . (c) K_2CO_3 , MeOH, CH_2Cl_2 , rt, 2 h, 88%

8 的合成如路线 3 所示。8 有一个被保护的苯硫酚端基，其可连在金表面上。分子的另一端是吡啶基，其有可能作为比苯基更好的顶层连接基。8 的合成是用 2,5-二溴-4-硝基乙酰苯胺和 1 偶联以中等的产率得到化合物 5。接着用 5 与三甲基甲硅烷基乙炔的偶联得到 6，

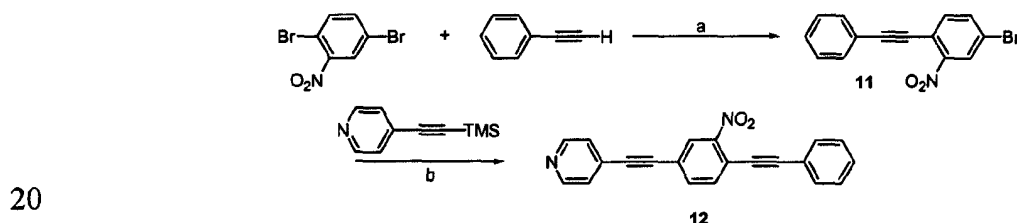
产率 49%，再用碳酸钾脱保护给出 7。合成的最后一步是与 4-硫代乙酰基碘苯偶联，得到较高产率的潜在装置 8（75%）。



路线 4 (a) K_2CO_3 , MeOH, $\text{Pd}(\text{PPh}_3)_2\text{Cl}_2$, PPh_3 , CuI, THF, rt, 2 d, 71%.

10 (b) Et_3N , $\text{Pd}(\text{PPh}_3)_2\text{Cl}_2$, PPh_3 , CuI, THF, 56°C , 36 h, 69%

10 和 12 的合成是为了研究在自组装过程中，硝基位置相对于“接线夹”的位置的重要性。10 的硝基取向朝向吡啶基（路线 4），10 的合成是先通过偶联 1 和 2,5-二溴硝基苯，并在原位去除 TMS 基团以较高产率给出 9。9 与苯基乙炔偶联得到 10。



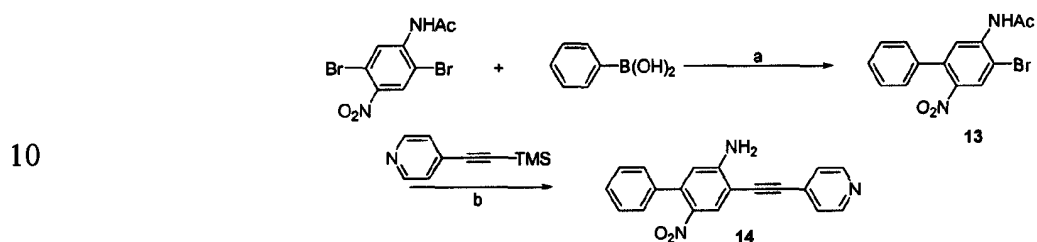
路线 5 (a) Et_3N , $\text{Pd}(\text{dba})_2$, PPh_3 , CuI, THF, rt, 48 h, 47%. (b) K_2CO_3 , MeOH, $\text{Pd}(\text{PPh}_3)_2\text{Cl}_2$, PPh_3 , CuI, THF, 64°C , 18 h, 79%

25 12 硝基指向远离吡啶基的方向，其合成（路线 5）与合成 10 的方法相似，除了步骤是相反的。这里，苯基乙炔先与 2,5-二溴硝基苯反应以中等的产率得到 11。1 接着再与 11 偶联得到 12，产率很高。

30 为了以最小的禁阻转移电子，这些有机低聚体最好是其所有苯环都在同一平面上。如果末端的苯基乙炔基被苯基取代，分子则变得稍

稍扭曲。为了研究这种转动势垒的影响，合成了 14。2,5-二溴-4-硝基乙酰苯胺与苯基硼酸的 Suzuki 偶联被用来合成化合物 13（路线 6），13 接着与 4-(三甲基甲硅烷基乙炔基)吡啶（1）偶联得到 14。

5 化合物 2、4、8、10、12 和 14 的结构用 IR、 ^1H NMR、 ^{13}C NMR 和 MS 确定。



路线 6 (a) $\text{Pd}(\text{dba})_2$, PPh_3 , Cs_2CO_3 , 甲苯, 67°C , 3 d, 51%. (b) K_2CO_3 , MeOH , $\text{Pd}(\text{PPh}_3)_2\text{Cl}_2$, PPh_3 , CuI , THF , 70°C , 3 d, 79%

15

总之，使用钯催化的偶联完成了用于分子电子装置的含吡啶单元的共轭芳香族分子的合成。

实施例 3

20 负微分电阻

再次参见图 3，在示例分子二极管 30 中观察到负微分电阻，尤其是在单硝基取代的低聚亚苯基 32，特别是 4,4'-二亚苯基亚乙炔基-2'-硝基-1-苯硫酚的分子中，以及在二硝基取代的低聚亚苯基 34，特别是 2',5'-二硝基-4,4'-二亚苯基亚乙炔基-1-苯硫酚的分子中。

25

现在参见图 4A 和 4B，其中画出了图 2 中所示分子的 $I(V)$ 响应曲线（其中 I 表示电流并且 V 表示电压）。这些曲线图通过测量分子 32 和分子 34 的自组装单分子层的响应来获得。在每个单分子层中，利用硫醇取代的端部连接金引线并且未被取代的相对端连接第二条金引线来定向该分子。

30

现在特别地参考图 4A, 对于分子 32, 起初该 I (V) 响应是处于“0”状态 (空心圆)。一旦施加 1.75V 脉冲, 该分子设成一种呈现负微分电阻 (NDR) 行为的新状态, “1” (黑圆), 此时电流随着电压的增加先升后降。

现在特别地参考图 4B, 对于分子 34, 起初该 I (V) 响应是处于呈现 NDR 的“1”状态 (实心圆)。一旦施加 1.5 V 脉冲, 该分子被设置成为一种新的状态, “0” (空心圆)。该初始状态通过施加负偏压而恢复。这是对分子 32 观察的初始 / 最后开关的反转, 如图 3A 所示。然而, 每个行为都是开关状态对偶性的示例。分子 34 的一个优点在于它是一个在室温下呈现负微分电阻的分子。此外, 开关状态的保持被观察 24 个小时。可以相信, 通过系统提高封装, 可以具有更长的保持时间。优选纳米单元 12 是全封闭的以提高该开关状态稳定性更长时间。

在图 4B 中显示的 NDR 曲线用于动态的纳米单元模拟以及如下所述的 SPICE 模拟。

实施例 4

本发明人已经发现: 基于纳米网络的模拟纳米单元可被训练以用作例示的公知逻辑装置, 其中该纳米网络包含通过纳米颗粒连接而排列的分子开关。分子开关是呈现以负微分电阻为特征的 I (V) 响应的分子。

可以相信模拟纳米单元表示实际的物理纳米单元。因此, 可以相信, 这是首次发现了一种用于编程一个实际纳米单元的技术。本发明人知道没有其它通过包含具有这些 I (V) 特征的“树枝状晶体”(使用传统对脑结构的模拟) 的网络来学习逻辑的演示。特别地, 脑的传统神经网络模型及其它模拟系统通常基于具有选自阶跃函数、双曲线

切线等的“树枝状晶体” $I(V)$ 曲线的系统的表示，这些都不具有负的微分电阻。

5 遗传算法用来全知地训练模拟的纳米单元。也就是说，该算法知道远程分子开关的状态。该算法通过全能开关训练该纳米单元，也就是说通过直接调整该开关的状态来训练纳米单元。尽管如此，可以相信这些结果表示可通过自适应算法获得的结果，该自适应算法通过调整在输入输出引线处的电压以人类方式构造远程分子开关。

10 一般编程

编程或者训练纳米单元的目的在于获取一个随机、固定的纳米单元并且打开或者关闭它的开关直到它起一个目标逻辑装置作用为止。每个分子开关的物理位置首先是固定的；也即该纳米单元的内部拓扑结构是静态的。该纳米单元在制造之后被接着训练。仅仅能够改变分子开关的“on”或者“off”状态。

20 在此我们引入与所用的编程算法有关的术语全知、全能以及人类开关。对于全知，我们意指在该纳米单元内的连接以及每个开关的位置和状态是已知的。全能意指检索算法知道每个分子开关的位置而且可以精确和有选择性的访问以可逆地设置它的“on”或者“off”状态。自然地，全能的定义包含全知。最后，对于人类开关，该算法不知道在该纳米单元内的连接或者开关的位置，而且开关局限于施加至输入/输出管脚的电压脉冲。一个实际的物理纳米单元优选以一种人类方式编程，而且仅仅经由沿着该外围的接触垫片之间的电压脉冲进行开关。

30 在此处给出的模拟中，我们指明存在如下开关状态，使得给定的纳米单元起一个目标逻辑装置的作用。给定纳米颗粒以及分子开关的一定密度，通过假定对开关状态可以绝对控制，可以确定任何随机的纳米单元是否能够被训练为某些目标逻辑装置。某些用于把该方法扩

展到人类开关的初级策略包含利用金颗粒的电容来更好地访问个人开关。可以相信在两个 I / O 管脚之间的一系列分子开关，其中确有一个开关以在两个金颗粒之间的某些电容，能够通过使用这些电容来设置成“on”和“off”状态的任何模式。在一个纳米单元内的分子开关和金颗粒网络比 I/O 管脚之间的一个简单列的开关要复杂得多；然而，模拟表明，用于某些逻辑门的解空间是非常密集的。这意味着不必要唯一地访问每个单个分子。实际上，如果在两个金颗粒之间有多个开关，则以一个方向定向的每个开关同时切换状态。然而，这将不会是一个问题，因为切换分子组很可能是足够的。

10

具有全能的纳米单元训练问题是一个组合最优化问题，其中搜索空间是一组用于某些既定纳米单元的所有可能切换状态。如果一个纳米单元包含 250 个纳米颗粒以及大约 750 个以一个适当方向用于切换的分子开关，则这个搜索空间的大小是 2^{750} （作为一个大小比较，在宇宙中的基本粒子数目估计为 2^{300} ）。遗传算法用来搜索这个空间。首先产生一个随机纳米单元并且定义一个目标逻辑装置（比如 NAND）。纳米单元开关的状态被存储为“1”和“0”的“染色体”。产生随机染色体的一个初始一代。每个染色体都对应于一个不同组的纳米单元开关状态，具有纳米颗粒和分子开关的固定位置。形成一个拟合函数，使得导致该纳米单元用作目标逻辑装置的开关状态接收低分，同时没有执行目标逻辑功能的那些开关状态接收高分。当一个开关状态的染色体获得零分时停止搜索，而且因此可接受地执行期望的逻辑。在第一代之后，通过在上代执行操作产生新染色体的每一代。高度拟合或者低得分的染色体成对组合以形成新的并且有希望更好地执行的染色体。用这样的方式，搜索该空间直到获得拟合为零的染色体为止。

20

25

在此我们给出两种模拟这个万能训练过程的方法。为了计算每个结构的拟合度，或者一个纳米单元内的开关状态组合，必须分析一系列的电路。这些电路中的每一个包含一个非线性电阻复合网络。输入

30

电压随时间变化的模式施加于某些输入 / 输出管脚，而且必须计算产生的输出电流随时间的变化。这涉及求解一系列非线性常微分方程。尽管求解这个系统是困难的，但是在此给出的模拟以两种方法解决这个问题。在我们给出的第二种模型中，电路工程软件，SPICE，用来分析纳米单元的每个结构。这个软件是高度准确但是费时的，因为它没有被设计成执行随机组合电路的迭代。在我们首先给出的动态纳米单元模型中，为了速度起见牺牲准确度；对纳米单元的电行为取近似，从而不求解复杂的方程式系统，但是能够获得一个有用的近似。

10 使用的遗传算法记录在附录中的文件 vzNanocell.cpp 中。

动态纳米单元模型

蜂窝式自动装置（CA）是具有时空离散值的动态系统。根据仅仅依赖于本地或者邻近蜂窝[c]状态的决定性规则同步地更新在一个常规点阵中的蜂窝状态。虽然每个蜂窝的状态经常局限于小的离散值集合，但扩展 CA 的概念以允许一个真实值的状态变量[d]也不罕见。动态纳米单元模型是一个其中六方点阵表示该纳米单元而且在该点阵中蜂窝表示单个金纳米颗粒的蜂窝式自动装置。用于一个蜂窝的真实值状态变量是该纳米颗粒的电位，而且用于在每个时间步长处改变状态变量的跃迁规则是调整该纳米颗粒的电位以使它 Kirchhoff 顺从于它的邻近蜂窝。已经说过，计算机科学家在物理学家使用受“场方程”控制的场论的地方使用蜂窝自动装置，而且使用 CA 的提供了一种可以优于传统方法许多数量级[a] [b]的替代计算方法。我们相信该动态纳米单元模型允许我们的检索算法以一种及时的方式执行而且仍然准确地模拟物理装置的电特性。

30 用于该动态纳米单元模型的跃迁规则考虑了在该 NDR 装置中的 I（V）曲线的非线性并且允许该模型模拟通过该纳米单元的电流，而不仅仅是流体流动。这提供了模拟更感兴趣的逻辑装置比如具有否定逻辑的那些逻辑装置的能力。

该动态模型以如下所述的一种递增方式计算。所有的金属纳米颗粒都用 0 电位初始化，然后将一个非零的电势施加于某些已经被指定作为输入 / 输出点的纳米颗粒。施加于该输入点的电位逐渐地增大直到它们到达表示到该纳米单元的布尔值输入的电平为止，然后贯穿该模拟过程保持不变。被影响的纳米颗粒发信号告知它们的邻近颗粒已经出现了一个改变。然后通过把纳米颗粒的电位和它们紧邻颗粒中每一个的电势进行比较，来重新计算它们自己的电势。每个相邻颗粒的电压差异连同介于其间的分子开关 $I(V)$ 特性一起确定流到每个相邻颗粒或者从每个相邻颗粒流出的电流数量。如果从某些相邻颗粒流入的电流总和不等于流出到其余相邻颗粒的电流总和，则因此调整该纳米颗粒的电位。如果执行调整，则相邻颗粒的纳米颗粒被发信号告知应重新计算它们的电势。继续这个过程直到纳米颗粒确信它们的进入电流等于它们的输出电流为止，由此使该系统 Kirchhoff 一致。最后，在每个输入 / 输出处计算电流。

遗传算法用来查找使该纳米单元用作期望的逻辑门的开关设置组合。因为开关要么在一个“on”要么在“关闭”状态，所以染色体模型是一组值为 0 或者 1 的数位，分别表示该蜂窝中的所有开关的状态，“off”或者“on”。采用了六角形阵列。也就是说纳米颗粒放置在三角形的角部，并且开关沿着三角形的边。遗传算法能够查找开关设置的一个组合以构造一个小的、 4×4 的纳米单元充当一个 XOR 装置。此外，已经指明，在一个具有半径为 1 的“圆”纳米单元（即由六个纳米颗粒的大致圆围绕的一个中心纳米颗粒）中，该纳米单元能够被训练用作 8 个 2×1 真值表（2 个输入、1 个输出）中的任何一个。更进一步，已经指明，在一个具有半径为 2 的“圆”纳米单元（即由 12 纳米颗粒组成的另一个大致圆围绕的上述半径为 1 的纳米单元）中，该纳米单元能够被训练用作 64 个 2×2 真值表（2 个输入、2 个输出）中的任何一个。这个动态纳米单元模型是简单的而且它相对快速地执行，从而使之成为一种用于研究该纳米单元的搜索技术和逻辑属

性的优秀工具。

下述文献的每一个在此引入作为参考：

5 [a] T. Toffoli, N.H.Margolus; "Invertible Cellular Automata: A Review", Cellular Automata: Theory and Experiment, H. Gutowitz 编辑; 1991, A Bradford Book, The MIT Press, Cambridge, Massachusetts, London England.

10 [b] T. Toffoli; "Cellular Automata as an Alternative to (Rather than An Approximation of) Differential Equations in Modeling Physics. PHYSICA D, Nonlinear Phenomena, Vol 10D (1984) Nos. 1 & 2, 1984 年 1 月

[c] H. Gutowitz; "Introduction", Cellular Automata: Theory and Experiment, H. Gutowitz 同[a]

15 [d] Chopard, Droz; Cellular Automata Modeling of Physical Systems; 1998, Cambridge University Press

SPICE 模型

SPICE 模型

20 SPICE 模型模拟一个纳米单元的复杂装置电路属性。我们构造 SPICE 以与在先前部分中描述的遗传算法进行接口。使用微软公司的 COM 平台，通过 OLE 和 Intusoft 公司的 ICAPS / 4 Windows SPICE 变体进行接口，形成一个纳米单元模拟器。还利用可以从 Avant 获得的 HSPICE v. 1999.2 执行计算。纳米单元模拟器随机地产生纳米单元并且把它们配置为起简单逻辑门的作用。给定纳米颗粒的密度和维数以及分子开关的平均密度，产生一个随机的纳米单元作为具有具体选择密度的金属颗粒六边形网络。连接相邻纳米颗粒的分子开关遵循基于给定平均密度的泊松分布（图 7）。在创建一个纳米单元之后，限定围绕输入 / 输出管脚（五个管脚占据四边中的每一个）的设置 20。每个输入 / 输出管脚都能够被设置为输入、输出、或者浮动并且因此具有类似一个纳米颗粒的性质。在 SPICE 引擎内部，使用非线性电阻电

25

30

路元件模拟单个分子。通过在该纳米颗粒之间包含预期的寄生电容来实现在 SPICE 中的收敛。该增加的电容阻止在模拟过程中出现电流突变，其更现实地模拟纳米单元体系结构并且有助于收敛。

5 在上述工作中，逻辑门是电压输入和电流输出电路。当把输入 / 输出管脚设置为“高”或者“低”时，我们令 V_{IL} 和 V_{IH} 分别为输入管脚的低和高电压。当一个输入的真值表值是 1 时， V_{IH} 伏电压施加于这个管脚。真值表值 0 表示施加了 V_{IL} 伏电压。类似地，我们令 I_{OL} 和 I_{OH} 分别为输出电流阈值。如果通过一个输出管脚的电流等于或低于 I_{OL} ，则管脚被认为是“off”，而且如果该电流等于或高于 I_{OH} ，则该管脚被认为是“on”。

15 给定多个双态的输入输出，一个真值表描述了期望的逻辑。测试每个真值是不够的。在真值之间的每次转变必须同样被测试。用于一个反相器、NAND、和半加法器反相的输入图形和相应真值表在一个稍后部分中显示。

20 通过从 SPICE 中分析输出，我们在每个时钟步骤处确定纳米单元的输出。我们然后把这些读数和 I_{OH} 和 I_{OL} 进行比较以确定输出管脚是否是“on”、“off”、或者都不是（在离散阈值设空值之间）。以这种方式，我们确定任何给定纳米单元的逻辑。通过把这个逻辑和期望的真值表进行比较，我们能够确定该纳米单元是否执行期望的逻辑功能。

25 对于一个固定的纳米单元，在开关状态的所有可能组合当中进行搜索是所希望的，其中每个开关的位置是固定的。换句话说，在纳米颗粒之间不能增加新的开关，而且现有的开关不能被去除。仅仅开关状态能够被改变。在 SPICE 模型中，管脚设置以及“on”和“off”电流阈值（ I_{OL} 和 I_{OH} ）是不变的。目的在于固定这些参数，使得特定纳
30 米颗粒和分子开关密度的任何随机纳米单元能够被训练为某些目标逻辑

辑门。对反相器、NAND 门和反相半加器确定示例设置。

5 在一个未编程状态中的一个示例纳米单元的代表性 SPICE 列表记录在附录的文件 Nanocell.doc 中。在图 7 中显示了未编程纳米单元的分子开关的 “on” - “off” 状态。

10 重新编程为用作反相器的同一个纳米单元的代表性 SPICE 列表被记录在附录的文件 Trained Nanocell.doc 中。在图 8 中显示了用作被编程反相器的纳米单元的 “on” - “off” 状态。

编程为用作 NAND 门的同一个纳米单元的代表性 SPICE 列表被记录在附录的文件 Trained Nanocell.doc 中。在图 9 中显示了用作被编程的 NAND 门的纳米单元的 “on” - “off” 状态。

15 编程为用作反相半加法器的同一个纳米单元的代表性 SPICE 列表被记录在附录的文件 Trained Nanocell.doc 中。在图 10 中显示了用作被编程的反相半加法器的纳米单元的 “on” - “off” 状态。

20 上述结果表明在图 7 中显示的示例纳米单元的可编程性和可重新编程性。

25 虽然已经显示和描述了本发明的优选实施例，但本领域技术人员能够对其进行改动而不背离本发明的精神或者教导。在此描述的实施例仅仅是例示性的而不是限制性的。装置、计算机、和方法的许多变化和改动是可能的，而且都落在本发明的范围之内。因此，保护范围不局限于在此描述的实施例，而是仅仅由所附的权利要求所限制，其范围应当包含权利要求技术主题的所有等同。

附页

adjust.cpp

```

/*
*****
*****
adjust.cpp
Van Zandt
*****
*/

#include "vzComputer.h"

void main()
{
    vzNanoCell *myCell;
    myCell= new vzNanoCell;

        StartGraphics();
        SetTimer(0);

    myCell->DoCreate(5,4,30,30,80,90);
    myCell->SetInReg("00100");
    myCell->SetOutReg("00000");
    myCell->DoDraw();
        readkey();

        rectfill(screen,0,0,5,5,WHITE);
        myCell->DoAdjust();
        rectfill(screen,0,0,5,5,BLACK);
    readkey();

    delete myCell;

        StopGraphics();

    return 0;
}

//*****
*****

```

conduct.cpp

```
/*
*****
*****
conduct.cpp
Van Zandt
*****
*****
*/

#include "vzComputer.h"

void main()
{
    vzNanoCell *myCell;
    myCell= new vzNanoCell;

        StartGraphics();
        SetTimer(5);

    myCell->DoCreate(5,9,30,30,75,75);
    myCell->SetInReg("01000");
    myCell->SetOutReg("01010");
    myCell->DoDraw();
        readkey();

        myCell->DoConduct();
    readkey();

    delete myCell;

        StopGraphics();

    return 0;
}

//*****
*****
```

draw.cpp

```
/*
*****
Draw.cpp
Van Zandt
*****
*/
#include "vzComputer.h"

void main()
{
    StartGraphics();

    line(screen,50,50,70,70,WHITE);
    line(screen,51,50,71,70,WHITE);
    line(screen,52,50,72,70,WHITE);
    circlefill(screen,50,50,10,RED);
    circlefill(screen,70,70,10,RED);

    line(screen,50,150,70,170,WHITE);
    circlefill(screen,50,150,10,LIGHTRED);
    circlefill(screen,70,170,10,LIGHTRED);

    readkey();
    StopGraphics();
}
```

flood.cpp

```
/*
*****
*****
flood.cpp
Van Zandt
*****
*****
*/

#include "vzComputer.h"

void    main()
{
    vzNanoCell  *myCell;
    myCell= new vzNanoCell;

        StartGraphics();
        SetTimer(1);

    myCell->DoCreate(5,4,30,30,80,80);
    myCell->SetInReg("00100");
    myCell->SetOutReg("01010");
    myCell->DoDraw();
    save_bitmap("flood1.bmp",screen,desktop_palette);
        readkey();

    //myCell->DoAdjust();
    //myCell->DoDraw();

        myCell->DoFlood();
    save_bitmap("flood2.bmp",screen,desktop_palette);
    readkey();

    delete myCell;

        StopGraphics();

    return 0;
}

//*****
*****
```

gal.cpp

```

/*
*****
*****
gal.cpp
Van Zandt
*****
*****
*/

#include "vzComputer.h"
#include "vzPopulation.h"
#include <iostream.h>

    vzNanoCell      *myCell;
        double      theEvaluator(vzIndividual *iPtr);

void    main()
{
    int                i;
    char                buffer[80];
    vzPopulation      *myPopulationPtr;

    StartGraphics();

    myCell= new vzNanoCell;

    SetTimer(0);

    myCell->DoCreate(5,4,30,30,80,90);

    myPopulationPtr = new vzPopulation(500, myCell->GetSwitchCo
unt(), 2, theEvaluator);

    myCell->DoClear();
    myCell->SetInReg("00100");
    myCell->SetOutReg("00000");
    myCell->DoDraw();
    readkey();

    myCell->DoFlood();
    save_bitmap("flood1.bmp", screen, desktop_palette);
    readkey();

    rectfill(screen,0,0,SCREEN_W,SCREEN_H,WHITE);
    myPopulationPtr->BuildRandom();
    myPopulationPtr->ReGenerate(100);
    readkey();

```

gal.cpp

```

        rectfill(screen,0,0,SCREEN_W,SCREEN_H,BLACK);

        myCell->DoClear();
        myCell->SetInReg("00100");
        myCell->SetOutReg("00000");
        myCell->LoadSwitch(myPopulationPtr->
PopulationPtr->SelectBest())->
GetIndividualPtr(my
GetChromosomePtr())
;
        myCell->DoDraw();
        readkey();

        myCell->DoFlood();
        save_bitmap("flood2.bmp",screen,desktop_palette);
        readkey();

        StopGraphics();

        delete myPopulationPtr;
        delete myCell;

        return 0;
}

double theEvaluator(vzIndividual *iPtr)
{
    static int i = 0;
    char buffer[80];
    double tmpFitness;
    double totFitness = 0.0;

    myCell->DoClear();
    myCell->LoadSwitch(iPtr->GetChromosomePtr());
    myCell->SetInReg("00100");
    myCell->SetOutReg("00000");
    myCell->DoAdjust();

    tmpFitness = (double)myCell->GetPinPtr(out,0)->GetPinDistan
ce(in,2);
    if ( tmpFitness == -1.00 ) tmpFitness = 0.00;
    else tmpFitness = 1000.0 - tmpFitness;
    totFitness += tmpFitness;

    tmpFitness = (double)myCell->GetPinPtr(out,1)->GetPinDistan
ce(in,2);
    if ( tmpFitness == -1.00 ) tmpFitness = 0.00;
    else tmpFitness = 1000.0 - tmpFitness;

```

gal.cpp

```

        totFitness += tmpFitness;

        tmpFitness = (double)myCell->GetPinPtr(out,3)->GetPinDistan
ce(in,2);
        if ( tmpFitness == -1.00 ) tmpFitness = 0.00;
        else tmpFitness = 1000.0 - tmpFitness;
        totFitness += tmpFitness;

        tmpFitness = (double)myCell->GetPinPtr(out,4)->GetPinDistan
ce(in,2);
        if ( tmpFitness == -1.00 ) tmpFitness = 0.00;
        else tmpFitness = 1000.0 - tmpFitness;
        totFitness += tmpFitness;

        tmpFitness = (double)myCell->GetPinPtr(in,1)->GetPinDistan
ce(in,2);
        if ( tmpFitness == -1.00 ) tmpFitness = 0.00;
        else tmpFitness = 1000.0 - tmpFitness;
        totFitness += tmpFitness;

        tmpFitness = (double)myCell->GetPinPtr(in,3)->GetPinDistan
ce(in,2);
        if ( tmpFitness == -1.00 ) tmpFitness = 0.00;
        else tmpFitness = 1000.0 - tmpFitness;
        totFitness += tmpFitness;

// =====

        tmpFitness = (double)myCell->GetPinPtr(out,2)->GetPinDistan
ce(in,2);
        if ( tmpFitness == -1.00 ) tmpFitness = 1000.00;
        else if ( tmpFitness < 50 ) tmpFitness = 0.00;
        else tmpFitness = tmpFitness - 50;
        totFitness += tmpFitness;

        tmpFitness = (double)myCell->GetPinPtr(in,0)->GetPinDistan
ce(in,2);
        if ( tmpFitness == -1.00 ) tmpFitness = 1000.00;
        else if ( tmpFitness < 50 ) tmpFitness = 0.00;
        else tmpFitness = tmpFitness - 50;
        totFitness += tmpFitness;

        tmpFitness = (double)myCell->GetPinPtr(in,4)->GetPinDistan
ce(in,2);
        if ( tmpFitness == -1.00 ) tmpFitness = 1000.00;
        else if ( tmpFitness < 50 ) tmpFitness = 0.00;
        else tmpFitness = tmpFitness - 50;
        totFitness += tmpFitness;

```

gal.cpp

```
        sprintf(buffer,"%d: %f",i++, totFitness);
        textout(screen,font,buffer,5,5,BLACK);

        return totFitness;
}

//*****
*****
```

parity.cpp

```

/*
*****
parity.cpp
Van Zandt
*****
*/
#include "vzClassifier.h"

int          gClassifierSetSize = 100;
double  gGeneralTaxRate = .01;
double  gBadMatchTaxRate = .10;
double  gHonorariumRate = .0;
int      gMutationPer1000 = 100;
int      gTournamentSize = 100;
int      gTournamentCount = 10;
int      gRegeneratePer1000 = 1000;

int      main()
{
    vzClassSystem  myClass(          gClassifierSetSize,          gGe
    neralTaxRate,          gBa
    dMatchTaxRate,          gHo
    norariumRate,          gMu
    tationPer1000,          gTo
    urnamentSize,          gTo
    urnamentCount,          gRe
    generatePer1000 );

    vzVector<int>  S0(3);
    vzVector<int>  S1(3);
    vzVector<int>  S2(3);
    vzVector<int>  S3(3);
    vzVector<int>  S4(3);
    vzVector<int>  S5(3);
    vzVector<int>  S6(3);
    vzVector<int>  S7(3);

    vzVector<int>  Solution(8);

    int            i;

```

parity.cpp

```
int          maxCycle = 1000;
int          theScore;

S0[0] = 0;
S0[1] = 0;
S0[2] = 0;
Solution[0] = 1;

S1[0] = 0;
S1[1] = 0;
S1[2] = 1;
Solution[1] = 0;

S2[0] = 0;
S2[1] = 1;
S2[2] = 0;
Solution[2] = 0;

S3[0] = 0;
S3[1] = 1;
S3[2] = 1;
Solution[3] = 1;

S4[0] = 1;
S4[1] = 0;
S4[2] = 0;
Solution[4] = 0;

S5[0] = 1;
S5[1] = 0;
S5[2] = 1;
Solution[5] = 1;

S6[0] = 1;
S6[1] = 1;
S6[2] = 0;
Solution[6] = 1;

S7[0] = 1;
S7[1] = 1;
S7[2] = 1;
Solution[7] = 0;

for (i=0; i<maxCycle; i++)
{
    theScore = 0;
```

parity.cpp

```
if ( Solution[0] == myClass.GetMove(S0) )
    { theScore++; myClass.ApplyReward(1000); }
if ( Solution[1] == myClass.GetMove(S1) )
    { theScore++; myClass.ApplyReward(1000); }
if ( Solution[2] == myClass.GetMove(S2) )
    { theScore++; myClass.ApplyReward(1000); }
if ( Solution[3] == myClass.GetMove(S3) )
    { theScore++; myClass.ApplyReward(1000); }
if ( Solution[4] == myClass.GetMove(S4) )
    { theScore++; myClass.ApplyReward(1000); }
if ( Solution[5] == myClass.GetMove(S5) )
    { theScore++; myClass.ApplyReward(1000); }
if ( Solution[6] == myClass.GetMove(S6) )
    { theScore++; myClass.ApplyReward(1000); }
if ( Solution[7] == myClass.GetMove(S7) )
    { theScore++; myClass.ApplyReward(1000); }

cout << i << " -> " << theScore << endl;
doPause(10);
}

return 0;

};
```

plot.cpp

```

/*
*****
Plot.cpp
Van Zandt
*****
*/
#include <math.h>
#include "vzComputer.h"

void main()
{
    double x, y;
    int i;

    StartGraphics();

    int myBorder = 20;
    int X0 = SCREEN_W/2;
    int Y0 = SCREEN_H/2;

    line(screen,X0,Y0-(Y0-myBorder),X0,Y0+(Y0-myBorder),WHITE);
    line(screen,X0-(X0-myBorder),Y0,X0+(X0-myBorder),Y0,WHITE);

    for ( i=-75; i<=75; i+=2 )
    {
        x = i + 0.00;
        y = 150.0 * sin((x*PI)/50.0);

        if ( x*PI/50.0 > PI/2 && y < 50.0 )
            y = 50.0;
        if ( x*PI/50.0 < -PI/2 && y > -50.0 )
            y = -50.0;
        circlefill(screen,X0+((int)x*4),Y0-(int)y,3,RED);

        x = i + 1.00;
        y = 150.0 * sin((x*PI)/50.0);

        if ( (x > 0.0 && y > 50.0) || x*PI/50.0 > PI/2 )
            y = 50.0;
        if ( (x < 0.0 && y < -50.0) || x*PI/50.0 < -PI/2 )
            y = -50.0;
        circlefill(screen,X0+((int)x*4),Y0-(int)y,3,LIGHTRED);
    }
    save_bitmap("plot.bmp",screen,desktop_palette);
    readkey();
    StopGraphics();
}

```

`plot.cpp`

ptest.cpp

```

/*
ptest.cpp
Van Zandt
*/
#include <iostream.h>
#include <conio.h>
#include <math.h>
#include "vzTools.h"

int    PoissonRandom(double expectedvalue);

void main()
{
    int    k = 5;
    int    i;
    int    n;
    int    result[20];
    double sum = 0.0;

    for ( i=0; i<20; i++ )
        result[i] = 0;

    for ( i=0; i<1000; i++ )
    {
        n = PoissonRandom((double)k);
        result[n]++;
        sum += n;
    }

    for ( i=0; i<20; i++ )
        cout << i << '\t' << result[i] << endl;

    cout << endl << endl << sum/1000.0 << endl;
    getch();
}

int    PoissonRandom(double expectedvalue)
{
    int    n = 0;
    double limit;
    double x;
    limit = exp(-expectedvalue);
    x = getRandom(1000000)/1000000.0;

    while ( x > limit )
    {
        n++;
        x *= getRandom(1000000)/1000000.0;
    }
}

```

`pctest.cpp`

```
    }  
    return n;  
}
```

Run.cpp

```

/*
*****
*****
vzRun.cpp
Van Zandt
*****
*/

#include "vzComputer.h"

void    main()
{
    vzNanoCell  *myCell;
    myCell= new vzNanoCell;

        StartGraphics();
        SetTimer(0);

    myCell->DoCreate(5,9,30,30,75,75);
    //myCell->SetInReg("01000");
    //myCell->SetOutReg("01010");
    myCell->DoDraw();
        readkey();

    myCell->DoAdjust();
    //myCell->DoDraw();
    readkey();

    delete myCell;

        StopGraphics();

    return 0;
}

//*****
*****

```

theSimulator.cpp

```

/*
*****
theSimulator.cpp
Van Zandt
*****
*/
#include      "vzComputer.h"
#include      "vzPopulation.h"

vzNanocell  *myCell;

double      theEvaluator1(vzIndividual *iPtr);
double      theEvaluator2(vzIndividual *iPtr);
double      doScore(int n, bool IsItAnd);

int          doNew();
int          doClear();
int          doRun();
int          doAbout();
int          doExit();
int          doSet10000();
int          doSet01000();
int          doSet11000();
int          doStore();
int          doLoad();
int          doPaint();
int          doSelect();
int          doTrainAND();
int          doTrainOR();
int          doGetAND();
int          doGetOR();
int          doClassifierAND();

MENU        theFileMenu[] =
{
    {          "Load   ",          doLoad,          NULL,
  NULL      },
    {          "Store   ",          doStore,          NULL,
  NULL      },
    {          "Paint   ",          doPaint,          NULL,
  NULL      },
    {          "Select  ",          doSelect,          NULL,
  NULL      },
    {          "        ",          NULL,          NULL,
  NULL      }
}

```

· theSimulator.cpp

```

NULL    },
        {      "Exit    ",          doExit,          NULL,
NULL    }
};

MENU    theTrainMenu[] =
{
    {      "AND    ",          doTrainAND,
NULL,    NULL    },
    {      "OR     ",          doTrainOR,
NULL,    NULL    },
    {      "      ",          NULL,
NULL,    NULL    },
    {      "AND Circuit",      doGetAND, -
NULL,    NULL    },
    {      "OR  Circuit",      doGetOR,
NULL,    NULL    },
    {      "      ",          NULL,
NULL,    NULL    },
    {      "AND Classifier",    doClassifierAND,    NUL
L,
};

MENU    theMenu[] =
{
    {      "File    ",          NULL,          theFileMenu
,
    {      "Train   ",          NULL,          theTrainMen
u,
    {      "      ",          "The Nanocell Simu
lator",
doAbout,    NULL,
NULL    }
};

MENU    theRunMenu[] =
{
    {      "New     ",          doNew,          NULL,
NULL    },
    {      "Run     ",          doRun,          NULL,
NULL    },
    {      "Clear   ",          doClear,        NULL,
NULL    },
    {      "      ",          NULL,          NULL,          NULL
},
    {      "Set[10000] ",      doSet10000,    NULL,          NULL
},
    {      "Set[01000] ",      doSet01000,    NULL,          NULL

```

theSimulator.cpp

```

},
    { "Set[11000] ", doSet11000, NULL, NUL
L    }
};

```

```

DIALOG the_dialog[] =
{
    /* (dialog proc)      (x)      (y)      (w)      (h)      (fg)      (bg)      (key)      (
flags) (d1)      (d2)      (dp)      (dp2) (dp3) */
    { d_box_proc,      0,      0,      639,      479,      255,      4,      0,      0
,      0,      0,      NULL,      NULL, NULL },
    { d_box_proc,      12,      12,      615,      455,      8,      6,      0,      0
,      0,      0,      NULL,      NULL, NULL },
    { d_menu_proc,      12,      13,      0,      0,      0,      0,      0,      0
,      0,      0,      (void *)theMenu,      NULL, NULL },
    { d_menu_proc,      12,      454,      0,      0,      0,      0,      0,      0
,      0,      0,      (void *)theRunMenu,      NULL, NULL },
    { NULL,      0,      0,      0,      0,      0,      0,      0,      0
,      0,      0,      NULL,      NULL, NULL }
};

```

```

int main()
{
    StartGraphics();
    SetTimer(0);
    TraceOff();

    do_dialog(the_dialog, -1);

    if ( myCell != NULL )
        delete myCell;

    StopGraphics();
    return 0;
}

```

```

int      doNew()
{
    show_mouse(NULL);

    if ( myCell != NULL )
        delete myCell;

    myCell= new vzNanocell;
}

```

theSimulator.cpp

```

myCell->DoCreate(5,4,30,30,85,90);
myCell->DoDraw();

    show_mouse(screen);

    return D_O_K;
}

int doAbout()
{
    char    str1[] = "The Nanocell Simulator";
    char    str2[] = "by Bill Van Zandt";
    char    str3[] = "and Eric Libby";

    alert(str1,str2,str3,"OK",NULL,0,0);

    return D_O_K;
}

int doClear()
{
    int      i;
    char     msg[20] = "          ";
    char     buffer[40] = "          ";

    if ( myCell != NULL )
    {
        show_mouse(NULL);
        text_mode(CYAN);

        for (i=0; i<myCell->GetPinNum(); i++)
        {
            textout(screen, font, msg, 540,myCell->PinH
eight(i),BLACK);
            textout(screen, font, msg, 540,myCell->PinHeight(i)
+10,BLACK);
            textout(screen, font, msg, 15, myCell->PinH
eight(i),BLACK);
            textout(screen, font, msg, 15, myCell->PinHeight(i)
+10,BLACK);
        }

        textout(screen,font,buffer,20,440,BLACK);

        TraceOn();
        myCell->ClearCell();
    }
}

```

```

                                theSimulator.cpp

                                TraceOff();

                                show_mouse(screen);
                            }
                        else
                        {
                            show_mouse(NULL);

                                rectfill(screen,25,25,615,445,CYAN);

                                show_mouse(screen);
                            }
                        return D_O_K;
                    }

int                                doRun()
{
    int                                i;
    int                                flow;
    char                                msg[20];

    if ( myCell != NULL )
    {
        show_mouse(NULL);
        text_mode(CYAN);

        myCell->DoFlow();

        for (i=0; i<myCell->GetPinNum(); i++)
        {
            flow=0;

            flow=myCell->GetInitialFlow(false,i);
            sprintf(msg, "init = %d", flow);
            textout(screen, font, msg, 540,myCell->PinH
eight(i),BLACK);

            flow=myCell->GetFlow(false,i);
            sprintf(msg, "flow = %d", flow);
            textout(screen, font, msg, 540,myCell->PinHeight(i)
+10,BLACK);

            flow=myCell->GetInitialFlow(true,i);
            sprintf(msg, "init = %d", flow);
            textout(screen, font, msg, 15, myCell->PinH
eight(i),BLACK);

```

theSimulator.cpp

```

        flow=myCell->GetFlow(true,i);
        sprintf(msg, "flow = %d",flow);
        textout(screen, font, msg, 15,myCell->PinHeight(i)+
10,BLACK);

        myCell->Drawpaths();
    }

    show_mouse(screen);
}

return D_O_K;
}

int doSelect()
{
    char path[81];

    file_select("Select File", path, NULL);

    doClear();

    return D_O_K;
}

int doSet10000()
{
    char msg[20];

    if ( myCell != NULL )
    {
        doClear();

        show_mouse(NULL);
        text_mode(CYAN);
        TraceOn();

        myCell->SetPin(true, 0, kStartFlow);
        sprintf(msg, "flow = %-5d", kStartFlow);
        textout(screen, font, msg, 16,myCell->PinHeight(0),BLACK);

        myCell->DeactivatePin(true, 1);
        myCell->DeactivatePin(true, 2);
        myCell->DeactivatePin(true, 3);
    }
}

```

theSimulator.cpp

```
        myCell->DeactivatePin(true, 4);

        TraceOff();
        show_mouse(screen);
    }

    return D_O_K;
}

int doSet01000()
{
    char msg[20];

    if ( myCell != NULL )
    {
        doClear();

        show_mouse(NULL);
        text_mode(CYAN);
        TraceOn();

        myCell->SetPin(true, 1, kStartFlow);
        sprintf(msg, "flow = %-5d", kStartFlow);
        textout(screen, font, msg, 16, myCell->PinHeight(1), BLACK);

        myCell->DeactivatePin(true, 0);
        myCell->DeactivatePin(true, 2);
        myCell->DeactivatePin(true, 3);
        myCell->DeactivatePin(true, 4);

        TraceOff();
        show_mouse(screen);
    }

    return D_O_K;
}

int doSet11000()
{
    char msg[20];

    if ( myCell != NULL )
    {
        doClear();

        show_mouse(NULL);
```

theSimulator.cpp

```

        text_mode(-1);
        TraceOn();

        myCell->SetPin(true, 0, kStartFlow);
        sprintf(msg, "flow = %-5d", kStartFlow);
        textout(screen, font, msg, 16, myCell->PinHeight(0), BLACK);

        myCell->SetPin(true, 1, kStartFlow);
        sprintf(msg, "flow = %-5d", kStartFlow);
        textout(screen, font, msg, 16, myCell->PinHeight(1), BLACK);

        myCell->DeactivatePin(true, 2);
        myCell->DeactivatePin(true, 3);
        myCell->DeactivatePin(true, 4);

        TraceOff();
        show_mouse(screen);
    }

    return D_O_K;
}

int doStore()
{
    if ( myCell != NULL )
        myCell->DoStore("Nanocell.out");

    return D_O_K;
}

int doLoad()
{
    show_mouse(NULL);

    doClear();

    if ( myCell != NULL )
        delete myCell;

    myCell = new vzNanocell;

    myCell->DoLoad("Nanocell.out");
    myCell->DoDraw();

    show_mouse(screen);
}

```

theSimulator.cpp

```
        return D_O_K;
    }

int          doGetAND()
{
    show_mouse(NULL);

    doClear();

    if ( myCell != NULL )
        delete myCell;

    myCell = new vzNanocell;

    myCell->DoLoad("Nanocell.and");
    myCell->DoDraw();

    show_mouse(screen);

    return D_O_K;
}

int          doGetOR()
{
    show_mouse(NULL);

    doClear();

    if ( myCell != NULL )
        delete myCell;

    myCell = new vzNanocell;

    myCell->DoLoad("Nanocell.or");
    myCell->DoDraw();

    show_mouse(screen);

    return D_O_K;
}

int          doPaint()
{
    char      buffer[40];
    static int bmp = 1;
```

theSimulator.cpp

```

        sprintf(buffer, "mcPaint%d.bmp", bmp);
        save_bitmap(buffer, screen, desktop_palette);
        bmp++;

        return D_O_K;
    }

int doTrainAND()
{
    vzPopulation *myPopulationPtr;

    if ( myCell != NULL )
    {
        doClear();

        TraceOff();
        myPopulationPtr = new vzPopulation(    20,
//mkTournamentCount
        32,          //mkTournamentSize
        400,        //mkMutate
        500,        //mkCrossover
        100,        //mkRandom,
        50,          //mkCount
        myCell->GetSwitchCo
unt(), //mkSize
        2,          //mkAlphabetCount
        the
Evaluator2);      //Evaluation Ptr

        myCell->ClearCell();

        myPopulationPtr->BuildRandom();
        myPopulationPtr->ReGenerate(50);

        myCell->ClearCell();
        myCell->LoadSwitch(myPopulationPtr->
            GetIndividualPtr(myPopulationPtr->S
electBest())->
            GetChromosomePtr());
    }
}

```

```

                                theSimulator.cpp

                                delete myPopulationPtr;
                                }
                                return D_O_K;
                                }

                                int doTrainOR()
                                {
                                    vzPopulation *myPopulationPtr;

                                    if ( myCell != NULL )
                                    {
                                        doClear();

                                        TraceOff();
                                        myPopulationPtr = new vzPopulation( 20,
                                        //mkTournamentCount
                                        32, //mkTournamentSize
                                        400, //mkMutate
                                        500, //mkCrossover
                                        100, //mkRandom,
                                        50, //mkCount
                                        myCell->GetSwitchCo
                                        unt(), //mkSize
                                        2, //mkAlphabetCount
                                        the
                                        Evaluator1); //Evaluation Ptr

                                        myCell->ClearCell();

                                        myPopulationPtr->BuildRandom();
                                        myPopulationPtr->ReGenerate(10);

                                        TraceOn();
                                        myCell->ClearCell();
                                        TraceOff();

                                        myCell->LoadSwitch(myPopulationPtr->
                                        GetIndividualPtr(myPopulationPtr->S
                                        electBest())->
                                        GetChromosomePtr());

```

```
theSimulator.cpp

        delete myPopulationPtr;
    }

    return D_O_K;
}

int doExit()
{
    return D_CLOSE;
}

double theEvaluator1(vzIndividual *iPtr)
{
    static int i = 0;
    char buffer[80];
    int tmpFlow;
    double totFitness = 0.0;

    text_mode(CYAN);

    myCell->LoadSwitch(iPtr->GetChromosomePtr());

    myCell->ClearCell();
    myCell->SetInReg("10000");
    myCell->SetOutReg("*****");
    myCell->DoFlow();
    totFitness += doScore(0, false);

    myCell->ClearCell();
    myCell->SetInReg("01000");
    myCell->SetOutReg("*****");
    myCell->DoFlow();
    totFitness += doScore(1, false);

    myCell->ClearCell();
    myCell->SetInReg("11000");
    myCell->SetOutReg("*****");
    myCell->DoFlow();
    totFitness += doScore(2, false);

    sprintf(buffer, "%d: %f", i++, totFitness);
    textout(screen, font, buffer, 20, 440, BLACK);
    if (totFitness==0)
        i=0;
}
```

theSimulator.cpp

```

        return totFitness;
    }

double theEvaluator2(vzIndividual *iPtr)
{
    static int i = 0;
    char buffer[80];
    int tmpFlow;
    double totFitness = 0.0;

    text_mode(CYAN);

    myCell->LoadSwitch(iPtr->GetChromosomePtr());

    myCell->ClearCell();
    myCell->SetInReg("10000");
    myCell->SetOutReg("*****");
    myCell->DoFlow();
    totFitness += doScore(0, true);

    myCell->ClearCell();
    myCell->SetInReg("01000");
    myCell->SetOutReg("*****");
    myCell->DoFlow();
    totFitness += doScore(1, true);

    myCell->ClearCell();
    myCell->SetInReg("11000");
    myCell->SetOutReg("*****");
    myCell->DoFlow();
    totFitness += doScore(2, true);

    sprintf(buffer, "%d: %f", i++, totFitness);
    textout(screen, font, buffer, 20, 440, BLACK);

    if (totFitness==0)
        i=0;

    return totFitness;
}

double doScore(int n, bool IsItAnd)
{
    //      AND: in("??****") --> out("*****")

```

theSimulator.cpp

```

        int          tmpFlow;
        double retFitness = 0.0;
        if (IsItAnd)
        {
            switch(n)
            {
                case 0: // in("10000")
                case 1: // in("01000")
                    tmpFlow = myCell->GetPinPtr(out,0)-
>GetFlow(true);
                    if ( tmpFlow <= kLowFlow )
                        retFitness += 0.00;
                    else
                        retFitness += (double)(tmpF
low - kLowFlow);
                    break;

                case 2: // in("11000")
                    tmpFlow = myCell->GetPinPtr(out,0)-
>GetFlow(true);
                    if ( tmpFlow >= kHighFlow )
                        retFitness += 0.00;
                    else
                        retFitness += (double)(kHig
hFlow - tmpFlow);
                    break;
            }
            tmpFlow = myCell->GetPinPtr(out,1)->GetFlow(true);
            if ( tmpFlow <= kLowFlow )
                retFitness += 0.00;
            else
                retFitness += (double)(tmpFlow - kLowFlow);

            tmpFlow = myCell->GetPinPtr(out,2)->GetFlow(true);
            if ( tmpFlow <= kLowFlow )
                retFitness += 0.00;
            else
                retFitness += (double)(tmpFlow - kLowFlow);

            tmpFlow = myCell->GetPinPtr(out,3)->GetFlow(true);
            if ( tmpFlow <= kLowFlow )
                retFitness += 0.00;
            else
                retFitness += (double)(tmpFlow - kLowFlow);

            tmpFlow = myCell->GetPinPtr(out,4)->GetFlow(true);
            if ( tmpFlow <= kLowFlow )
                retFitness += 0.00;

```

```

theSimulator.cpp

    else
        retFitness += (double)(tmpFlow - kLowFlow);
}
else
{
    switch(n)
    {
        case 0: // in("10000")
        case 1: // in("01000")
        case 2: // in("11000")
            tmpFlow = myCell->GetPinPtr(out,0)->GetFlow
(tmpFlow);
            if ( tmpFlow >= kHighFlow )
                retFitness += 0.00;
            else
                retFitness += (double)(kHighFlow -
tmpFlow);
            break;
        }

        tmpFlow = myCell->GetPinPtr(out,1)->GetFlow(true);
        if ( tmpFlow <= kLowFlow )
            retFitness += 0.00;
        else
            retFitness += (double)(tmpFlow - kLowFlow);

        tmpFlow = myCell->GetPinPtr(out,2)->GetFlow(true);
        if ( tmpFlow <= kLowFlow )
            retFitness += 0.00;
        else
            retFitness += (double)(tmpFlow - kLowFlow);

        tmpFlow = myCell->GetPinPtr(out,3)->GetFlow(true);
        if ( tmpFlow <= kLowFlow )
            retFitness += 0.00;
        else
            retFitness += (double)(tmpFlow - kLowFlow);

        tmpFlow = myCell->GetPinPtr(out,4)->GetFlow(true);
        if ( tmpFlow <= kLowFlow )
            retFitness += 0.00;
        else
            retFitness += (double)(tmpFlow - kLowFlow);
    }
    return retFitness;
}

```

theSimulator.cpp

```
int          doClassifierAND()  
{  
    return  D_O_K;  
}
```

vzGrid.cpp

```

/*
*****
vzGrid.cpp
Van Zandt
*****
*/

vzGrid::vzGrid()
{
}

vzGrid::~vzGrid()
{
}

void vzGrid::DoCreate(int NumConn, int NumSwit)
{
    mNumConn=NumConn;
    mNumSwit=NumSwit;
    mBoxHeight=190/(2*mNumConn);
    mBoxWidth=320/(2*mNumSwit);
    mCount=-1;
}

void vzGrid::DoDraw(int CurrentConn, int CurrentSwit, int Successval)
{
    int x,y;

    x=CurrentConn;
    y=CurrentSwit;
    mCount++;

    if ((Successval>=90)&&(Successval<100))
        box(mBoxWidth*2*x+5,2*y*mBoxHeight+10,mBoxWidth*(2*
x+1)+5,
            mBoxHeight*(2*y+1)+10,RED,FILLED);

    if ((Successval>=80)&&(Successval<90))
        box(mBoxWidth*2*x+5,2*y*mBoxHeight+10,mBoxWidth*(2*
x+1)+5,
            mBoxHeight*(2*y+1)+10,MAGENTA,FILLED);

    if ((Successval>=70)&&(Successval<80))
        box(mBoxWidth*2*x+5,2*y*mBoxHeight+10,mBoxWidth*(2*
x+1)+5,

```

vzGrid.cpp

```
        mBoxHeight*(2*y+1)+10,YELLOW,FILLED);  
    if ((Successval>=60)&&(Successval<70))  
        box(mBoxWidth*2*x+5,2*y*mBoxHeight+10,mBoxWidth*(2*  
x+1)+5,  
            mBoxHeight*(2*y+1)+10,GREEN,FILLED);  
    if ((Successval>=50)&&(Successval<60))  
        box(mBoxWidth*2*x+5,2*y*mBoxHeight+10,mBoxWidth*(2*  
x+1)+5,  
            mBoxHeight*(2*y+1)+10,CYAN,FILLED);  
    if ((Successval<50))  
        box(mBoxWidth*2*x+5,2*y*mBoxHeight+10,mBoxWidth*(2*  
x+1)+5,  
            mBoxHeight*(2*y+1)+10,BLUE,FILLED);  
    if (Successval==100)  
        box(mBoxWidth*2*x+5,2*y*mBoxHeight+10,mBoxWidth*(2*  
x+1)+5,  
            mBoxHeight*(2*y+1)+10,WHITE,FILLED);  
}
```

vzGrid.h

```

/*
*****
vzGrid.h
Van Zandt
*****
*/

#ifndef _vzGrid_h
#define _vzGrid_h

#include <conio.h>

class vzGrid
{
private:
    int mBoxHeight;
    int mBoxWidth;
    int mNumSwit;
    int mNumConn;
    int mCount;

public:
    vzGrid();
    ~vzGrid();
    void DoCreate(int NumConn,int NumSwit);
    void DoDraw(int CurrentConn, int CurrentSwit, int Successva
1);
};

#include "vzGrid.cpp"

#endif // _vzGrid_h

```

vzPercolation.cpp

```

/*
*****
*****
vzPercolation.cpp
Van Zandt
*****
*****
*/

const int kRegisterWidth=15;

#include "vzNanoCell.h"
#include <iostream.h>
#include "vzGrid.h"

void main()
{
    int connx;
    int swity;
    int x,y,p;
    float success;
    int i;
    int testnum;
    vzGrid myGrid;

    vgainit();
    cls(BLACK);

    testnum=1;
    i=-1;
    x=-1;
    vzNanoCell *myCell;
    myGrid.DoCreate(30,30);

    for (connx=40;connx<101;connx=connx+2)
    {
        x++;
        y=-1;

        for (swity=40;swity<101;swity=swity+2)
        {
            y++;
            success=0;
            i++;

            for (p=0;p<testnum;p++)

```

vzPercolation.cpp

```
{
    myCell= new vzNanoCell;
    myCell->DoCreate(4,3,40,40,connx,swity);
    myCell->DoLoad("1000");
    myCell->DoFlood();

    if (myCell->DoCheck())
        success++;

    delete myCell;

    myCell= new vzNanoCell;
    myCell->DoCreate(4,3,40,40,connx,swity);
    myCell->DoLoad("0100");
    myCell->DoFlood();

    if (myCell->DoCheck())
        success++;

    delete myCell;

    myCell= new vzNanoCell;
    myCell->DoCreate(4,3,40,40,connx,swity);
    myCell->DoLoad("0010");
    myCell->DoFlood();

    if (myCell->DoCheck())
        success++;

    delete myCell;

    myCell= new vzNanoCell;
    myCell->DoCreate(4,3,40,40,connx,swity);
    myCell->DoLoad("0001");
    myCell->DoFlood();

    if (myCell->DoCheck())
        success++;

    delete myCell;

}

myGrid.DoDraw(x,y,success/testnum*25.00);
}
```

vzPercolation.cpp

```
        getch();  
        vgaexit();  
    return 0;  
}  
  
//*****  
*****
```

vzClassifier.cpp

```

/*
*****
*****
Van Zandt
vzClassifier.cpp
*****
*****
*/

vzClassifier::vzClassifier(      int          kClassifierSetSize,
                                dou
ble      kGeneralTaxRate,                                dou
ble      kBadMatchTaxRate,                                dou
ble      kHonorariumRate,                                int
                                kMutationPer1000,           int
                                kTournamentSize,            int
                                kTournamentCount,           int
                                kRegeneratePer1000          )
:   mClassifierSetSize(kClassifierSetSize),
    mGeneralTaxRate(kGeneralTaxRate),
    mBadMatchTaxRate(kBadMatchTaxRate),
    mHonorariumRate(kHonorariumRate),
    mMutationPer1000(kMutationPer1000),
    mTournamentSize(kTournamentSize),
    mTournamentCount(kTournamentCount),
    mRegeneratePer1000(kRegeneratePer1000)
{
    mClassifierSet.resize(mClassifierSetSize);
    mMatchSet.resize(mClassifierSetSize);
    mActionSet.resize(mClassifierSetSize);
    mOldActionSet.resize(mClassifierSetSize);

    //BuildRandom();
    mMatchSize = 0;
    mActionSize = 0;
    mOldActionSize = 0;
}

/*

```

vzClassifier.cpp

```

void vzClassifier::PrintvzClassifiers(ofstream out)
{
    int i;

    for (i=0; i<kvzClassifierSetSize; i++)
    {
        out << i << " :--> ";
        mClassifierSet[i].PrintToFile(out);
    }
}
*/

void vzClassifier::BuildMatchSet(vzVector<int> theSensor)
{
    int i;

    mMatchSize = 0;

    for (i=0; i<mClassifierSet.length(); i++)
        if ( mClassifierSet[i].IsMatch(theSensor) )
        {
            mMatchSet[mMatchSize] = i;
            mMatchSize++;
        }

    if ( mMatchSize == 0 )
    {
        i = BuildCover(theSensor);
        mMatchSet[mMatchSize] = i;
        mMatchSize++;
    }
}

int vzClassifier::GetMove(vzVector<int> theSensor)
{
    BuildMatchSet(theSensor);
    return BuildActionSet();
}

int vzClassifier::BuildCover(vzVector<int> theSensor)
{
    int i;
    double average;

    average = 0.0;

```

vzClassifier.cpp

```

    for (i=0; i<mClassifierSetSize; i++)
        average += mClassifierSet[i].GetFitness();

    average /= mClassifierSetSize;

    i = SelectWeak(mClassifierSet, mClassifierSet.length());
    mClassifierSet[i].BuildCover(theSensor, average);

    return i;
}

void vzClassifier::ApplyReward(int score)
{
    double tmpHonorarium = 0.0;
    double totHonorarium = 0.0;
    int bestAction;
    int i;
    double tmpFitness;

    bestAction = mClassifierSet[mActionSet[0]].GetAction();

    for (i=0; i<mClassifierSet.length(); i++)
    {
        tmpFitness = mClassifierSet[i].GetFitness();
        mClassifierSet[i].SetFitness( tmpFitness *
                                     (1.0-mGener
alTaxRate) );
    }

    for (i=0; i<mMatchSize; i++)
        if ( mClassifierSet[mMatchSet[i]].GetAction() != be
stAction )
        {
            tmpFitness = mClassifierSet[mMatchSet[i]].G
etFitness();
            mClassifierSet[mMatchSet[i]].SetFitness( tm
pFitness *
                                     (1.0-mBadMa
tchTaxRate) );
        }

    /*
    for (i=0; i<mActionSize; i++)
    {
        tmpHonorarium = mClassifierSet[mActionSet[i]].GetFi
tness() *
                                     kHonorariumRate;

```

vzClassifier.cpp

```

        mClassifierSet[mActionSet[i]].SetFitness(
            mClassifierSet[mActionSet[i]].GetFitness()
- tmpHonorarium)
        totHonorarium += tmpHonorarium;
    }

    for (i=0; i<mOldActionSize; i++)
        mClassifierSet[mOldActionSet[i]].SetFitness(tmpHono
rarium /
                                                    mOldActionS
ize);
    */

    for (i=0; i<mActionSize; i++)
    {
        tmpFitness = mClassifierSet[mActionSet[i]].GetFitne
ss();
        tmpFitness *= 1.0+mGeneralTaxRate;
        mClassifierSet[mActionSet[i]].SetFitness( tmpFitnes
s +
                                                    (((double)s
core)/mActionSize) );
    }

    DoTournament();
}

```

```

int          vzClassifier::SelectStrongAction(vzVector<int> theS
et, int size)
{
    int          i;
    int          r;
    double  TotalFitness;

    TotalFitness = 0.0;
    for ( i=0; i<size; i++ )
        TotalFitness += mClassifierSet[theSet[i]].GetFitnes
s();

    r = getRandom( (int)TotalFitness );

    TotalFitness = 0.0;
    for ( i=0; i<size; i++ )
    {
        TotalFitness += mClassifierSet[theSet[i]].GetFitnes

```

vzClassifier.cpp

```

s();

        if ( TotalFitness >= r )
            break;
    }

    return i;
}

int          vzClassifier::BuildActionSet()
{
    int          retSelection;
    int          i;

    for (i=0; i<mActionSize; i++)
        mOldActionSet[i] = mActionSet[i];

    mOldActionSize = mActionSize;

    retSelection = SelectStrongAction(mMatchSet, mMatchSize);

    mActionSize = 0;
    for (i=0; i<mMatchSize; i++)
    {
        if (mClassifierSet[mMatchSet[i]].GetAction() ==
            mClassifierSet[mMatchSet[retSelection]].GetAction())
        {
            mActionSet[mActionSize] = mMatchSet[i];
            mActionSize++;
        }
    }

    return mClassifierSet[mMatchSet[retSelection]].GetAction();
}

//*****
*****

```

vzClassifier.h

```

/*
*****
*****
Van Zandt
vzClassifier.h
*****
*****
*/
#if !defined(_vzClassifier_h)
#define _ClassSystem_h

#include "vzPopulation.h"
#include "vzLib\vzTools.h"
#include "vzLib\vzVector.h"

class vzClassifier
{
private:
    int mClassifierSetSize;
    double mGeneralTaxRate;
    double mBadMatchTaxRate;
    double mHonorariumRate;
    int mMutationPer1000;
    int mTournamentSize;
    int mTournamentCount;
    int mRegeneratePer1000;
    int mMatchSize;
    int mActionSize;
    int mOldActionSize;

    vzPopulation *mpClassifierSetPtr;
    vzVector<int> mMatchSet;
    vzVector<int> mActionSet;
    vzVector<int> mOldActionSet;

public:
    vzClassifier( int kClassifierSetSize,
                  double kGeneralTaxRate,
                  double kBadMatchTaxRate,
                  double kHonorariumRate,
                  int kMutationPe
r1000,
                  int kTournament
Size,
                  int kTournament

```

vzClassifier.h

```

Count,
Per1000 );

int kRegenerate

int BuildCover(vzVector<int> theSensor);
void BuildMatchSet(vzVector<int> theSensor);
int GetMove(vzVector<int> theSensor);
int BuildActionSet();
int SelectStrongAction(vzVector<int> theSet, in
t size);
void ApplyReward(int score);
// void PrintClassifiers(ofstream out);
};

#include "vzClassifier.cpp"
#endif // _vzClassifier_h
//*****
*****

```

vzComputer.h

```

/*
*****
*****
vzComputer.h
Van Zandt
*****
*/

#if !defined(_vzComputer_h)
#define _vzComputer_h

#include <allegro.h>
#include <fstream.h>

// Global Variables
static bool    GraphicsOn = false;
static int     TimerInterval = 0;
static bool    Trace = true;
static bool    New = false;
static int     Summation=0;
static int     Counter=0;

const int      kHighFlow = 80;
const int      kLowFlow = 40;
const int      kHighCapacity = 100;
const int      kLowCapacity = 20;
const int      kStartFlow = 120;

enum vzCOLOR {    WHITE, RED, GREEN, YELLOW, BLUE, MAGENTA, C
YAN, DARKGRAY,
                                LIGHTGRAY, LIGHTRED, LIGHTG
REEN, LIGHTYELLOW, LIGHTBLUE,
                                LIGHTMAGENTA, LIGHTCYAN, BL
ACK };

void    StartGraphics()
{
    allegro_init();
    install_keyboard();
    install_mouse();
    install_timer();
    set_gfx_mode(GFX_AUTODETECT,640,480,0,0);
    set_pallette(desktop_pallette);

    GraphicsOn = true;
}

```

vzComputer.h

```
void    StopGraphics()
{
    allegro_exit();

    GraphicsOn = false;
}
/*
void    CapacityChange()
{
    kHighCapacity = kHighCapacity-1;
    kLowCapacity = kLowCapacity++;
}

void    Sum(int x)
{
    Summation=Summation+x;
    Counter++;
}

int     Average()
{
    int Average;
    Average=Summation/Counter;
    Counter=0;
    Summation=0;
    return Average;
}
void    FlowChange()
{
    kHighFlow = kHighFlow-1;
    kLowFlow = kLowFlow++;
}
*/
void    Reset()
{
    New=true;
}

void    NoReset()
{
    New=false;
}

void    SetTimer(int n)
{
    TimerInterval = n;
}
```

vzComputer.h

```
void    TraceOff()
{
Trace=false;
}

void    TraceOn()
{
Trace=true;
}

#include "vzNanoCell.h"

#endif // _vzComputer_h
//*****
*****
```

vzConnector.cpp

```

/*
*****
*****
vzConnector.cpp
Van Zandt
*****
*****
*/

vzConnector::vzConnector()
{
}

vzConnector::~vzConnector()
{
}

void vzConnector::DoCreate(int wordsize, int x0, int y0, int size, int x, int y,
                           bool alive, vzEvent
                           Q *apEventQ)
{
    int i;

    mWordSize = wordsize;
    mSize = size;
    mX = x0 + (3*mSize*x); // + (3*mSize);
    mY = y0 + (3*mSize*y) + ( (x%2) * (3*mSize/2) );
    mAlive = alive;
    mRow = y;
    mColumn = x;
    mpEventQ = apEventQ;
    mColor = cOffColor;
    mFlow = 0;
    mLabel = false;
    mpSource = NULL;
}

void vzConnector::DoDraw()
{
    if (mAlive)
        circlefill(screen,mX,mY,mSize,mColor);
}

void vzConnector::DoFlood()
{
    int i;

```

vzConnector.cpp

```
    if ((mColor==cLowColor) && (mAlive))
    {
        mColor=cHighColor;

        if (Trace)
            DoDraw();

        for (i=0;i<mNeighborCount;i++)
            mpEventQ->UpdateFuture(mNeighborArray[i]);
    }
}

int vzConnector::GetX()
{
    return mX;
}

int vzConnector::GetY()
{
    return mY;
}

int vzConnector::GetSize()
{
    return mSize;
}

void vzConnector::DoClear()
{
    if ( mAlive )
    {
        mColor = cOffColor;

        if (Trace)
            DoDraw();

        mLabel = false;
        mpSource = NULL;
        mFlow = 0;
    }
}

void vzConnector::GiveFlow(int flow,bool meaningless, vzDevice *p)
{
    mFlow = flow;
    mpSource = p;
}
```

vzConnector.cpp

```

bool vzConnector::GetLabel()
{
    return mLabel;
}

int vzConnector::DoAugment(int x)
{
    int j;

    for(j=0;j<mNeighborCount;j++)
        if (mNeighborArray[j]==mpSource)
        {
            mColor=cLowColor;

            if (Trace)
                DoDraw();

            mpEventQ->UpdateFuture(mNeighborArray[j]);
        }

    return x;
}

void vzConnector::CheckFlow()
{
    int j;
    int sum;
    sum =0;
    for (j=0;j<mNeighborCount;j++)
        sum=sum+mNeighborArray[j]->GetFlow(true)+mNeighborA
rray[j]->GetFlow(false);
    sum=sum/2;
    if (sum==0)
        mColor=cOffColor;
    else if (sum<=kLowFlow)
        mColor=cLowColor;
    else if (sum>=kHighFlow)
        mColor=cHighColor;
    else
        mColor=cMiddleColor;
}

int vzConnector::DoLabel(bool lefttoright)
{
    int j, space;

    if ( (mLabel==false) && (mFlow>0) )
    {

```

vzConnector.cpp

```

        mColor = cHighColor;

        if (Trace)
            DoDraw();

        for (j=0;j<mNeighborCount;j++)
        {
            if (mNeighborArray[j]->GetLabel() == false)
            {
                space = mNeighborArray[j]->GetSpace(lefttoright);
                if ( space != 0 )

                    {
                        if (space<mFlow)
                            mNeighborAr
ray[j]->GiveFlow(space, lefttoright, this);
                        else
                            mNeighborAr
ray[j]->GiveFlow(mFlow, lefttoright, this);

                        mpEventQ->UpdateFuture(mNeighborArray[j]);
                    }
            }
        }
        mLabel = true;
        return 0;
    }

void vzConnector::ClearStart()
{
    mColor = cOffColor;
    mFlow = 0;
    mLabel = false;
    mpSource = NULL;
}
//*****
*****

```

vzConnector.h

```

/*
*****
vzConnector.h
Van Zandt
*****
*/

#ifndef defined(_vzConnector_h)
#define _vzConnector_h

#include "vzLib\vzVector.h"
#include "vzDevice.h"
#include "vzEventQ.h"

const int      cOffColor = DARKGRAY;
const int      cLowColor = YELLOW;
const int      cMiddleColor = LIGHTBLUE;
const int      cHighColor = BLUE;

class vzRegister;

class vzConnector : public vzDevice
{
private:
    int          mX;
    int          mY;
    int          mSize;
    int          mAlive;
    int          mRow;
    int          mColumn;
    int          mColor;
    int          mWordSize;
    vzEventQ     *mpEventQ;
    int          mFlow;
    bool         mLabel;
    vzDevice     *mpSource;

public:
    vzConnector();
    ~vzConnector();

    void          DoCreate(int wordsize, int x0, int y0, int size, in
t x, int y,
                                bool alive, vzEventQ *apEve
ntQ);
    void          DoDraw();

```

vzConnector.h

```

bool        IsAlive() {return mAlive;};
    int      GetRow() { return mRow; };
    int      GetColumn() { return mColumn; };
void        DoFlood();
    void      DoClear();
int         GetX();
int         GetY();
int         GetSize();
    void      GiveFlow(int flow, bool meaningless, vzDevice *p);
    bool      GetLabel();
    int        DoAugment(int x);
    bool      IsADrain() {return false;};
    int        GetSpace(bool lefttoright) {return 0;};
    int        GetFlow(bool lefttoright) {return 0;};
    void      CheckFlow();
int         DoLabel(bool lefttoright);
    void      ClearStart();
};

#include "vzConnector.cpp"

#endif // _vzConnector_h

//*****
*****

```

vzDevice.cpp

```

/*
*****
vzDevice.cpp
Van Zandt
*****
*/

vzDevice::vzDevice()
: mStatus(0),mNeighborCount(0)
{
    int i;

    mNeighborArray.resize(2);
    mNeighborCurrent.resize(2);

    for (i=0;i<2;i++)
    {
        mNeighborArray[i]=NULL;
        mNeighborCurrent[i] = 0;
    }
}

vzDevice::~vzDevice()
{
}

void vzDevice::AddNeighbor(vzDevice *N)
{
    int i;

    mNeighborArray[mNeighborCount]=N;
    mNeighborCount++;

    if ( mNeighborCount >= mNeighborArray.length() )
    {
        mNeighborArray.resize(mNeighborCount+4);
        mNeighborCurrent.resize(mNeighborCount+4);

        for (i=0; i<4; i++)
            mNeighborCurrent[mNeighborCount+i] = 0;
    }
}

vzDevice *vzDevice::GetNeighbor(int n)
{
    return mNeighborArray[n];
}

```

vzDevice.cpp

```
}

int vzDevice::GetCurrent(vzDevice *dPtr)
{
    int i;
    int retCode = 0;

    for ( i=0; i<mNeighborCount; i++ )
        if ( dPtr == mNeighborArray[i] )
            retCode = mNeighborCurrent[i];

    return retCode;
}
```

vzDevice.h

```

/*
*****
*****
vzDevice.h
Van Zandt
*****
*****
*/

#ifndef _vzDevice_h
#define _vzDevice_h

#include "vzLib\vzVector.h"

enum    pType { in, out };
enum    vzCommand { Flood };

class vzDevice
{
protected:
    int                mStatus;
    int                mNeighborCount;
    vzVector<vzDevice*> mNeighborArray;
    vzVector<int>       mNeighborCurrent;

public:
    vzDevice();
    ~vzDevice();
    void        AddNeighbor(vzDevice *N);
    vzDevice    *GetNeighbor(int n);
    int         GetCurrent(vzDevice *dPtr);

    virtual void DoDraw() {};
    virtual void DoFlood() {};
    virtual int  DoAugment(int x) {};
    virtual bool GetLabel() {};
    virtual void DoClear() {};
    virtual void GiveFlow(int f, bool m, vzDevice *p) {};
    virtual bool IsADrain() {};
    virtual int  GetFlow(bool m) {};
    virtual int  DoLabel(bool m) {};
    virtual int  GetSpace(bool m) {};
};

#include "vzDevice.cpp"

#endif // _vzDevice_h

```

vzEventQ.cpp

```

/*
*****
*****
vzEventQ.cpp
Van Zandt
*****
*****
*/

vzEventQ::vzEventQ()
{
    mFutureCount=0;
    mFutureList.resize(50);
}

vzEventQ::~vzEventQ()
{
}

void vzEventQ::UpdateFuture(vzDevice *N)
{
    if (mFutureList.length()-1<mFutureCount)
        mFutureList.resize(mFutureList.length()+50);

    mFutureList[mFutureCount]=N;
    mFutureCount++;
}

void vzEventQ::DoFire(int command)
{
    int mCurrentCount;
    int firecounter;
    vzVector<vzDevice*> mCurrentList;

    mCurrentList=mFutureList;
    mCurrentCount=mFutureCount;
    mFutureCount=0;

    while (mCurrentCount>0)
    {
        for (firecounter=0;firecounter<mCurrentCount;fireco
unter++)
        {
            switch (command)
            {
            case Flood:                mCurrentList[fireco
unter]->DoFlood();
                                break;

```

vzEventQ.cpp

```

    }

    doPause(TimerInterval);
    mCurrentCount=mFutureCount;
    mCurrentList=mFutureList;

//    rectfill(screen,0,0,10,10,mColor++);
//    if ( mColor > 16 )
//        mColor = 0;

    mFutureCount=0;
}

void vzEventQ::DoLabel(bool lefttoright)
{
    int    mCurrentCount;
    int    firecounter;
    int    temp;
    vzVector<vzDevice*> mCurrentList;

    mCurrentList = mFutureList;
    mCurrentCount = mFutureCount;
    mFutureCount = 0;
    temp = 0;

    while ( mCurrentCount > 0 )
    {
        for (firecounter=0;firecounter<mCurrentCount;firecounter++)
        {
            temp=mCurrentList[firecounter]->DoLabel(lefttoright);

            if ( temp == 1 )
            {
                mCurrentCount =0 ;
                mFutureCount =0 ;
                break;
            }
        }

        doPause(TimerInterval);
        mCurrentCount = mFutureCount;
        mCurrentList = mFutureList;
        /* rectfill(screen,0,0,10,10,mColor++);

```

vzEventQ.cpp

```

        if (mColor>16)
            mColor=0;        */
            mFutureCount=0;
        }.
    }
}

void vzEventQ::DoAugment(int minflow)
{
    int      mCurrentCount;
    int      firecounter;
    int      temp;
    vzVector<vzDevice*> mCurrentList;

    mCurrentList = mFutureList;
    mCurrentCount = mFutureCount;
    mFutureCount = 0;
    temp = 0;

    while ( mCurrentCount > 0 )
    {
        for (firecounter=0;firecounter<mCurrentCount;firecounter++)
        {
            temp = mCurrentList[firecounter]->DoAugment
(minflow);
            if ( temp == 0 )
            {
                mCurrentCount = 0;
                mFutureCount = 0;
                break;
            }
        }

        doPause(TimerInterval);
        mCurrentCount=mFutureCount;
        mCurrentList=mFutureList;
        /*      rectfill(screen,0,0,10,10,mColor++);
        if ( mColor > 16 )
            mColor = 0;*/

        mFutureCount = 0;
    }
}

```

vzEventQ.h

```

/*
*****
*****
vzEventQ.h
Van Zandt
*****
*****
*/

#ifndef _vzEventQ_h
#define _vzEventQ_h

#include "vzLib\ vzVector.h"
#include "vzDevice.h"
#include "vzLib\ vzTools.h"

class vzEventQ
{
private:
    int mFutureCount;
    vzVector<vzDevice*> mFutureList;
    int mColor;

public:
    vzEventQ();
    ~vzEventQ();
    void UpdateFuture(vzDevice *D);
    void DoFire(int command);
    void DoLabel(bool lefttoright);
    void DoAugment(int minflow);
};

#include "vzEventQ.cpp"

#endif // _vzEventQ_h

```

vzIndividual.cpp

```
/*
*****
*****
Van Zandt
vzIndividual.cpp
*****
*****
*/

vzIndividual::vzIndividual(int size, int alpha_count)
: mSize(size), mAlphabetCount(alpha_count)
{
    mChromosome.resize(mSize);
}

vzIndividual::~vzIndividual()
{
}

void vzIndividual::DoResize(int size)
{
    mChromosome.resize(size);
}

/*
void vzIndividual::PrintToFile(ofstream out)
{
}
*/

/*
void vzIndividual::DoPrint(int x, int y)
{
}
*/

void vzIndividual::DoMutate()
{
    int i;
    int r;
    int c;

    for (i=0; i<100; i++)
    {
        r = getRandom(mChromosome.length());
        c = getRandom(mAlphabetCount);
    }
}
```

vzIndividual.cpp

```

        while ( mChromosome[r] == c )
            c = getRandom(mAlphabetCount);

        mChromosome[r] = c;
    }
}

void vzIndividual::DoCrossover(vzIndividual *cPtr)
{
    int i;
    int r;

    r = getRandom(mChromosome.length());

    if ( getRandom(2) )
        for (i=0; i<r; i++)
            mChromosome[i] = cPtr->mChromosome[i];
    else
        for (i=r; i<mChromosome.length(); i++)
            mChromosome[i] = cPtr->mChromosome[i];
}

void vzIndividual::BuildRandom()
{
    int i;

    for (i=0; i<mChromosome.length(); i++)
        mChromosome[i] = getRandom(mAlphabetCount);
}

vzIndividual vzIndividual::operator=(const vzIndividual rhs)
{
    int i;

    for (i=0; i<mChromosome.length(); i++)
        mChromosome[i] = rhs.mChromosome[i];

    mFitness = rhs.mFitness;

    return *this;
}

//*****
*****

```

vzIndividual.h

```

/*
*****
*****
Van Zandt
vzIndividual.h
*****
*****
*/
#ifdef _vzIndividual_h
#define _vzIndividual_h

#include "vzLib\vzTools.h"
#include "vzLib\vzVector.h"

class vzIndividual
{
private:
    const int      mSize;
    const int      mAlphabetCount;
    vzVector<int>   mChromosome;
    double         mFitness;

public:
    vzIndividual(int size, int alpha_count);
    ~vzIndividual();

    void DoResize(int size);
    void BuildRandom();
    double GetFitness() { return mFitness; };
    void SetFitness(double fitness) { mFitness=fitness; };
    int GetValue(int i) { return mChromosome[i]; };
    vzVector<int> *GetChromosomePtr() { return &mChromosome; };

    void DoMutate();
    void DoCrossover(vzIndividual *cPtr);

    vzIndividual operator=(const vzIndividual rhs);
    // void DoPrint(int x, int y);
    // void PrintToFile(ofstream out);
};

#include "vzIndividual.cpp"
#endif // _vzIndividual_h
//*****
*****

```

vzNanocell.cpp

```

/*
*****
*****
vzNanocell.cpp
Van Zandt
*****
*****
*/

vzNanocell::vzNanocell()
: mActive(false), mWordSize(0), mRowCount(0), mColumnCount(0)
{
    int i;

    mSwitchMax = 50;
    mSwitchCount = 0;
    mpSwitch.resize(mSwitchMax);

    for (i=0;i<mSwitchMax;i++)
        mpSwitch[i]=new vzSwitch;
}

vzNanocell::~vzNanocell()
{
    int i;

    for (i=0;i<mSwitchMax;i++)
        if (mpSwitch[i] != NULL)
            delete mpSwitch[i];

    if (mpInRegister != NULL)
        delete mpInRegister;

    if (mpOutRegister != NULL)
        delete mpOutRegister;
}

void vzNanocell::DoCreate(int WordSize, int devicesize, int rowcount,
                          int columncount, int density, int switchdensity)
{
    int    r, c;
    int    border;
    int    temp;
    bool    alive;

    mWordSize = WordSize;

```

vzNanocell.cpp

```

mDeviceSize = devicesize;
mRowCount = rowcount;
mColumnCount = columncount;
    mConnector.resize(mRowCount,mColumnCount);

mDensity = density;
mSwitchDensity=switchdensity;

mpInRegister = new vzRegister(mWordSize, mDeviceSize);
mpOutRegister = new vzRegister(mWordSize, mDeviceSize);

border = 1 + mDeviceSize;

mX1 = (SCREEN_W/2) - ((mColumnCount+1) * 3 * mDeviceSize)/2 - b
order;
mY1 = (SCREEN_H/2) - (mRowCount*3*mDeviceSize)/2 - border;

mX2 = (SCREEN_W/2) + ((mColumnCount+3) * 3 * mDeviceSize)/2 + b
order;
mY2 = (SCREEN_H/2) + (mRowCount*3*mDeviceSize)/2 + border;

mActive = true;

mpInRegister->DoCreate(mX1-(8*mDeviceSize),mY1,mY2, true,&mEven
tQ);
mpOutRegister->DoCreate(mX2-(4*mDeviceSize),mY1,mY2, false,&mEv
entQ);

    for (r=0; r<mRowCount; r++)
    for (c=0; c<mColumnCount; c++)
    {
        if ( getRandom(100) < mDensity )
        {
            alive = true;
        }
        else
        {
            alive = false;
        }
        mConnector[r][c].DoCreate(mWordSize,mX1+6*mDeviceSi
ze,mY1+2*mDeviceSize,
        mDeviceSize,c,r,alive,&mEventQ);
    }

    for (r=0;r<mRowCount;r++)
        for (c=0;c<mColumnCount;c++)

```

vzNanocell.cpp

```

{
    if (mConnector[r][c].IsAlive())
    {
        if ((c==0)&&(r<mRowCount-1))
        {
            if (mConnector[r][c+1].IsAl
                SetSwitch(r,c,r,c+1
        );

            if (mConnector[r+1][c].IsAl
                SetSwitch(r,c,r+1,c
        );
        }

        if ((c>0)&&(c<mColumnCount-1)&&(c%2
            {
                if (mConnector[r+1][c-1].Is
                    SetSwitch(r,c,r+1,c
                -1);

                if (mConnector[r+1][c].IsAl
                    SetSwitch(r,c,r+1,c
                );

                if (mConnector[r+1][c+1].Is
                    SetSwitch(r,c,r+1,c
                +1);
            }

            if ((c>0)&&(c<mColumnCount-1)&&(c%2
                {
                    if (mConnector[r][c-1].IsAl
                        SetSwitch(r,c,r,c-1
                    );

```

vzNanocell.cpp

```

    if (mConnector[r+1][c].IsAl
ive()
        SetSwitch(r,c,r+1,c
);

    if (mConnector[r][c+1].IsAl
ive()
        SetSwitch(r,c,r,c+1
);
}
if ((r==mRowCount-1)&&(c<mColumnCou
nt-1)&&(c%2==0))
{
    if (mConnector[r][c+1].IsAl
ive()
        SetSwitch(r,c,r,c+1
);
}
if ((r==mRowCount-1)&&(c<mColumnCo
unt-1)&&(c%2>0))
{
    if (mConnector[r][c+1].IsAl
ive()
        SetSwitch(r,c,r,c+1
);
}
if ((r<mRowCount-1)&&(c==mColumnCou
nt-1)&&(c%2==0))
{
    if (mConnector[r][c-1].IsAl
ive()
        SetSwitch(r,c,r,c-1
);

    if (mConnector[r+1][c].IsAl
ive()
        SetSwitch(r,c,r+1,c
);
}
if ((r<mRowCount-1)&&(c==mColumnCou
nt-1)&&(c%2>0))

```

121

```

vzNanocell.cpp

{

    if (mConnector[r+1][c-1].Is
Alive()
        SetSwitch(r,c,r+1,c
-1);

    if (mConnector[r+1][c].IsAl
ive()
        SetSwitch(r,c,r+1,c
);
}

}

for (r=0;r<mRowCount;r++)
{
    if (mConnector[r][0].IsAlive())
    {
        temp=mpInRegister->ConnectToPin(&mC
onconnector[r][0], mSwitchDensity*2);
        if (temp>=0)
            SetPinSwitch(mpInRegister->
GetPinPtr(temp),&mConnector[r][0]);
    }

    if (mConnector[r][mColumnCount-1].IsAlive()
)
    {
        temp=mpOutRegister->ConnectToPin(&m
Connector[r][mColumnCount-1], mSwitchDensity*2);
        if (temp>=0)
            SetPinSwitch(mpOutRegister->
>GetPinPtr(temp),&mConnector[r][mColumnCount-1]);
    }

}

DoDraw();
}

void vzNanocell::SetSwitch(int r1, int c1, int r2, int c2)
{
    int i;
    int state;

    if ( mSwitchCount >= mSwitchMax )
    {

```

vzNanocell.cpp

```

        mpSwitch.resize(mSwitchMax+50);

        for (i=0; i<50; i++)
            mpSwitch[mSwitchCount+i] = new vzSwitch;

        mSwitchMax += 50;
    }

    if ( getRandom(100) < mSwitchDensity )
    {
        state = 1 + getRandom(2);
        mpSwitch[mSwitchCount]->DoCreate(mConnector[r1][c1]
        .GetX(),
        mC
        onnector[r1][c1].GetY(),
        mC
        onnector[r2][c2].GetX(),
        mC
        onnector[r2][c2].GetY(),
        st
        ate, &mEventQ);

        mConnector[r1][c1].AddNeighbor((vzDevice *)mpSwitch
        [mSwitchCount]);
        mpSwitch[mSwitchCount]->AddNeighbor(&mConnector[r1]
        [c1]);
        mpSwitch[mSwitchCount]->AddNeighbor(&mConnector[r2]
        [c2]);
        mConnector[r2][c2].AddNeighbor((vzDevice *)mpSwitch
        [mSwitchCount]);

        mpSwitch[mSwitchCount]->SetNeighbor1Type(1);
        mpSwitch[mSwitchCount]->SetNeighbor1Parm1(r1);
        mpSwitch[mSwitchCount]->SetNeighbor1Parm2(c1);

        mpSwitch[mSwitchCount]->SetNeighbor2Type(1);
        mpSwitch[mSwitchCount]->SetNeighbor2Parm1(r2);
        mpSwitch[mSwitchCount]->SetNeighbor2Parm2(c2);

        mSwitchCount++;
    }
}

void vzNanocell::SetSwitch2(int r1, int c1, int r2, int c2, int sta
te)
{
    mpSwitch[mSwitchCount] = new vzSwitch;

```

vzNanocell.cpp

```

        mpSwitch[mSwitchCount]->DoCreate(mConnector[r1][c1].GetX(),
                                                    mC
onconnector[r1][c1].GetY(),
                                                    mC
onconnector[r2][c2].GetX(),
                                                    mC
onconnector[r2][c2].GetY(),
                                                    st
ate, &mEventQ);

        mConnector[r1][c1].AddNeighbor((vzDevice *)mpSwitch[mSwitch
Count]);
        mpSwitch[mSwitchCount]->AddNeighbor(&mConnector[r1][c1]);
        mpSwitch[mSwitchCount]->AddNeighbor(&mConnector[r2][c2]);
        mConnector[r2][c2].AddNeighbor((vzDevice *)mpSwitch[mSwitch
Count]);

        mpSwitch[mSwitchCount]->SetNeighbor1Type(1);
        mpSwitch[mSwitchCount]->SetNeighbor1Parm1(r1);
        mpSwitch[mSwitchCount]->SetNeighbor1Parm2(c1);

        mpSwitch[mSwitchCount]->SetNeighbor2Type(1);
        mpSwitch[mSwitchCount]->SetNeighbor2Parm1(r2);
        mpSwitch[mSwitchCount]->SetNeighbor2Parm2(c2);

        mSwitchCount++;
    }

void vzNanocell::SetPinSwitch2(vzPin *aPin, vzConnector *aConnector
, int state)
{
    mpSwitch[mSwitchCount] = new vzSwitch;

    mpSwitch[mSwitchCount]->DoCreate(aConnector->GetX(),
                                                    aC
onconnector->GetY(),
                                                    aP
in->GetOutX2(),
                                                    aC
onconnector->GetY(),
                                                    st
ate, &mEventQ);

    aConnector->AddNeighbor((vzDevice *)mpSwitch[mSwitchCount]);
    mpSwitch[mSwitchCount]->AddNeighbor(aConnector);
    mpSwitch[mSwitchCount]->AddNeighbor(aPin);
    aPin->AddNeighbor(mpSwitch[mSwitchCount]);
    mpSwitch[mSwitchCount]->SetState(state);
}

```

vzNanocell.cpp

```

        mpSwitch[mSwitchCount]->SetNeighbor1Type(1);
        mpSwitch[mSwitchCount]->SetNeighbor1Parm1(aConnector->GetRo
w());
        mpSwitch[mSwitchCount]->SetNeighbor1Parm2(aConnector->GetCo
lumn());

        mpSwitch[mSwitchCount]->SetNeighbor2Type(0);
        mpSwitch[mSwitchCount]->SetNeighbor2Parm1(aPin->GetType());
        mpSwitch[mSwitchCount]->SetNeighbor2Parm2(aPin->GetID());

        mSwitchCount++;
    }

void vzNanocell::SetPinSwitch(vzPin *aPin, vzConnector *aConnector)
{
    int            i;
    int            state;

    state = 1 + getRandom(2);

    if ( mSwitchCount >= mSwitchMax )
    {
        mpSwitch.resize(mSwitchMax+50);

        for (i=0; i<50; i++)
            mpSwitch[mSwitchCount+i] = new vzSwitch;

        mSwitchMax += 50;
    }

    mpSwitch[mSwitchCount]->DoCreate(aConnector->GetX(),
onconnector->GetY(),
aC
in->GetOutX2(),
aP
onconnector->GetY(),
aC
ate, &mEventQ);
st

    aConnector->AddNeighbor((vzDevice *)mpSwitch[mSwitchCount]);
    mpSwitch[mSwitchCount]->AddNeighbor(aConnector);
    mpSwitch[mSwitchCount]->AddNeighbor(aPin);
    aPin->AddNeighbor(mpSwitch[mSwitchCount]);

    mpSwitch[mSwitchCount]->SetNeighbor1Type(1);
    mpSwitch[mSwitchCount]->SetNeighbor1Parm1(aConnector->GetRo

```

vzNanocell.cpp

```

w());
    mpSwitch[mSwitchCount]->SetNeighbor1Parm2(aConnector->GetCo
lumn());

    mpSwitch[mSwitchCount]->SetNeighbor2Type(0);
    mpSwitch[mSwitchCount]->SetNeighbor2Parm1(aPin->GetType());
    mpSwitch[mSwitchCount]->SetNeighbor2Parm2(aPin->GetID());

    mSwitchCount++;
}

void vzNanocell::DoClear()
{
    int      r, c;
    int      i;

    mpInRegister->DoClear();
    mpOutRegister->DoClear();

    for (r=0; r<mRowCount; r++)
        for (c=0; c<mColumnCount; c++)
        {
            if ( mConnector[r][c].IsAlive() )
                mConnector[r][c].DoClear();
        }

    for (i=0; i<mSwitchCount; i++)
        mpSwitch[i]->DoClear();
}

void vzNanocell::DoDraw()
{
    int r, c;

    if ( mActive )
    {
        rectfill(screen,mX1,mY1,mX2,mY2,LIGHTGRAY);

        mpInRegister->DoDraw();
        mpOutRegister->DoDraw();

        for (r=0; r<mRowCount; r++)
            for (c=0; c<mColumnCount; c++)
                mConnector[r][c].DoDraw();

        for (r=0; r<mSwitchCount; r++)
            mpSwitch[r]->DoDraw();
    }
}

```

vzNanocell.cpp

```

    }
}

void vzNanocell::DoFlood()
{
    mpInRegister->DoFlood();
    mpOutRegister->DoFlood();

    mEventQ.DoFire(Flood);
}

void vzNanocell::DoFlow()
{
    int minflow = 0;
    int r, c;
    bool abletolabelone = true;
    bool abletolabeltwo = true;
    bool abletolabelthree = true;
    bool abletolabelfour = true;
    bool breakerone = true;
    bool breakertwo = true;

    while ((breakerone) || (breakertwo))
    {
        mpInRegister->DoLabel(true);
        mEventQ.DoLabel(true);
        abletolabelone=mpOutRegister->CheckLabel();
        abletolabeltwo=mpInRegister->CheckLabel();

        if ((abletolabelone) || (abletolabeltwo))
        {
            if (abletolabelone)
                minflow=mpOutRegister->DoAugment();
            else
                minflow=mpInRegister->DoAugment();
            mEventQ.DoAugment(minflow);

            mpInRegister->DoClear();
            mpOutRegister->DoClear();

            for (r=0; r<mRowCount; r++)
                for (c=0; c<mColumnCount; c++)
                    mConnector[r][c].DoClear();

            for (r=0; r<mSwitchCount; r++)
                mpSwitch[r]->DoClear();
        }
        else
    }
}

```

vzNanocell.cpp

```

        breakerone=false;

mpOutRegister->DoLabel(false);
    mEventQ.DoLabel(false);
    abletolabelthree=mpInRegister->CheckLabel();
    abletolabelfour=mpOutRegister->CheckLabel();
    if ((abletolabelthree)|| (abletolabelfour))
    {
        if (abletolabelthree)
            minflow=mpInRegister->DoAugment();
        if (abletolabelfour)
            minflow=mpOutRegister->DoAugment();
        mEventQ.DoAugment(minflow);
        mpInRegister->DoClear();
        mpOutRegister->DoClear();

        for (r=0; r<mRowCount; r++)
            for (c=0; c<mColumnCount; c++)
                mConnector[r][c].DoClear();

        for (r=0; r<mSwitchCount; r++)
            mpSwitch[r]->DoClear();
    }
    else
        breakertwo=false;
}

int vzNanocell::GetPinNum()
{
    return mWordSize;
}

int vzNanocell::OnPin(int x1,int y1)
{
    int FirstReply;
    int SecondReply;
    FirstReply=mpInRegister->OnPin(x1,y1);
    if (FirstReply>=0)
        return FirstReply;
    else
        SecondReply=mpOutRegister->OnPin(x1,y1);
    if (SecondReply>=0)
        return SecondReply+mWordSize;
    else
        return -1;
}

```

vzNanocell.cpp

```

void vzNanocell::SetPin(bool InRegister,int WhichPin, int flow)
{
    if (InRegister)
        mpInRegister->SetPin(WhichPin,flow);
    else
        mpOutRegister->SetPin(WhichPin,flow);
}

int vzNanocell::PinHeight(int WhichPin)
{
    return (mpInRegister->PinHeight(WhichPin));
}

int vzNanocell::GetFlow(bool WhichRegister,int WhichPin)
{
    int flow;
    if (WhichRegister)
        flow=mpInRegister->GetPinFlow(WhichPin);
    else
        flow=mpOutRegister->GetPinFlow(WhichPin);
    return flow;
}

void vzNanocell::DeactivatePin(bool WhichRegister,int WhichPin)
{
    if (WhichRegister)
        mpInRegister->DeactivatePin(WhichPin);
    else
        mpOutRegister->DeactivatePin(WhichPin);
}

void vzNanocell::Drawpaths()
{
    int r,c;
    for (r=0; r<mRowCount; r++)
        for (c=0; c<mColumnCount; c++)
        {
            mConnector[r][c].CheckFlow();
            mConnector[r][c].DoDraw();
        }

    for (r=0;r<mSwitchCount;r++)
        mpSwitch[r]->DoDraw();
}

int vzNanocell::GetInitialFlow(bool WhichRegister,int WhichPin)
{
    int flow;

```

vzNanocell.cpp

```

        if (WhichRegister)
            flow=mpInRegister->GetInitialFlow(WhichPin);
        else
            flow=mpOutRegister->GetInitialFlow(WhichPin);
        return flow;
    }

void vzNanocell::ClearCell()
{
    int r,c;
    mpInRegister->ClearStart();
    mpOutRegister->ClearStart();

    for (r=0; r<mRowCount; r++)
        for (c=0; c<mColumnCount; c++)
            mConnector[r][c].ClearStart();

        for (r=0;r<mSwitchCount;r++)
            mpSwitch[r]->ClearStart();

    if ( Trace )
        DoDraw();
}

void vzNanocell::SetInReg(vzString aFireCode)
{
    mpInRegister->Write(aFireCode);
}

void vzNanocell::SetOutReg(vzString aFireCode)
{
    mpOutRegister->Write(aFireCode);
}

void vzNanocell::LoadSwitch(vzVector<int> *vPtr)
{
    int i;

    if ( vPtr->length() == mSwitchCount )
    {
        for (i=0; i<mSwitchCount; i++)
            mpSwitch[i]->SetState(1+(*vPtr)[i]);
    }
}

vzPin *vzNanocell::GetPinPtr(pType type, int i)
{
    if ( type == in )

```

vzNanocell.cpp

```
        return mpInRegister->GetPinPtr(i);
    else
        return mpOutRegister->GetPinPtr(i);
}

int    vzNanocell::GetConnectorData(int x)
{
    return mConnectorData[x];
}

int    vzNanocell::GetColumnCount()
{
    return mColumnCount;
}

int    vzNanocell::GetWordSize()
{
    int r,c,temp;
    temp=mSwitchData.length();
    c=0;
    for (r=0;r<temp;r++)
    {
        if ((mSwitchData[r]>0)&&(c<mSwitchCount))
        {
            mSwitchData[r]= mpSwitch[c]->GetState();
            c++;
        }
    }
    return mWordSize;
}

int    vzNanocell::GetRowCount()
{
    return mRowCount;
}

int    vzNanocell::GetSwitchDataSize()
{
    return mSwitchData.length();
}

int    vzNanocell::GetSwitchData(int y)
{
    return mSwitchData[y];
}
```

vzNanocell.cpp

```

void vzNanocell::DoLoad(vzString filename)
{
    int i;
    int r, c;
    int border;
    int temp;
    bool alive;
    int tmpState;
    int tmpNeighbor1Type, tmpNeighbor1Parm1
, tmpNeighbor1Parm2;
    int tmpNeighbor2Type, tmpNeighbor2Parm1
, tmpNeighbor2Parm2;
    ifstream infile;

    infile.open( filename.c_str() );

    infile >> mWordSize;
    infile >> mDeviceSize;
    infile >> mRowCount;
    infile >> mColumnCount;
    mConnector.resize(mRowCount,mColumnCount);

    infile >> mDensity;
    infile >> mSwitchDensity;

    infile >> mSwitchCount;
    mSwitchMax = mSwitchCount;
    mpSwitch.resize(mSwitchMax);
    mSwitchCount = 0;

    mpInRegister = new vzRegister(mWordSize, mDeviceSize);
    mpOutRegister = new vzRegister(mWordSize, mDeviceSize);

    border = 1 + mDeviceSize;

    mX1 = (SCREEN_W/2) - ((mColumnCount+1) * 3 * mDeviceSize)/2 - b
order;
    mY1 = (SCREEN_H/2) - (mRowCount*3*mDeviceSize)/2 - border;

    mX2 = (SCREEN_W/2) + ((mColumnCount+3) * 3 * mDeviceSize)/2 + b
order;
    mY2 = (SCREEN_H/2) + (mRowCount*3*mDeviceSize)/2 + border;

    mActive = true;

    mpInRegister->DoCreate(mX1-(8*mDeviceSize),mY1,mY2, true,&mEven

```

vzNanocell.cpp

```

tQ);
    mpOutRegister->DoCreate(mX2-(4*mDeviceSize),mY1,mY2, false,&mEventQ);

for (r=0; r<mRowCount; r++)
    for (c=0; c<mColumnCount; c++)
    {
        infile >> alive;
        mConnector[r][c].DoCreate(mWordSize,mX1+6*mDeviceSize,
mDeviceSize,c,r,alive,&mEventQ);
    }

for (i=0; i<mSwitchMax; i++)
{
    infile >> tmpState;

    infile >> tmpNeighbor1Type;
    infile >> tmpNeighbor1Parm1;
    infile >> tmpNeighbor1Parm2;

    infile >> tmpNeighbor2Type;
    infile >> tmpNeighbor2Parm1;
    infile >> tmpNeighbor2Parm2;

    if ( tmpNeighbor2Type == 1 )
        SetSwitch2(tmpNeighbor1Parm1, tmpNeighbor1Parm2,
tmpNeighbor2Parm1, tmpNeighbor2Parm2, tmpState);
    else
    {
        if ( tmpNeighbor2Parm1 == 0 )
            SetPinSwitch2(mpInRegister->GetPinPtr(tmpNeighbor2Parm2),
&mConnector[tmpNeighbor1Parm1][tmpNeighbor1Parm2], tmpState);
        else
            SetPinSwitch2(mpOutRegister->GetPinPtr(tmpNeighbor2Parm2),
&mConnector[tmpNeighbor1Parm1][tmpNeighbor1Parm2], tmpState);
    }
}

```

vzNanocell.cpp

```

        }
    }

    infile.close();
}

void vzNanocell::DoStore(vzString filename)
{
    int i;
    int r,c;
    ofstream outfile;

    outfile.open( filename.c_str() );

    outfile << mWordSize << endl;
    outfile << mDeviceSize << endl;
    outfile << mRowCount << endl;
    outfile << mColumnCount << endl;
    outfile << mDensity << endl;
    outfile << mSwitchDensity << endl;

    outfile << mSwitchCount << endl;

    for (r=0; r<mRowCount; r++)
        for (c=0; c<mColumnCount; c++)
            outfile << mConnector[r][c].IsAlive() << endl;

    for (i=0; i<mSwitchCount; i++)
    {
        outfile << mpSwitch[i]->GetState() << endl;

        outfile << mpSwitch[i]->GetNeighbor1Type() << endl;
        outfile << mpSwitch[i]->GetNeighbor1Parm1() << endl;

        outfile << mpSwitch[i]->GetNeighbor1Parm2() << endl;

        outfile << mpSwitch[i]->GetNeighbor2Type() << endl;
        outfile << mpSwitch[i]->GetNeighbor2Parm1() << endl;

        outfile << mpSwitch[i]->GetNeighbor2Parm2() << endl;
    }

    outfile.close();
}

```

vzNanocell.cpp

```
//*****  
*****
```

vzNanocell.h

```

/*
*****
*****
vzNanocell.h
Van Zandt
*****
*/

#ifndef _vzNanocell_h
#define _vzNanocell_h

#include "vzEventQ.h"
#include "vzRegister.h"
#include "vzConnector.h"
#include "vzLib\vzTools.h"
#include "vzLib\vzString.h"
#include "vzSwitch.h"
#include "vzLib\vzMatrix.h"
#include "vzLib\vzVector.h"

class vzNanocell
{
private:
    bool        mActive;
    int          mWordSize;
    int          mRowCount;
    int          mColumnCount;
    int          mDeviceSize;
    int          mDensity;
    int          mSwitchDensity;
    int          mSwitchMax;
    int          mX1;
    int          mY1;
    int          mX2;
    int          mY2;
    int          mSwitchCount;
    vzRegister   *mpInRegister;
    vzRegister   *mpOutRegister;
    int          sizecount;
    vzMatrix<vzConnector> mConnector;
    vzVector<vzSwitch *> mpSwitch;
    vzEventQ     mEventQ;
    vzVector<int> mConnectorData;
    vzVector<int> mSwitchData;
public:
    vzNanocell();
    ~vzNanocell();

```

vzNanocell.h

```

void      DoCreate(int WordSize, int devicesize, int rowcount
               int columncount, int density, int switchdensity);
void      SetSwitch(int r1, int c1, int r2, int c2);
void      SetSwitch2(int r1, int c1, int r2, int c2, int state);
void      SetPinSwitch2(vzPin *aPin, vzConnector *aConnector,
int state);
void      SetPinSwitch(vzPin *aPin, vzConnector *aConnector);
int        GetSwitchCount() { return mSwitchCount; };
void      DoClear();
void      DoDraw();
void      DoFlood();
void      DoFlow();
int        GetPinNum();
int        OnPin(int x1, int y1);
void      SetPin(bool InRegister, int WhichPin, int flow);
int        PinHeight(int WhichPin);
int        GetFlow(bool WhichRegister, int WhichPin);
void      DeactivatePin(bool WhichRegister, int WhichPin);
void      Drawpaths();
int        GetInitialFlow(bool WhichRegister, int Which
Pin);
void      ClearCell();
void      SetInReg(vzString aFireCode);
void      SetOutReg(vzString aFireCode);
void      LoadSwitch(vzVector<int> *vPtr);
vzPin      *GetPinPtr(pType type, int i);
int        GetConnectorData(int x);
int        GetColumnCount();
int        GetWordSize();
int        GetRowCount();
int        GetSwitchDataSize();
int        GetSwitchData(int y);
void      DoLoad(vzString filename);
void      DoStore(vzString filename);
};

#include "vzNanocell.cpp"

#endif // _vzNanocell_h
/*****
*****/

```

vzPin.cpp

```

/*
*****
*****
vzPin.cpp
Van Zandt
*****
*/

vzPin::vzPin()
:vzDevice()
{
}

vzPin::~vzPin()
{
}

void vzPin::DoCreate(int id, pType type, int x1, int y1, int x2, in
t.y2,
                                int x12, int y12, i
nt x22, int y22,
                                int wordsize, vzEve
ntQ *apEventQ)
{
    int i;

    mID = id;
    mType = type;
    mColor = pnDrainColor;
    mInX1 = x1;
    mInX2 = x2;
    mInY1 = y1;
    mInY2 = y2;
    mOutX1 = x12;
    mOutX2 = x22;
    mOutY1 = y12;
    mOutY2 = y22;
    mWordSize = wordsize;
    mpEventQ = apEventQ;
    mFlow=0;
    mIsADrain=true;
    mIsASource=false;
    mTempFlow=0;
    mLabel=false;
    mDeactivated=false;
    mInitialFlow=0;

```

vzPin.cpp

```
}

void vzPin::DoDraw()
{
    if (mIsADrain)
    {
        if (mFlow==0)
            mColor=pnDrainColor;
        else if (mFlow<=kLowFlow)
            mColor=pnLowColor;
        else if (mFlow>=kHighFlow)
            mColor=pnHighColor;
        else
            mColor=pnMiddleColor;
    }

    if (mDeactivated)
        mColor=pnOffColor;

    if (mIsASource)
        mColor=pnSourceColor;

    rectfill(screen,mInX1,mInY1,mInX2,mInY2,mColor);
    rectfill(screen,mOutX1,mOutY1,mOutX2,mOutY2,mColor);
}

int vzPin::GetOutX1()
{
    return mOutX1;
}

int vzPin::GetOutY1()
{
    return mOutY1;
}

int vzPin::GetOutX2()
{
    return mOutX2;
}

int vzPin::GetOutY2()
{
    return mOutY2;
}
```

vzPin.cpp

```
void vzPin::DoFlood()
{
    int i;
    if (mColor == pnLowColor)
    {
        mColor=pnHighColor;
        if (Trace)
            DoDraw();
        for (i=0;i<mNeighborCount;i++)
            mpEventQ->UpdateFuture(mNeighborArray[i]);
    }
}

void vzPin::DoClear()
{
    if (mIsADrain)
        mColor=pnDrainColor;
    if (mDeactivated)
        mColor=pnOffColor;
    if (mIsASource)
        mColor=pnSourceColor;
    if (Trace)
        DoDraw();
    mpSource = NULL;
    mLabel = false;
    mTempFlow = 0;
}

bool vzPin::GetLabel()
{
    return mLabel;
}

bool vzPin::IsADrain()
{
    return mIsADrain;
}

void vzPin::GiveFlow(int flow, bool meaningless, vzDevice *p)
{
    mTempFlow = flow;
    mpSource = p;
    mLabel = true;
    mColor=pnSourceColor;
    if (Trace)
        DoDraw();
}
```

vzPin.cpp

```
int vzPin::DoAugment(int x)
{
    int    j;
    int    returncode;

    if (mIsADrain)
    {
        mFlow=mFlow+mTempFlow;

        for (j=0;j<mNeighborCount;j++)
            if (mNeighborArray[j]==mpSource)
                mpEventQ->UpdateFuture(mNeighborArray[j]);

        returncode=mTempFlow;
    }
    else
        if (mDeactivated)
        {
            for (j=0;j<mNeighborCount;j++)
                if (mNeighborArray[j]==mpSource)
                    mpEventQ->UpdateFuture(mNeighborArray[j]);

            returncode=mTempFlow;
        }
        else
        {
            mFlow=mFlow-x;
            returncode=0;
        }

    return returncode;
}

int vzPin::GetFlow(bool lefttoright)
{
    if (mIsASource)
        return (mInitialFlow-mFlow);
    else
        return mFlow;
}

int vzPin::GetInitialFlow()
{
    return mInitialFlow;
}

void vzPin::SetStart(int flow)
{

```

vzPin.cpp

```

        mIsASource = true;
        mIsADrain = false;
        mFlow = flow;
        mInitialFlow = flow;
        mTempFlow = 0;
        mColor = pnSourceColor;

        if (Trace)
            DoDraw();

        mColor = pnDrainColor;
    }

void vzPin::DeactivatePin()
{
    mDeactivated = true;
    mColor = pnOffColor;

    if (Trace)
        DoDraw();

    mIsADrain = false;
}

int vzPin::DoLabel(bool lefttoright)
{
    int j, space;
    int returncode = 0;
    int flow, extra;
    extra=0;
    flow=0;
    if (mIsASource)
    {
        if ( (mLabel==false) && (mFlow>0) && (mNeighborCount!=0))
        {
            flow=mFlow/mNeighborCount;
            extra=mFlow%mNeighborCount;
            mColor = pnSourceColor;
            if (Trace)
                DoDraw();
            for (j=0;j<mNeighborCount;j++)
            {
                if (mNeighborArray[j]->GetLabel() =
= false)
                {
                    space = mNeighborArray[j]->

```

vzPin.cpp

```
        mLabel=true;
        return 0;
    }
}

void    vzPin::ClearStart()
{
    mColor=pinDrainColor;
    mFlow=0;
    mIsADrain=true;
    mIsASource=false;
    mTempFlow=0;
    mLabel=false;
    mDeactivated=false;
    mInitialFlow=0;
}
```

vzPin.h

```

/*
*****
*****
vzPin.h
Van Zandt
*****
*/

#ifndef _vzPin_h
#define _vzPin_h

#include "vzLib\vzVector.h"
#include "vzDevice.h"
#include "vzEventQ.h"

const int      pnOffColor  = DARKGRAY;
const int      pnDrainColor = WHITE;
const int      pnSourceColor = BLUE;
const int      pnLowColor  = YELLOW;
const int      pnMiddleColor = LIGHTBLUE;
const int      pnHighColor = BLUE;

class vzPin: public vzDevice
{
private:
    int      mID;
    pType    mType;
    int      mInX1;
    int      mInX2;
    int      mInY1;
    int      mInY2;
    int      mOutX1;
    int      mOutX2;
    int      mOutY1;
    int      mOutY2;
    int      mWordSize;
    int      mColor;
    int      mCurrent;
    vzEventQ *mpEventQ;
    bool     mAmIInPin;
    bool     mLabel;
    int      mFlow;
    bool     mIsADrain;
    bool     mIsASource;
    vzDevice *mpSource;

```

vzPin.h

```

        int            mTempFlow;
        bool           mDeactivated;
        int            mInitialFlow;
public:
    vzPin();
    ~vzPin();
    void              DoCreate(int id, pType type, int x1,int y1,int x2,i
nt y2,
                                int x12, int y12, int x22,
int y22,
                                int wordsize, vzEventQ *apE
ventQ);
    void              DoDraw();
    int               GetOutX1();
    int               GetOutY1();
    int               GetOutX2();
    int               GetOutY2();
        int            GetType() { return mType; };
        int            GetID() { return mID; };
    void              DoFlood();
    void              DoClear();
    bool              GetLabel();
    bool              IsADrain();
    void              GiveFlow(int flow, bool meaningless, vzDevice *p);
    int               DoAugment(int x);
    int               GetSpace(bool m) {return 0;};
    int               GetFlow(bool lefttoright);
    int               GetInitialFlow();
    void              SetStart(int flow);
    void              DeactivatePin();
    int               DoLabel(bool lefttoright);
    void              ClearStart();
};

#include "vzPin.cpp"
#endif // _vzPin_h
//*****
*****

```

vzPopulation.cpp

```

/*
*****
*****
Van Zandt
vzPopulation.cpp
*****
*****
*/

vzPopulation::vzPopulation(.    int tournamentcount,
                                int tournam
entSize,
                                int mutate,
                                int crossov
er,
                                int    ran
dom,
                                int count,
                                int size,
                                int alpha_s
ize,
                                double (* e
val)(vzIndividual *iPtr) )
:   mkTournamentCount(tournamentcount),
    mkTournamentSize(tournamentSize),
    mkMutate(mutate),
    mkCrossover(crossover),
    mkRandom(random),
    mkTotal(mutate + crossover + random),
    mkCount(count),
    mkSize(size),
    mkAlphabetCount(alpha_size),
    DoEvaluate(eval)
{
    int          i;

    mpIndividual.resize(mkCount);

    for (i=0; i<mkCount; i++)
        mpIndividual[i] = new vzIndividual(mkSize, mkAlphab
etCount);
}

vzPopulation::~vzPopulation()
{
    int          i;

    for (i=0; i<mkCount; i++)

```

vzPopulation.cpp

```

        delete mpIndividual[i];
    }

    /**
void vzPopulation::PrintToFile(ofstream out)
{
}
*/
    /**
void vzPopulation::DoPrint(int x, int y)
{
}
*/

void vzPopulation::ReGenerate(int cycle_count)
{
    int i, j;

    for (i=0; i<cycle_count; i++)
    {
        if ( mpIndividual[SelectBest()]->GetFitness() == 0.
0 )
            break;

        for (j=0; j<mkTournamentCount; j++)
        {
            if ( mpIndividual[SelectBest()]->GetFitness
() == 0.0 )
                break;

            DoTournament();
        }
    }
}

int vzPopulation::SelectBest()
{
    int i;
    int retBestIndex;
    double retBestFitness;

    for (i=0; i<mkCount; i++)
    {
        if ( i == 0 )
        {
            retBestIndex = 0;

```

vzPopulation.cpp

```

retBestFitness = mpIndividual[0]->GetFitness();
s();
    }
    else
    {
        if ( mpIndividual[i]->GetFitness() < retBestFitness )
        {
            retBestIndex = i;
            retBestFitness = mpIndividual[i]->GetFitness();
        }
    }
}

return retBestIndex;
}

void vzPopulation::DoTournament()
{
    int i;
    int r;
    int first = 0;
    int second = 0;
    int last = 0;
    double tmpFitness = 0.0;
    double firstFit = 1000000.0;
    double secondFit = 1000000.0;
    double lastFit = 0.0;

    for (i=0; i<mkTournamentSize; i++)
    {
        r = getRandom(mkCount);
        tmpFitness = mpIndividual[r]->GetFitness();

        if ( tmpFitness < firstFit )
        {
            secondFit = firstFit;
            firstFit = tmpFitness;
            second = first;
            first = r;
        }
        else if ( tmpFitness < secondFit )
        {
            secondFit = tmpFitness;
            second = r;
        }
        else if ( tmpFitness > lastFit )

```

vzPopulation.cpp

```

        {
            lastFit = tmpFitness;
            last = r;
        }
    }

    if ( first != last )
    {
        r = getRandom(mkTotal);

        if ( r < mkMutate )
        {
            DoClone(first, last);
            DoMutate(last);
        }
        else if ( r < mkMutate+mkCrossover )
        {
            if ( second != last )
            {
                DoClone(first, last);
                DoCrossover(last, second);
            }
        }
        else // ( r < mkMutate+mkCrossover+mkRandom )
        {
            mpIndividual[last]->BuildRandom();
        }

        mpIndividual[last]->SetFitness(DoEvaluate(mpIndividual[
    ual[last])));
    }

}

void    vzPopulation::DoClone(int from, int to)
{
    *(mpIndividual[to]) = *(mpIndividual[from]);
}

void    vzPopulation::DoMutate(int i)
{
    mpIndividual[i]->DoMutate();
}

void    vzPopulation::DoCrossover(int i, int j)
{
    mpIndividual[i]->DoCrossover(mpIndividual[j]);
}

```

vzPopulation.cpp

```
}

void vzPopulation::BuildRandom()
{
    int i;

    for (i=0; i<mkCount; i++)
    {
        mpIndividual[i]->BuildRandom();
        mpIndividual[i]->SetFitness(DoEvaluate(mpIndividual
[i]));
    }
}

//*****
*****
```

vzPopulation.h

```

/*
*****
*****
Van Zandt
vzPopulation.h
*****
*****
*/
#ifndef _vzPopulation_h
#define _vzPopulation_h

#include "vzLib\vzTools.h"
#include "vzLib\vzVector.h"
#include "vzIndividual.h"

class vzPopulation
{
private:
    const int mkTournamen
tCount;
    const int mkTournamen
tSize;
    const int mkMutate;
    const int mkCrossover;
;
    const int mkRandom;
    const int mkTotal;
    const int mkCount;
    const int mkSize;
    const int mkAlphabetC
ount;
    vzVector<vzIndividual *> mpIndividual;

public:
    vzPopulation( int tournamentcount,
                  int tournamentsize,
                  int mutate,
                  int crossover,
                  int random,
                  int count,
                  int size,
                  int alpha_size,
                  double (* eval)(vzIndividua
l *iPtr) );
    ~vzPopulation();

```

vzPopulation.h.

```

void    BuildRandom();
void    ReGenerate(int cycle_count);
void    DoTournament();
int      SelectBest();
void    DoClone(int from, int to);
void    DoMutate(int i);
void    DoCrossover(int i, int j);
vzIndividual *GetIndividualPtr(int i) { return mpIndividual[i]; };
double (* DoEvaluate)(vzIndividual *iPtr);
// void DoPrint(int x, int y);
// void PrintToFile(ofstream out);
};

#include      "vzPopulation.cpp"
#endif // _vzPopulation_h
//*****
*****

```

vzRegister.cpp

```

/*
*****
*****
vzRegister.cpp
Van Zandt
*****
*****
*/

vzRegister::vzRegister(int WordSize, int devicesize)
: mWordSize(WordSize), mWidth(devicesize*4), mX1(0), mY1(0), mX2(0)
, mY2(0)
{
    mPin= new vzPin[mWordSize];
}

vzRegister::~vzRegister()
{
    delete[] mPin;
    mPin=NULL;
}

void vzRegister::DoCreate(int x1, int y1, int y2, bool ingrid,
                          vzEventQ *a
pEventQ)
{
    int counter; // in for loop that draws yellow boxes
    int boxlength; // the calculated average length of each box
    int extra; // the leftover space for boxes
    int y1adder; // adders distribute leftover space to boxes
    int y2adder;

    mX1 = x1 + mWidth; //mWidth;
    mY1 = y1;
    mX2 = mX1 + mWidth; //mWidth;
    mY2 = y2;
    mpEventQ = apEventQ;

    boxlength=(mY2-mY1)/(2*mWordSize+1);
    extra=(mY2-mY1)%(2*mWordSize+1); // the space reserved
between ends
    y2adder=extra/2;
    y1adder=extra/2;

    if ( ingrid )
    {
        for (counter = 0;counter<mWordSize;counter++)
        {

```

vzRegister.cpp

```

        mPin[counter].DoCreate(counter,in,mX1+3,mY1+boxlength+y1adder+(boxlength+boxlength)*counter,
        mX2-3,mY1+boxlength+y2adder+(boxlength+boxlength)*counter+boxlength,
        mX1+mWidth+3,mY1+boxlength+y1adder+(boxlength+boxlength)*counter,
        mX1+2*mWidth-3,mY1+boxlength+y2adder+(boxlength+boxlength)*counter+boxlength,
        mWordSize,mpEventQ);

        if (counter==mWordSize-2)
        {
            if (extra%2>0)
                y2adder=extra/2+1;
            else
                y2adder=extra/2;
        }
    }
    else
    {
        for (counter = 0;counter<mWordSize;counter++)
        {
            mPin[counter].DoCreate(counter,out,mX1+3,mY1+boxlength+y1adder+(boxlength+boxlength)*counter,
            mX2-3,mY1+boxlength+y2adder+(boxlength+boxlength)*counter+boxlength,
            mX1-mWidth+3,mY1+boxlength+y1adder+(boxlength+boxlength)*counter,
            mX1-3,mY1+boxlength+y2adder+(boxlength+boxlength)*counter+boxlength,
            mWordSize,mpEventQ);

            if (counter==mWordSize-2)
            {
                if (extra%2>0)
                    y2adder=extra/2+1;
                else
                    y2adder=extra/2;
            }
        }
    }
}

void vzRegister::DoDraw()
{
    int counter;

```

vzRegister.cpp

```

        for (counter = 0;counter<mWordSize;counter++)
            mPin[counter].DoDraw();
    }

void vzRegister::DoFlood() :
{
    int counter;

    for (counter=0;counter<mWordSize;counter++)
        if (mPin[counter].GetFlow(true)>0)
            mPin[counter].DoFlood();
}

void vzRegister::DoClear()
{
    int counter;

    for (counter=0;counter<mWordSize;counter++)
        mPin[counter].DoClear();
}

int vzRegister::ConnectToPin(vzConnector *pAConnector, int SwitchDensity)
{
    int any;
    int connectorsize;
    int counter;
    int whichpin;

    any = pAConnector->GetY();
    connectorsize = pAConnector->GetSize();
    whichpin=-1;
    for (counter=0;counter<mWordSize;counter++)
    {
        if ((any>=mPin[counter].GetOutY1()) && (any<=mPin[counter].GetOutY2()))
            if ( getRandom(100) < SwitchDensity )
                whichpin=counter;
    }
    return whichpin;
}

vzPin *vzRegister::GetPinPtr(int i)
{
    return &mPin[i];
}

```

vzRegister.cpp

```
void vzRegister::DoLabel(bool lefttoright)
{
    int counter;

    for (counter=0;counter<mWordSize;counter++)
        mPin[counter].DoLabel(lefttoright);
}

int vzRegister::DoAugment()
{
    int counter, minflow;

    for (counter=0;counter<mWordSize;counter++)
    {
        if ( (mPin[counter].GetLabel()) && (mPin[counter].I
sADrain()) )
        {
            minflow = mPin[counter].DoAugment(0);
            break;
        }
    }

    return minflow;
}

bool vzRegister::CheckLabel()
{
    int counter;
    int check = 0;

    for (counter=0;counter<mWordSize;counter++)
        if (( mPin[counter].GetLabel() )&& (mPin[counter].I
sADrain()))
            check++;

    if ( check > 0 )
        return true;
    else
        return false;
}

int vzRegister::OnPin(int x1,int y1)
{
    int j;
    int answer;
    answer=-1;
    for (j=0;j<mWordSize;j++)
```

vzRegister.cpp

```
{
    if (mPin[j].GetOutX1() <= x1)
    if (mPin[j].GetOutX2() >= x1)
    if (mPin[j].GetOutY1() <= y1)
    if (mPin[j].GetOutY2() >= y1)
    {
        answer=j;
        break;
    }
}
return answer;
}

void vzRegister::SetPin(int WhichPin, int flow)
{
    mPin[WhichPin].SetStart(flow);
}

int vzRegister::PinHeight(int WhichPin)
{
    return (mPin[WhichPin].GetOutY1());
}

int vzRegister::GetPinFlow(int WhichPin)
{
    return (mPin[WhichPin].GetFlow(true));
}

void vzRegister::DeactivatePin(int WhichPin)
{
    mPin[WhichPin].DeactivatePin();
}

int vzRegister::GetInitialFlow(int WhichPin)
{
    return (mPin[WhichPin].GetInitialFlow());
}

void vzRegister::ClearStart()
{
    int counter;

    for (counter=0; counter<mWordSize; counter++)
        mPin[counter].ClearStart();
}

void vzRegister::Write(vzString string)
{

```

vzRegister.cpp

```
int counter;
int size;

if ( string.length() < mWordSize )
    size = string.length();
else
    size = mWordSize;

for (counter=0;counter<size;counter++)
    if (string[counter]=='0')
        mPin[counter].DeactivatePin();
    else if (string[counter]=='1')
        mPin[counter].SetStart(kStartFlow);
}

//*****
*****
```

vzRegister.h

```

/*
*****
*****
vzRegister.h
Van Zandt
*****
*****
*/

#ifndef _vzRegister_h
#define _vzRegister_h

#include "vzEventQ.h"
#include "vzPin.h"
#include "vzConnector.h"
#include "vzLib\vzString.h"

//const int kRegisterWidth=15;

class vzRegister
{
private:
    const int      mWordSize;
                int      mWidth;
                int      mX1;
                int      mY1;
                int      mX2;
                int      mY2;
    vzPin          *mPin;
    vzEventQ       *mpEventQ;
                bool      mAmIInReg;

public:
    vzRegister(int WordSize, int devicesize);
    ~vzRegister();
    void      DoCreate(int x1, int y1, int y2, bool ingrid, vzEventQ *apEventQ);
    void      DoDraw();
    void      DoFlood();
    int      ConnectToPin(vzConnector *pAConnector, int SwitchDensity);
    void      DoClear();
    vzPin     *GetPinPtr(int i);
    void      DoLabel(bool lefttoright);
    int      DoAugment();
    bool      CheckLabel();
    int      OnPin(int x1, int y1);
    void      SetPin(int WhichPin, int flow);

```

```
                                vzRegister.h

int      PinHeight(int WhichPin);
int      GetPinFlow(int WhichPin);
void     DeactivatePin(int WhichPin);
int      GetInitialFlow(int WhichPin);
void     ClearStart();
void     Write(vzString string);
};

#include "vzRegister.cpp"

#endif // _vzRegister_h

//*****
*****
```

vzRegister.h

```

/*
*****
*****
vzRegister.h
Van Zandt
*****
*****
*/

#ifndef _vzRegister_h
#define _vzRegister_h

#include "vzEventQ.h"
#include "vzPin.h"
#include "vzConnector.h"
#include "vzLib\vzString.h"

//const int kRegisterWidth=15;

class vzRegister
{
private:
    const int      mWordSize;
                   int      mWidth;
    int            mX1;
    int            mY1;
    int            mX2;
    int            mY2;
    vzPin          *mPin;
    vzEventQ       *mpEventQ;
    bool           mAmIInReg;

public:
    vzRegister(int WordSize, int devicesize);
    ~vzRegister();
    void        DoCreate(int x1, int y1, int y2, bool ingrid,vzEventQ *apEventQ);
    void        DoDraw();
    void        DoFlood();
    int         ConnectToPin(vzConnector *pAConnector, int SwitchDensity);
    void        DoClear();
    vzPin       *GetPinPtr(int i);
    void        DoLabel(bool lefttoright);
    int         DoAugment();
    bool        CheckLabel();
    int         OnPin(int x1,int y1);
    void        SetPin(int WhichPin, int flow);

```

vzRegister.h

```
int      PinHeight(int WhichPin);
int      GetPinFlow(int WhichPin);
void     DeactivatePin(int WhichPin);
int      GetInitialFlow(int WhichPin);
void     ClearStart();
void     Write(vzString string);
};

#include "vzRegister.cpp"

#endif // _vzRegister_h

//*****
*****
```

vzSwitch.cpp

```

/*
*****
*****
vzSwitch.cpp
Van Zandt
*****
*/

vzSwitch::vzSwitch()
{
}

vzSwitch::~vzSwitch()
{
}

void vzSwitch::DoCreate(int x1, int y1, int x2, int y2,
                        int state, vzEventQ
                        *apEventQ)
{
    mX1 = x1;
    mY1 = y1;
    mX2 = x2;
    mY2 = y2;
    mpEventQ = apEventQ;
    mState = state;

    if ( mState == 0 )
        mInCapacity = 0;
    else if (mState==1)
        mInCapacity=kLowCapacity;
    else
        mInCapacity=kHighCapacity;

    mOutCapacity = mInCapacity;
    mInFlow = 0;
    mOutFlow = 0;
    mLabel = false;
    mpSource = NULL;
    mTempFlow = 0;
    mColor = swLowColor;
}

void vzSwitch::DoDraw()
{
    int          i;

```

vzSwitch.cpp

```

    if (mState==2)
        mColor=swHighColor;

    if (mState==1)
        mColor=swLowColor;

    if (mState==0)
        mColor=swOffColor;

    line(screen,mX1,mY1,mX2,mY2,mColor);
    for (i=0;i<mNeighborCount;i++)
        mNeighborArray[i]->DoDraw();
}

void vzSwitch::DoFlood()
{
    int i;

    if ( mColor == swLowColor )
    {
        mColor = swHighColor;
        if (Trace)
            DoDraw();
        for (i=0;i<mNeighborCount;i++)
            mpEventQ->UpdateFuture(mNeighborArray[i]);
    }
}

void vzSwitch::DoClear()
{
    mColor = swLowColor;
    if (Trace)
        DoDraw();
    mLabel = false;
    mpSource = NULL;
    mTempFlow = 0;
    mAddFlow = true;
}

bool vzSwitch::GetLabel()
{
    return mLabel;
}

int vzSwitch::GetSpace(bool lefttoright)
{
    if (lefttoright)

```

vzSwitch.cpp

```

        return (mInCapacity-mInFlow);
    else
        return (mOutCapacity-mOutFlow);
}

int vzSwitch::GetFlow(bool lefttoright)
{
    if (lefttoright)
        return mOutFlow;
    else
        return mInFlow;
}

void vzSwitch::GiveFlow(int flow,bool inflow,vzDevice *p)
{
    mpSource = p;
    mTempFlow = flow;

    if (inflow)
        mAddFlow = true;
    else
        mAddFlow = false;
}

int vzSwitch::DoAugment(int x)
{
    int j;

    mColor=swHighColor;
    if (Trace)
        DoDraw();

    if (mAddFlow)
        mInFlow=mInFlow+x;
    else
        mOutFlow=mOutFlow+x;

    for (j=0;j<mNeighborCount;j++)
        if (mNeighborArray[j] == mpSource)
        {
            mpEventQ->UpdateFuture(mNeighborArray[j]);
            break;
        }

    return x;
}

int vzSwitch::DoLabel(bool lefttoright)

```

vzSwitch.cpp

```

{
    int    j, returncode;
    returncode=0;

    if ( (mLabel==false) && (mTempFlow>0) )
    {
        for (j=0;j<mNeighborCount;j++)
        {
            if ((mNeighborArray[j]!=mpSource)&&(mNeighborArray[j]->GetLabel()==false))
            {
                if (mNeighborArray[j]->IsADrain())
                {
                    mNeighborArray[j]->GiveFlow(mTempFlow, lefttoright, this);
                    returncode = 1;
                    break;
                }
                else
                {
                    mNeighborArray[j]->GiveFlow(mTempFlow, lefttoright, this);
                    mpEventQ->UpdateFuture(mNeighborArray[j]);
                }
            }
        }
        mLabel=true;
    }
    return returncode;
}

void vzSwitch::ClearStart()
{
    mInFlow = 0;
    mOutFlow = 0;
    mLabel = false;
    mpSource = NULL;
    mTempFlow = 0;
    mColor = swLowColor;
}

void vzSwitch::SetState(int n)
{
    if ( n == 0 )
    {
        mState = 0;
    }
}

```

vzSwitch.cpp

```
        mColor = swOffColor;
        mInCapacity=0;
    }
    else if ( n == 1 )
    {
        mState = 1;
        mColor = swLowColor;
        mInCapacity=kLowCapacity;
    }
    else
    {
        mState = 2;
        mColor = swHighColor;
        mInCapacity=kHighCapacity;
    }
    mOutCapacity=mInCapacity;
}
```

vzSwitch.h

```

/*
*****
vzSwitch.h
Van Zandt
*****
*/
..
#ifndef _vzSwitch_h
#define _vzSwitch_h

#include "vzDevice.h"

const int      swOffColor  = DARKGRAY;
const int      swLowColor  = BLACK;
const int      swHighColor = WHITE;

class vzSwitch: public vzDevice
{
private:
    int          mX1;
    int          mY1;
    int          mX2;
    int          mY2;
    int          mColor;
    vzEventQ     *mpEventQ;
    int          mInCapacity;
    int          mOutCapacity;
    int          mInFlow;
    int          mOutFlow;
    bool         mLabel;
    vzDevice     *mpSource;
    int          mTempFlow;
    bool         mAddFlow;
    int          mState;
    int          mNeighbor1Type;
    int          mNeighbor1Parm1;
    int          mNeighbor1Parm2;
    int          mNeighbor2Type;
    int          mNeighbor2Parm1;
    int          mNeighbor2Parm2;

public:
    vzSwitch();
    ~vzSwitch();
    void        DoCreate(int x1, int y1, int x2, int y2,

```

vzSwitch.h

```

                                int state, vzEventQ *apEven
tQ);
void      DoDraw();
void      DoFlood();
void      DoClear();
bool      GetLabel();
int        GetSpace(bool lefttoright);
int        GetFlow(bool lefttoright);
void      GiveFlow(int flow, bool inflow, vzDevice *p);
bool      IsADrain() {return false;};
int        DoAugment(int x);
int        DoLabel(bool lefttoright);
void      ClearStart();
void      SetState(int n);
void      SetNeighbor1Type(int n) { mNeighbor1Type = n; };
void      SetNeighbor1Parm1(int n) { mNeighbor1Parm1 = n; };
void      SetNeighbor1Parm2(int n) { mNeighbor1Parm2 = n; };
void      SetNeighbor2Type(int n) { mNeighbor2Type = n; };
void      SetNeighbor2Parm1(int n) { mNeighbor2Parm1 = n; };
void      SetNeighbor2Parm2(int n) { mNeighbor2Parm2 = n; };
int        GetState() {return mState;};
int        GetNeighbor1Type() { return mNeighbor1Type;
};
int        GetNeighbor1Parm1() { return mNeighbor1Parm
1; };
int        GetNeighbor1Parm2() { return mNeighbor1Parm
2; };
int        GetNeighbor2Type() { return mNeighbor2Type;
};
int        GetNeighbor2Parm1() { return mNeighbor2Parm
1; };
int        GetNeighbor2Parm2() { return mNeighbor2Parm
2; };
};

#include "vzSwitch.cpp"

#endif // _vzSwitch_h

```

未经训练的纳米单元连接表

Moletronics

.OPTIONS RELTOL=.01 ITL4=10 SCALE=1 POST METHOD=Gear DCSTEP=10

.OPTIONS ACCT=0 INGOLD=1 NOMOD PROBE

.TRAN 10000n 60000n

.Rout1 1 0 1

.PRINT TRAN I(Rout1)

C1 1 0 10f

C2 2 0 10f

C3 3 0 10f

C4 4 0 10f

C5 5 0 10f

C6 6 0 10f

C7 7 0 10f

C8 8 0 10f

C9 9 0 10f

C10 10 0 10f

C11 11 0 10f

C12 12 0 10f

C13 13 0 10f

C14 14 0 10f

VInputA15 15 0 PWL(0n,0V 20000n,0V 20001n,0.5V 30000n,0.5V 30001n,2V
40000n,2V

+ 40001n,0.5V 50000n,0.5V 50001n,0.5V 60000n,0.5V)

.PRINT TRAN 15

C15 15 0 10f

C16 16 0 10f

C17 17 0 10f

C18 18 0 10f

C19 19 0 10f

C20 20 0 10f

C24 24 0 5f

C27 27 0 5f

C29 29 0 5f

C32 32 0 5f

C34 34 0 5f

C35 35 0 5f

C37 37 0 5f

C39 39 0 5f

C42 42 0 5f

C44 44 0 5f

C47 47 0 5f

C49 49 0 5f

C50 50 0 5f

C52 52 0 5f

C53 53 0 5f

C54 54 0 5f

C55 55 0 5f

C56 56 0 5f

C57 57 0 5f

C58 58 0 5f

C59 59 0 5f

C60 60 0 5f

C61 61 0 5f

C62 62 0 5f

C63 63 0 5f

C64 64 0 5f

C67 67 0 5f

C68 68 0 5f
C69 69 0 5f
C70 70 0 5f
C71 71 0 5f
C72 72 0 5f
C74 74 0 5f
C75 75 0 5f
C76 76 0 5f
C77 77 0 5f
C78 78 0 5f
C79 79 0 5f
C82 82 0 5f
C83 83 0 5f
C84 84 0 5f
C85 85 0 5f
C86 86 0 5f
C87 87 0 5f
C88 88 0 5f
C89 89 0 5f
C90 90 0 5f
C92 92 0 5f
C93 93 0 5f
C94 94 0 5f
C97 97 0 5f
C99 99 0 5f
C101 101 0 5f
C102 102 0 5f
C103 103 0 5f
C104 104 0 5f
C105 105 0 5f
C106 106 0 5f
C108 108 0 5f
C109 109 0 5f
C111 111 0 5f
C112 112 0 5f
C113 113 0 5f
C114 114 0 5f
C115 115 0 5f
C116 116 0 5f
C119 119 0 5f
C120 120 0 5f
C121 121 0 5f
C122 122 0 5f
C123 123 0 5f
C124 124 0 5f
C125 125 0 5f
C128 128 0 5f
C129 129 0 5f
C130 130 0 5f
C131 131 0 5f
C132 132 0 5f
C133 133 0 5f
C134 134 0 5f
C135 135 0 5f
C136 136 0 5f
C137 137 0 5f
C138 138 0 5f
C139 139 0 5f
C142 142 0 5f
C143 143 0 5f
C144 144 0 5f

C145 145 0 5f
C146 146 0 5f
C147 147 0 5f
C149 149 0 5f
C150 150 0 5f
C151 151 0 5f
C152 152 0 5f
C153 153 0 5f
C154 154 0 5f
C157 157 0 5f
C158 158 0 5f
C159 159 0 5f
C160 160 0 5f
C161 161 0 5f
C162 162 0 5f
C163 163 0 5f
C164 164 0 5f
C165 165 0 5f
C166 166 0 5f
C167 167 0 5f
C168 168 0 5f
C169 169 0 5f
C172 172 0 5f
C173 173 0 5f
C175 175 0 5f
C176 176 0 5f
C178 178 0 5f
C179 179 0 5f
C180 180 0 5f
C181 181 0 5f
C182 182 0 5f
C183 183 0 5f
C184 184 0 5f
C186 186 0 5f
C187 187 0 5f
C188 188 0 5f
C190 190 0 5f
C191 191 0 5f
C192 192 0 5f
C193 193 0 5f
C194 194 0 5f
C195 195 0 5f
C196 196 0 5f
C197 197 0 5f
C198 198 0 5f
C199 199 0 5f
C200 200 0 5f
C202 202 0 5f
C203 203 0 5f
C204 204 0 5f
C205 205 0 5f
C206 206 0 5f
C207 207 0 5f
C208 208 0 5f
C209 209 0 5f
C210 210 0 5f
C212 212 0 5f
C214 214 0 5f
C217 217 0 5f
C219 219 0 5f
C222 222 0 5f

C224 224 0 5f
C229 229 0 5f
C231 231 0 5f
C232 232 0 5f
C234 234 0 5f
C237 237 0 5f
C239 239 0 5f
C242 242 0 5f
C244 244 0 5f
X512 20 244 NDRDIODE_0
X511 242 20 NDRDIODE_1
X510 19 242 NDRDIODE_0
X508 239 19 NDRDIODE_0
X507 18 239 NDRDIODE_1
X506 237 18 NDRDIODE_0
X505 237 17 NDRDIODE_0
X503 234 17 NDRDIODE_1
X502 234 16 NDRDIODE_1
X501 16 232 NDRDIODE_0
X500 232 231 NDRDIODE_1
X499 5 229 NDRDIODE_1
X498 229 244 NDRDIODE_0
X497 20 229 NDRDIODE_0
X496 20 229 NDRDIODE_0
X490 239 19 NDRDIODE_1
X489 224 19 NDRDIODE_1
X488 224 239 NDRDIODE_0
X487 18 224 NDRDIODE_0
X486 18 239 NDRDIODE_0
X484 18 237 NDRDIODE_0
X483 222 18 NDRDIODE_0
X482 237 222 NDRDIODE_1
X481 222 17 NDRDIODE_0
X480 237 17 NDRDIODE_1
X475 219 17 NDRDIODE_1
X474 219 17 NDRDIODE_0
X473 219 234 NDRDIODE_1
X472 219 16 NDRDIODE_0
X471 16 219 NDRDIODE_1
X469 217 16 NDRDIODE_0
X468 217 16 NDRDIODE_0
X467 217 232 NDRDIODE_0
X466 217 231 NDRDIODE_0
X465 10 231 NDRDIODE_0
X464 10 217 NDRDIODE_1
X462 214 5 NDRDIODE_0
X461 214 5 NDRDIODE_1
X460 214 229 NDRDIODE_1
X459 214 20 NDRDIODE_1
X458 212 20 NDRDIODE_0
X457 212 19 NDRDIODE_0
X456 210 19 NDRDIODE_0
X455 19 210 NDRDIODE_0
X454 210 224 NDRDIODE_0
X453 210 209 NDRDIODE_0
X452 209 224 NDRDIODE_0
X451 209 208 NDRDIODE_1
X450 224 208 NDRDIODE_0
X449 18 208 NDRDIODE_0
X448 222 208 NDRDIODE_1
X447 207 208 NDRDIODE_0

X446 222 207 NDRDIODE_0
X445 207 206 NDRDIODE_0
X444 206 222 NDRDIODE_0
X443 206 17 NDRDIODE_0
X442 206 17 NDRDIODE_0
X441 206 205 NDRDIODE_0
X440 205 17 NDRDIODE_0
X439 204 205 NDRDIODE_0
X438 17 204 NDRDIODE_1
X437 204 219 NDRDIODE_0
X436 16 204 NDRDIODE_1
X435 203 204 NDRDIODE_0
X434 203 16 NDRDIODE_1
X433 203 202 NDRDIODE_0
X432 16 202 NDRDIODE_1
X431 217 202 NDRDIODE_0
X430 202 10 NDRDIODE_1
X428 10 202 NDRDIODE_1
X427 200 5 NDRDIODE_1
X426 199 200 NDRDIODE_0
X425 199 5 NDRDIODE_0
X424 199 214 NDRDIODE_0
X423 199 198 NDRDIODE_0
X422 197 198 NDRDIODE_1
X421 212 197 NDRDIODE_1
X420 196 197 NDRDIODE_0
X419 195 196 NDRDIODE_1
X418 195 210 NDRDIODE_0
X417 195 209 NDRDIODE_0
X416 194 195 NDRDIODE_0
X415 194 209 NDRDIODE_1
X414 194 193 NDRDIODE_1
X413 193 209 NDRDIODE_1
X412 193 208 NDRDIODE_0
X411 207 193 NDRDIODE_1
X410 193 192 NDRDIODE_0
X409 207 192 NDRDIODE_1
X408 191 192 NDRDIODE_0
X407 207 191 NDRDIODE_0
X406 191 206 NDRDIODE_1
X405 205 191 NDRDIODE_0
X404 190 191 NDRDIODE_0
X403 205 190 NDRDIODE_0
X402 188 203 NDRDIODE_1
X401 188 187 NDRDIODE_1
X400 187 203 NDRDIODE_1
X399 202 187 NDRDIODE_1
X398 10 187 NDRDIODE_0
X397 186 10 NDRDIODE_1
X396 187 186 NDRDIODE_0
X395 200 4 NDRDIODE_0
X394 4 184 NDRDIODE_1
X393 184 200 NDRDIODE_0
X392 184 199 NDRDIODE_0
X391 184 198 NDRDIODE_0
X390 183 184 NDRDIODE_0
X389 198 183 NDRDIODE_1
X388 183 182 NDRDIODE_0
X387 182 198 NDRDIODE_1
X386 197 182 NDRDIODE_1
X385 182 196 NDRDIODE_0

X384 182 181 NDRDIODE_0
X383 196 181 NDRDIODE_1
X382 181 180 NDRDIODE_0
X381 196 180 NDRDIODE_1
X380 195 180 NDRDIODE_1
X379 194 180 NDRDIODE_1
X378 179 180 NDRDIODE_1
X377 179 194 NDRDIODE_0
X376 179 178 NDRDIODE_0
X375 178 194 NDRDIODE_1
X374 178 193 NDRDIODE_1
X373 192 178 NDRDIODE_1
X372 192 176 NDRDIODE_0
X371 191 176 NDRDIODE_0
X370 190 176 NDRDIODE_0
X369 175 176 NDRDIODE_0
X368 190 175 NDRDIODE_1
X367 188 173 NDRDIODE_0
X366 173 172 NDRDIODE_0
X365 188 172 NDRDIODE_0
X364 172 187 NDRDIODE_0
X363 172 186 NDRDIODE_0
X362 186 9 NDRDIODE_0
X361 9 172 NDRDIODE_1
X359 169 4 NDRDIODE_0
X358 169 4 NDRDIODE_1
X357 184 169 NDRDIODE_0
X356 183 169 NDRDIODE_0
X355 168 169 NDRDIODE_1
X354 183 168 NDRDIODE_1
X353 167 168 NDRDIODE_1
X352 167 183 NDRDIODE_0
X351 167 182 NDRDIODE_1
X350 181 167 NDRDIODE_0
X349 167 166 NDRDIODE_0
X348 181 166 NDRDIODE_1
X347 165 166 NDRDIODE_1
X346 181 165 NDRDIODE_1
X345 165 180 NDRDIODE_0
X344 179 165 NDRDIODE_1
X343 165 164 NDRDIODE_1
X342 179 164 NDRDIODE_1
X341 164 163 NDRDIODE_0
X340 179 163 NDRDIODE_0
X339 178 163 NDRDIODE_0
X338 162 163 NDRDIODE_1
X337 162 161 NDRDIODE_1
X336 161 176 NDRDIODE_1
X335 161 175 NDRDIODE_1
X334 161 160 NDRDIODE_0
X333 175 160 NDRDIODE_1
X332 160 159 NDRDIODE_0
X331 159 175 NDRDIODE_1
X330 173 159 NDRDIODE_1
X329 158 159 NDRDIODE_0
X328 158 173 NDRDIODE_1
X327 158 157 NDRDIODE_0
X326 157 173 NDRDIODE_0
X325 157 172 NDRDIODE_0
X324 157 9 NDRDIODE_0
X322 157 9 NDRDIODE_1

X320 3 154 NDRDIODE_1
X319 154 4 NDRDIODE_0
X318 154 169 NDRDIODE_0
X317 168 154 NDRDIODE_1
X316 153 154 NDRDIODE_1
X315 168 153 NDRDIODE_0
X314 152 153 NDRDIODE_0
X313 152 168 NDRDIODE_0
X312 152 167 NDRDIODE_0
X311 152 166 NDRDIODE_1
X310 152 151 NDRDIODE_1
X309 151 166 NDRDIODE_0
X308 150 151 NDRDIODE_1
X307 166 150 NDRDIODE_1
X306 165 150 NDRDIODE_1
X305 164 150 NDRDIODE_1
X304 150 149 NDRDIODE_0
X303 149 164 NDRDIODE_1
X302 147 162 NDRDIODE_0
X301 147 146 NDRDIODE_0
X300 162 146 NDRDIODE_0
X299 146 161 NDRDIODE_1
X298 146 160 NDRDIODE_1
X297 145 146 NDRDIODE_0
X296 145 160 NDRDIODE_1
X295 144 145 NDRDIODE_0
X294 160 144 NDRDIODE_1
X293 144 159 NDRDIODE_1
X292 144 158 NDRDIODE_0
X291 143 144 NDRDIODE_1
X290 158 143 NDRDIODE_1
X289 143 142 NDRDIODE_0
X288 158 142 NDRDIODE_0
X287 157 142 NDRDIODE_0
X286 9 142 NDRDIODE_0
X284 8 142 NDRDIODE_0
X282 139 3 NDRDIODE_1
X281 3 139 NDRDIODE_0
X280 139 154 NDRDIODE_1
X279 139 153 NDRDIODE_1
X278 139 138 NDRDIODE_1
X277 138 153 NDRDIODE_1
X276 137 138 NDRDIODE_1
X275 137 153 NDRDIODE_1
X274 152 137 NDRDIODE_0
X273 151 137 NDRDIODE_1
X272 136 137 NDRDIODE_0
X271 136 151 NDRDIODE_0
X270 136 135 NDRDIODE_1
X269 151 135 NDRDIODE_1
X268 150 135 NDRDIODE_0
X267 149 135 NDRDIODE_0
X266 135 134 NDRDIODE_0
X265 134 149 NDRDIODE_1
X264 134 133 NDRDIODE_1
X263 133 149 NDRDIODE_0
X262 133 147 NDRDIODE_0
X261 132 133 NDRDIODE_0
X260 147 132 NDRDIODE_0
X259 131 132 NDRDIODE_1
X258 147 131 NDRDIODE_0

X257 146 131 NDRDIODE_1
X256 131 145 NDRDIODE_0
X255 131 130 NDRDIODE_1
X254 145 130 NDRDIODE_1
X253 130 129 NDRDIODE_1
X252 129 145 NDRDIODE_0
X251 129 144 NDRDIODE_0
X250 143 129 NDRDIODE_1
X249 128 129 NDRDIODE_1
X248 143 128 NDRDIODE_0
X246 3 125 NDRDIODE_1
X245 124 125 NDRDIODE_0
X244 3 124 NDRDIODE_0
X243 124 139 NDRDIODE_1
X242 138 124 NDRDIODE_1
X241 124 123 NDRDIODE_1
X240 138 123 NDRDIODE_0
X239 122 123 NDRDIODE_0
X238 122 138 NDRDIODE_1
X237 122 137 NDRDIODE_0
X236 136 122 NDRDIODE_0
X235 121 122 NDRDIODE_0
X234 121 136 NDRDIODE_0
X233 121 120 NDRDIODE_1
X232 136 120 NDRDIODE_1
X231 120 135 NDRDIODE_1
X230 120 134 NDRDIODE_1
X229 120 119 NDRDIODE_1
X228 134 119 NDRDIODE_0
X227 115 116 NDRDIODE_1
X226 115 130 NDRDIODE_0
X225 114 115 NDRDIODE_1
X224 114 130 NDRDIODE_0
X223 129 114 NDRDIODE_0
X222 114 128 NDRDIODE_1
X221 113 114 NDRDIODE_0
X220 113 128 NDRDIODE_1
X219 112 113 NDRDIODE_1
X218 112 128 NDRDIODE_0
X217 112 8 NDRDIODE_0
X216 111 8 NDRDIODE_0
X215 112 111 NDRDIODE_1
X214 2 125 NDRDIODE_0
X213 2 109 NDRDIODE_0
X212 125 109 NDRDIODE_0
X211 109 124 NDRDIODE_1
X210 109 123 NDRDIODE_0
X209 108 109 NDRDIODE_1
X208 123 108 NDRDIODE_0
X207 121 106 NDRDIODE_1
X206 106 105 NDRDIODE_1
X205 121 105 NDRDIODE_0
X204 120 105 NDRDIODE_1
X203 119 105 NDRDIODE_0
X202 105 104 NDRDIODE_0
X201 119 104 NDRDIODE_0
X200 104 103 NDRDIODE_0
X199 103 119 NDRDIODE_0
X198 102 103 NDRDIODE_0
X197 101 102 NDRDIODE_1
X196 101 116 NDRDIODE_1

X195 115 101 NDRDIODE_0
X194 99 115 NDRDIODE_1
X193 114 99 NDRDIODE_1
X192 113 99 NDRDIODE_0
X191 97 113 NDRDIODE_0
X190 112 97 NDRDIODE_0
X189 111 97 NDRDIODE_0
X188 111 7 NDRDIODE_1
X187 97 7 NDRDIODE_0
X185 2 94 NDRDIODE_0
X184 2 94 NDRDIODE_1
X183 94 109 NDRDIODE_1
X182 94 108 NDRDIODE_1
X181 93 94 NDRDIODE_0
X180 93 108 NDRDIODE_1
X179 93 92 NDRDIODE_1
X178 108 92 NDRDIODE_0
X177 106 92 NDRDIODE_0
X176 106 90 NDRDIODE_0
X175 105 90 NDRDIODE_0
X174 90 104 NDRDIODE_1
X173 89 90 NDRDIODE_1
X172 104 89 NDRDIODE_1
X171 89 88 NDRDIODE_0
X170 88 104 NDRDIODE_1
X169 103 88 NDRDIODE_1
X168 88 102 NDRDIODE_0
X167 87 88 NDRDIODE_1
X166 87 102 NDRDIODE_1
X165 86 87 NDRDIODE_0
X164 86 102 NDRDIODE_0
X163 101 86 NDRDIODE_0
X162 86 85 NDRDIODE_1
X161 84 85 NDRDIODE_1
X160 84 99 NDRDIODE_1
X159 84 83 NDRDIODE_1
X158 83 82 NDRDIODE_1
X157 82 97 NDRDIODE_1
X156 82 7 NDRDIODE_1
X154 7 82 NDRDIODE_1
X152 1 79 NDRDIODE_1
X151 2 79 NDRDIODE_1
X150 94 79 NDRDIODE_1
X149 93 79 NDRDIODE_0
X148 78 79 NDRDIODE_1
X147 78 93 NDRDIODE_0
X146 78 77 NDRDIODE_1
X145 77 93 NDRDIODE_0
X144 92 77 NDRDIODE_1
X143 77 76 NDRDIODE_1
X142 76 75 NDRDIODE_1
X141 90 75 NDRDIODE_1
X140 75 89 NDRDIODE_0
X139 74 75 NDRDIODE_1
X138 74 89 NDRDIODE_1
X137 87 72 NDRDIODE_0
X136 72 71 NDRDIODE_0
X135 87 71 NDRDIODE_1
X134 71 86 NDRDIODE_1
X133 71 85 NDRDIODE_0
X132 70 71 NDRDIODE_0

X131 85 70 NDRDIODE_1
X130 70 69 NDRDIODE_0
X129 85 69 NDRDIODE_1
X128 69 84 NDRDIODE_0
X127 69 83 NDRDIODE_1
X126 68 69 NDRDIODE_0
X125 83 68 NDRDIODE_0
X124 68 67 NDRDIODE_0
X123 83 67 NDRDIODE_0
X122 67 82 NDRDIODE_1
X121 7 67 NDRDIODE_0
X119 6 67 NDRDIODE_1
X117 64 1 NDRDIODE_1
X116 64 1 NDRDIODE_0
X115 79 64 NDRDIODE_0
X114 64 78 NDRDIODE_0
X113 63 64 NDRDIODE_0
X112 78 63 NDRDIODE_1
X111 63 62 NDRDIODE_0
X110 78 62 NDRDIODE_1
X109 62 77 NDRDIODE_1
X108 62 76 NDRDIODE_1
X107 62 61 NDRDIODE_0
X106 76 61 NDRDIODE_0
X105 60 61 NDRDIODE_0
X104 76 60 NDRDIODE_0
X103 60 75 NDRDIODE_1
X102 74 60 NDRDIODE_1
X101 59 60 NDRDIODE_1
X100 59 74 NDRDIODE_1
X99 59 58 NDRDIODE_0
X98 58 74 NDRDIODE_0
X97 72 58 NDRDIODE_0
X96 57 58 NDRDIODE_1
X95 72 57 NDRDIODE_1
X94 56 57 NDRDIODE_1
X93 72 56 NDRDIODE_1
X92 56 71 NDRDIODE_0
X91 56 70 NDRDIODE_0
X90 56 55 NDRDIODE_0
X89 55 70 NDRDIODE_0
X88 54 55 NDRDIODE_1
X87 70 54 NDRDIODE_0
X86 69 54 NDRDIODE_1
X85 54 68 NDRDIODE_1
X84 54 53 NDRDIODE_0
X83 68 53 NDRDIODE_0
X82 52 53 NDRDIODE_0
X81 68 52 NDRDIODE_0
X80 67 52 NDRDIODE_1
X79 52 6 NDRDIODE_1
X77 6 52 NDRDIODE_1
X76 1 50 NDRDIODE_0
X75 50 49 NDRDIODE_0
X74 49 1 NDRDIODE_1
X73 64 49 NDRDIODE_1
X72 63 49 NDRDIODE_1
X71 15 49 NDRDIODE_0
X70 15 63 NDRDIODE_1
X69 15 47 NDRDIODE_1
X68 63 47 NDRDIODE_1

```

X67 62 47 NDRDIODE_0
X66 47 61 NDRDIODE_0
X65 47 14 NDRDIODE_1
X64 61 14 NDRDIODE_0
X62 14 61 NDRDIODE_1
X61 60 14 NDRDIODE_0
X60 14 59 NDRDIODE_1
X59 14 44 NDRDIODE_1
X58 44 59 NDRDIODE_1
X57 13 44 NDRDIODE_1
X56 13 59 NDRDIODE_1
X55 13 58 NDRDIODE_0
X54 13 57 NDRDIODE_1
X53 13 42 NDRDIODE_1
X52 57 42 NDRDIODE_0
X51 12 42 NDRDIODE_1
X50 57 12 NDRDIODE_0
X49 12 56 NDRDIODE_1
X48 55 12 NDRDIODE_1
X46 12 55 NDRDIODE_1
X45 12 39 NDRDIODE_0
X44 55 39 NDRDIODE_1
X43 39 54 NDRDIODE_0
X42 53 39 NDRDIODE_1
X41 11 39 NDRDIODE_1
X40 11 53 NDRDIODE_0
X39 37 11 NDRDIODE_0
X38 53 37 NDRDIODE_0
X37 37 52 NDRDIODE_0
X36 6 37 NDRDIODE_0
X35 35 50 NDRDIODE_0
X34 35 34 NDRDIODE_1
X33 50 34 NDRDIODE_1
X32 34 49 NDRDIODE_0
X31 15 34 NDRDIODE_1
X30 32 15 NDRDIODE_0
X29 32 47 NDRDIODE_1
X28 32 14 NDRDIODE_1
X27 32 14 NDRDIODE_1
X22 44 14 NDRDIODE_0
X21 14 29 NDRDIODE_1
X20 29 44 NDRDIODE_0
X19 13 29 NDRDIODE_0
X18 44 13 NDRDIODE_1
X16 13 42 NDRDIODE_0
X15 13 27 NDRDIODE_1
X14 42 27 NDRDIODE_1
X13 27 12 NDRDIODE_0
X12 12 42 NDRDIODE_1
X7 12 24 NDRDIODE_0
X6 12 24 NDRDIODE_1
X5 39 24 NDRDIODE_0
X4 11 24 NDRDIODE_0
X3 11 24 NDRDIODE_0

```

```

.SUBCKT NDRDIODE_1 IN OUT
G1 IN OUT PWL(1) IN OUT -1V,-1n -.8V,-1n -.625V,-760n 0V,0n
.625V,760n .8V,1n 1V,1n SMOOTH=1
.ENDS

```

```
.SUBCKT NDRDIODE_0 IN OUT
G1 IN OUT PWL(1) IN OUT -2V,-.8n -1V,-.8n -.75V,-.55n -.5V,-.21n
0V,0n .5V,.21n .75V,.55n 1V,.8n 2V,.8n SMOOTH=1
.ENDS
```

```
.END
```

反相器连接表

Moletronics

.OPTIONS RELTOL=.01 ITL4=10 SCALE=1 POST METHOD=Gear DCSTEP=10

.OPTIONS ACCT=0 INGOLD=1 NOMOD PROBE

.TRAN 10000n 60000n

ROut1 1 0 1

.PRINT TRAN I(Rout1)

C1 1 0 10f

C2 2 0 10f

C3 3 0 10f

C4 4 0 10f

C5 5 0 10f

C6 6 0 10f

C7 7 0 10f

C8 8 0 10f

C9 9 0 10f

C10 10 0 10f

C11 11 0 10f

C12 12 0 10f

C13 13 0 10f

C14 14 0 10f

VInputA15 15 0 PWL(0n,0V 20000n,0V 20001n,0.5V 30000n,0.5V 30001n,2V
40000n,2V

+ 40001n,0.5V 50000n,0.5V 50001n,0.5V 60000n,0.5V)

.PRINT TRAN 15

C15 15 0 10f

C16 16 0 10f

C17 17 0 10f

C18 18 0 10f

C19 19 0 10f

C20 20 0 10f

C24 24 0 5f

C27 27 0 5f

C29 29 0 5f

C32 32 0 5f

C34 34 0 5f

C35 35 0 5f

C37 37 0 5f

C39 39 0 5f

C42 42 0 5f

C44 44 0 5f

C47 47 0 5f

C49 49 0 5f

C50 50 0 5f

C52 52 0 5f

C53 53 0 5f

C54 54 0 5f

C55 55 0 5f

C56 56 0 5f

C57 57 0 5f

C58 58 0 5f

C59 59 0 5f

C60 60 0 5f

C61 61 0 5f

C62 62 0 5f

C63 63 0 5f

C64 64 0 5f

C67 67 0 5f

C68 68 0 5f

C69 69 0 5f
C70 70 0 5f
C71 71 0 5f
C72 72 0 5f
C74 74 0 5f
C75 75 0 5f
C76 76 0 5f
C77 77 0 5f
C78 78 0 5f
C79 79 0 5f
C82 82 0 5f
C83 83 0 5f
C84 84 0 5f
C85 85 0 5f
C86 86 0 5f
C87 87 0 5f
C88 88 0 5f
C89 89 0 5f
C90 90 0 5f
C92 92 0 5f
C93 93 0 5f
C94 94 0 5f
C97 97 0 5f
C99 99 0 5f
C101 101 0 5f
C102 102 0 5f
C103 103 0 5f
C104 104 0 5f
C105 105 0 5f
C106 106 0 5f
C108 108 0 5f
C109 109 0 5f
C111 111 0 5f
C112 112 0 5f
C113 113 0 5f
C114 114 0 5f
C115 115 0 5f
C116 116 0 5f
C119 119 0 5f
C120 120 0 5f
C121 121 0 5f
C122 122 0 5f
C123 123 0 5f
C124 124 0 5f
C125 125 0 5f
C128 128 0 5f
C129 129 0 5f
C130 130 0 5f
C131 131 0 5f
C132 132 0 5f
C133 133 0 5f
C134 134 0 5f
C135 135 0 5f
C136 136 0 5f
C137 137 0 5f
C138 138 0 5f
C139 139 0 5f
C142 142 0 5f
C143 143 0 5f
C144 144 0 5f
C145 145 0 5f

C146 146 0 5f
C147 147 0 5f
C149 149 0 5f
C150 150 0 5f
C151 151 0 5f
C152 152 0 5f
C153 153 0 5f
C154 154 0 5f
C157 157 0 5f
C158 158 0 5f
C159 159 0 5f
C160 160 0 5f
C161 161 0 5f
C162 162 0 5f
C163 163 0 5f
C164 164 0 5f
C165 165 0 5f
C166 166 0 5f
C167 167 0 5f
C168 168 0 5f
C169 169 0 5f
C172 172 0 5f
C173 173 0 5f
C175 175 0 5f
C176 176 0 5f
C178 178 0 5f
C179 179 0 5f
C180 180 0 5f
C181 181 0 5f
C182 182 0 5f
C183 183 0 5f
C184 184 0 5f
C186 186 0 5f
C187 187 0 5f
C188 188 0 5f
C190 190 0 5f
C191 191 0 5f
C192 192 0 5f
C193 193 0 5f
C194 194 0 5f
C195 195 0 5f
C196 196 0 5f
C197 197 0 5f
C198 198 0 5f
C199 199 0 5f
C200 200 0 5f
C202 202 0 5f
C203 203 0 5f
C204 204 0 5f
C205 205 0 5f
C206 206 0 5f
C207 207 0 5f
C208 208 0 5f
C209 209 0 5f
C210 210 0 5f
C212 212 0 5f
C214 214 0 5f
C217 217 0 5f
C219 219 0 5f
C222 222 0 5f
C224 224 0 5f

C229 229 0 5f
C231 231 0 5f
C232 232 0 5f
C234 234 0 5f
C237 237 0 5f
C239 239 0 5f
C242 242 0 5f
C244 244 0 5f
X512 20 244 NDRDIODE_0
X511 242 20 NDRDIODE_0
X510 19 242 NDRDIODE_1
X508 239 19 NDRDIODE_1
X507 18 239 NDRDIODE_1
X506 237 18 NDRDIODE_0
X505 237 17 NDRDIODE_1
X503 234 17 NDRDIODE_0
X502 234 16 NDRDIODE_0
X501 16 232 NDRDIODE_1
X500 232 231 NDRDIODE_0
X499 5 229 NDRDIODE_0
X498 229 244 NDRDIODE_0
X497 20 229 NDRDIODE_0
X496 20 229 NDRDIODE_0
X490 239 19 NDRDIODE_1
X489 224 19 NDRDIODE_1
X488 224 239 NDRDIODE_1
X487 18 224 NDRDIODE_1
X486 18 239 NDRDIODE_0
X484 18 237 NDRDIODE_1
X483 222 18 NDRDIODE_1
X482 237 222 NDRDIODE_0
X481 222 17 NDRDIODE_0
X480 237 17 NDRDIODE_0
X475 219 17 NDRDIODE_1
X474 219 17 NDRDIODE_0
X473 219 234 NDRDIODE_1
X472 219 16 NDRDIODE_0
X471 16 219 NDRDIODE_0
X469 217 16 NDRDIODE_1
X468 217 16 NDRDIODE_0
X467 217 232 NDRDIODE_0
X466 217 231 NDRDIODE_0
X465 10 231 NDRDIODE_1
X464 10 217 NDRDIODE_0
X462 214 5 NDRDIODE_0
X461 214 5 NDRDIODE_1
X460 214 229 NDRDIODE_1
X459 214 20 NDRDIODE_0
X458 212 20 NDRDIODE_0
X457 212 19 NDRDIODE_1
X456 210 19 NDRDIODE_1
X455 19 210 NDRDIODE_0
X454 210 224 NDRDIODE_1
X453 210 209 NDRDIODE_1
X452 209 224 NDRDIODE_1
X451 209 208 NDRDIODE_1
X450 224 208 NDRDIODE_0
X449 18 208 NDRDIODE_1
X448 222 208 NDRDIODE_0
X447 207 208 NDRDIODE_1
X446 222 207 NDRDIODE_0

X445 207 206 NDRDIODE_0
X444 206 222 NDRDIODE_1
X443 206 17 NDRDIODE_0
X442 206 17 NDRDIODE_1
X441 206 205 NDRDIODE_1
X440 205 17 NDRDIODE_0
X439 204 205 NDRDIODE_0
X438 17 204 NDRDIODE_0
X437 204 219 NDRDIODE_1
X436 16 204 NDRDIODE_1
X435 203 204 NDRDIODE_0
X434 203 16 NDRDIODE_1
X433 203 202 NDRDIODE_1
X432 16 202 NDRDIODE_1
X431 217 202 NDRDIODE_0
X430 202 10 NDRDIODE_0
X428 10 202 NDRDIODE_1
X427 200 5 NDRDIODE_0
X426 199 200 NDRDIODE_0
X425 199 5 NDRDIODE_1
X424 199 214 NDRDIODE_0
X423 199 198 NDRDIODE_1
X422 197 198 NDRDIODE_0
X421 212 197 NDRDIODE_1
X420 196 197 NDRDIODE_0
X419 195 196 NDRDIODE_1
X418 195 210 NDRDIODE_0
X417 195 209 NDRDIODE_0
X416 194 195 NDRDIODE_0
X415 194 209 NDRDIODE_0
X414 194 193 NDRDIODE_1
X413 193 209 NDRDIODE_1
X412 193 208 NDRDIODE_0
X411 207 193 NDRDIODE_0
X410 193 192 NDRDIODE_0
X409 207 192 NDRDIODE_0
X408 191 192 NDRDIODE_1
X407 207 191 NDRDIODE_1
X406 191 206 NDRDIODE_1
X405 205 191 NDRDIODE_0
X404 190 191 NDRDIODE_1
X403 205 190 NDRDIODE_0
X402 188 203 NDRDIODE_1
X401 188 187 NDRDIODE_0
X400 187 203 NDRDIODE_1
X399 202 187 NDRDIODE_1
X398 10 187 NDRDIODE_1
X397 186 10 NDRDIODE_1
X396 187 186 NDRDIODE_0
X395 200 4 NDRDIODE_1
X394 4 184 NDRDIODE_0
X393 184 200 NDRDIODE_1
X392 184 199 NDRDIODE_1
X391 184 198 NDRDIODE_0
X390 183 184 NDRDIODE_0
X389 198 183 NDRDIODE_1
X388 183 182 NDRDIODE_0
X387 182 198 NDRDIODE_0
X386 197 182 NDRDIODE_0
X385 182 196 NDRDIODE_0
X384 182 181 NDRDIODE_0

X383 196 181 NDRDIODE_1
X382 181 180 NDRDIODE_1
X381 196 180 NDRDIODE_1
X380 195 180 NDRDIODE_0
X379 194 180 NDRDIODE_1
X378 179 180 NDRDIODE_1
X377 179 194 NDRDIODE_1
X376 179 178 NDRDIODE_0
X375 178 194 NDRDIODE_1
X374 178 193 NDRDIODE_1
X373 192 178 NDRDIODE_1
X372 192 176 NDRDIODE_1
X371 191 176 NDRDIODE_0
X370 190 176 NDRDIODE_0
X369 175 176 NDRDIODE_1
X368 190 175 NDRDIODE_0
X367 188 173 NDRDIODE_1
X366 173 172 NDRDIODE_0
X365 188 172 NDRDIODE_1
X364 172 187 NDRDIODE_1
X363 172 186 NDRDIODE_1
X362 186 9 NDRDIODE_0
X361 9 172 NDRDIODE_1
X359 169 4 NDRDIODE_1
X358 169 4 NDRDIODE_1
X357 184 169 NDRDIODE_0
X356 183 169 NDRDIODE_1
X355 168 169 NDRDIODE_1
X354 183 168 NDRDIODE_0
X353 167 168 NDRDIODE_1
X352 167 183 NDRDIODE_0
X351 167 182 NDRDIODE_1
X350 181 167 NDRDIODE_1
X349 167 166 NDRDIODE_1
X348 181 166 NDRDIODE_1
X347 165 166 NDRDIODE_1
X346 181 165 NDRDIODE_1
X345 165 180 NDRDIODE_0
X344 179 165 NDRDIODE_0
X343 165 164 NDRDIODE_1
X342 179 164 NDRDIODE_1
X341 164 163 NDRDIODE_0
X340 179 163 NDRDIODE_0
X339 178 163 NDRDIODE_0
X338 162 163 NDRDIODE_1
X337 162 161 NDRDIODE_0
X336 161 176 NDRDIODE_1
X335 161 175 NDRDIODE_0
X334 161 160 NDRDIODE_1
X333 175 160 NDRDIODE_1
X332 160 159 NDRDIODE_1
X331 159 175 NDRDIODE_0
X330 173 159 NDRDIODE_0
X329 158 159 NDRDIODE_1
X328 158 173 NDRDIODE_1
X327 158 157 NDRDIODE_1
X326 157 173 NDRDIODE_1
X325 157 172 NDRDIODE_1
X324 157 9 NDRDIODE_1
X322 157 9 NDRDIODE_0
X320 3 154 NDRDIODE_1

X319 154 4 NDRDIODE_0
X318 154 169 NDRDIODE_1
X317 168 154 NDRDIODE_1
X316 153 154 NDRDIODE_1
X315 168 153 NDRDIODE_0
X314 152 153 NDRDIODE_0
X313 152 168 NDRDIODE_0
X312 152 167 NDRDIODE_0
X311 152 166 NDRDIODE_0
X310 152 151 NDRDIODE_0
X309 151 166 NDRDIODE_0
X308 150 151 NDRDIODE_0
X307 166 150 NDRDIODE_1
X306 165 150 NDRDIODE_0
X305 164 150 NDRDIODE_0
X304 150 149 NDRDIODE_0
X303 149 164 NDRDIODE_0
X302 147 162 NDRDIODE_0
X301 147 146 NDRDIODE_0
X300 162 146 NDRDIODE_0
X299 146 161 NDRDIODE_0
X298 146 160 NDRDIODE_0
X297 145 146 NDRDIODE_1
X296 145 160 NDRDIODE_1
X295 144 145 NDRDIODE_1
X294 160 144 NDRDIODE_1
X293 144 159 NDRDIODE_0
X292 144 158 NDRDIODE_0
X291 143 144 NDRDIODE_0
X290 158 143 NDRDIODE_1
X289 143 142 NDRDIODE_1
X288 158 142 NDRDIODE_0
X287 157 142 NDRDIODE_1
X286 9 142 NDRDIODE_0
X284 8 142 NDRDIODE_0
X282 139 3 NDRDIODE_1
X281 3 139 NDRDIODE_0
X280 139 154 NDRDIODE_1
X279 139 153 NDRDIODE_0
X278 139 138 NDRDIODE_0
X277 138 153 NDRDIODE_1
X276 137 138 NDRDIODE_1
X275 137 153 NDRDIODE_1
X274 152 137 NDRDIODE_1
X273 151 137 NDRDIODE_1
X272 136 137 NDRDIODE_0
X271 136 151 NDRDIODE_0
X270 136 135 NDRDIODE_0
X269 151 135 NDRDIODE_1
X268 150 135 NDRDIODE_0
X267 149 135 NDRDIODE_0
X266 135 134 NDRDIODE_1
X265 134 149 NDRDIODE_0
X264 134 133 NDRDIODE_1
X263 133 149 NDRDIODE_1
X262 133 147 NDRDIODE_1
X261 132 133 NDRDIODE_0
X260 147 132 NDRDIODE_0
X259 131 132 NDRDIODE_0
X258 147 131 NDRDIODE_0
X257 146 131 NDRDIODE_0

X256 131 145 NDRDIODE_1
X255 131 130 NDRDIODE_0
X254 145 130 NDRDIODE_0
X253 130 129 NDRDIODE_1
X252 129 145 NDRDIODE_1
X251 129 144 NDRDIODE_0
X250 143 129 NDRDIODE_1
X249 128 129 NDRDIODE_0
X248 143 128 NDRDIODE_0
X246 3 125 NDRDIODE_0
X245 124 125 NDRDIODE_0
X244 3 124 NDRDIODE_0
X243 124 139 NDRDIODE_1
X242 138 124 NDRDIODE_1
X241 124 123 NDRDIODE_0
X240 138 123 NDRDIODE_1
X239 122 123 NDRDIODE_0
X238 122 138 NDRDIODE_0
X237 122 137 NDRDIODE_1
X236 136 122 NDRDIODE_1
X235 121 122 NDRDIODE_1
X234 121 136 NDRDIODE_0
X233 121 120 NDRDIODE_1
X232 136 120 NDRDIODE_1
X231 120 135 NDRDIODE_1
X230 120 134 NDRDIODE_1
X229 120 119 NDRDIODE_0
X228 134 119 NDRDIODE_0
X227 115 116 NDRDIODE_0
X226 115 130 NDRDIODE_1
X225 114 115 NDRDIODE_1
X224 114 130 NDRDIODE_0
X223 129 114 NDRDIODE_0
X222 114 128 NDRDIODE_1
X221 113 114 NDRDIODE_0
X220 113 128 NDRDIODE_0
X219 112 113 NDRDIODE_1
X218 112 128 NDRDIODE_1
X217 112 8 NDRDIODE_1
X216 111 8 NDRDIODE_1
X215 112 111 NDRDIODE_0
X214 2 125 NDRDIODE_0
X213 2 109 NDRDIODE_0
X212 125 109 NDRDIODE_0
X211 109 124 NDRDIODE_1
X210 109 123 NDRDIODE_0
X209 108 109 NDRDIODE_0
X208 123 108 NDRDIODE_1
X207 121 106 NDRDIODE_0
X206 106 105 NDRDIODE_0
X205 121 105 NDRDIODE_0
X204 120 105 NDRDIODE_1
X203 119 105 NDRDIODE_0
X202 105 104 NDRDIODE_0
X201 119 104 NDRDIODE_0
X200 104 103 NDRDIODE_0
X199 103 119 NDRDIODE_1
X198 102 103 NDRDIODE_0
X197 101 102 NDRDIODE_1
X196 101 116 NDRDIODE_1
X195 115 101 NDRDIODE_1

X194 99 115 NDRDIODE_0
X193 114 99 NDRDIODE_0
X192 113 99 NDRDIODE_1
X191 97 113 NDRDIODE_0
X190 112 97 NDRDIODE_1
X189 111 97 NDRDIODE_1
X188 111 7 NDRDIODE_1
X187 97 7 NDRDIODE_1
X185 2 94 NDRDIODE_1
X184 2 94 NDRDIODE_0
X183 94 109 NDRDIODE_0
X182 94 108 NDRDIODE_0
X181 93 94 NDRDIODE_0
X180 93 108 NDRDIODE_0
X179 93 92 NDRDIODE_1
X178 108 92 NDRDIODE_0
X177 106 92 NDRDIODE_0
X176 106 90 NDRDIODE_0
X175 105 90 NDRDIODE_0
X174 90 104 NDRDIODE_1
X173 89 90 NDRDIODE_0
X172 104 89 NDRDIODE_1
X171 89 88 NDRDIODE_1
X170 88 104 NDRDIODE_0
X169 103 88 NDRDIODE_0
X168 88 102 NDRDIODE_1
X167 87 88 NDRDIODE_0
X166 87 102 NDRDIODE_0
X165 86 87 NDRDIODE_1
X164 86 102 NDRDIODE_1
X163 101 86 NDRDIODE_0
X162 86 85 NDRDIODE_1
X161 84 85 NDRDIODE_0
X160 84 99 NDRDIODE_0
X159 84 83 NDRDIODE_0
X158 83 82 NDRDIODE_0
X157 82 97 NDRDIODE_1
X156 82 7 NDRDIODE_0
X154 7 82 NDRDIODE_1
X152 1 79 NDRDIODE_0
X151 2 79 NDRDIODE_0
X150 94 79 NDRDIODE_1
X149 93 79 NDRDIODE_0
X148 78 79 NDRDIODE_1
X147 78 93 NDRDIODE_0
X146 78 77 NDRDIODE_0
X145 77 93 NDRDIODE_0
X144 92 77 NDRDIODE_0
X143 77 76 NDRDIODE_0
X142 76 75 NDRDIODE_0
X141 90 75 NDRDIODE_1
X140 75 89 NDRDIODE_1
X139 74 75 NDRDIODE_1
X138 74 89 NDRDIODE_1
X137 87 72 NDRDIODE_0
X136 72 71 NDRDIODE_0
X135 87 71 NDRDIODE_0
X134 71 86 NDRDIODE_0
X133 71 85 NDRDIODE_1
X132 70 71 NDRDIODE_1
X131 85 70 NDRDIODE_0

X130 70 69 NDRDIODE_1
X129 85 69 NDRDIODE_1
X128 69 84 NDRDIODE_1
X127 69 83 NDRDIODE_1
X126 68 69 NDRDIODE_1
X125 83 68 NDRDIODE_0
X124 68 67 NDRDIODE_1
X123 83 67 NDRDIODE_0
X122 67 82 NDRDIODE_0
X121 7 67 NDRDIODE_1
X119 6 67 NDRDIODE_1
X117 64 1 NDRDIODE_1
X116 64 1 NDRDIODE_0
X115 79 64 NDRDIODE_0
X114 64 78 NDRDIODE_0
X113 63 64 NDRDIODE_1
X112 78 63 NDRDIODE_0
X111 63 62 NDRDIODE_1
X110 78 62 NDRDIODE_1
X109 62 77 NDRDIODE_0
X108 62 76 NDRDIODE_0
X107 62 61 NDRDIODE_0
X106 76 61 NDRDIODE_1
X105 60 61 NDRDIODE_1
X104 76 60 NDRDIODE_1
X103 60 75 NDRDIODE_1
X102 74 60 NDRDIODE_0
X101 59 60 NDRDIODE_0
X100 59 74 NDRDIODE_1
X99 59 58 NDRDIODE_1
X98 58 74 NDRDIODE_1
X97 72 58 NDRDIODE_0
X96 57 58 NDRDIODE_0
X95 72 57 NDRDIODE_1
X94 56 57 NDRDIODE_0
X93 72 56 NDRDIODE_1
X92 56 71 NDRDIODE_1
X91 56 70 NDRDIODE_0
X90 56 55 NDRDIODE_0
X89 55 70 NDRDIODE_1
X88 54 55 NDRDIODE_1
X87 70 54 NDRDIODE_1
X86 69 54 NDRDIODE_1
X85 54 68 NDRDIODE_1
X84 54 53 NDRDIODE_1
X83 68 53 NDRDIODE_1
X82 52 53 NDRDIODE_0
X81 68 52 NDRDIODE_1
X80 67 52 NDRDIODE_1
X79 52 6 NDRDIODE_1
X77 6 52 NDRDIODE_0
X76 1 50 NDRDIODE_1
X75 50 49 NDRDIODE_1
X74 49 1 NDRDIODE_1
X73 64 49 NDRDIODE_0
X72 63 49 NDRDIODE_0
X71 15 49 NDRDIODE_1
X70 15 63 NDRDIODE_1
X69 15 47 NDRDIODE_0
X68 63 47 NDRDIODE_1
X67 62 47 NDRDIODE_1

```

X66 47 61 NDRDIODE_1
X65 47 14 NDRDIODE_1
X64 61 14 NDRDIODE_1
X62 14 61 NDRDIODE_0
X61 60 14 NDRDIODE_1
X60 14 59 NDRDIODE_1
X59 14 44 NDRDIODE_0
X58 44 59 NDRDIODE_1
X57 13 44 NDRDIODE_1
X56 13 59 NDRDIODE_1
X55 13 58 NDRDIODE_1
X54 13 57 NDRDIODE_0
X53 13 42 NDRDIODE_1
X52 57 42 NDRDIODE_0
X51 12 42 NDRDIODE_1
X50 57 12 NDRDIODE_0
X49 12 56 NDRDIODE_1
X48 55 12 NDRDIODE_0
X46 12 55 NDRDIODE_0
X45 12 39 NDRDIODE_0
X44 55 39 NDRDIODE_0
X43 39 54 NDRDIODE_1
X42 53 39 NDRDIODE_1
X41 11 39 NDRDIODE_0
X40 11 53 NDRDIODE_0
X39 37 11 NDRDIODE_0
X38 53 37 NDRDIODE_1
X37 37 52 NDRDIODE_1
X36 6 37 NDRDIODE_0
X35 35 50 NDRDIODE_1
X34 35 34 NDRDIODE_1
X33 50 34 NDRDIODE_0
X32 34 49 NDRDIODE_0
X31 15 34 NDRDIODE_0
X30 32 15 NDRDIODE_0
X29 32 47 NDRDIODE_0
X28 32 14 NDRDIODE_0
X27 32 14 NDRDIODE_0
X22 44 14 NDRDIODE_1
X21 14 29 NDRDIODE_0
X20 29 44 NDRDIODE_1
X19 13 29 NDRDIODE_0
X18 44 13 NDRDIODE_0
X16 13 42 NDRDIODE_0
X15 13 27 NDRDIODE_0
X14 42 27 NDRDIODE_1
X13 27 12 NDRDIODE_0
X12 12 42 NDRDIODE_1
X7 12 24 NDRDIODE_1
X6 12 24 NDRDIODE_1
X5 39 24 NDRDIODE_1
X4 11 24 NDRDIODE_0
X3 11 24 NDRDIODE_1

```

```

.SUBCKT NDRDIODE_1 IN OUT
G1 IN OUT PWL(1) IN OUT -1V,-1n -.8V,-1n -.625V,-760n 0V,0n
.625V,760n .8V,1n 1V,1n SMOOTH=1
.ENDS

```

```

.SUBCKT NDRDIODE_0 IN OUT

```

```
G1 IN OUT PWL(1) IN OUT -2V,-.8n -1V,-.8n -.75V,-.55n -.5V,-.21n
0V,0n .5V,.21n .75V,.55n 1V,.8n 2V,.8n SMOOTH=1
.ENDS
```

```
.END
```

Nand 连接表

```

Moletronics
.OPTIONS RELTOL=.01 ITL4=10 SCALE=1 POST METHOD=Gear DCSTEP=10
.OPTIONS ACCT=0 INGOLD=1 NOMOD PROBE
.TRAN 10000n 100000n

ROut1 1 0 1
.PRINT TRAN I(Rout1)
C1 1 0 10f
VInputB2 2 0 PWL(0n,0V 20000n,0V 20001n,0.5V 30000n,0.5V 30001n,2V
40000n,2V
+ 40001n,0.5V 50000n,0.5V 50001n,0.5V 60000n,0.5V
+ 60001n,0.5V 70000n,0.5V 70001n,2V 80000n,2V
+ 80001n,0.5V 90000n,0.5V 90001n,0.5V 100000n,0.5V )
.PRINT TRAN 2
C2 2 0 10f
C3 3 0 10f
C4 4 0 10f
C5 5 0 10f
C6 6 0 10f
C7 7 0 10f
C8 8 0 10f
C9 9 0 10f
C10 10 0 10f
C11 11 0 10f
C12 12 0 10f
C13 13 0 10f
C14 14 0 10f
VInputA15 15 0 PWL(0n,0V 20000n,0V 20001n,0.5V 30000n,0.5V
30001n,0.5V 40000n,0.5V
+ 40001n,0.5V 50000n,0.5V 50001n,2V 60000n,2V
+ 60001n,0.5V 70000n,0.5V 70001n,2V 80000n,2V
+ 80001n,0.5V 90000n,0.5V 90001n,0.5V 100000n,0.5V )
.PRINT TRAN 15
C15 15 0 10f
C16 16 0 10f
C17 17 0 10f
C18 18 0 10f
C19 19 0 10f
C20 20 0 10f
C24 24 0 5f
C27 27 0 5f
C29 29 0 5f
C32 32 0 5f
C34 34 0 5f
C35 35 0 5f
C37 37 0 5f
C39 39 0 5f
C42 42 0 5f
C44 44 0 5f
C47 47 0 5f
C49 49 0 5f
C50 50 0 5f
C52 52 0 5f
C53 53 0 5f
C54 54 0 5f
C55 55 0 5f
C56 56 0 5f
C57 57 0 5f
C58 58 0 5f

```

C59 59 0 5f
C60 60 0 5f
C61 61 0 5f
C62 62 0 5f
C63 63 0 5f
C64 64 0 5f
C67 67 0 5f
C68 68 0 5f
C69 69 0 5f
C70 70 0 5f
C71 71 0 5f
C72 72 0 5f
C74 74 0 5f
C75 75 0 5f
C76 76 0 5f
C77 77 0 5f
C78 78 0 5f
C79 79 0 5f
C82 82 0 5f
C83 83 0 5f
C84 84 0 5f
C85 85 0 5f
C86 86 0 5f
C87 87 0 5f
C88 88 0 5f
C89 89 0 5f
C90 90 0 5f
C92 92 0 5f
C93 93 0 5f
C94 94 0 5f
C97 97 0 5f
C99 99 0 5f
C101 101 0 5f
C102 102 0 5f
C103 103 0 5f
C104 104 0 5f
C105 105 0 5f
C106 106 0 5f
C108 108 0 5f
C109 109 0 5f
C111 111 0 5f
C112 112 0 5f
C113 113 0 5f
C114 114 0 5f
C115 115 0 5f
C116 116 0 5f
C119 119 0 5f
C120 120 0 5f
C121 121 0 5f
C122 122 0 5f
C123 123 0 5f
C124 124 0 5f
C125 125 0 5f
C128 128 0 5f
C129 129 0 5f
C130 130 0 5f
C131 131 0 5f
C132 132 0 5f
C133 133 0 5f
C134 134 0 5f
C135 135 0 5f

C136 136 0 5f
C137 137 0 5f
C138 138 0 5f
C139 139 0 5f
C142 142 0 5f
C143 143 0 5f
C144 144 0 5f
C145 145 0 5f
C146 146 0 5f
C147 147 0 5f
C149 149 0 5f
C150 150 0 5f
C151 151 0 5f
C152 152 0 5f
C153 153 0 5f
C154 154 0 5f
C157 157 0 5f
C158 158 0 5f
C159 159 0 5f
C160 160 0 5f
C161 161 0 5f
C162 162 0 5f
C163 163 0 5f
C164 164 0 5f
C165 165 0 5f
C166 166 0 5f
C167 167 0 5f
C168 168 0 5f
C169 169 0 5f
C172 172 0 5f
C173 173 0 5f
C175 175 0 5f
C176 176 0 5f
C178 178 0 5f
C179 179 0 5f
C180 180 0 5f
C181 181 0 5f
C182 182 0 5f
C183 183 0 5f
C184 184 0 5f
C186 186 0 5f
C187 187 0 5f
C188 188 0 5f
C190 190 0 5f
C191 191 0 5f
C192 192 0 5f
C193 193 0 5f
C194 194 0 5f
C195 195 0 5f
C196 196 0 5f
C197 197 0 5f
C198 198 0 5f
C199 199 0 5f
C200 200 0 5f
C202 202 0 5f
C203 203 0 5f
C204 204 0 5f
C205 205 0 5f
C206 206 0 5f
C207 207 0 5f
C208 208 0 5f

```
C209 209 0 5f
C210 210 0 5f
C212 212 0 5f
C214 214 0 5f
C217 217 0 5f
C219 219 0 5f
C222 222 0 5f
C224 224 0 5f
C229 229 0 5f
C231 231 0 5f
C232 232 0 5f
C234 234 0 5f
C237 237 0 5f
C239 239 0 5f
C242 242 0 5f
C244 244 0 5f
X512 20 244 NDRDIODE_0
X511 242 20 NDRDIODE_0
X510 19 242 NDRDIODE_0
X508 239 19 NDRDIODE_0
X507 18 239 NDRDIODE_1
X506 237 18 NDRDIODE_1
X505 237 17 NDRDIODE_1
X503 234 17 NDRDIODE_1
X502 234 16 NDRDIODE_0
X501 16 232 NDRDIODE_1
X500 232 231 NDRDIODE_0
X499 5 229 NDRDIODE_0
X498 229 244 NDRDIODE_0
X497 20 229 NDRDIODE_0
X496 20 229 NDRDIODE_0
X490 239 19 NDRDIODE_0
X489 224 19 NDRDIODE_1
X488 224 239 NDRDIODE_1
X487 18 224 NDRDIODE_0
X486 18 239 NDRDIODE_0
X484 18 237 NDRDIODE_1
X483 222 18 NDRDIODE_0
X482 237 222 NDRDIODE_1
X481 222 17 NDRDIODE_0
X480 237 17 NDRDIODE_1
X475 219 17 NDRDIODE_0
X474 219 17 NDRDIODE_1
X473 219 234 NDRDIODE_1
X472 219 16 NDRDIODE_0
X471 16 219 NDRDIODE_0
X469 217 16 NDRDIODE_1
X468 217 16 NDRDIODE_0
X467 217 232 NDRDIODE_0
X466 217 231 NDRDIODE_0
X465 10 231 NDRDIODE_0
X464 10 217 NDRDIODE_0
X462 214 5 NDRDIODE_0
X461 214 5 NDRDIODE_1
X460 214 229 NDRDIODE_1
X459 214 20 NDRDIODE_1
X458 212 20 NDRDIODE_1
X457 212 19 NDRDIODE_1
X456 210 19 NDRDIODE_1
X455 19 210 NDRDIODE_0
X454 210 224 NDRDIODE_0
```

X453 210 209 NDRDIODE_1
X452 209 224 NDRDIODE_0
X451 209 208 NDRDIODE_0
X450 224 208 NDRDIODE_1
X449 18 208 NDRDIODE_0
X448 222 208 NDRDIODE_1
X447 207 208 NDRDIODE_1
X446 222 207 NDRDIODE_1
X445 207 206 NDRDIODE_1
X444 206 222 NDRDIODE_0
X443 206 17 NDRDIODE_0
X442 206 17 NDRDIODE_0
X441 206 205 NDRDIODE_0
X440 205 17 NDRDIODE_0
X439 204 205 NDRDIODE_1
X438 17 204 NDRDIODE_0
X437 204 219 NDRDIODE_0
X436 16 204 NDRDIODE_0
X435 203 204 NDRDIODE_0
X434 203 16 NDRDIODE_0
X433 203 202 NDRDIODE_1
X432 16 202 NDRDIODE_1
X431 217 202 NDRDIODE_0
X430 202 10 NDRDIODE_0
X428 10 202 NDRDIODE_1
X427 200 5 NDRDIODE_0
X426 199 200 NDRDIODE_1
X425 199 5 NDRDIODE_1
X424 199 214 NDRDIODE_1
X423 199 198 NDRDIODE_0
X422 197 198 NDRDIODE_1
X421 212 197 NDRDIODE_0
X420 196 197 NDRDIODE_0
X419 195 196 NDRDIODE_1
X418 195 210 NDRDIODE_1
X417 195 209 NDRDIODE_1
X416 194 195 NDRDIODE_1
X415 194 209 NDRDIODE_0
X414 194 193 NDRDIODE_0
X413 193 209 NDRDIODE_0
X412 193 208 NDRDIODE_1
X411 207 193 NDRDIODE_0
X410 193 192 NDRDIODE_1
X409 207 192 NDRDIODE_0
X408 191 192 NDRDIODE_0
X407 207 191 NDRDIODE_0
X406 191 206 NDRDIODE_1
X405 205 191 NDRDIODE_1
X404 190 191 NDRDIODE_1
X403 205 190 NDRDIODE_1
X402 188 203 NDRDIODE_1
X401 188 187 NDRDIODE_1
X400 187 203 NDRDIODE_1
X399 202 187 NDRDIODE_0
X398 10 187 NDRDIODE_0
X397 186 10 NDRDIODE_0
X396 187 186 NDRDIODE_1
X395 200 4 NDRDIODE_0
X394 4 184 NDRDIODE_1
X393 184 200 NDRDIODE_0
X392 184 199 NDRDIODE_1

X391 184 198 NDRDIODE_1
X390 183 184 NDRDIODE_1
X389 198 183 NDRDIODE_0
X388 183 182 NDRDIODE_0
X387 182 198 NDRDIODE_0
X386 197 182 NDRDIODE_0
X385 182 196 NDRDIODE_1
X384 182 181 NDRDIODE_0
X383 196 181 NDRDIODE_1
X382 181 180 NDRDIODE_1
X381 196 180 NDRDIODE_1
X380 195 180 NDRDIODE_0
X379 194 180 NDRDIODE_0
X378 179 180 NDRDIODE_0
X377 179 194 NDRDIODE_1
X376 179 178 NDRDIODE_1
X375 178 194 NDRDIODE_1
X374 178 193 NDRDIODE_1
X373 192 178 NDRDIODE_0
X372 192 176 NDRDIODE_0
X371 191 176 NDRDIODE_1
X370 190 176 NDRDIODE_1
X369 175 176 NDRDIODE_0
X368 190 175 NDRDIODE_0
X367 188 173 NDRDIODE_1
X366 173 172 NDRDIODE_1
X365 188 172 NDRDIODE_0
X364 172 187 NDRDIODE_0
X363 172 186 NDRDIODE_1
X362 186 9 NDRDIODE_0
X361 9 172 NDRDIODE_1
X359 169 4 NDRDIODE_0
X358 169 4 NDRDIODE_0
X357 184 169 NDRDIODE_1
X356 183 169 NDRDIODE_1
X355 168 169 NDRDIODE_1
X354 183 168 NDRDIODE_1
X353 167 168 NDRDIODE_1
X352 167 183 NDRDIODE_1
X351 167 182 NDRDIODE_1
X350 181 167 NDRDIODE_0
X349 167 166 NDRDIODE_0
X348 181 166 NDRDIODE_0
X347 165 166 NDRDIODE_1
X346 181 165 NDRDIODE_1
X345 165 180 NDRDIODE_0
X344 179 165 NDRDIODE_1
X343 165 164 NDRDIODE_0
X342 179 164 NDRDIODE_0
X341 164 163 NDRDIODE_0
X340 179 163 NDRDIODE_1
X339 178 163 NDRDIODE_1
X338 162 163 NDRDIODE_0
X337 162 161 NDRDIODE_0
X336 161 176 NDRDIODE_1
X335 161 175 NDRDIODE_0
X334 161 160 NDRDIODE_0
X333 175 160 NDRDIODE_0
X332 160 159 NDRDIODE_0
X331 159 175 NDRDIODE_0
X330 173 159 NDRDIODE_1

X329 158 159 NDRDIODE_0
X328 158 173 NDRDIODE_0
X327 158 157 NDRDIODE_0
X326 157 173 NDRDIODE_0
X325 157 172 NDRDIODE_1
X324 157 9 NDRDIODE_0
X322 157 9 NDRDIODE_1
X320 3 154 NDRDIODE_0
X319 154 4 NDRDIODE_0
X318 154 169 NDRDIODE_1
X317 168 154 NDRDIODE_1
X316 153 154 NDRDIODE_0
X315 168 153 NDRDIODE_0
X314 152 153 NDRDIODE_0
X313 152 168 NDRDIODE_1
X312 152 167 NDRDIODE_0
X311 152 166 NDRDIODE_1
X310 152 151 NDRDIODE_0
X309 151 166 NDRDIODE_1
X308 150 151 NDRDIODE_0
X307 166 150 NDRDIODE_1
X306 165 150 NDRDIODE_0
X305 164 150 NDRDIODE_1
X304 150 149 NDRDIODE_1
X303 149 164 NDRDIODE_1
X302 147 162 NDRDIODE_1
X301 147 146 NDRDIODE_1
X300 162 146 NDRDIODE_0
X299 146 161 NDRDIODE_0
X298 146 160 NDRDIODE_0
X297 145 146 NDRDIODE_1
X296 145 160 NDRDIODE_0
X295 144 145 NDRDIODE_0
X294 160 144 NDRDIODE_1
X293 144 159 NDRDIODE_0
X292 144 158 NDRDIODE_1
X291 143 144 NDRDIODE_0
X290 158 143 NDRDIODE_0
X289 143 142 NDRDIODE_0
X288 158 142 NDRDIODE_1
X287 157 142 NDRDIODE_0
X286 9 142 NDRDIODE_0
X284 8 142 NDRDIODE_1
X282 139 3 NDRDIODE_1
X281 3 139 NDRDIODE_1
X280 139 154 NDRDIODE_0
X279 139 153 NDRDIODE_1
X278 139 138 NDRDIODE_1
X277 138 153 NDRDIODE_1
X276 137 138 NDRDIODE_1
X275 137 153 NDRDIODE_1
X274 152 137 NDRDIODE_0
X273 151 137 NDRDIODE_0
X272 136 137 NDRDIODE_1
X271 136 151 NDRDIODE_1
X270 136 135 NDRDIODE_0
X269 151 135 NDRDIODE_1
X268 150 135 NDRDIODE_1
X267 149 135 NDRDIODE_1
X266 135 134 NDRDIODE_0
X265 134 149 NDRDIODE_1

X264 134 133 NDRDIODE_0
X263 133 149 NDRDIODE_0
X262 133 147 NDRDIODE_0
X261 132 133 NDRDIODE_1
X260 147 132 NDRDIODE_1
X259 131 132 NDRDIODE_1
X258 147 131 NDRDIODE_0
X257 146 131 NDRDIODE_1
X256 131 145 NDRDIODE_0
X255 131 130 NDRDIODE_0
X254 145 130 NDRDIODE_0
X253 130 129 NDRDIODE_1
X252 129 145 NDRDIODE_0
X251 129 144 NDRDIODE_1
X250 143 129 NDRDIODE_0
X249 128 129 NDRDIODE_1
X248 143 128 NDRDIODE_1
X246 3 125 NDRDIODE_1
X245 124 125 NDRDIODE_1
X244 3 124 NDRDIODE_1
X243 124 139 NDRDIODE_1
X242 138 124 NDRDIODE_0
X241 124 123 NDRDIODE_0
X240 138 123 NDRDIODE_0
X239 122 123 NDRDIODE_1
X238 122 138 NDRDIODE_1
X237 122 137 NDRDIODE_0
X236 136 122 NDRDIODE_1
X235 121 122 NDRDIODE_1
X234 121 136 NDRDIODE_1
X233 121 120 NDRDIODE_0
X232 136 120 NDRDIODE_1
X231 120 135 NDRDIODE_1
X230 120 134 NDRDIODE_0
X229 120 119 NDRDIODE_0
X228 134 119 NDRDIODE_0
X227 115 116 NDRDIODE_1
X226 115 130 NDRDIODE_0
X225 114 115 NDRDIODE_0
X224 114 130 NDRDIODE_0
X223 129 114 NDRDIODE_0
X222 114 128 NDRDIODE_1
X221 113 114 NDRDIODE_1
X220 113 128 NDRDIODE_0
X219 112 113 NDRDIODE_0
X218 112 128 NDRDIODE_0
X217 112 8 NDRDIODE_1
X216 111 8 NDRDIODE_1
X215 112 111 NDRDIODE_0
X214 2 125 NDRDIODE_1
X213 2 109 NDRDIODE_0
X212 125 109 NDRDIODE_1
X211 109 124 NDRDIODE_0
X210 109 123 NDRDIODE_1
X209 108 109 NDRDIODE_1
X208 123 108 NDRDIODE_0
X207 121 106 NDRDIODE_0
X206 106 105 NDRDIODE_1
X205 121 105 NDRDIODE_1
X204 120 105 NDRDIODE_0
X203 119 105 NDRDIODE_0

X202 105 104 NDRDIODE_0
X201 119 104 NDRDIODE_0
X200 104 103 NDRDIODE_1
X199 103 119 NDRDIODE_1
X198 102 103 NDRDIODE_1
X197 101 102 NDRDIODE_0
X196 101 116 NDRDIODE_1
X195 115 101 NDRDIODE_0
X194 99 115 NDRDIODE_1
X193 114 99 NDRDIODE_1
X192 113 99 NDRDIODE_0
X191 97 113 NDRDIODE_1
X190 112 97 NDRDIODE_1
X189 111 97 NDRDIODE_0
X188 111 7 NDRDIODE_1
X187 97 7 NDRDIODE_1
X185 2 94 NDRDIODE_1
X184 2 94 NDRDIODE_0
X183 94 109 NDRDIODE_1
X182 94 108 NDRDIODE_1
X181 93 94 NDRDIODE_0
X180 93 108 NDRDIODE_1
X179 93 92 NDRDIODE_0
X178 108 92 NDRDIODE_1
X177 106 92 NDRDIODE_0
X176 106 90 NDRDIODE_0
X175 105 90 NDRDIODE_1
X174 90 104 NDRDIODE_0
X173 89 90 NDRDIODE_0
X172 104 89 NDRDIODE_0
X171 89 88 NDRDIODE_0
X170 88 104 NDRDIODE_0
X169 103 88 NDRDIODE_0
X168 88 102 NDRDIODE_0
X167 87 88 NDRDIODE_0
X166 87 102 NDRDIODE_0
X165 86 87 NDRDIODE_0
X164 86 102 NDRDIODE_0
X163 101 86 NDRDIODE_1
X162 86 85 NDRDIODE_0
X161 84 85 NDRDIODE_1
X160 84 99 NDRDIODE_0
X159 84 83 NDRDIODE_1
X158 83 82 NDRDIODE_1
X157 82 97 NDRDIODE_1
X156 82 7 NDRDIODE_0
X154 7 82 NDRDIODE_0
X152 1 79 NDRDIODE_1
X151 2 79 NDRDIODE_1
X150 94 79 NDRDIODE_1
X149 93 79 NDRDIODE_0
X148 78 79 NDRDIODE_0
X147 78 93 NDRDIODE_1
X146 78 77 NDRDIODE_0
X145 77 93 NDRDIODE_0
X144 92 77 NDRDIODE_1
X143 77 76 NDRDIODE_0
X142 76 75 NDRDIODE_0
X141 90 75 NDRDIODE_0
X140 75 89 NDRDIODE_0
X139 74 75 NDRDIODE_0

X138 74 89 NDRDIODE_0
X137 87 72 NDRDIODE_1
X136 72 71 NDRDIODE_1
X135 87 71 NDRDIODE_0
X134 71 86 NDRDIODE_1
X133 71 85 NDRDIODE_0
X132 70 71 NDRDIODE_0
X131 85 70 NDRDIODE_1
X130 70 69 NDRDIODE_1
X129 85 69 NDRDIODE_0
X128 69 84 NDRDIODE_0
X127 69 83 NDRDIODE_1
X126 68 69 NDRDIODE_0
X125 83 68 NDRDIODE_1
X124 68 67 NDRDIODE_0
X123 83 67 NDRDIODE_0
X122 67 82 NDRDIODE_0
X121 7 67 NDRDIODE_0
X119 6 67 NDRDIODE_1
X117 64 1 NDRDIODE_1
X116 64 1 NDRDIODE_1
X115 79 64 NDRDIODE_1
X114 64 78 NDRDIODE_1
X113 63 64 NDRDIODE_1
X112 78 63 NDRDIODE_0
X111 63 62 NDRDIODE_0
X110 78 62 NDRDIODE_1
X109 62 77 NDRDIODE_1
X108 62 76 NDRDIODE_0
X107 62 61 NDRDIODE_0
X106 76 61 NDRDIODE_1
X105 60 61 NDRDIODE_0
X104 76 60 NDRDIODE_1
X103 60 75 NDRDIODE_0
X102 74 60 NDRDIODE_0
X101 59 60 NDRDIODE_0
X100 59 74 NDRDIODE_0
X99 59 58 NDRDIODE_0
X98 58 74 NDRDIODE_1
X97 72 58 NDRDIODE_1
X96 57 58 NDRDIODE_1
X95 72 57 NDRDIODE_1
X94 56 57 NDRDIODE_0
X93 72 56 NDRDIODE_1
X92 56 71 NDRDIODE_0
X91 56 70 NDRDIODE_1
X90 56 55 NDRDIODE_1
X89 55 70 NDRDIODE_1
X88 54 55 NDRDIODE_1
X87 70 54 NDRDIODE_0
X86 69 54 NDRDIODE_0
X85 54 68 NDRDIODE_1
X84 54 53 NDRDIODE_0
X83 68 53 NDRDIODE_1
X82 52 53 NDRDIODE_1
X81 68 52 NDRDIODE_0
X80 67 52 NDRDIODE_1
X79 52 6 NDRDIODE_1
X77 6 52 NDRDIODE_0
X76 1 50 NDRDIODE_1
X75 50 49 NDRDIODE_1

X74 49 1 NDRDIODE_0
X73 64 49 NDRDIODE_0
X72 63 49 NDRDIODE_1
X71 15 49 NDRDIODE_0
X70 15 63 NDRDIODE_1
X69 15 47 NDRDIODE_0
X68 63 47 NDRDIODE_0
X67 62 47 NDRDIODE_1
X66 47 61 NDRDIODE_0
X65 47 14 NDRDIODE_0
X64 61 14 NDRDIODE_0
X62 14 61 NDRDIODE_0
X61 60 14 NDRDIODE_0
X60 14 59 NDRDIODE_1
X59 14 44 NDRDIODE_1
X58 44 59 NDRDIODE_1
X57 13 44 NDRDIODE_1
X56 13 59 NDRDIODE_0
X55 13 58 NDRDIODE_1
X54 13 57 NDRDIODE_0
X53 13 42 NDRDIODE_1
X52 57 42 NDRDIODE_1
X51 12 42 NDRDIODE_0
X50 57 12 NDRDIODE_1
X49 12 56 NDRDIODE_1
X48 55 12 NDRDIODE_1
X46 12 55 NDRDIODE_1
X45 12 39 NDRDIODE_1
X44 55 39 NDRDIODE_0
X43 39 54 NDRDIODE_1
X42 53 39 NDRDIODE_1
X41 11 39 NDRDIODE_1
X40 11 53 NDRDIODE_1
X39 37 11 NDRDIODE_1
X38 53 37 NDRDIODE_0
X37 37 52 NDRDIODE_0
X36 6 37 NDRDIODE_1
X35 35 50 NDRDIODE_1
X34 35 34 NDRDIODE_0
X33 50 34 NDRDIODE_1
X32 34 49 NDRDIODE_0
X31 15 34 NDRDIODE_1
X30 32 15 NDRDIODE_1
X29 32 47 NDRDIODE_0
X28 32 14 NDRDIODE_1
X27 32 14 NDRDIODE_1
X22 44 14 NDRDIODE_0
X21 14 29 NDRDIODE_1
X20 29 44 NDRDIODE_0
X19 13 29 NDRDIODE_0
X18 44 13 NDRDIODE_0
X16 13 42 NDRDIODE_0
X15 13 27 NDRDIODE_0
X14 42 27 NDRDIODE_1
X13 27 12 NDRDIODE_1
X12 12 42 NDRDIODE_0
X7 12 24 NDRDIODE_0
X6 12 24 NDRDIODE_0
X5 39 24 NDRDIODE_0
X4 11 24 NDRDIODE_1
X3 11 24 NDRDIODE_0

```
.SUBCKT NDRDIODE_1 IN OUT
G1 IN OUT PWL(1) IN OUT -1V,-1n -.8V,-1n -.625V,-760n 0V,0n
.625V,760n .8V,1n 1V,1n SMOOTH=1
.ENDS
.SUBCKT NDRDIODE_0 IN OUT
G1 IN OUT PWL(1) IN OUT -2V,-.8n -1V,-.8n -.75V,-.55n -.5V,-.21n
0V,0n .5V,.21n .75V,.55n 1V,.8n 2V,.8n SMOOTH=1
.ENDS

.END
```

反相半加法器连接表

```

Moletronics
.OPTIONS RELTOL=.01 ITL4=10 SCALE=1 POST METHOD=Gear DCSTEP=10
.OPTIONS ACCT=0 INGOLD=1 NOMOD PROBE
.TRAN 10000n 100000n

Rout1 1 0 1
.PRINT TRAN I(Rout1)
C1 1 0 10f
VInputB2 2 0 PWL(0n,0V 20000n,0V 20001n,0.5V 30000n,0.5V 30001n,2V
40000n,2V
+ 40001n,0.5V 50000n,0.5V 50001n,0.5V 60000n,0.5V
+ 60001n,0.5V 70000n,0.5V 70001n,2V 80000n,2V
+ 80001n,0.5V 90000n,0.5V 90001n,0.5V 100000n,0.5V )
.PRINT TRAN 2
C2 2 0 10f
C3 3 0 10f
C4 4 0 10f
C5 5 0 10f
VInputB6 6 0 PWL(0n,0V 20000n,0V 20001n,0.5V 30000n,0.5V 30001n,2V
40000n,2V
+ 40001n,0.5V 50000n,0.5V 50001n,0.5V 60000n,0.5V
+ 60001n,0.5V 70000n,0.5V 70001n,2V 80000n,2V
+ 80001n,0.5V 90000n,0.5V 90001n,0.5V 100000n,0.5V )
.PRINT TRAN 6
C6 6 0 10f
C7 7 0 10f
Rout2 8 0 1
.PRINT TRAN I(Rout2)
C8 8 0 10f
C9 9 0 10f
VInputA10 10 0 PWL(0n,0V 20000n,0V 20001n,0.5V 30000n,0.5V
30001n,0.5V 40000n,0.5V
+ 40001n,0.5V 50000n,0.5V 50001n,2V 60000n,2V
+ 60001n,0.5V 70000n,0.5V 70001n,2V 80000n,2V
+ 80001n,0.5V 90000n,0.5V 90001n,0.5V 100000n,0.5V )
.PRINT TRAN 10
C10 10 0 10f
VDInputA11 11 0 PWL(0n,0V 20000n,0V 20001n,2V 30000n,2V 30001n,2V
40000n,2V 40001n,2V 50000n,2V 50001n,0.5V 60000n,0.5V 60001n,2V
70000n,2V 70001n,0.5V 80000n,0.5V 80001n,2V 90000n,2V 90001n,2V
100000n,2V )
.PRINT TRAN 11
C11 11 0 10f
C12 12 0 10f
C13 13 0 10f
C14 14 0 10f
VInputA15 15 0 PWL(0n,0V 20000n,0V 20001n,0.5V 30000n,0.5V
30001n,0.5V 40000n,0.5V
+ 40001n,0.5V 50000n,0.5V 50001n,2V 60000n,2V
+ 60001n,0.5V 70000n,0.5V 70001n,2V 80000n,2V
+ 80001n,0.5V 90000n,0.5V 90001n,0.5V 100000n,0.5V )
.PRINT TRAN 15
C15 15 0 10f
VDInputB16 16 0 PWL(0n,0V 20000n,0V 20001n,2V 30000n,2V 30001n,0.5V
40000n,0.5V 40001n,2V 50000n,2V 50001n,2V 60000n,2V 60001n,2V
70000n,2V 70001n,0.5V 80000n,0.5V 80001n,2V 90000n,2V 90001n,2V
100000n,2V )
.PRINT TRAN 16
C16 16 0 10f

```

C17 17 0 10f
C18 18 0 10f
C19 19 0 10f
C20 20 0 10f
C24 24 0 5f
C27 27 0 5f
C29 29 0 5f
C32 32 0 5f
C34 34 0 5f
C35 35 0 5f
C37 37 0 5f
C39 39 0 5f
C42 42 0 5f
C44 44 0 5f
C47 47 0 5f
C49 49 0 5f
C50 50 0 5f
C52 52 0 5f
C53 53 0 5f
C54 54 0 5f
C55 55 0 5f
C56 56 0 5f
C57 57 0 5f
C58 58 0 5f
C59 59 0 5f
C60 60 0 5f
C61 61 0 5f
C62 62 0 5f
C63 63 0 5f
C64 64 0 5f
C67 67 0 5f
C68 68 0 5f
C69 69 0 5f
C70 70 0 5f
C71 71 0 5f
C72 72 0 5f
C74 74 0 5f
C75 75 0 5f
C76 76 0 5f
C77 77 0 5f
C78 78 0 5f
C79 79 0 5f
C82 82 0 5f
C83 83 0 5f
C84 84 0 5f
C85 85 0 5f
C86 86 0 5f
C87 87 0 5f
C88 88 0 5f
C89 89 0 5f
C90 90 0 5f
C92 92 0 5f
C93 93 0 5f
C94 94 0 5f
C97 97 0 5f
C99 99 0 5f
C101 101 0 5f
C102 102 0 5f
C103 103 0 5f
C104 104 0 5f
C105 105 0 5f

C106 106 0 5f
C108 108 0 5f
C109 109 0 5f
C111 111 0 5f
C112 112 0 5f
C113 113 0 5f
C114 114 0 5f
C115 115 0 5f
C116 116 0 5f
C119 119 0 5f
C120 120 0 5f
C121 121 0 5f
C122 122 0 5f
C123 123 0 5f
C124 124 0 5f
C125 125 0 5f
C128 128 0 5f
C129 129 0 5f
C130 130 0 5f
C131 131 0 5f
C132 132 0 5f
C133 133 0 5f
C134 134 0 5f
C135 135 0 5f
C136 136 0 5f
C137 137 0 5f
C138 138 0 5f
C139 139 0 5f
C142 142 0 5f
C143 143 0 5f
C144 144 0 5f
C145 145 0 5f
C146 146 0 5f
C147 147 0 5f
C149 149 0 5f
C150 150 0 5f
C151 151 0 5f
C152 152 0 5f
C153 153 0 5f
C154 154 0 5f
C157 157 0 5f
C158 158 0 5f
C159 159 0 5f
C160 160 0 5f
C161 161 0 5f
C162 162 0 5f
C163 163 0 5f
C164 164 0 5f
C165 165 0 5f
C166 166 0 5f
C167 167 0 5f
C168 168 0 5f
C169 169 0 5f
C172 172 0 5f
C173 173 0 5f
C175 175 0 5f
C176 176 0 5f
C178 178 0 5f
C179 179 0 5f
C180 180 0 5f
C181 181 0 5f

C182 182 0 5f
C183 183 0 5f
C184 184 0 5f
C186 186 0 5f
C187 187 0 5f
C188 188 0 5f
C190 190 0 5f
C191 191 0 5f
C192 192 0 5f
C193 193 0 5f
C194 194 0 5f
C195 195 0 5f
C196 196 0 5f
C197 197 0 5f
C198 198 0 5f
C199 199 0 5f
C200 200 0 5f
C202 202 0 5f
C203 203 0 5f
C204 204 0 5f
C205 205 0 5f
C206 206 0 5f
C207 207 0 5f
C208 208 0 5f
C209 209 0 5f
C210 210 0 5f
C212 212 0 5f
C214 214 0 5f
C217 217 0 5f
C219 219 0 5f
C222 222 0 5f
C224 224 0 5f
C229 229 0 5f
C231 231 0 5f
C232 232 0 5f
C234 234 0 5f
C237 237 0 5f
C239 239 0 5f
C242 242 0 5f
C244 244 0 5f
X512 20 244 NDRDIODE_0
X511 242 20 NDRDIODE_0
X510 19 242 NDRDIODE_1
X508 239 19 NDRDIODE_0
X507 18 239 NDRDIODE_0
X506 237 18 NDRDIODE_0
X505 237 17 NDRDIODE_1
X503 234 17 NDRDIODE_0
X502 234 16 NDRDIODE_0
X501 16 232 NDRDIODE_1
X500 232 231 NDRDIODE_1
X499 5 229 NDRDIODE_0
X498 229 244 NDRDIODE_0
X497 20 229 NDRDIODE_0
X496 20 229 NDRDIODE_0
X490 239 19 NDRDIODE_0
X489 224 19 NDRDIODE_1
X488 224 239 NDRDIODE_0
X487 18 224 NDRDIODE_0
X486 18 239 NDRDIODE_0
X484 18 237 NDRDIODE_1

X483 222 18 NDRDIODE_1
X482 237 222 NDRDIODE_0
X481 222 17 NDRDIODE_0
X480 237 17 NDRDIODE_0
X475 219 17 NDRDIODE_0
X474 219 17 NDRDIODE_1
X473 219 234 NDRDIODE_0
X472 219 16 NDRDIODE_0
X471 16 219 NDRDIODE_1
X469 217 16 NDRDIODE_0
X468 217 16 NDRDIODE_0
X467 217 232 NDRDIODE_0
X466 217 231 NDRDIODE_0
X465 10 231 NDRDIODE_0
X464 10 217 NDRDIODE_0
X462 214 5 NDRDIODE_1
X461 214 5 NDRDIODE_0
X460 214 229 NDRDIODE_0
X459 214 20 NDRDIODE_0
X458 212 20 NDRDIODE_0
X457 212 19 NDRDIODE_1
X456 210 19 NDRDIODE_1
X455 19 210 NDRDIODE_1
X454 210 224 NDRDIODE_1
X453 210 209 NDRDIODE_1
X452 209 224 NDRDIODE_1
X451 209 208 NDRDIODE_1
X450 224 208 NDRDIODE_1
X449 18 208 NDRDIODE_0
X448 222 208 NDRDIODE_0
X447 207 208 NDRDIODE_0
X446 222 207 NDRDIODE_0
X445 207 206 NDRDIODE_0
X444 206 222 NDRDIODE_0
X443 206 17 NDRDIODE_1
X442 206 17 NDRDIODE_1
X441 206 205 NDRDIODE_1
X440 205 17 NDRDIODE_0
X439 204 205 NDRDIODE_1
X438 17 204 NDRDIODE_1
X437 204 219 NDRDIODE_1
X436 16 204 NDRDIODE_0
X435 203 204 NDRDIODE_0
X434 203 16 NDRDIODE_1
X433 203 202 NDRDIODE_1
X432 16 202 NDRDIODE_1
X431 217 202 NDRDIODE_1
X430 202 10 NDRDIODE_1
X428 10 202 NDRDIODE_1
X427 200 5 NDRDIODE_1
X426 199 200 NDRDIODE_0
X425 199 5 NDRDIODE_1
X424 199 214 NDRDIODE_0
X423 199 198 NDRDIODE_1
X422 197 198 NDRDIODE_1
X421 212 197 NDRDIODE_1
X420 196 197 NDRDIODE_0
X419 195 196 NDRDIODE_0
X418 195 210 NDRDIODE_0
X417 195 209 NDRDIODE_0
X416 194 195 NDRDIODE_0

X415 194 209 NDRDIODE_1
X414 194 193 NDRDIODE_0
X413 193 209 NDRDIODE_0
X412 193 208 NDRDIODE_0
X411 207 193 NDRDIODE_0
X410 193 192 NDRDIODE_0
X409 207 192 NDRDIODE_1
X408 191 192 NDRDIODE_0
X407 207 191 NDRDIODE_0
X406 191 206 NDRDIODE_0
X405 205 191 NDRDIODE_0
X404 190 191 NDRDIODE_0
X403 205 190 NDRDIODE_1
X402 188 203 NDRDIODE_0
X401 188 187 NDRDIODE_1
X400 187 203 NDRDIODE_0
X399 202 187 NDRDIODE_1
X398 10 187 NDRDIODE_0
X397 186 10 NDRDIODE_1
X396 187 186 NDRDIODE_1
X395 200 4 NDRDIODE_1
X394 4 184 NDRDIODE_1
X393 184 200 NDRDIODE_1
X392 184 199 NDRDIODE_0
X391 184 198 NDRDIODE_1
X390 183 184 NDRDIODE_1
X389 198 183 NDRDIODE_1
X388 183 182 NDRDIODE_1
X387 182 198 NDRDIODE_0
X386 197 182 NDRDIODE_0
X385 182 196 NDRDIODE_1
X384 182 181 NDRDIODE_0
X383 196 181 NDRDIODE_0
X382 181 180 NDRDIODE_1
X381 196 180 NDRDIODE_0
X380 195 180 NDRDIODE_0
X379 194 180 NDRDIODE_1
X378 179 180 NDRDIODE_0
X377 179 194 NDRDIODE_1
X376 179 178 NDRDIODE_0
X375 178 194 NDRDIODE_1
X374 178 193 NDRDIODE_1
X373 192 178 NDRDIODE_0
X372 192 176 NDRDIODE_0
X371 191 176 NDRDIODE_1
X370 190 176 NDRDIODE_0
X369 175 176 NDRDIODE_0
X368 190 175 NDRDIODE_0
X367 188 173 NDRDIODE_1
X366 173 172 NDRDIODE_0
X365 188 172 NDRDIODE_1
X364 172 187 NDRDIODE_1
X363 172 186 NDRDIODE_0
X362 186 9 NDRDIODE_0
X361 9 172 NDRDIODE_0
X359 169 4 NDRDIODE_1
X358 169 4 NDRDIODE_0
X357 184 169 NDRDIODE_0
X356 183 169 NDRDIODE_1
X355 168 169 NDRDIODE_0
X354 183 168 NDRDIODE_0

X353 167 168 NDRDIODE_1
X352 167 183 NDRDIODE_1
X351 167 182 NDRDIODE_0
X350 181 167 NDRDIODE_0
X349 167 166 NDRDIODE_0
X348 181 166 NDRDIODE_1
X347 165 166 NDRDIODE_1
X346 181 165 NDRDIODE_1
X345 165 180 NDRDIODE_0
X344 179 165 NDRDIODE_0
X343 165 164 NDRDIODE_0
X342 179 164 NDRDIODE_0
X341 164 163 NDRDIODE_1
X340 179 163 NDRDIODE_0
X339 178 163 NDRDIODE_0
X338 162 163 NDRDIODE_0
X337 162 161 NDRDIODE_1
X336 161 176 NDRDIODE_1
X335 161 175 NDRDIODE_0
X334 161 160 NDRDIODE_1
X333 175 160 NDRDIODE_1
X332 160 159 NDRDIODE_1
X331 159 175 NDRDIODE_1
X330 173 159 NDRDIODE_0
X329 158 159 NDRDIODE_0
X328 158 173 NDRDIODE_1
X327 158 157 NDRDIODE_1
X326 157 173 NDRDIODE_1
X325 157 172 NDRDIODE_1
X324 157 9 NDRDIODE_0
X322 157 9 NDRDIODE_0
X320 3 154 NDRDIODE_1
X319 154 4 NDRDIODE_0
X318 154 169 NDRDIODE_0
X317 168 154 NDRDIODE_0
X316 153 154 NDRDIODE_0
X315 168 153 NDRDIODE_0
X314 152 153 NDRDIODE_1
X313 152 168 NDRDIODE_1
X312 152 167 NDRDIODE_1
X311 152 166 NDRDIODE_0
X310 152 151 NDRDIODE_1
X309 151 166 NDRDIODE_1
X308 150 151 NDRDIODE_0
X307 166 150 NDRDIODE_1
X306 165 150 NDRDIODE_1
X305 164 150 NDRDIODE_1
X304 150 149 NDRDIODE_0
X303 149 164 NDRDIODE_1
X302 147 162 NDRDIODE_0
X301 147 146 NDRDIODE_0
X300 162 146 NDRDIODE_0
X299 146 161 NDRDIODE_1
X298 146 160 NDRDIODE_0
X297 145 146 NDRDIODE_0
X296 145 160 NDRDIODE_0
X295 144 145 NDRDIODE_0
X294 160 144 NDRDIODE_0
X293 144 159 NDRDIODE_1
X292 144 158 NDRDIODE_1
X291 143 144 NDRDIODE_1

X290 158 143 NDRDIODE_0
X289 143 142 NDRDIODE_0
X288 158 142 NDRDIODE_1
X287 157 142 NDRDIODE_1
X286 9 142 NDRDIODE_1
X284 8 142 NDRDIODE_1
X282 139 3 NDRDIODE_1
X281 3 139 NDRDIODE_1
X280 139 154 NDRDIODE_1
X279 139 153 NDRDIODE_0
X278 139 138 NDRDIODE_1
X277 138 153 NDRDIODE_0
X276 137 138 NDRDIODE_1
X275 137 153 NDRDIODE_0
X274 152 137 NDRDIODE_0
X273 151 137 NDRDIODE_0
X272 136 137 NDRDIODE_1
X271 136 151 NDRDIODE_0
X270 136 135 NDRDIODE_1
X269 151 135 NDRDIODE_1
X268 150 135 NDRDIODE_0
X267 149 135 NDRDIODE_0
X266 135 134 NDRDIODE_0
X265 134 149 NDRDIODE_1
X264 134 133 NDRDIODE_1
X263 133 149 NDRDIODE_1
X262 133 147 NDRDIODE_0
X261 132 133 NDRDIODE_1
X260 147 132 NDRDIODE_0
X259 131 132 NDRDIODE_0
X258 147 131 NDRDIODE_1
X257 146 131 NDRDIODE_0
X256 131 145 NDRDIODE_0
X255 131 130 NDRDIODE_1
X254 145 130 NDRDIODE_0
X253 130 129 NDRDIODE_1
X252 129 145 NDRDIODE_0
X251 129 144 NDRDIODE_0
X250 143 129 NDRDIODE_0
X249 128 129 NDRDIODE_1
X248 143 128 NDRDIODE_0
X246 3 125 NDRDIODE_1
X245 124 125 NDRDIODE_1
X244 3 124 NDRDIODE_0
X243 124 139 NDRDIODE_0
X242 138 124 NDRDIODE_0
X241 124 123 NDRDIODE_1
X240 138 123 NDRDIODE_1
X239 122 123 NDRDIODE_0
X238 122 138 NDRDIODE_0
X237 122 137 NDRDIODE_0
X236 136 122 NDRDIODE_0
X235 121 122 NDRDIODE_0
X234 121 136 NDRDIODE_1
X233 121 120 NDRDIODE_0
X232 136 120 NDRDIODE_0
X231 120 135 NDRDIODE_1
X230 120 134 NDRDIODE_1
X229 120 119 NDRDIODE_0
X228 134 119 NDRDIODE_0
X227 115 116 NDRDIODE_0

X226 115 130 NDRDIODE_1
X225 114 115 NDRDIODE_1
X224 114 130 NDRDIODE_0
X223 129 114 NDRDIODE_0
X222 114 128 NDRDIODE_1
X221 113 114 NDRDIODE_1
X220 113 128 NDRDIODE_1
X219 112 113 NDRDIODE_0
X218 112 128 NDRDIODE_1
X217 112 8 NDRDIODE_1
X216 111 8 NDRDIODE_1
X215 112 111 NDRDIODE_0
X214 2 125 NDRDIODE_0
X213 2 109 NDRDIODE_1
X212 125 109 NDRDIODE_1
X211 109 124 NDRDIODE_1
X210 109 123 NDRDIODE_1
X209 108 109 NDRDIODE_0
X208 123 108 NDRDIODE_1
X207 121 106 NDRDIODE_1
X206 106 105 NDRDIODE_1
X205 121 105 NDRDIODE_0
X204 120 105 NDRDIODE_0
X203 119 105 NDRDIODE_1
X202 105 104 NDRDIODE_1
X201 119 104 NDRDIODE_1
X200 104 103 NDRDIODE_0
X199 103 119 NDRDIODE_1
X198 102 103 NDRDIODE_0
X197 101 102 NDRDIODE_0
X196 101 116 NDRDIODE_0
X195 115 101 NDRDIODE_1
X194 99 115 NDRDIODE_0
X193 114 99 NDRDIODE_0
X192 113 99 NDRDIODE_1
X191 97 113 NDRDIODE_0
X190 112 97 NDRDIODE_1
X189 111 97 NDRDIODE_0
X188 111 7 NDRDIODE_0
X187 97 7 NDRDIODE_0
X185 2 94 NDRDIODE_1
X184 2 94 NDRDIODE_0
X183 94 109 NDRDIODE_1
X182 94 108 NDRDIODE_1
X181 93 94 NDRDIODE_1
X180 93 108 NDRDIODE_1
X179 93 92 NDRDIODE_1
X178 108 92 NDRDIODE_0
X177 106 92 NDRDIODE_1
X176 106 90 NDRDIODE_1
X175 105 90 NDRDIODE_0
X174 90 104 NDRDIODE_0
X173 89 90 NDRDIODE_1
X172 104 89 NDRDIODE_0
X171 89 88 NDRDIODE_0
X170 88 104 NDRDIODE_0
X169 103 88 NDRDIODE_0
X168 88 102 NDRDIODE_0
X167 87 88 NDRDIODE_1
X166 87 102 NDRDIODE_1
X165 86 87 NDRDIODE_0

X164 86 102 NDRDIODE_1
X163 101 86 NDRDIODE_1
X162 86 85 NDRDIODE_1
X161 84 85 NDRDIODE_0
X160 84 99 NDRDIODE_0
X159 84 83 NDRDIODE_0
X158 83 82 NDRDIODE_1
X157 82 97 NDRDIODE_1
X156 82 7 NDRDIODE_1
X154 7 82 NDRDIODE_1
X152 1 79 NDRDIODE_1
X151 2 79 NDRDIODE_0
X150 94 79 NDRDIODE_1
X149 93 79 NDRDIODE_1
X148 78 79 NDRDIODE_1
X147 78 93 NDRDIODE_1
X146 78 77 NDRDIODE_1
X145 77 93 NDRDIODE_0
X144 92 77 NDRDIODE_0
X143 77 76 NDRDIODE_1
X142 76 75 NDRDIODE_0
X141 90 75 NDRDIODE_1
X140 75 89 NDRDIODE_1
X139 74 75 NDRDIODE_1
X138 74 89 NDRDIODE_0
X137 87 72 NDRDIODE_0
X136 72 71 NDRDIODE_0
X135 87 71 NDRDIODE_1
X134 71 86 NDRDIODE_1
X133 71 85 NDRDIODE_0
X132 70 71 NDRDIODE_0
X131 85 70 NDRDIODE_1
X130 70 69 NDRDIODE_1
X129 85 69 NDRDIODE_1
X128 69 84 NDRDIODE_1
X127 69 83 NDRDIODE_1
X126 68 69 NDRDIODE_1
X125 83 68 NDRDIODE_0
X124 68 67 NDRDIODE_0
X123 83 67 NDRDIODE_0
X122 67 82 NDRDIODE_0
X121 7 67 NDRDIODE_1
X119 6 67 NDRDIODE_0
X117 64 1 NDRDIODE_1
X116 64 1 NDRDIODE_0
X115 79 64 NDRDIODE_0
X114 64 78 NDRDIODE_0
X113 63 64 NDRDIODE_0
X112 78 63 NDRDIODE_0
X111 63 62 NDRDIODE_0
X110 78 62 NDRDIODE_0
X109 62 77 NDRDIODE_0
X108 62 76 NDRDIODE_0
X107 62 61 NDRDIODE_1
X106 76 61 NDRDIODE_1
X105 60 61 NDRDIODE_0
X104 76 60 NDRDIODE_0
X103 60 75 NDRDIODE_1
X102 74 60 NDRDIODE_0
X101 59 60 NDRDIODE_0
X100 59 74 NDRDIODE_1

X99 59 58 NDRDIODE_1
X98 58 74 NDRDIODE_1
X97 72 58 NDRDIODE_0
X96 57 58 NDRDIODE_0
X95 72 57 NDRDIODE_0
X94 56 57 NDRDIODE_1
X93 72 56 NDRDIODE_1
X92 56 71 NDRDIODE_0
X91 56 70 NDRDIODE_1
X90 56 55 NDRDIODE_1
X89 55 70 NDRDIODE_1
X88 54 55 NDRDIODE_0
X87 70 54 NDRDIODE_0
X86 69 54 NDRDIODE_1
X85 54 68 NDRDIODE_0
X84 54 53 NDRDIODE_0
X83 68 53 NDRDIODE_1
X82 52 53 NDRDIODE_1
X81 68 52 NDRDIODE_0
X80 67 52 NDRDIODE_1
X79 52 6 NDRDIODE_1
X77 6 52 NDRDIODE_1
X76 1 50 NDRDIODE_0
X75 50 49 NDRDIODE_0
X74 49 1 NDRDIODE_1
X73 64 49 NDRDIODE_0
X72 63 49 NDRDIODE_1
X71 15 49 NDRDIODE_0
X70 15 63 NDRDIODE_0
X69 15 47 NDRDIODE_1
X68 63 47 NDRDIODE_0
X67 62 47 NDRDIODE_1
X66 47 61 NDRDIODE_0
X65 47 14 NDRDIODE_0
X64 61 14 NDRDIODE_1
X62 14 61 NDRDIODE_1
X61 60 14 NDRDIODE_0
X60 14 59 NDRDIODE_1
X59 14 44 NDRDIODE_1
X58 44 59 NDRDIODE_1
X57 13 44 NDRDIODE_1
X56 13 59 NDRDIODE_0
X55 13 58 NDRDIODE_0
X54 13 57 NDRDIODE_0
X53 13 42 NDRDIODE_0
X52 57 42 NDRDIODE_0
X51 12 42 NDRDIODE_1
X50 57 12 NDRDIODE_1
X49 12 56 NDRDIODE_0
X48 55 12 NDRDIODE_0
X46 12 55 NDRDIODE_0
X45 12 39 NDRDIODE_1
X44 55 39 NDRDIODE_0
X43 39 54 NDRDIODE_1
X42 53 39 NDRDIODE_1
X41 11 39 NDRDIODE_1
X40 11 53 NDRDIODE_0
X39 37 11 NDRDIODE_0
X38 53 37 NDRDIODE_1
X37 37 52 NDRDIODE_0
X36 6 37 NDRDIODE_0

```

X35 35 50 NDRDIODE_0
X34 35 34 NDRDIODE_0
X33 50 34 NDRDIODE_1
X32 34 49 NDRDIODE_1
X31 15 34 NDRDIODE_1
X30 32 15 NDRDIODE_1
X29 32 47 NDRDIODE_0
X28 32 14 NDRDIODE_0
X27 32 14 NDRDIODE_1
X22 44 14 NDRDIODE_1
X21 14 29 NDRDIODE_0
X20 29 44 NDRDIODE_0
X19 13 29 NDRDIODE_0
X18 44 13 NDRDIODE_0
X16 13 42 NDRDIODE_0
X15 13 27 NDRDIODE_1
X14 42 27 NDRDIODE_0
X13 27 12 NDRDIODE_0
X12 12 42 NDRDIODE_0
X7 12 24 NDRDIODE_0
X6 12 24 NDRDIODE_0
X5 39 24 NDRDIODE_1
X4 11 24 NDRDIODE_1
X3 11 24 NDRDIODE_1

```

```

.SUBCKT NDRDIODE_1 IN OUT
G1 IN OUT PWL(1) IN OUT -1V,-1n -.8V,-1n -.625V,-760n 0V,0n
.625V,760n .8V,1n 1V,1n SMOOTH=1
.ENDS

```

```

.SUBCKT NDRDIODE_0 IN OUT
G1 IN OUT PWL(1) IN OUT -2V,-.8n -1V,-.8n -.75V,-.55n -.5V,-.21n
0V,0n .5V,.21n .75V,.55n 1V,.8n 2V,.8n SMOOTH=1
.ENDS

```

```

.END

```

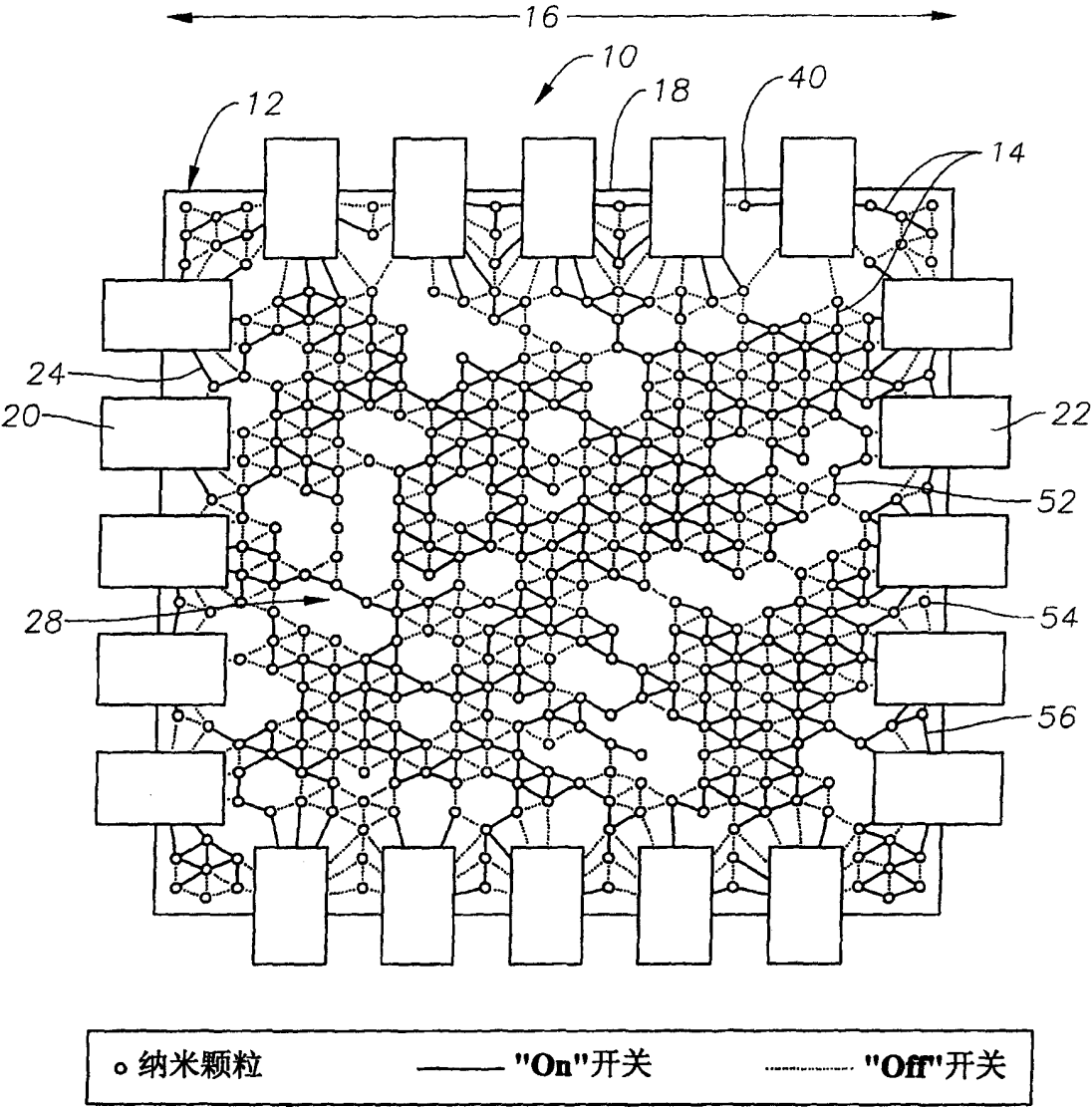


图1

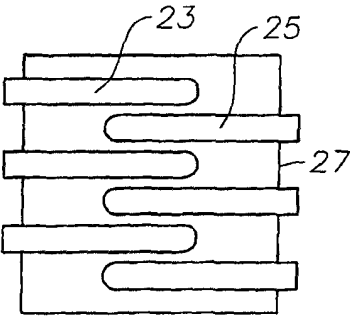


图2A

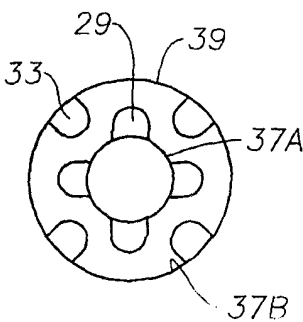


图2B

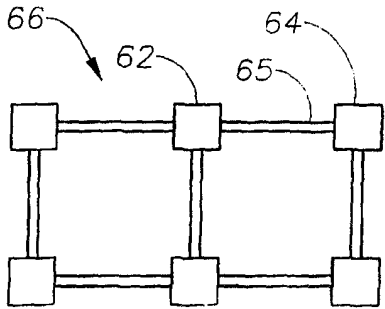
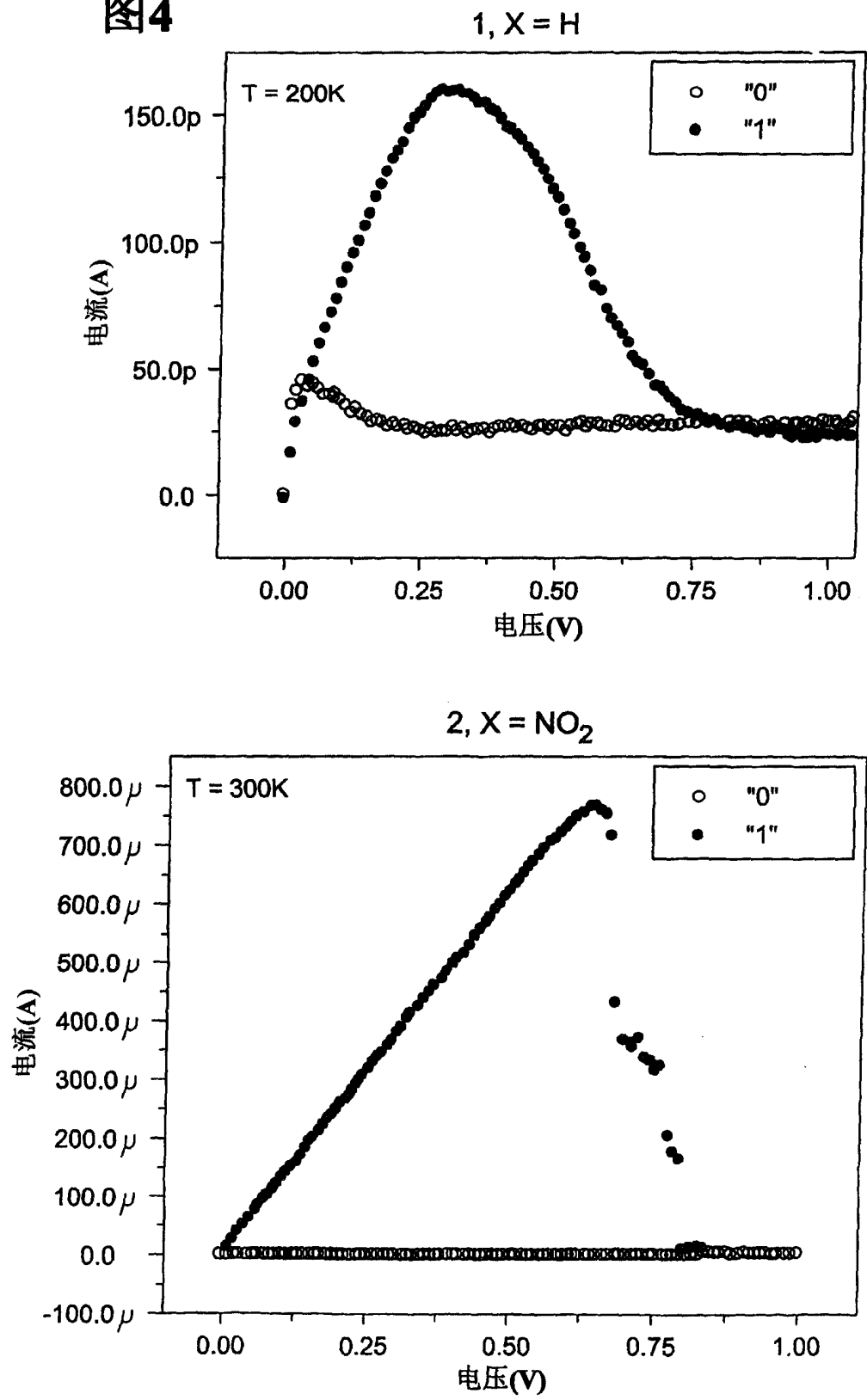


图5

图4



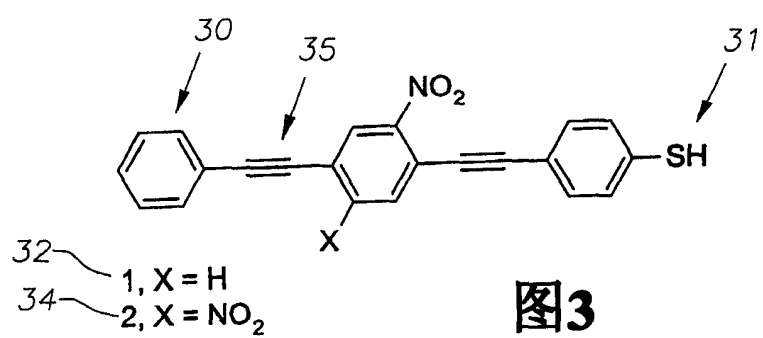


图3

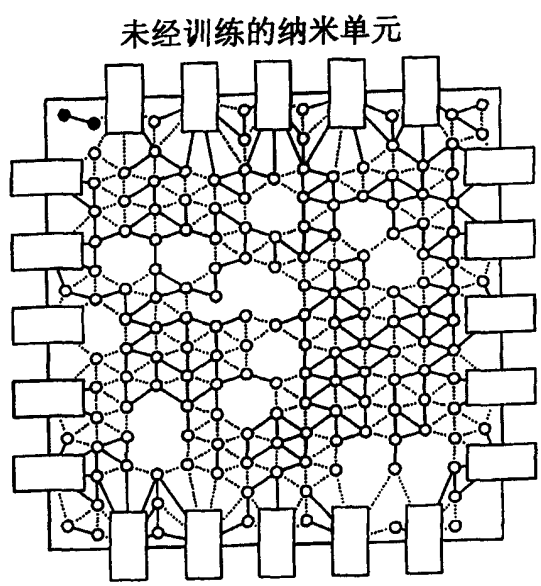


图7

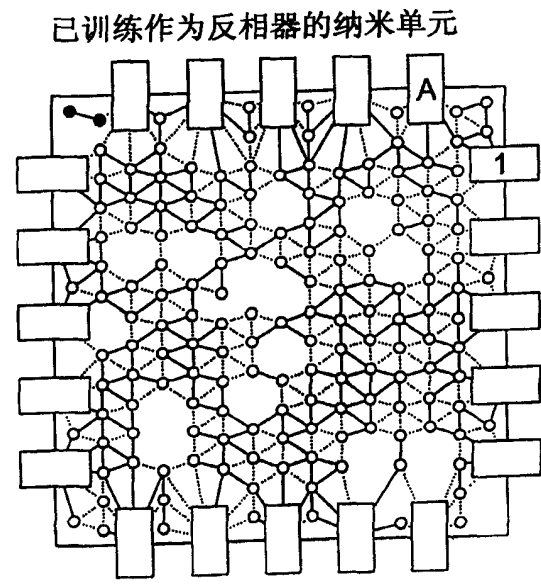


图8

反相器真值表	
输入A	输出1
0	1
1	0

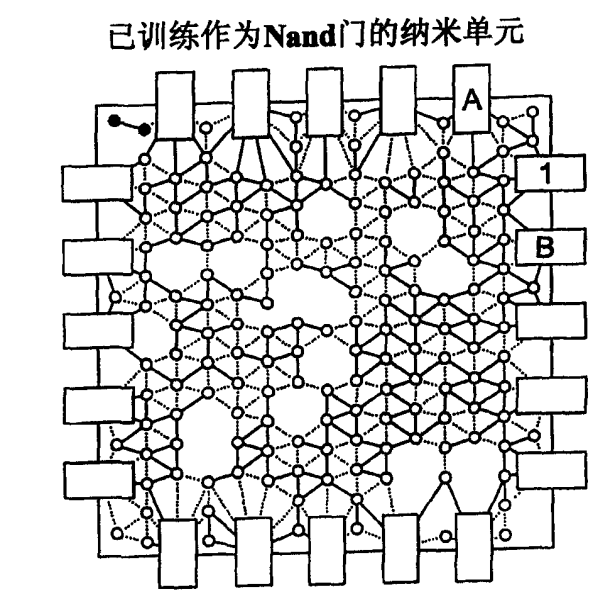
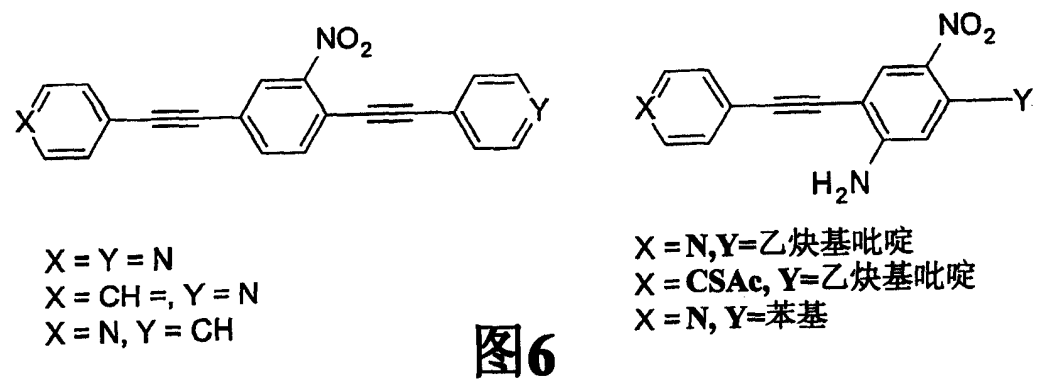


图9

Nand门真值表		
输入A	输入B	输出1
0	0	1
0	1	1
1	0	1
1	1	0

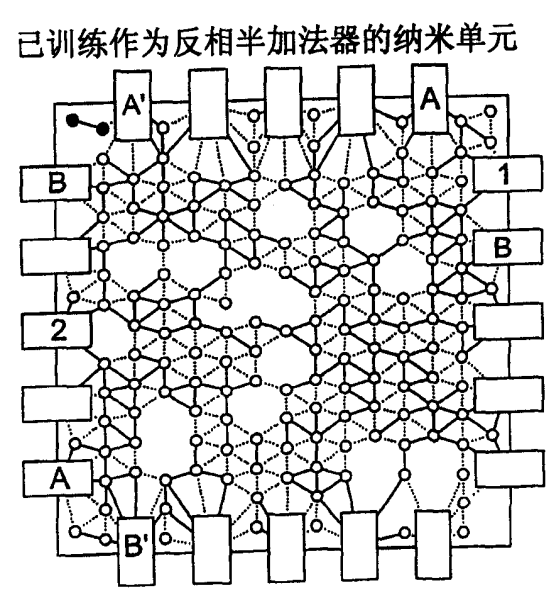


图10

半加法器真值表的反相			
输入A	输入B	输出1	输出2
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	1