[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 1/22 (2006.01)
G06F 15/78 (2006.01)
G06F 9/44 (2006.01)

[12] 发明专利申请公布说明书

[21] 申请号 200580006325.X

[43] 公开日 2007 年 5 月 30 日

[11] 公开号 CN 1973253A

[22] 申请日 2005.2.25
[21] 申请号 200580006325.X
[30] 优先权
[32] 2004.2.27 [33] GB [31] 0404443.4
[86] 国际申请 PCT/GB2005/000711 2005.2.25
[87] 国际公布 WO2005/083582 英 2005.9.9
[85] 进入国家阶段日期 2006.8.28
[71] 申请人 赛恩控股有限公司
地址 英国剑桥
[72] 发明人 P·T·约翰逊

[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 王 岳 王 勇

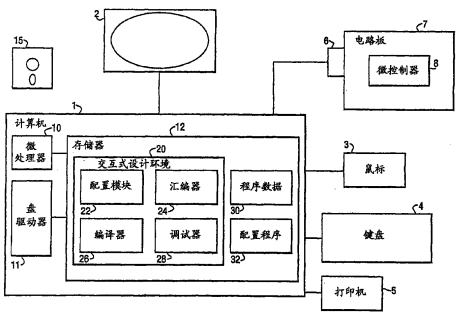
权利要求书 10 页 说明书 31 页 附图 20 页

[54] 发明名称

用于产生配置数据的方法和设备

[57] 摘要

描述了一种用于编程并测试微控制器(8)的计算机系统(1)。计算机系统(1)包括交互式设计环境(20)，包括配置模块(22)、汇编器(24)、编译器(26)和调试器(28)。编译器(26)、汇编器(24)和调试器(28)都使高级程序数据(30)能够被转换为二进制代码并且存储在程序存储装置内以便控制在微控制器(8)内所包括的 CPU 起作用。配置模块(22)是这样以致使用户能够产生配置程序(30)，用于把值存储在寄存器中以便控制微控制器(8)中的内置外围装置的设置和往返于所述外围装置的信号的路由。在两阶段过程中识别用于路由信号的寄存器值。首先利用配置模块(22)，用户识别将启用哪些外围装置。然后产生用户接口以图解用于路由为所识别外围装置组所必须的信号的可能配置。然后用户可以选择应当使用所识别的哪些配置数据组来产生配置程序(20)。



1. 一种微控制器配置设备，可操作来识别用于存储在包括多个外围装置的微控制器的寄存器内的一组寄存器值，所述微控制器可操作来根据在所述寄存器中所存储的寄存器值来经由一组端口往返于所述外围装置路由不同的信号选择，所述配置设备包括：

第一用户输入接口，可操作来使用户能够输入用于识别从在所述微控制器中所包括的多个外围装置中的外围装置选择的数据；

确定模块，可操作来为用于识别外围装置选择的输入数据来确定用于往返于所选外围装置路由所要求的一组信号，以便使每个所述外围装置能够起作用，所述确定模块还可操作来识别寄存器值的多个组合，所述寄存器值当存储在微控制器的寄存器内时使所有确定的信号能够经由所述微控制器的端口被路由；

第二用户输入接口，可操作来使用户能够选择由所述确定模块所识别的寄存器值的所述多个组合之一；和

输出模块，可操作来输出用于识别在微控制器中的寄存器内存储的一组寄存器值的数据，包括与所选寄存器值的组合相对应的寄存器值。

2. 如权利要求1所述的设备，其中所述确定模块包括：

多个配置表，当与所述端口相关联的寄存器被设置为定义值时，每个配置表均用于为微控制器的端口识别要经由与所述端口相关联的一个或多个引脚接收或输出的信号；

外围装置数据库，被配置为存储数据，所述数据用于为多个外围装置中的每个识别用于往返于所选外围装置路由所要求的信号以便使每个外围装置能够起作用；和

配置检查模块，可操作来利用所述配置表和所述外围装置数据库来识别寄存器值的组合，所述寄存器值当存储在微控制器的寄存器内时使由所识别的外围装置选择要求的所有信号能够经由所述微控制器的端口被路由。

3. 如权利要求1所述的设备，其中所述确定模块可操作来确定寄存器值的任何组合是否当存储在微控制器的寄存器内时使由所识别的外围装置选择所要求的所有信号能够经由所述微控制器的端口被路由，并且如果没有任何寄存器值的组合使由所识别的外围装置

选择所要求的所有信号能够经由所述微控制器的端口被路由，那么输出警告数据。

4. 如权利要求3所述的设备，其中所述警告数据包括用于识别所识别的外围装置选择中的哪些外围装置要求把不兼容的寄存器值存储在微控制器的寄存器内使得由所识别的外围装置选择所要求的所有信号能够经由所述微控制器的端口被路由的数据。

5. 如权利要求3所述的设备，其中所述警告数据包括用于识别候选微控制器的数据，所述候选微控制器包括所识别的外围装置并且能够存储一组寄存器值，所述寄存器值使由所识别的外围装置选择所要求的所有信号能够经由所述候选微控制器的端口被路由。

6. 如权利要求1所述的设备，其中所述第二用户输入接口可操作来产生所述微控制器的图示，用于图解微控制器的引脚的相对物理位置，当包括与所述寄存器值组合的寄存器值相对应的寄存器值被存储在所述微控制器的寄存器内时，该相对物理位置用来往返于所识别的外围装置选择而路由信号。

7. 如权利要求6所述的设备，其中所述第二用户输入接口可操作来根据用户从所排序的组合选择中选择组合来排序寄存器值的多个组合并且产生所述图示。

8. 如权利要求7所述的设备，其中所述第二用户输入接口可操作来当所述寄存器值的组合被存储在所述微控制器的寄存器内时，根据所述微控制器专用于路由通用输入和输出的信号的引脚数目来排序寄存器值的多个组合。

9. 如权利要求7所述的设备，其中所述第二用户输入接口可操作来当所述寄存器值的组合被存储在所述微控制器的寄存器内时，根据在显示内所选外围装置的表示和与把信号路由到所述各自外围装置相关联的引脚表示之间的相对距离来排序寄存器值的多个组合。

10. 如权利要求1所述的设备，其中所述微控制器还包括一个或多个寄存器，可操作来存储用于所述多个外围装置的设置数据，所述设备还包括第三用户输入接口，可操作来使用户能够输入用于识别存储在与所述外围装置选择相关联的一个或多个寄存器中的寄存器值的数据。

11. 如权利要求1所述的设备，其中所述输出模块可操作来输出用于识别一组寄存器值的数据，包括与所选寄存器值的组合相对应的寄存器值和没有为其识别任何其它寄存器值的寄存器的默认值。

12. 如权利要求1所述的设备，其中所述输出模块可操作来产生计算机程序，所述计算机程序可操作来当存储在与所述微控制器相关联的存储器中时，使所述微控制器存储在所述微控制器的寄存器中所识别的寄存器值组相对应的寄存器值。

13. 如权利要求1所述的设备，其中所述输出模块可操作来输出用于定义所述微控制器的图示的数据，包括用于识别往返于所选外围装置路由所要求的信号组的标记以使每个所述外围装置能够起作用。

14. 如权利要求13所述的设备，其中所述输出模块可操作来输出用于定义图示的数据，其中所述标记按照与所识别的信号相关联的外围装置而在所述表示中分组。

15. 如权利要求13所述的设备，其中所述输出模块可操作来输出用于定义图示的数据，其中所述标记在所述表示中被分组以便表示用于转送所述信号的引脚的物理位置。

16. 如权利要求1所述的设备，其中所述微控制器的至少一个端口包括可操作来转送根据在与所述端口相关联的寄存器中存储的寄存器值所选择的单个信号的端口。

17. 如权利要求16所述的设备，其中所述微控制器的每个端口包括可操作来转送根据在与所述端口相关联的寄存器中存储的寄存器值所选择的单个信号的端口，所述设备可操作来识别寄存器值组，所述寄存器值当存储在所述寄存器中时使所识别的外围装置选择起作用所要求的信号能够经由所述端口被转送。

18. 一种用于识别存储在包括多个外围装置的微控制器的寄存器内的一组寄存器值的方法，所述微控制器可操作来根据在所述寄存器中所存储的寄存器值来经由一组端口往返于所述外围装置路由不同的信号选择，所述方法包括：

接收用于识别从在所述微控制器所包括的多个外围装置中选择外围装置的数据；

为所接收的用于识别外围装置选择的数据而确定用于往返于所

选外围装置路由所要求的一组信号以便使每个外围装置能够起作用；

识别寄存器值的多个组合，所述寄存器值当存储在微控制器的寄存器内时使所有确定的信号能够经由所述微控制器的端口被路由；

接收用于识别选择多个识别的寄存器值组合之一的数据；并且

输出用于识别在微控制器中的寄存器内存储的一组寄存器值的数据，包括与所选寄存器值的组合相对应的寄存器值。

19. 如权利要求18所述的方法，还包括：

存储多个配置表，当与所述端口相关联的寄存器被设置为定义值时，每个配置表均用于为微控制器的端口识别要经由与所述端口相关联的一个或多个引脚接收或输出的信号；

存储外围装置数据库，被配置为存储数据，所述数据用于为多个外围装置中的每个识别用于往返于所选外围装置路由所要求的信号以便使每个外围装置能够起作用；并且

利用所述配置表和所述外围装置数据库来识别寄存器值的组合，所述寄存器值当存储在微控制器的寄存器内时使由所识别的外围装置选择要求的所有信号能够经由所述微控制器的端口被路由。

20. 如权利要求18所述的方法，还包括：

如果无法识别任何可兼容的寄存器值组，那么输出用于识别所识别的外围装置选择中的哪些外围装置会要求把不兼容的寄存器值存储在微控制器的寄存器内使得由所识别的外围装置选择所要求的所有信号能够经由所述微控制器的端口被路由的数据。

21. 如权利要求18所述的方法，还包括：

如果无法识别任何可兼容的寄存器值组，那么输出用于识别候选微控制器的数据，所述候选微控制器包括所识别的外围装置，能够存储一组寄存器值，所述寄存器值使由所识别的外围装置选择所要求的所有信号能够经由所述候选微控制器的端口被路由。

22. 如权利要求18所述的方法，还包括：

产生用于定义所述微控制器的图示的数据，用于图解微控制器的引脚的相对物理位置，当包括与所述寄存器值组合相对应的寄存器值的寄存器值被存储在所述微控制器的寄存器内时，该相对物理位置用来往返于所识别的外围装置选择而路由信号。

23. 如权利要求22所述的方法，还包括排序所识别的寄存器值的组合；并且

根据从所排序的组合选择中所接收的用户输入组合选择来产生所述图示。

24. 如权利要求23所述的方法，其中所述排序寄存器值的多个组合包括：当所述寄存器值的组合被存储在所述微控制器的寄存器内时，根据所述微控制器专用于路由通用输入和输出的信号的引脚数目来排序。

25. 如权利要求23所述的方法，其中所述排序寄存器值的多个组合包括：当所述寄存器值的组合被存储在所述微控制器的寄存器内时，根据在所述微控制器的表示内所选外围装置的表示和与把信号路由到所述各自外围装置相关联的引脚表示之间的相对距离来进行排序。

26. 如权利要求18所述的方法，其中所述微控制器还包括一个或多个寄存器，可操作来存储用于所述多个外围装置的设置数据，所述方法还包括接收用于识别存储在与所述外围装置选择相关联的一个或多个寄存器中的寄存器值的数据。

27. 如权利要求18所述的方法，包括输出用于识别一组寄存器值的数据，包括与所选寄存器值的组合相对应的寄存器值和没有为其识别任何其它寄存值的寄存器的默认值。

28. 如权利要求18所述的方法，还包括产生计算机程序，所述计算机程序可操作来当存储在与所述微控制器相关联的存储器中时，使所述微控制器存储与在所述微控制器的寄存器中所识别的寄存器值组相对应的寄存器值。

29. 如权利要求18所述的方法，还包括输出用于定义所述微控制器的图示的数据，包括用于识别往返于所选外围装置路由所要求的信号组的标记以使每个所述外围装置能够起作用。

30. 如权利要求29所述的方法，其中所述标记由与所识别的信号相关联的外围装置来在所述表示中分组。

31. 如权利要求29所述的方法，其中所述标记在所述表示中被分组以便表示用于转送所述信号的引脚的物理位置。

32. 如权利要求18所述的方法，其中所述微控制器的至少一个

端口包括可操作来转送根据在与所述端口相关联的寄存器中存储的寄存器值所选择的单个信号的端口。

33. 如权利要求32所述的方法，其中所述微控制器的每个端口包括可操作来转送根据在与所述端口相关联的寄存器中存储的寄存器值所选择的单个信号的端口，所述方法包括识别寄存器值组，所述寄存器值当存储在所述寄存器中时使所识别的外围装置选择起作用所要求的信号能够经由所述端口被转送。

34. 一种存储有计算机可执行指令的计算机可读介质，所述计算机可执行指令使可编程计算机执行用于识别在包括多个外围装置的微控制器的寄存器内存储的一组寄存器值的方法，所述微控制器可操作来根据在所述寄存器中所存储的寄存器值来经由一组端口往返于所述外围装置路由不同的信号选择，所述方法包括:

接收用于识别从在所述微控制器所包括的多个外围装置中选择外围装置的数据;

为所接收的用于识别外围装置选择的数据而确定用于往返于所选外围装置路由所要求的一组信号以便使每个外围装置能够起作用;

识别寄存器值的多个组合,所述寄存器值当存储在微控制器的寄存器内时使所有确定的信号能够经由所述微控制器的端口被路由;

接收用于识别选择多个识别的寄存器值组合之一的数据; 并且

输出用于识别在微控制器中的寄存器内存储的一组寄存器值的数据，包括与所选寄存器值组合相对应的寄存器值。

35. 如权利要求34所述的计算机可读介质，其中所述方法还包括:

存储多个配置表,当与所述端口相关联的寄存器被设置为定义值时，每个配置表均用于为微控制器的端口识别要经由与所述端口相关联的一个或多个引脚接收或输出的信号;

存储外围装置数据库，被配置为存储数据，所述数据用于为多个外围装置中的每个识别用于往返于所选外围装置路由所要求的信号以便使每个外围装置能够起作用; 并且

利用所述配置表和所述外围装置数据库来识别寄存器值的组合，所述寄存器值当存储在微控制器的寄存器内时使由所识别的外

围装置选择所要求的所有信号能够经由所述微控制器的端口被路由。

36. 如权利要求34所述的计算机可读介质，其中所述方法还包括：

如果没有识别任何可兼容的寄存器值组，那么输出用于识别所识别的外围装置选择中的哪些外围装置要求把不兼容的寄存器值存储在微控制器的寄存器内使得由所识别的外围装置选择所要求的所有信号能够经由所述微控制器的端口被路由的数据。

37. 如权利要求34所述的计算机可读介质，其中所述方法还包括：

如果无法识别任何可兼容的寄存器值组，那么输出用于识别候选微控制器的数据，所述候选微控制器包括所识别的外围装置，能够存储一组寄存器值，所述寄存器值使由所识别的外围装置选择所要求的所有信号能够经由所述候选微控制器的端口被路由。

38. 如权利要求34所述的计算机可读介质，还包括：

产生用于定义所述微控制器的图示的数据，用于图解微控制器的引脚的相对物理位置，当包括与所述寄存器值组合相对应的寄存器值的寄存器值被存储在所述微控制器的寄存器内时，该相对物理位置用来往返于所识别的外围装置选择而路由信号的。

39. 如权利要求38所述的计算机可读介质，还包括：

排序所识别的寄存器值的组合；并且

根据从所排序的组合选择中接收的用户输入组合选择来产生所述图示。

40. 如权利要求39所述的计算机可读介质，其中所述排序寄存器值的多个组合包括：当所述寄存器值的组合被存储在所述微控制器的寄存器内时，根据所述微控制器专用于路由通用输入和输出的信号的引脚数目来排序。

41. 如权利要求39所述的计算机可读介质，其中所述排序寄存器值的多个组合包括：当所述寄存器值的组合被存储在所述微控制器的寄存器内时，根据在所述微控制器的表示内所选外围装置的表示和与把信号路由到所述各自外围装置相关联的引脚表示之间的相对距离来进行排序。

42. 如权利要求34所述的计算机可读介质，其中所述微控制器还包括一个或多个寄存器，可操作来存储用于所述多个外围装置的设置数据，所述方法还包括接收用于识别存储在与所述外围装置选择相关联的一个或多个寄存器中的寄存器值的数据。

43. 如权利要求34所述的计算机可读介质，其中所述方法还包括输出用于识别所述组寄存器值的数据，包括与所选寄存器值的组合相对应的寄存器值和没有为其识别任何其它寄存器值的寄存器的默认值。

44. 如权利要求34所述的计算机可读介质，其中所述方法还包括产生一种计算机程序，所述计算机程序可操作来当存储在与所述微控制器相关联的存储器中时使所述微控制器存储与在所述微控制器的寄存器中所识别的寄存器值组相对应的寄存器值。

45. 如权利要求34所述的计算机可读介质，其中所述方法还包括输出用于定义所述微控制器的图示的数据，包括用于识别往返于所选外围装置路由所要求的信号组的标记以使每个所述外围装置能够起作用。

46. 如权利要求45所述的计算机可读介质，其中所述标记由与所识别的信号相关联的外围装置来在所述表示中分组。

47. 如权利要求45所述的计算机可读介质，其中所述标记在所述表示中被分组以便表示用于转送所述信号的引脚的物理位置。

48. 如权利要求34所述的计算机可读介质，包括盘。

49. 如权利要求48所述的盘，包括磁、磁光或光盘。

50. 如权利要求34所述的计算机可读介质，包括计算机网络内的电信号。

51. 一种用于为包括多个外围装置的微控制器编程的方法，所述微控制器可操作来根据在所述微控制器的寄存器内所存储的寄存器值来经由一组端口往返于所述外围装置路由不同的信号选择，所述方法包括：

依照如权利要求28所述的方法来产生计算机程序；

把所产生的计算机程序存储在与微控制器相关联的存储器中；并且

运行所存储的程序以便使所述微控制器把对应于所识别的寄存

器值组的寄存器值存储在所述微控制器的寄存器中。

52. 一种微控制器，包括：

多个外围装置；

多个选择器，可操作来根据在与一组端口相关联的寄存器中所存储的寄存器值来经由所述端口往返于外围装置路由信号；

存储器，可操作来存储配置程序；和

处理部件，可操作来处理在所述存储器中所存储的配置程序以便把寄存器值存储在所述寄存器中；其中所述微控制器的每个端口包括可操作来转送根据在与所述端口相关联的寄存器中所存储的寄存器值所选择的单个信号的端口。

53. 如权利要求1所述的设备，其中所述第一用户输入接口可操作来使用户能够输入用于识别在微控制器以外的外部外围装置选择的数据，并且利用所述外部外围装置选择来识别从在所述微控制器中所包括的多个外围装置中的外围装置选择，其中所述微控制器被要求来与所述外部外围装置选择相对接。

54. 如权利要求53所述的设备，其中所述第一用户输入接口可操作来确定与外部外围装置输入选择相对接所必须的外围装置选择是否要求在微控制器中未包括的外围装置，其中所述输出模块可操作来输出用于产生计算机程序的数据以便使所述微控制器仿真除在所述微控制器中包括的外围装置之外的一个或多个附加外围装置。

55. 如权利要求53所述的设备，其中所述一个或多个外部外围装置包括可操作来存储设置数据的寄存器，所述设备还包括用户输入接口，可操作来使用户能够输入用于识别存储在与所述外部外围装置选择相关联的一个或多个寄存器中的寄存器值的数据。

56. 如权利要求55所述的设备，其中所述另外的用户接口可操作来把每个外部外围装置选择与内部外围装置相关联并且使输出模块输出用于识别彼此相兼容的外部外围装置和相关联的内部外围装置的一组寄存器值的数据。

57. 如权利要求18所述的方法，其中所述用于识别外围装置选择的接收数据包括用于识别在微控制器以外的一个或多个外围装置的数据，所述方法还包括处理所述接收数据以识别从在所述微控制器中包括的多个外围装置中的外围装置选择，其中所述微控制器被

要求来与所述外部外围装置相对接。

58. 如权利要求57所述的方法，其中如果与外部外围装置相对接所必须的外围装置确定包括在所述微控制器中未包括的一个或多个外围装置，其中所述输出步骤包括输出用于产生计算机程序的数据以便使所述微控制器仿真除在所述微控制器中包括的外围装置之外的一个或多个附加外围装置。

59. 如权利要求57所述的方法，还接收用于识别存储在与所述外部外围装置选择相关联的一个或多个寄存器中的寄存器值的数据，其中所述输出步骤包括识别存储在对应于所接收数据的一个或多个外部外围装置的寄存器内的寄存器值。

60. 如权利要求59所述的设备，还包括把外部外围装置与内部外围装置相关联，其中所述输出步骤包括输出用于识别彼此相兼容的外部外围装置和相关联的内部外围装置的一组寄存器值的数据。

61. 如权利要求34所述的计算机可读介质，其中所述用于识别外围装置选择的接收数据包括用于识别在微控制器以外的一个或多个外围装置的数据，所述方法还包括处理所述接收数据以识别从在所述微控制器中包括的多个外围装置中的外围装置选择，其中所述微控制器被要求来与所述外部外围装置相对接。

62. 如权利要求61所述的计算机可读介质，其中如果与外部外围装置相对接所必须的外围装置确定包括在所述微控制器中未包括的一个或多个外围装置，其中所述输出步骤包括输出用于产生计算机程序的数据以便使所述微控制器仿真除在所述微控制器中包括的外围装置之外的一个或多个附加外围装置。

63. 如权利要求62所述的计算机可读介质，还接收用于识别存储在与所述外部外围装置选择相关联的一个或多个寄存器中的寄存器值的数据，其中所述输出步骤包括识别存储在对应于所接收数据的一个或多个外部外围装置的寄存器内的寄存器值。

64. 如权利要求63所述的计算机可读介质，还包括把外部外围装置与内部外围装置相关联，其中所述输出步骤包括输出用于识别彼此相兼容的外部外围装置和相关联的内部外围装置的一组寄存器值的数据。

用于产生配置数据的方法和设备

技术领域

本申请涉及用于产生配置数据的方法和设备。本申请尤其涉及产生用于配置微控制器的配置数据。

背景技术

许多设备要求嵌入式微处理器系统或微控制器能够控制所述设备的处理。这种微控制器包括具有多个引脚的微芯片以便使信号能够进入并退出所述微芯片。微芯片典型情况下包括中央处理器、存储器和用于在微控制器内从事特殊任务的多个内置外围装置。

在现代微控制器设计中，内置外围装置可以接收、与之相交互并输出的信号数目常常超过微芯片上可用的引脚数目。这是因为通常在任何特定的应用中只需要利用可用外围装置的子集。在这种环境中，微控制器的编程具有两个部分。第一，用于控制微控制器的CPU的处理的计算机程序需要被写入并存储在所述微控制器的存储器中。通常，这通过用高级编程语言编写软件并且把所产生的代码编译为二进制代码来实现，其中所述二进制代码被存储在微控制器的存储器内。

编程微控制器的第二方面包括为微控制器内可用的外围装置确定适当设置，以便所利用的外围装置可以相交互并且经由所述微控制器上可用的引脚来输出信号。此编程微控制器的第二方面很难，这是因为不兼容的设置可能会产生很难识别并纠正的不正确操作。因此需要一种能够更容易地从事编程此微控制器的第二方面的系统。

发明内容

依照本发明的一个方面，提供了一种微控制器配置设备，可操作来识别用于存储在包括多个外围装置的微控制器的寄存器内的一组寄存器值，所述微控制器可操作来根据在所述寄存器中所存储的寄存器值来往返于所述外围装置路由不同的信号选择，所述配置设备

包括:

第一用户输入接口生成器,可操作来产生第一用户接口以便使用户能够输入用于识别从在所述微控制器中所包括的多个外围装置中的外围装置选择的数据;

确定模块,可操作来为用于识别经由所述第一用户接口所输入的外围装置选择的数据来确定为往返于所选择的外围装置路由所要求的一组信号,以便使所述每个外围装置能够起作用,所述确定模块还可操作来识别寄存器值的多个组合,所述寄存器值当存储在微控制器的寄存器内时使所有确定的信号能够被往返于所选外围装置路由;第二用户输入接口生成器,可操作来产生第二用户接口以便使用户能够选择由所述确定模块所识别的寄存器值的所述多个组合之一;和

输出模块,可操作来输出用于识别在微控制器中的寄存器内存储的一组寄存器值的数据,包括与经由所述第二用户接口所选择的寄存器值的组合相对应的寄存器值。

附图说明

参考附图本发明的进一步方面和实施例将变得更加清楚,其中:

图1是依照本发明第一实施例的用于内嵌测试微控制器的计算机系统的示意性框图,包括用于产生配置数据的配置模块;

图2是示例性的微控制器的示意性框图;

图3是图2的微控制器的时钟模块组件的示意性框图;

图4是图1的配置模块的子模块的示意性框图;

图5是在图4的配置模块的外围装置数据库内外围装置记录的示意性框图;

图6是用于识别示例性微控制器的端口的信道设置的配置表的例图;

图7是在图4的配置模块的所选外围装置数据库内所选外围装置记录的示意性框图;

图8是图4的配置模块处理的流程图;

图9是借助图4的配置模块所产生的用户接口的示例性例图;

图10是图4的配置模块的配置检查模块进行处理以便为微控制器

内的外围装置选择确定可兼容的配置设置的流程图；

图11是在微控制器上可用的多个外围装置已经被选择启动之后用户接口的示例性例图；

图12是图4的配置模块的配置检查模块进行处理以便为所选的一组外围装置计算一组可能的可兼容配置设置的流程图；

图13是用于为在图3中所示的时钟模块选择设置的用户接口菜单的示例性例图；

图14是依照本发明的第二实施例用于内嵌测试微控制器的计算机系统的示意性框图，包括用于产生配置数据的所修改配置模块；

图15是图14的修改配置模块的子模块的示意性框图；

图16是在图14的修改配置模块的外部外围装置数据库内的外部外围装置记录的示意性框图；

图17是在图14的修改配置模块的外部外围装置存储内的所选外部外围装置记录的示意性框图；

图18是借助图14的修改配置模块所产生的用户接口的示例性例图；

图19是图14的修改配置模块进行处理的流程图；和

图20和21是借助图14的修改配置模块所产生的用户接口的进一步示例性例图。

具体实施方式

第一实施例

图1是用于依照本发明的第一实施例执行内嵌测试微控制器的计算机系统的例图。特别地是，图1是包括计算机1的计算机系统的例图，所述计算机1被编程来使用户比现有技术中能够更容易地编程并测试微控制器。

如图1所示，计算机1连接到显示屏2、鼠标3、键盘4和打印机5。计算机1还经由接口6连接到电路板7，在所述电路板7上提供了微控制器8。计算机1本身包括微处理器10、盘驱动器11和存储器12。

盘驱动器被配置成接收计算机盘15，所述计算机盘15使存储器12被配置为用于定义交互式设计环境20的功能模块，包括配置模块22、汇编器24、编译器26和调试器28。

在此实施例中，汇编器24、编译器26和调试器28均包括常规的汇编器、编译器和调试器，被安排来处理经由盘驱动器11从盘15下载的以及在存储器12中作为程序数据30存储的高级计算机程序高级计算机程序，并且把所述高级计算机程序转换为二进制代码，所述二进制代码被经由接口6下载到微控制器8的存储器中。

如稍后详细描述，交互式设计环境20的配置模块22被安排来使用户能够经由键盘4和鼠标3输入指令以便产生用于微控制器8的配置程序32。此配置程序32这样配置微控制器8以致能够特殊选择微控制器8上的外围装置来启用，并且接收并输出到用于把微控制器8连接到电路板7的特殊引脚的信号。通过提供此配置模块22，由于简化了配置程序32的产生，所以显著地使编程和测试微控制器8的过程易于进行。

更具体地说，如将要描述，配置模块22通过为用户把任务划分为两个阶段来简化配置数据的产生。

首先，向用户展示用户接口，所述用户接口使用户能够识别要在微控制器8内启用的一组外围装置以及用于那些所选外围装置的设置。然后配置模块22利用用户对外围装置的选择来识别可能配置设置的有限子集，所述设置适于在微控制器8内外路由所选外围装置组的信号。

同时，向用户展示用于图解微控制器8的输入和输出的物理位置的显示，所述物理位置由所识别的可能配置设置组产生。因而胜于必须考虑微控制器8的所有可能设置，用户可以使用对不同的配置设置怎样改变物理位置的可用反馈来从所图解的子集中进行想要的选择，在所述物理位置处由微控制器8接收并输出用于所选外围装置的信号。当用户识别所想要的设置组时，那么可以根据由配置模块22所存储的数据来自动产生配置程序32。

然后可以通过由编译器26和汇编器24和配置程序32把来源于程序数据30的二进制代码从计算机的存储器12中下载到微控制器8的存储器中，来测试所述微控制器8的性能。然后配置程序32使微控制器8采用由用户经由配置模块22所选择的配置设置。微控制器8于是将利用所识别的配置设置来运行二进制代码并且所述微控制器8的输出可以依照常规方式来分析。

最后，当二进制代码和配置程序32的性能被认为是满意的时，被编程来利用程序数据30和配置程序32的微控制器8可以被生成并包括在更大设备的电路板内。

在描述配置模块22的结构和功能之前，首先参考图2和3来描述示例性微控制器8的结构。

示例性微控制器的结构

参照微控制器8的示意性框图图2，每个微控制器包括主芯片35，在所述主芯片35上提供有时钟36、中央处理器37、程序存储装置38和一组外围装置40-1－40-N，每个外围装置包括被安排来在所述芯片上进行特殊任务的功能块。典型的功能块可以是红外检测器、计数器、捕获计时器或UART等。在特定芯片上可用的精密外围装置在不同的微控制器之间可能会发生变化。

主芯片35借助一系列引脚42连接到电路板7。这些引脚42被安排来把信号从电路板的其余部分路由到主芯片35。常常，设置微控制器8以便可以由主芯片35上的CPU和外围装置40-1－40-N处理的输入和输出数目超过可用来路由进出主芯片35信号的引脚42的数目。这是因为通常在任何特定的应用中只会利用外围装置40-1－40-N的子集。

为了提供关于可以一起使用哪些外围装置组的灵活性以及为了使信号能够路由到不同的引脚42，所述引脚42被分成引脚分组。这些分组在图2中被示为端口A-K 45-56和包括一组控制信号57的端口。这些端口中的每个包括一组引脚，其中图2中的端口被示为除端口K 50和端口L 49具有四个引脚并且端口57包括具有十六个引脚的一组控制信号引脚之外都包括八个引脚。除与控制信号相关联的引脚组57之外，其余每个端口55 － 56与选择器60和寄存器62相关联。每个选择器60经由一组线路63连接到主芯片35上的外围装置40-1－40-N。然后每个选择器60被安排为根据在与该选择器60相关联的寄存器62中所存储的数据来选择把经由线路63所转送的哪些信号连接到被连接到所述选择器60的引脚。

因而通过把适当的寄存器值存储在寄存器62中，可以配置微控制器8以致把来自外围装置40-1－40-N的不同信号连接到所选的引脚

42，所述引脚42用于把微控制器8连接到电路板7。

除需要被编程来使适当的信号组能够经由引脚42来路由的寄存器62之外，在典型的微控制器中，时钟36和每个外围装置40-1－40-N还均具有与它们相关联的一组寄存器64、65-1－65-N，所述寄存器使所述时钟36和外围装置的单个参数能够适合于所述微控制器8将要执行的特定功能。

举例来说，图3是在微控制器8上可用的典型时钟36的示意性框图。典型的时钟可以包括被安排来产生具有32Khz频率的时钟信号的低振荡器80和被安排来产生具有5Mhz频率的时钟信号的高振荡器82。另外时钟可以包括被安排来使用来自低振荡器80的32Khz信号来产生5Mhz时钟信号的低PLL 84和被安排来把由高振荡器82所产生的时钟信号转换为具有100Mhz频率的时钟信号的高PLL 86。依照这种方式，时钟36将能产生一组四个时钟信号，每个时钟信号可以形成用于微控制器8的主时钟的基础。

为了使用户能够选择应当把哪个信号利用为主时钟信号，于是把低振荡器80、低PLL 84、高振荡器82和高PLL 86都连接到选择器87，所述选择器87随后连接到寄存器88，所述寄存器88在此例子中可以是两比特寄存器。通过把适当值存储在寄存器88中，选择器87被安排来确定由低振荡器88、低PLL 84、高振荡器82或高PLL 86所产生的哪个时钟信号将被用为基础时钟频率。

另外，时钟36还可能包括一对除法器90、91，被安排来通过将时钟除以预置值以根据由选择器87所选择的主时钟信号来导出不同频率的时钟信号。这些预置值被存储在与每个除法器90、91相关联的另外的寄存器92、94内。因而依照这种方式，都基于由选择器87所选主时钟信号的一组不同的时钟信号被时钟模块36导出并输出，并且由主芯片35上的CPU 37和外围装置40-1－40-N利用。因而为了使时钟模块36能够输出适当信号，需要把另外的配置数据存储在时钟的寄存器92、94中。

为了编程微控制器8，因此需要产生并存储三组数据。首先，二进制代码需要被产生并存储在主芯片35的程序存储装置38内，所述二进制代码然后可以由所述主芯片35的CPU 37处理并运行。依照常规方式，此代码可以由编译器26、汇编器27根据程序数据30产生并

且由调试器28修正。

然而另外，用于每个外围装置40-1－40-N和时钟36的配置数据需要被创建并存储在寄存器64、65-1－65-N内以便配置所述外围装置40-1－40-N和时钟36，从而设置这些外围装置和时钟使得执行它们所被安排来执行的专用功能。配置数据还需要被存储在与每个端口45-56相关联的寄存器62内，以便使由各个外围装置所利用的信号能够往返于用于把微控制器8连接到电路板7的适当引脚42被路由。

现在参考图4－13描述配置模块22的结构和功能，所述配置模块22使用户能够容易且有效地产生所有要求的配置数据。

配置模块的功能组件的概述

图4是依照本发明的此实施例的配置模块22的子组件的示意性框图。

在此实施例中，配置模块22包括：接口生成器100，被安排来经由键盘4和鼠标3接收指令并且在显示屏2上显示用户接口；配置检查模块102，被安排来确定用于与微控制器8的端口45-56相关联的寄存器62的哪些设置与所选外围装置组兼容；代码生成模块104，被安排来产生用于使适当值存储在寄存器62中以便特定选择外围装置40-1－40-N的配置程序32；和输出模块106，被安排来利用用于识别所选外围装置组和端口配置的数据来产生并输出用于所识别的微控制器8配置的示意图和印刷电路板图。

除上述功能模块之外，配置模块22还包括：外围装置数据库108，用于存储用来定义在特定微控制器8上可用的每个外围装置40-1－40-N和时钟36的信号和潜在设置的数据；一组配置表110，包括用于表明当不同的值存储在与所述微控制器8的端口45-56相关联的寄存器62中时哪些信号连接到微控制器8上的哪些引脚42的表；优选输入表112，用于表明哪些引脚42优选由外围装置40-1－40-N用来接收输入信号；规则数据库114，包括用于定义在单个外围装置40-1-10-N和时钟36内不同设置的兼容性要求的数据；一组默认引脚名称116，是用于识别与用于把微控制器8连接到电路板7的每个引脚42相关联的默认名称的标记；和当没有预先规定寄存器的其它值时，用于定义一组默认配置值118以供存储在所述微控制器的寄存器62中的数

据。

最后，在使用中配置模块22还被安排来存储用于识别可接受的配置值的配置数据119以便存储在寄存器62中以供特定选择外围装置，以及存储所选外围装置数据库120以便存储用于识别那些所选外围装置的一组所选外围装置记录以及所建议的寄存器值以便连同用于时钟36及其寄存器64的记录一起存储在与所选外围装置相关联的寄存器65-1-65-N中。

图5是配置模块22的外围装置数据库108内的记录的示意性框图。在此实施例中，外围装置记录121连同与时钟36相关联的附加记录以及任何其它全局参数一起为在微控制器8中可用的每个外围装置40-1－40-N存储在外围装置数据库108内，所述全局参数诸如与微控制器8的总输入输出引脚相关联的数据。

在此实施例中，每个外围装置记录121包括：名称数据122，用于识别外围装置的名称；信号列表123，用于表明与该记录的外围装置相关联的输入和输出；和状态数据124，用于为信号列表123中的每个条目识别特定的信号是输入还是输出还是两者都是。

因而例如用于异步接收和发送模块的外围装置记录可以包括以下数据：

外围装置名称：UARTA

信号列表：UARTA_TX，UARTA_RX

状态数据：输出，输入

另外，每个外围装置记录121还包括寄存器名称列表125、一组寄存器记录126和一组变量值127。寄存器名称125、寄存器记录126和变量值127数目在各种外围装置之间可能会发成变化。

寄存器名称列表识别在外围装置记录表示的外围装置中所包括的每个寄存器64、65-1－65-N的功能。因而在用于图3的示例性时钟的时钟记录的情况下，可以存储以下寄存器名称：

低频率时钟频率

高频率时钟频率

内时钟频率

自由运行时钟频率

CPU时钟频率

寄存器记录126包括用于与外围装置相关联的寄存器64、65-1；65-N的每个可能值的记录，所述外围装置记录121为所述外围装置表示哪个能够采用固定数目的设置之一。每个寄存器记录121包括寄存器号码130，用于识别寄存器记录126所涉及的外围装置内的寄存器；由所述记录126所识别的寄存器的可能设置的值132；和把由寄存器号码130所识别的寄存器设置为由所述记录126所识别的值132的效果的文本解释的设置134。

因而例如在图3的示例性时钟的情况下，可以相对于用于选择源信号的寄存器88来存储以下寄存器记录126:

寄存器号: 88

值: 00

设置: 低频率时钟

寄存器号: 88

值: 01

设置: 高频率时钟

等等

类似地,外围装置记录121的变量值127包括用于识别把与外围装置记录121的外围装置相关联的变量设置为不同值的效果的文本数据。

配置表

图6是依照本发明此实施例的配置表110的示例性例图。

在此实施例中，为微控制器8的每个端口45-56存储配置表，为所述微控制器把配置模块22安排来产生配置数据。每个配置表包括一组引脚号，包括用于识别与该特定端口相关联的引脚号135和与所述端口相关联的一系列信号标记136，当被识别为信道值的特定值被存储在用于该端口的寄存器62中时。

通常，从所指定的芯片边角逆时针计数来分配芯片的引脚号。尽管在图2的示意图中，与相同端口相关联的引脚被集中在一起示出，然而在实际的微控制器中并不要求与所述端口相关联的引脚必须是在物理上相邻的引脚，相反，端口的定义包括与相同的选择器60和寄存器62相关联的一组引脚。因而在图6的例图中，用于四个引脚的

示例性端口的一组引脚号135被示为编号为62、64、65和66的四个引脚，编号为63的引脚被分配给不同的端口。

除一组引脚号135之外，每个配置表还包括当对应于信道值的值被存储在寄存器62中时用于识别哪些信号被潜在地连接到与特定端口相关联的哪个引脚的数据136。

因而在图6的例子中，当零值被存储在与由该表所表示的端口的选择器60相关联的寄存器62内时，如果串行输入接口已经被选为要由微控制器8利用的外围装置之一，那么用于唤醒CPU 37的唤醒信号WAKEUP被分配给引脚62，并且流确认STREAM_ACK、流状态STREAM_STS和流请求STREAM_REQ信号被串行网络接口经由引脚64、65和66输入和输出。

如从图6的示例性表中可以看出，改变在寄存器62内所存储的值可以使不同的信号连接到不同的引脚上。相反例如在图6的表中的唤醒信号WAKEUP的情况下，当与所述表的端口相关联的寄存器被设置为零或者作为选择被设置为三时，此信号由引脚62接收。如稍后将要更详细地描述，在配置表110内所存储的数据使配置检查模块102能够识别可能的寄存器值组，其使所选外围装置组能够在来自外围装置40-1－40-N的信号彼此不冲突的情况下（当它们视图在相同的引脚上被输出或接收时）在微控制器8内利用。

所选的外围装置数据库

图7是所选外围装置数据库120内所存储的数据的示意性框图。最初在此实施例中，所选外围装置数据库被安排来存储用于微控制器的时钟模块36的单个外围装置记录140。如将要描述，在使用中当不同的外围装置由用户经由配置模块22识别以被要求使用时，附加外围装置记录140被添加到数据库120。

在所选外围装置数据库120中所存储的外围装置记录140都包括外围装置名称141，其对应于从来自外围装置数据库108的外围装置记录121中所选外围装置的外围装置名称122；一组寄存器值142和变量值143，是用于与由所述记录的外围装置名称141识别的外围装置相关联的寄存器和变量的一组当前设置；位置144，是用于识别在由接口生成器100所产生的屏幕显示内应当在哪示出用于显示外围装

置名称141的框以便表示所识别的外围装置存在能够在微控制器8内启用的一对x y坐标。

配置模块的处理

现在参考图8 - 13描述配置模块22的处理。

参照作为配置模块22的处理的流程图的图8，最初当配置模块22被首先调用时，接口生成器100使（S8-1）初始用户接口屏幕显示在显示屏2上。

图9是依照本发明此实施例的初始用户接口显示145的示例性例图。从左至右，用户接口显示145包括外围装置窗口150，用于显示一系列图标151，所述图标151与在微控制器8内所存在的每个外围装置40-1 - 40-N相关联，其中可以为所述微控制器8产生配置数据；还包括主窗口152，在其内示出了微控制器8的图示154，其中将为所述微控制器8产生配置数据。

在此实施例中，图示154在其边缘具有微控制器的引脚42的表示，每个引脚与标记156相关联，所述标记156最初对应于与引脚号相关联的默认引脚名称116，所述引脚号由在配置模块22内存储的默认引脚名称数据116来识别。这些引脚名称被依照引脚号次序来安排由此图解输入和输出的相对位置，所述位置由引脚标记156相对于在微控制器8的一个边角的唯一标记157来识别，所述唯一的标记157还被图解为在控制器8的示意图154上的标记157。

最后，用户接口显示145在右手侧还包括配置窗口160，在所述配置窗口160内显示端口配置数据162，用于为微控制器8的端口45-56识别每个寄存器62的当前寄存器设置，其中将为所述微控制器8连同用于表明与当前配置163内的通用输入和输出相关联的引脚数目的数目一起产生配置数据。最初所述设置将是对应于在配置模块22内所存储的默认配置值118的设置。

在此端口配置窗口160的底部边缘是一组配置按钮164，用于在由配置模块22内所存储的配置数据119来表示的当前适用配置之间进行选择。当设备被首次调用时，所存储的配置数据119包括用于识别单个配置的数据，包括一组空值，每个值用于一个端口，如将要描述所述空值将使端口配置窗口160显示对应于所存储的默认配置118

的端口配置数据162和引脚标记156以便对应于所存储的默认引脚名称116。

在鼠标3的控制下利用指针165，于是用户能够选择用户接口显示屏145的各个区域，所述用户接口显示屏145使配置模块22改变所显示的用户接口显示145和在所选外围装置数据库120中存储的数据和配置数据119，这样用户可以选择用于微控制器8的适当寄存器设置，现在将要进行描述。

返回到图8，一旦接口生成器100已经在屏幕2上显示初始用户接口显示145，那么接口生成器（S8-2）检查用户是否已经在鼠标3的控制下使用指针165选择外围装置窗口150中的外围装置151的任何表示。在此实施例中，利用选择外围装置151的表示来使用户能够识别他们想要特定的外围装置可在微控制器8上使用。

如果是这种情况，那么接口生成器100（S8-3）创建新选择的外围装置记录140并且把所述记录140存储在所选外围装置数据库120中。在新近产生的所选外围装置记录140中，对应于从外围装置列表151所选择的外围装置，外围装置名称141被设置为外围装置记录121的外围装置名称122。然后寄存器值和变量值142、143被设置为所选外围装置的默认值，并且x y位置被设置为跟踪指针165直到用户放开鼠标3上的按键。

因而例如如果将选择在外围装置窗口顶部的SCI智能卡接口图标，那么可能会产生以下记录来存储：

外围装置名称：SCI

寄存器值：0，0，0，0，0，0

变量值：无

位置：指针的x y坐标

然后接口生成器100使以由新的外围装置记录140所识别的位置144为中心的正方块显示在屏幕2上，在所述屏幕2内对应于新创建的外围装置记录140的外围装置名称141的文本出现在所述块中。

然后配置模块22调用配置检查模块102来确定（S8-4）一组可能的端口配置，所述端口配置启用对应于外围装置组的信号输入和输出，现在已经为所述外围装置存储了外围装置记录140。

参照配置检查模块102的处理的流程图图10，在已经把新的外围

装置记录140存储在所选外围装置数据库120中之后，配置检查模块102最初继续确定（S10-1）端口信道设置，所述端口信道设置可能受到包括此附加外围装置的影响。

更具体地说，配置检查模块102最初识别外围装置记录121的信号列表123，所述外围装置记录121具有对应于新近产生的所选外围装置记录140的外围装置名称141的外围装置名称122。

因而例如在存储用于先前描述的外围装置UART A的新选外围装置记录140的情况下，信号列表123可能会包括列表：UART A_RX和UART A_TX。

然后配置检查模块102继续识别配置表110和信道值136，其与任何所识别的信号相关联。这通过把配置表110用为查找表以便确定一组可能的端口和信道值来实现，其可能受到添加新的外围装置的影响。

因而例如对于特定的外围装置，配置检查模块102可以识别由该外围装置所利用的信号当信道值被设置为1时与端口D的表相关联，以及当信道值被设置为2时与端口K的表相关联。

然后配置检查模块102继续识别所识别的信道设置组的所有可能组合。因而在上面例子的情况下，可能会识别三个组合即：把端口D设置到信道1，把端口K设置到信道2或者把端口D设置到信道1并且把端口K设置到信道2。

在已经识别此可能影响的端口设置组之后，配置检查模块102（S10-2）为每个识别的端口设置组合确定与那些端口设置相关联的完整信号列表。使用所识别的组合从配置表110读出这些值。

然后配置检查模块102检查对于每个组合来说，信号列表是否包括与外围装置相关联的信号列表123中的所有条目，为所述外围装置刚好存储新选择的外围装置记录140。如果情况不是这样，那么删除所识别的端口设置组合。

因而依照这种方式配置检查模块102识别一组端口设置组合，所述端口设置组合足以使由新近选择的外围装置所要求的所有信号能够输入和输出。

然后配置检查模块102（S10-3）随后继续处理每个识别的可接受组合。特别地，把与特定组合相关联的端口设置与配置数据库119中

的每个现有条目相比较，在此实施例中所述端口设置都包括一系列值，每个值用于每个端口45-56，其中所述条目可以为相关联的端口采取任何信道设置，或者作为选择包括空值。如果当前正处理的端口配置组合识别配置数据项中的空值或者作为选择把相同的端口设置值识别为由配置数据项为特定端口所识别的端口设置，那么产生并存储新的配置数据项。

在此实施例中，每个新产生的配置数据项包括配置数据项的拷贝，被利用对应于所影响的识别端口的任何空值的端口设置值来处理，所述端口被设置为由当前考虑的组合所识别的相应端口设置。

因而例如,相对于以下存储的配置数据项来处理以下识别的端口设置组合的可能影响组:

D=1

K=2

D = 1并且K = 2

	A	B	C	D	E	F	G	H	I	J	K	L
配置 1	1	_	_	_	_	_	_	_	_	_	1	_
配置 2	_	_	_	_	4	_	_	_	5	_	_	_

其中_表明信道的空值；在所有组合已经被处理之后，可能会存储以下配置数据项:

	A	B	C	D	E	F	G	H	I	J	K	L
配置 1	1	_	_	1	_	_	_	_	_	_	1	_
配置 2	_	_	_	1	4	_	_	_	5	_	_	_
配置 3	_	_	_	_	4	_	_	_	5	_	2	_
配置 4	_	_	_	1	4	_	_	_	5	_	2	_

在配置检查模块102已经处理所有识别影响的端口设置组合之后，所述配置检查模块102然后确定（S10-4）是否已经存储了任何新的配置数据项。如果没有存储新的项，那么这表明添加新的外围装置与先前所选择的外围装置组是不兼容的并且没有可能的端口设

置，所述端口设置可以使所有选择的外围装置组能够接收并输出与那些外围装置相关联的所有信号。然后配置检查模块102注意到此错误。

返回到图8，如果在配置检查模块102处理之后确定没有可能的配置组可以适应所选择的外围装置组（S8-5），那么接口生成器100继续删除新近产生的所选外围装置记录140并且显示用于识别为什么无法把所选择的外围装置添加到当前选择列表的警告。通过利用所产生的新外围装置的影响端口设置列表并且识别哪些其它所选外围装置使所识别的端口将被设置为候选值来产生此警告。

返回到图10，如果产生新的配置数据项，那么使用这些来代替先前存储的配置数据119。然后配置检查模块102为每个确认数据项计算相关联的引脚名称组。在此实施例中，这通过随后获取与端口45-46相关联的每个引脚并且利用配置表识别该引脚的引脚标记，所述配置表使用由配置数据所识别的端口设置来与该引脚和该引脚的引脚标记有关。

当已经产生完整的引脚名称列表时，检查每组引脚标记，所述标记对应于总输入/输出标记还是对应于一个外围装置的信号列表123中的信号，其中所述外围装置具有对应于任何所选外围装置记录140的外围装置名称141的外围装置名称122。如果情况不是这样，那么使用所存储的默认引脚名称116把引脚的标记复位为该引脚的默认标记。最后，继而根据在与每个配置数据项相关联的引脚名称列表中所包括的通用输入和输出的数目来分类所述配置数据项。

一旦已经产生引脚标记并且依照所包括的通用输入和输出的数目增加的次序来排序配置数据119，那么配置检查模块102（S10-7）处理每个产生的引脚名称列表来识别所述配置数据是否都包括任何信号的任何重复表示，其中所述任何信号被外围装置数据库108中的状态数据125识别为输入信号。如果情况是这样，那么配置检查模块102利用优选的输入列表112来把除优选输入之外的任何重复输入的引脚标记设置为如配置模块102的默认引脚名称116所表示的那些引脚的默认引脚名称。

最后，配置检查模块102通过利用引脚标记156来重新标记所述引脚的表示并且通过经由线路把用于表示所选外围装置的框连接到与

该所选外围装置的信号相关联的引脚，来使（S10-8）接口产生模块100更新用户接口显示145，其中所述引脚标记156对应于与最大通用输入和输出数目相关联的配置数据项的引脚标记。

图11是在已经使用用户接口145选择了多个外围装置之后用户接口显示145的示例性例图。把图11与图9相比较，可以看出在主窗口152内所示出的是九个正方块，每个正方块与先前所选的不同外围装置相关联。还把图9和11相比较，可以看出已经更新了引脚标记156以便反映当启用所图解的外围装置时这些引脚的目的变化。最后，比较图9和11中的配置窗口160，配置数据162已经被改变以便对应于当前所选配置数据项的端口设置，其用于表明对于那些端口来说已经用默认值118来代替空值并且所计算的总输入输出线路163的数目也已经被更新为具有作为部分配置按钮164组显示的可用配置的数目。

返回到图8，如果接口生成器100确定（S8-2）用户没有在外围装置窗口150中选择外围装置列表151中的条目之一，那么接口生成器100进行检查（S8-7）看看用户是否已经在主窗口152内选择了用于表示所选外围装置的一个框167。

如果情况是这样，那么接口生成器100检查是否已经利用鼠标左或右键选择了框167。如果已经利用鼠标左键同时按下该键选择了框167，那么接口生成器100继续利用指针165的位置来改变所选外围装置的位置坐标144。

当放开左键时，接口生成器100检查（S8-10）指针165的坐标144是否识别了在主窗口152的区域之外的位置。如果情况不是这样，那么接口生成器通过重新定位用于表示所选外围装置的框167以便其以指针165的坐标为中心，来更新（S8-11）用户接口显示。

相反地，如果接口生成器100确定（S8-10）指针165的坐标识别了在主窗口152区域之外的位置，那么这被视为表明用户想要从当前被示为在微控制器8内活动的外围装置组中移除所选外围装置。因此接口生成器100从所选外围装置数据库120内删除所选外围装置的所选外围装置记录140，并且通过删除用于表示所选外围装置的框167来更新主窗口152内的显示。然后接口生成器100调用配置检查模块102来更新配置数据119以便说明删除所选外围装置并且据此重新标

记出现在显示145中的引脚152，现在将参考图12来进行描述。

参照由配置检查模块102执行的处理的流程图图12，此时所选外围装置数据库120已经在其中存储了一组所选外围装置记录140，所述所选外围装置记录140表示彼此兼容的外围装置的选择。即存在至少一组端口配置，其使所有所选外围装置能够接收并输出所有它们所要求的信号。

为了产生一组新的配置数据119，配置检查模块102最初（S12-1）选择外围装置的第一外围装置记录140并且继续依照先前所描述的相同方式来识别（S12-2）用于输出并接收与由第一外围装置记录所识别的外围装置相关的信号以便相对于图10中的步骤（S10-1）更新配置数据。

然后配置检查模块102通过识别（S12-3）哪些所识别的影响端口设置的可能组合使第一外围装置的所有输入和输出能够被接收并输出。然后配置检查模块102为每个端口设置组合产生并存储配置数据项119，为此可以接收并输出所选外围装置的一组完整的输入和输出。这些产生的配置数据项119包括配置数据119，其中端口设置为所识别的组合识别了端口设置并且所有其余的端口设置由空值来表示。

然后配置检查模块102（S12-4）检查是否已经到达了所选外围装置数据库120内的最终外围装置记录140。如果情况不是这样，那么处理（S12-5）下一所选外围装置记录140继而识别（S12-6）该外围装置的一组影响的端口设置。然后该外围装置受影响的端口设置的可能组合被过滤（S12-7）以便除去那些不能够使完整的信号组被当前所考虑的外围装置接收并输出的组合。所识别的这组可能组合用于通过依照与先前已经相对于图10中的步骤（S10-3）所描述的相同方式确定哪些组合与当前存储的配置数据项兼容，来产生另外的配置数据项（S12-8）。

对于每个连续的所选外围装置记录140重复此过程（S12-4－S12-8）直到到达所选外围装置数据库120内最终的外围装置记录140。

此时配置模块22已经在其中存储了配置数据119，所述配置数据119识别了所有可能的端口配置使当前所选外围装置组所要求的所

有信号能够被输入或输出。依照与先前已经描述的相同方式，相对于步骤（S10-6－S10-8），然后配置检查模块102（S12-9）确定所存储的配置数据119的每个项的引脚名称列表继而按照存在于每个引脚名称列表中的通用输入和输出数目的组合来排序。然后配置检查模块102（S12-10）检查每个所产生的引脚名称列表来识别任何所选外围装置的任何输入信号的任何重复表示并且没到检测到此时配置检查模块102利用优选的输入列表112来识别哪个输入是优选的输入并且把任何其余重复输入的引脚名称修改为那些引脚的默认引脚名称116。

最后，配置检查模块102使（S12-11）接口生成器100使用配置数据和与最大数目的通用输入和输出相关联的配置数据项119的引脚名称组来在主窗口152中重新标记156所述引脚表示，并且更新配置窗口160中的端口配置数据162。然后接口生成器100使用于把用来表示外围装置167的每个框连接到与那些外围装置相关联的各自输入和输出的线路表示被更新使得图解当前所选择的配置，其中所述那些外围装置如由引脚数据156标记。

因而依照这种方式通过从外围装置列表151中选择外围装置组并且从主窗口152内除去外围装置，用户能够使用户所想要的外围装置组的识别在微控制器8上启用。同时对于每个选择，配置检查模块102确定用于识别所要求端口设置的一组配置数据项以便使与当前所选外围装置组相关联的信号能够被输出，并且识别了何时进行了不兼容的外围装置选择并且为什么这种选择的外围装置组是不兼容的。

此外如进一步论述，配置模块22还使微控制器8的图示154连同一组引脚名称156一起显示，其用于表明当选择特定的端口设置组时对于给定外围装置组来说微控制器的每个引脚目的，其中所述端口设置组与所识别的外围装置选择相兼容。

返回到图8，如果接口产生模块100确定用户没有选择用于表示所选外围装置（S8-7）的一个框167，那么接口生成器100（S8-13）检查是否已经选择了微控制器154的表示区域。

如果情况是这样，那么在此实施例中这用于使用户能够输入用来识别存在于微控制器8中的时钟模块36的设置。

更具体地说，最初用户接口生成器100对应于时钟模块来识别外

围装置数据库108以及所选外围装置数据库120内的记录（S8-14）继而使接口生成器100产生显示菜单（S8-15）以便使用户能够改变当前的时钟设置。

图13是用于改变微控制器8上时钟36的设置的用户接口显示菜单的示例性例图。如图13所示，菜单包括对应于来自外围装置数据库108的所选外围装置记录121的寄存器名称125的寄存器名称列表200，接下来显示一系列当前值202，所述当前值202包括设置解释134，其是对应于所选外围装置的变量值142、143和寄存器值的解释。

当在鼠标3的控制下使用指针165选择单个寄存器时，那么下拉菜单204呈现识别用于寄存器的所有可能的设置134，或者作为选择显示使用户能够输入变量值127的框。然后用户可以通过输入用于从菜单204选择条目的数据或者经由键盘4输入数据来改变当前设置。

如果在改变所述值之后用户对那些设置表示满意，那么用户使用指针165来选择OK按钮206。然后这使接口生成器100更新所选外围装置的寄存器值142和变量值143以便对应于所选择的设置。作为选择，用户可以选择取消按钮207，在这种情况下并不更新所选外围装置的寄存器值142。在选择了OK按钮206或取消按钮207之后，接口生成器100恢复成显示主显示145。

依照类似方式，当接口生成器102确定（S8-8）已经使用鼠标右键选择了被表示为现有外围装置的框167时，产生类似的显示菜单以便输入并改变与所选外围装置相关联的设置。然后依照相同方式更新用于所选外围装置的所选外围装置记录140的寄存器值142和变量值143，其中时钟的变量已经被描述为正在更新。类似地，在已经进行任何改变并且选择OK按钮之后，更新相应的所选外围装置记录140的寄存器值142和变量值143（S8-16）。

因而依照这种方式用户能够改变与单个外围装置记录140相关联的默认设置并且因而识别所选外围装置所要求存储在寄存器65-1-65-N；88；92、94中的值。

返回到图8，如果接口生成器100确定（S8-13）用户没有选择对应于芯片154的区域，那么接口生成器100（S8-17）检查是否已经选择了用户接口屏幕上的任何配置控制164。

在此实施例中，配置控制164包括用于选择由配置模块所存储的第一、前一个、下一个或最后配置数据项119的一系列按钮。当选择一个按钮时，利用由所选按钮相对于当前正用来产生显示的配置数据项所识别的配置数据项来产生（S8-19）新的用户接口显示145。这通过使用与新近选择的配置数据项相关联的引脚标记来重新标记引脚156，更新配置窗口160并且重新连接用于表示外围装置的块167来实现。

因而通过在配置控制面板164上选择不同的按钮，用户能够在当前所选外围装置组的不同可接受的配置之间进行卷动。这样做时，当显示上的引脚标记对应于实际微控制器8上物理引脚的物理位置时，用户能够根据不同的端口设置看出其中存在外围装置的各个不同输入和输出的物理位置。因而依照这种方式可以回顾可能的端口设置并且可以对于特定的所选外围装置组识别优选的端口设置组。

由于只对于适于接收并输出当前所选外围装置组的信号的端口设置才存储配置数据，所以用户只需要回顾有限数目个可能的端口设置组合。在本实施例中通过利用任何端口设置的默认值来进一步减少此有限的选择，其中所述端口设置对应于配置数据项中的空值。因而用户不必回顾任何重复组合，其改变的端口设置对当前所选外围装置组的输入或输出没有影响。

在检查（S8-17）是否已经选择了任何配置按钮164之后，接口生成器100然后检查用户是否已经输入了用于构建配置程序32的指令来配置微控制器8，所述配置对应于当前所选设置和端口配置。如果情况不是这样，那么接口生成器100再次检查用户是否已经利用指针165选择了在外围装置窗口150内所显示的任何外围装置（S8-2）。

当接口生成器100确定（S8-20）已经输入用于产生配置程序32的指令时，然后调用输出模块106。输出模块106最初（S8-21）检查用于单个外围装置和时钟的寄存器设置是否是可兼容的。这通过利用规则数据库114内的规则和所选外围装置数据库120内的所选外围装置记录140的寄存器值142和变量值143来实现。存在某种不兼容性的典型原因可能是为外围装置所选择的波特率与可能的时钟信号不兼容，所述时钟信号可以是根据时钟36的设置而产生的。

如果输出模块106确定任何寄存器设置是不兼容的，那么输出模

块106显示（S8-22）用于识别设置不兼容性的错误消息（S8-22）以便用户可以据此调整这些设置。

如果输出模块106确定（S8-21）在不同外围装置和时钟的设置之间没有不匹配，那么输出模块106调用代码生成模块104来产生配置程序32和所选外围装置数据库140内的外围装置记录140的寄存器值142和变量值143以便设置微控制器8的寄存器62、64、65-1；65-N内的相应值，使得依照对应于由用户接口所图解的方式来配置微控制器8，其中所述代码生成模块104利用当前所选择的配置项来产生显示。

更具体地说，根据在所选外围装置数据库120中所存储的所选外围装置记录140的寄存器值142和变量值143来确定所选外围装置的每个寄存器65-1-65-N寄存器值。类似地，根据与时钟相关联的所选外围装置记录140的寄存器值142和变量值143来确定与时钟36相关联的寄存器64的寄存器值。最后，根据当前用来产生用户接口显示145的配置数据项来确定与每个端口45-46相关联的寄存器62的寄存器值，其中任何空值被复位为由所存储的默认配置数据118识别的默认值。

因此代码生成模块104利用所存储的所选外围装置记录140和当前选择的配置数据项119来产生配置程序32以便使微控制器8把所识别的寄存器值存储在微控制器8的适当寄存器62、64、65-1-N－65-N中。

然后把所产生的此配置程序32存储在计算机的存储器11中。然后配置程序32可以经由接口6被下载到电路板7上的微控制器8中，其中所述接口6依照对应于所选配置数据的方式来配置微控制器8，以便可以利用微控制器8的所选寄存器设置来测试其它程序数据30来确定这些设置和所编译程序数据30的组合是否会使微控制器8执行所想要的功能。

在此实施例中，除产生用于使微控制器采用一组识别设置的配置程序32之外，输出模块106还被安排成以微控制器8的设计图示的形式来产生用于打印微控制器8表示的数据，所述设计图示把微控制器8的输入和输出示为由所选配置引脚标记来识别，其中依照与所选外围装置组相关联的信号列表123来分组所述引脚标记，以及被示为微

控制器8或印刷电路板的物理表示，其中依照与用户接口显示145中的图示154类似的方式按照引脚的物理位置来标记所述引脚。

此外在此实施例中，输出模块106还被安排来产生用于以它们可以被输出到软件包的形式来表示这些例图，例如以便设计供微控制器8使用的印刷电路板并且对将存在于电路板7上的配置微控制器8及其它芯片进行电兼容性检查，其中所述微控制器8已经按照配置数据和寄存器设置的确定来设置。

第二实施例

现在将参考图14-21描述本发明的第二实施例。

在第一实施例中描述了用于执行内嵌测试微控制器的计算机系统，其中测试电路板7上的单个微控制器8。常常微控制器8被附于在相同电路板上所提供的多个外部外围装置上。在此实施例中，将描述其中可以为微控制器的配置编程并且其中可以另外产生用于外部外围装置的配置数据的系统。

依照本发明第二实施例参照用于执行内嵌测试的计算机系统的例图图14，图14的计算机系统的大部分与相对于第一实施例所描述的系统相同，并且已经用相同的附图标记来识别先前出现在第一实施例的系统中的元件。然而与第一实施例对比，在此实施例中代替测试其上单独附有微控制器8的电路板7，在此实施例中，所测试的电路板7其上提供有微控制器8和多个外部外围装置芯片300-1 - 300-N。在芯片上所提供的典型外部外围装置可以是Zigbee芯片、以太网连接和一个或多个USB端口。

当将为包括微控制器和多个外部外围装置300-1 - 300-N的电路板编程时，除微控制器8的内部外围装置的配置之外，还要求在微控制器8的存储器内所存储的任何程序被用来依照与处理内部外围装置相兼容的方式来编程外部外围装置300-1 - 300-N的寄存器。因而例如在Zigbee芯片的情况下，要求UART内部外围装置在微控制器内可用，并且必须使Zigbee芯片和内部UART的配置在波特率、奇偶校验等方面相匹配。为此，依照本发明的此实施例，代替先前实施例的配置模块22，提供修改的配置模块302，用于便于为微控制器8编程以便选择适当的内部外围装置并且确定用于所选内部外围装置以

及所连接的外部外围装置301-300-N的适当配置数据，现在将要进行描述。

修改的配置模块

图15是依照本发明的此实施例的修改配置模块302的子组件的示意性框图。

修改配置模块302基本上与先前实施例的配置模块22相同，不过其另外还包括外部外围装置数据库304，用于存储用来定义由交互式设计环境20所支持的外部外围装置的信号和潜在设置的数据；还包括外部外围装置存储装置306，可操作来存储用户为与所述外部外围装置相关联的寄存器所选择的值，和选择模块308，用于处理与外部外围装置相关的用户输入，如稍后将进一步地详细描述。

外部外围装置数据库

图16是在外部外围装置数据库304内所存储的外部外围装置记录310的示意性框图。对于交互式设计环境20被安排来支持的每个多个外部外围装置300-1 - 300-N来说，在外部外围装置数据库304内存储外部外围装置记录310。外部外围装置记录310的结构类似于在外围装置数据库108内所存储的外围装置记录212的结构。依照与先前描述的外围装置记录121相同的方式，在此实施例中，外部外围装置记录310包括用于识别外围装置名称的名称数据122、寄存器名称列表125、一组寄存器记录126和一组变量值127，其中寄存器记录126和变量值与先前已经描述的完全相同。

然而，代替具有信号列表123和状态数据124，在此实施例中，外部外围装置记录310均包括优选的连接数据312，用于为外部外围装置识别优选在微控制器8内可用的内部外围装置以便使所述微控制器8能够与外部外围装置通信；连接要求列表313，用于为所述外部外围装置识别寄存器值和诸如波特率之类的变量，其被要求在内部外围装置和外部外围装置之间匹配；和一组虚拟要求314用于如果无法使适当的内部外围装置通信可用，那么识别用于在微控制器和外部外围装置之间直接通信的一组输入和输出引脚。

因而例如在外部外围装置包括Zigbee芯片的情况下，所述Zigbee

芯片通常经由UART连接与微控制器通信，用于表示Zigbee芯片的外部外围装置记录310的优选连接数据312可以包括：

优选的连接：UART A，UART B

这种条目可能会表明每当在交互式设计环境20内选择Zigbee芯片时，优选经由内部外围装置UART A把所选择的第一Zigbee芯片连接到微控制器8。如果UART A将不可用并且将选择Zigbee芯片来用于连接，那么可能优选经由内部外围装置UART B来连接。最后如果UART A和UART B都不可用，那么交互式设计环境20将试图配置微控制器8以便仿真另外的UART连接，如稍后将进一步描述。

外部外围装置存储装置

图17是在外部外围装置存储装置306内所存储的多个所选外部外围装置记录340的示意性框图。当首先调用交互式的设计环境时，外部外围装置存储装置306不存储任何数据。如稍后所述由用户经由用户接口选择外部外围装置的表示时，把新的所选外部外围装置记录340存储在外部外围装置存储装置306内。

外部外围装置存储装置306中所存储的所选外部外围装置记录340类似于在用于存储内部所选外围装置记录140的所选外围装置数据库120中所存储的外围装置记录140。在此实施例中，所选外部外围装置记录340均包括外围装置名称341，其对应于来自外部外围装置数据库304的外部外围装置记录310的所选外围装置的外围装置名称122；一组寄存器值342和变量值343，是当前与由所述记录所识别的外部外围装置相关联的寄存器和变量的一组设置；位置数据344，是一对x，y坐标，用于识别在由接口生成器100所产生的屏幕显示内应当在哪里示出用来显示外部外围装置名称341的框以便表示存在所识别的外部外围装置；和连接数据345，用于识别内部外围装置或引脚连接，所述引脚连接用于把由所述记录所表示的外部外围装置连接到微控制器8，如稍后将要详细描述。

使用修改配置模块的处理

当首先调用修改配置模块302时，接口生成器100使初始用户接口屏幕显示在显示屏2上。图18是依照本发明此实施例的初始用户接口

显示145的示例性例图。用户接口显示几乎与最初在第一实施例中所产生的用户接口显示完全相同，只是在用户接口的左手边上显示的外围装置窗口150被分成两个部分：用于显示一系列图标151的第一窗口150，所述每个图标151与微控制器8内存在的每个内部外围装置40-1－40-N相关联，其中可以为所述微控制器8产生配置数据；和用于显示与每个外部外围装置相关联的一系列图标的第二窗口350，所述交互式设计环境20可操作来支持所述外部外围装置，为此把外部外围装置记录310存储在外部外围装置数据库304内。

如在第一实施例的情况下，用户能够在鼠标3的控制下使用指针165来选择用户接口显示屏145内的各个区域，这使配置模块22改变所显示的用户接口显示145和在所选外围装置数据库120内所存储的数据、配置数据119以及在此实施例中在外部外围装置存储装置306内所存储的数据。

一旦已经显示初始屏幕，那么在此实施例中的处理类似于在第一实施例中所进行的处理。然而当接口生成器100检查用户是否已经在外围装置窗口150或外部外围装置窗口350中选择外围装置的任何表示时，修改先前相对于图8所描述的处理，现在将参考图19来进行描述。

在确定已经选择用于表示内部或外部外围装置的一个图标之后，接口生成器100确定（S19-1）所选外围装置是从第一窗口151中所选择的内部外围装置还是从窗口350中所选择的外部外围装置。如果确定所选择的图标表示内部外围装置，那么所执行的处理依照与先前相对于图8的步骤S8-3到S8-6所描述的相同方式来继续进行。

如果确定已经选择外部外围装置的表示，那么接口产生模块100（S19-2）显示用于表示新近选择的外部外围装置的块并且创建新选择的外部外围装置记录340，所述外部外围装置记录340被存储在外部外围装置存储装置306中。在新近产生的所选外部外围装置记录340中，对应于从外部外围装置列表350所选择的外围装置，外围装置名称341被设置为外部外围装置记录310的外围装置名称122。然后寄存器值342和变量值343被设置为所选外围装置的默认值，并且x，y位置被设置为跟踪指针165直到用户放开所述按钮或鼠标。

当放开鼠标3上的按钮时，接口生成器100调用选择模块308，于

是所述选择模块308利用具有外围装置名称122的外部外围装置记录310的优选连接数据312来选择适当的内部外围装置以便连接到所选的外部外围装置，其中所述外围装置名称122对应于新近产生的所选外部外围装置记录340的外围装置名称341。

更具体地说，选择模块308最初识别在外部外围装置记录310的优选连接列表312中出现的第一优选连接，所述外部外围装置记录310具有对应于新近产生的所选外部外围装置记录340的外围装置名称341的外围装置名称122。然后选择模块308检查（S19-4）外围装置记录140是否被存储在具有外围装置名称122的所选外围装置数据库120内，所述外围装置名称122对应于在正处理的优选连接列表312中的条目。如果情况是这样，那么这表明所识别的内部外围装置已经被分配用于其它用途并且选择模块308于是将（S19-5）检查以便确定是否已经到达在优选的连接列表312上的最后条目。如果情况不是这样，那么选择下一条目并且选择模块308确定是否在所选外围装置数据库120内的任何条目对应于该下一条目。

如果选择模块308确定（S19-4）所考虑的优选连接尚未被分配给用户，那么选择模块308继续（S19-6）依照与先前相对于第一实施例所描述的完全相同的方式，来确定对应于所识别的内部外围装置添加新的外围装置记录是否会导致与预先存在的端口配置出现冲突。如果确定所识别的内部外围装置无法与由所选外围装置数据库120中的记录所识别的内部外围装置共存，那么选择模块308（S19-5）检查是否已经到达优选的连接列表312中的最终外围装置并且如果情况不是这样，那么考虑选择（S19-4）所述列表中的下一可能的外围装置。

然而如果选择模块308确定可以允许激活所识别的内部外围装置并且不会导致冲突，那么选择模块308使接口产生模块100把所识别的内部外围装置的表示添加到屏幕显示并且把用于识别所识别的内部外围装置的所选外围装置记录140存储在所选外围装置数据库120中。

当存储新的所选外围装置记录140时，使新记录140的寄存器值142和变量值143与新选择的外部外围装置记录340中的相应值相匹配，其中所述新记录140由新选择的外部外围装置记录340的连接要

求313识别。用于识别新近存储的外围装置记录140的外围装置名称121的连接数据345也被存储在新选择的外部外围装置记录340内。最后，接口产生模块100更新用于经由引脚示出在新选择的内部和外部外围装置之间的连接，当前配置数据119表明内部外围装置接收和输出信号。

图20是在用户已经选择对应于Zigbee芯片的图标之后用户接口的示例性例图。在图20的例图中，示出了用于表示Zigbee芯片框355，其与用于表示适当内部外围装置的框360相关联，在此例子中是UART连接器UART A。

一旦已经把外部外围装置340和相关联的内部外围装置140的记录存储在所选外围装置数据库120和外部外围装置存储装置306内，那么就依照与先前相对于第一实施例所描述的类似方式来继续接口产生模块100的处理。

如果用户选择用于表示外围装置的框355和360之一并且使指针165拖动到主窗口152外面，那么这表明所关注的外围装置将被删除。如果所关注的框表示外部外围装置，那么在此实施例中，这还被视为用于把与该外围装置相连接的内部外围装置删除的指令，其由在循环外部外围装置记录中所存储的数据识别为连接数据345。类似地当删除内部外围装置时，接口产生模块继续检查是否在外部外围装置存储装置306中所存储的任何外部外围装置记录340具有用于识别所删除外围装置的连接数据345，并且如果是的话那么在用户接口中删除该记录和该记录的表示。

如果用户依照与先前描述的类似方式使用鼠标右击来选择外部外围装置355或相关联的内部外围装置360的表示，那么可以更新与该外围装置相关联的值中的寄存器条目。然而，为了确保对于内部和外部外围装置都存储了可兼容的配置数据，在此实施例中当更新外围装置的寄存器值或变量值时，接口产生模块100继续利用连接数据345和选择要求313来把相应值输入到相关联的内部或外部外围装置记录的寄存器值和变量值中以便确定哪些值和寄存器要求更新。

返回到图19，如果（S19-5）选择模块308发现在连接列表312中所有优选的外围装置由在所选外围装置数据库120内所存储的外围装置记录140识别或者作为选择如果选择任何可用的内部外围装置

的话不可以产生适当的配置数据，那么在此实施例中，选择模块308继续确定（S19-8）微控制器本身是否可以仿真所要求的外围装置并且在任何可用的引脚上输出信号。

如先前所声明，外部外围装置记录310的虚拟要求列表314识别由微控制器所要求的输入和输出来仿真所要求的虚拟外围装置。在已经确定不可以利用任何一个内置内部外围装置之后，在此实施例中，选择模块308检查当前的引脚名称列表来识别与总输入输出相关联的引脚。在此实施例中，选择模块308然后识别可用的总输入输出引脚，所述总输入输出引脚在与新近产生的外围装置记录340的数据344的当前位置距离最近的位置。一旦这些引脚已经被识别，那么接口产生模块100使框包括在用于表示特殊类型的虚拟（即软件模拟）板上外围装置的微芯片154的边界内，并且把用于识别所述引脚的数据在外部外围装置存储装置306中作为连接数据345存储在新近存储的所选外部外围装置记录340中。

图21是用户接口的说明性例子，其中在板上外围装置UARTA和UARTB已经被选择并包括在所选接口数据库120内并且在用户接口内由框370和372表示之后，用户在用户接口350中从外部外围装置列表中选择Zigbee芯片，其要求与UART外围装置相对接。由于在此特定的例子中没有另外的UART可用，所以把用于表示虚拟UART 375的框包括在用户接口中并且虚拟UART框375被示出为连接到用于表示外部Zigbee芯片的框355。

如在第一实施例的情况下，当从用户接口中选择构建时（图8中的步骤8 - 20），调用输出模块106。如在第一实施例中，输出模块106最初检查（S8-21）单个外围装置和时钟的寄存器设置是否是可兼容的并且如果情况是这样，那么继续产生（S8-23）配置程序32以便设置微控制器的寄存器内的值，使得依照对应于由用户接口所图解的方式来配置微控制器。另外在此实施例中，配置程序32还被安排来为与外部外围装置相关联的寄存器的设置编程以及产生用于仿真外围装置的软件代码并且经由所识别引脚输入和输出信号，其中在用户接口内表示虚拟外围装置，所述外部外围装置由在外部外围装置存储装置306中所存储的所选外部外围装置记录340的寄存器值342和变量值343表示。

进一步的实施例和修正

尽管在上述实施例中，已经描述了这样一种系统，其中配置模块22被安排来存储用于定义单个微控制器8表示的数据并且产生特别用于该微控制器8的配置程序32，然而应当理解在其它实施例中，可以提供配置模块22以产生用于各种微控制器8的配置程序32。

在这种系统中，配置模块22可能会需要为每个可能的微控制器8存储配置表110和外围装置数据库108。最初用户可以输入数据并且识别优选的微控制器8，将为所述微控制器8产生配置数据119。然后配置模块的处理22可以依照与先前所描述的相同方法来继续进行。

这种系统的优点在于当配置模块22确定无法在特定的微控制器8上利用特定的外围装置选择时，配置模块22可以利用被存储来用于表示其它微控制器8的数据来确定任何那些其它微控制器是否可以支持所想要的外围装置组。然后可以提示用户改变他们对微控制器8的选择并且配置模块22可以继续为候选微控制器8产生配置程序32。

尽管在上述实施例中，配置模块22已经被描述为根据通用输入和输出的数目来排序配置数据项119，其被引脚标记认为是特殊配置，然而可以利用其它用于排序配置数据项的手段。

因而例如在候选实施例中，可以对于每个配置数据项来确定并计算用于表示在主窗口150中识别块167和引脚之间的连接的线路长度。于是将根据所计算的值来排序配置数据项。

这种系统的优点在于通过把用于表示特定外围装置的块167放置在微控制器154的图示上的确定位置，用户可以识别微控制器的区域，其中用户更喜欢使该所选外围装置的输入和输出为可用的。

在上述实施例中，描述了用于产生微控制器8的配置数据寄存器62的系统，其用于控制信号的路由，其中引脚42被分组为多个端口45-56。在上述实施例中，与每个端口相关联的引脚42的数目被描述为四个或八个。

应当理解，与任何特定端口相关联的引脚数目可以潜在地包括任意数目的引脚42。最重要地是，端口可以包括单个引脚而不是一组引脚。

在其中单个选择器60和寄存器62与单个引脚42相关联的系统中，提供了用于改变用来转送哪些信号的那些引脚位置的高度灵活的系统。所描述的用于产生配置数据的系统于是使这种高度灵活的微控制器能够被适当地编程，而不管所增加的寄存器数目会依照常规方式来确定特别复杂任务的配置数据。

尽管在上述实施例中，已经描述了产生要存储在微控制器8的内部程序存储装置38内的配置程序32，然而应当理解，可以把所产生的用于使微控制器8存储所识别的配置值组的程序存储在电路板7中微控制器8之外的存储器内，而不是存储在内部程序存储装置38内。

尽管在上述实施例中寄存器64、65-1－65-N的寄存器值被描述为根据存储寄存器和变量值142、143来确定，然而应当理解，可以根据寄存器和变量值142、143而不是这些值的直接拷贝来计算所存储的实际值。因而例如通过输入32 kHz波特率的变量值，用户可以使配置模块22计算需要被存储在寄存器中、用于使得能够获得所想要的波特率的特定值。作为选择，诸如波特率、容差等多个值的用户条目可以被用来计算要存储在微控制器8的寄存器64；65-1；...；65-N中的单个值。

尽管在上述描述中，已经参考用于控制较大设备功能的微控制器8，然而应当理解，本发明适用于包括多个外围装置的任何微芯片的设计和配置，其中通过在微芯片的寄存器内的存储值来控制路由那些外围装置的信号。因此术语微控制器应当被视为包含任何这种可配置的微芯片，所述微芯片包括这种外围装置和寄存器。

尽管参考附图所描述的本发明的实施例包括计算机设备和在计算机设备中所执行的过程，然而本发明还扩展到计算机程序，特别是适于实施本发明的载体上或中的计算机程序。所述程序可以是源或对象代码的形式或适于与实现依照本发明过程一起使用的任何其它形式。所述载体可以是能够携带程序的任何实体或设备。

例如，所述载体可以包括存储介质，诸如例如CD ROM或半导体ROM之类的ROM，或例如软盘或硬盘之类的磁记录媒体。此外，所述载体可以是可发射的载体，诸如可以经由电或光缆或用无线电或其它装置传播的电或光信号。

当程序被实现为可以借助电缆或其它设备或装置直接传播的信

号时，所述载体可以由这种电缆或其它设备或装置构成。

作为选择，所述载体可以是其中嵌入有程序的集成电路，适于执行或用于实现相关过程的集成电路。

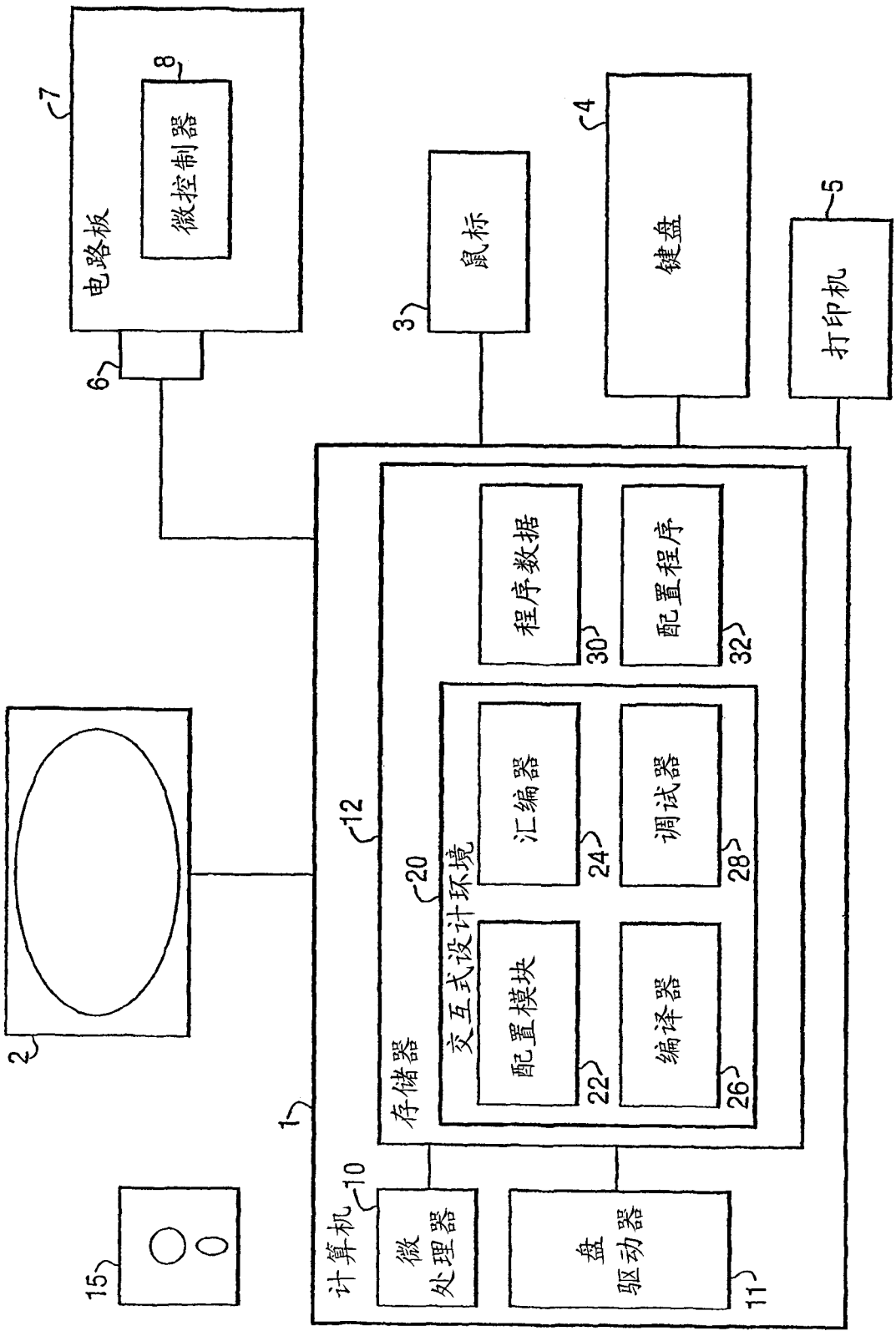

电路板
微控制器
8
7
6
鼠标
3
键盘
4
打印机
5
2
1
计算机
10
微处理器
存储器
12
交互式设计环境
20
配置模块
22
汇编器
24
编译器
26
调试器
28
程序数据
30
配置程序
32
盘驱动器
11
15

图 1

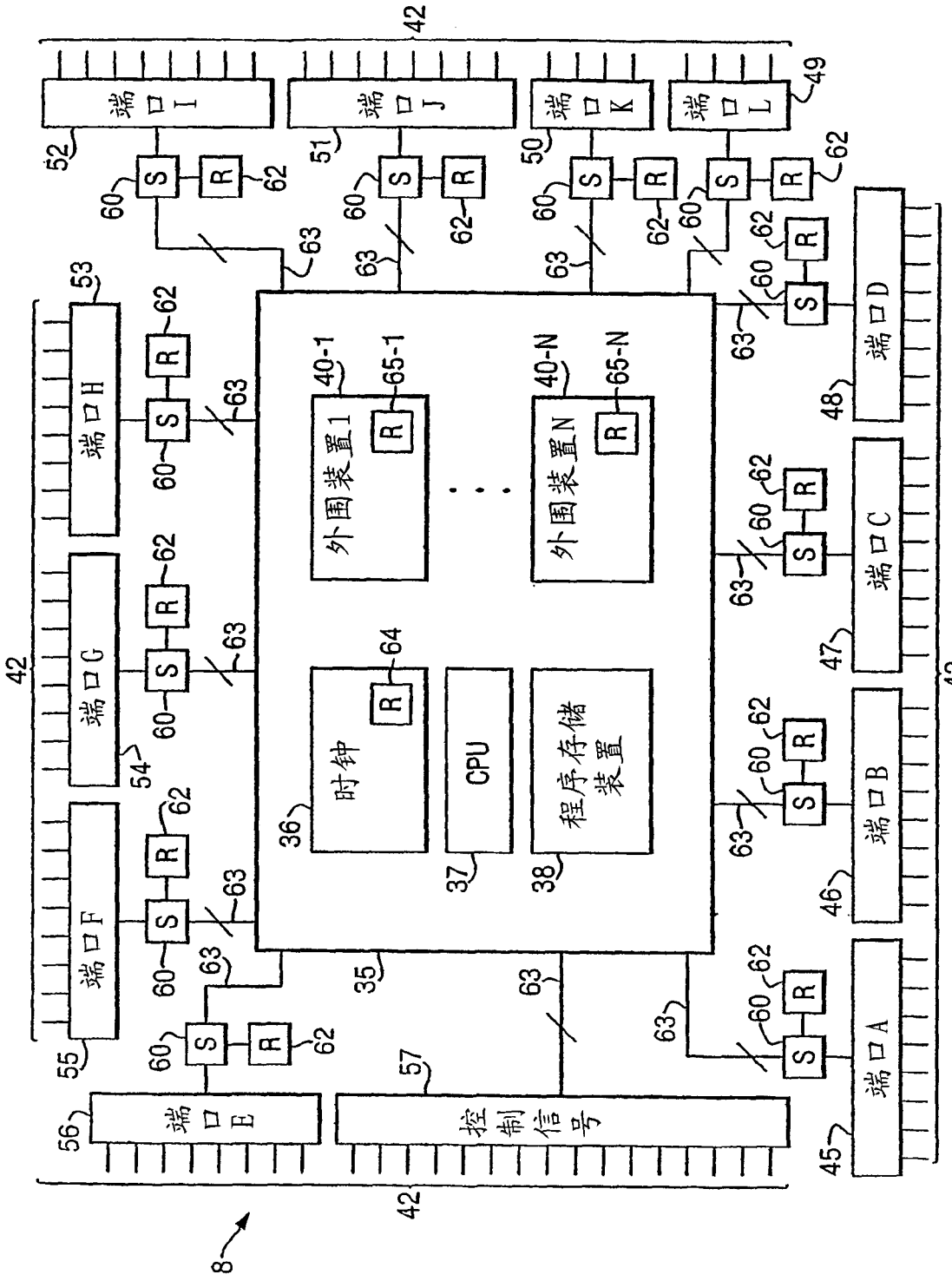

端口I
端口J
端口K
端口L
端口H
端口G
端口F
端口E
端口D
端口C
端口B
端口A
外围装置1
外围装置N
时钟
CPU
程序存储装置
控制信号
S
R

图 2

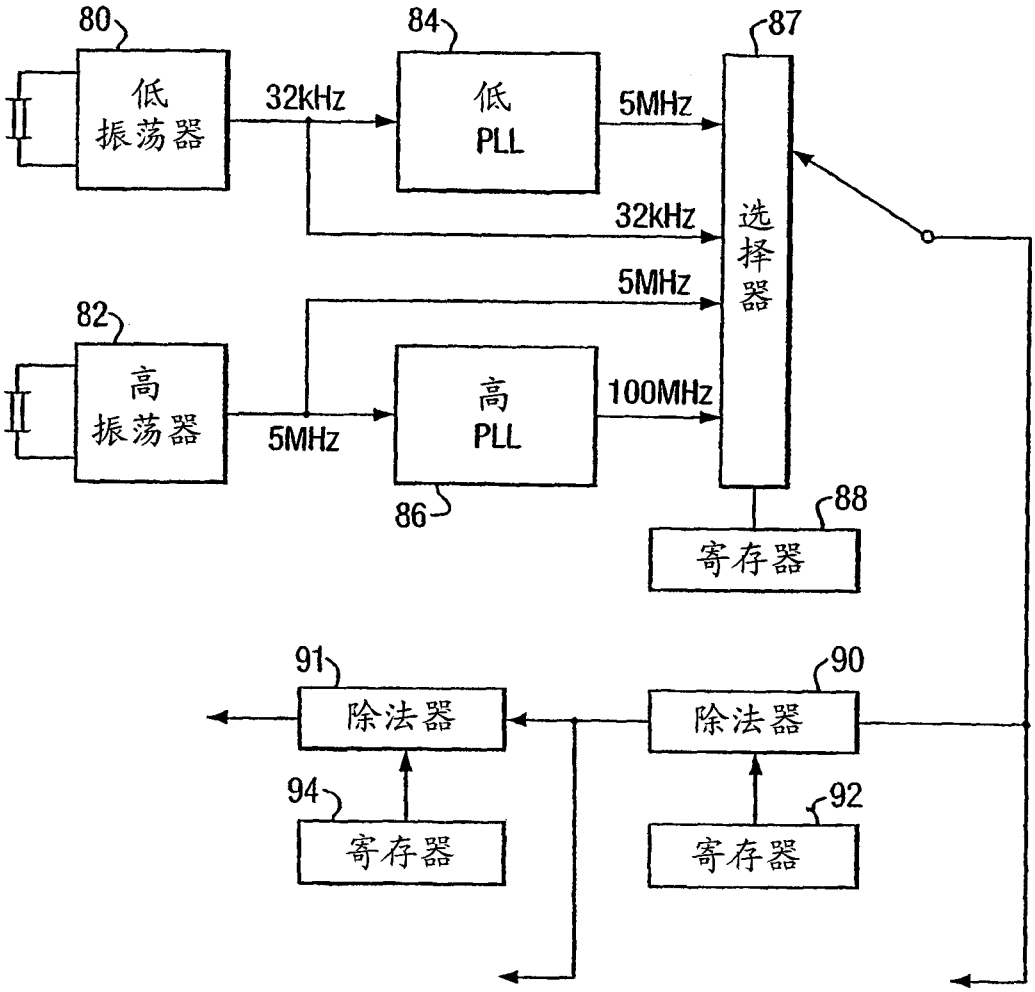

80
低
振荡器
32kHz
84
低
PLL
5MHz
87
选
择
器
32kHz
5MHz
82
高
振荡器
5MHz
高
PLL
100MHz
86
88
寄存器
91
除法器
90
除法器
94
寄存器
92
寄存器

图 3

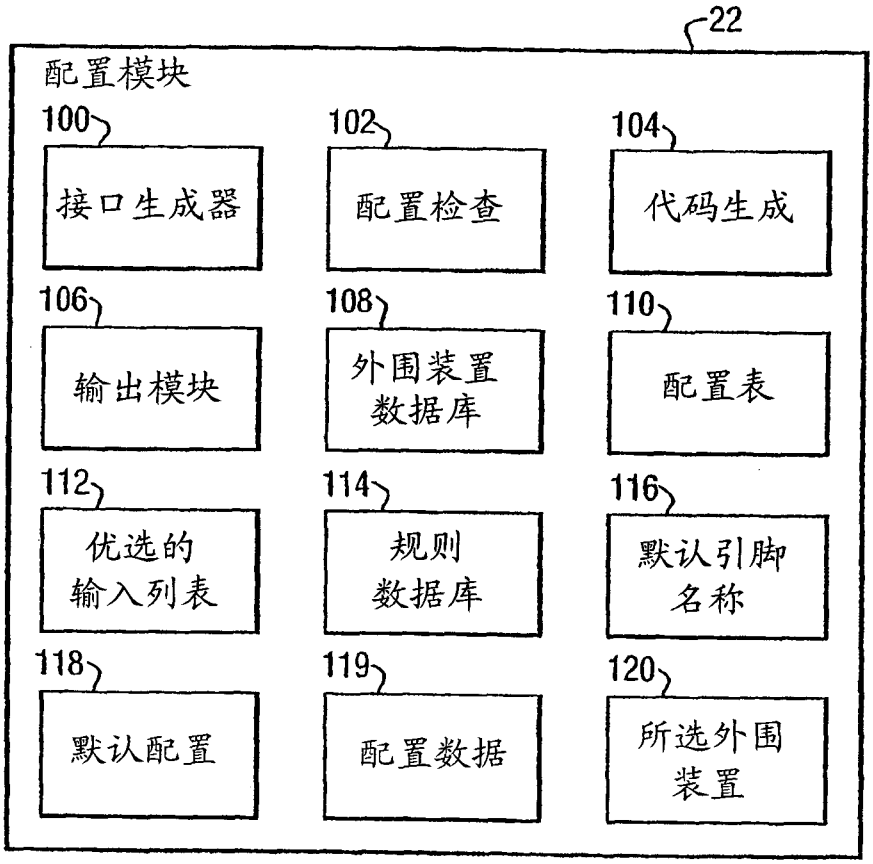

22
配置模块
100
接口生成器
102
配置检查
104
代码生成
106
输出模块
108
外围装置数据库
110
配置表
112
优选的输入列表
114
规则数据库
116
默认引脚名称
118
默认配置
119
配置数据
120
所选外围装置

图 4

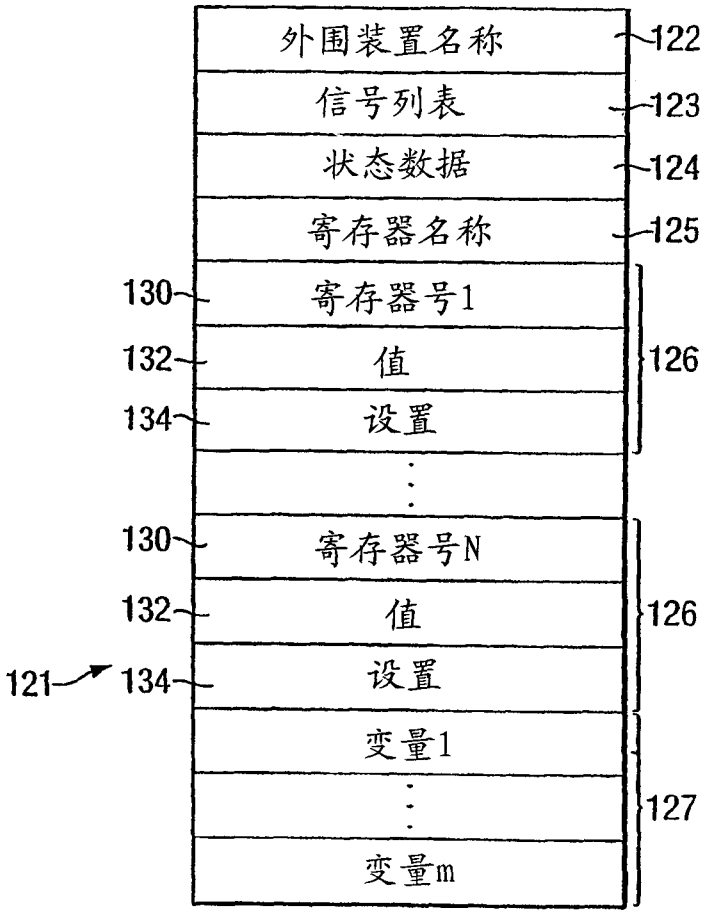

外围装置名称
122
信号列表
123
状态数据
124
寄存器名称
125
130
寄存器号1
132
值
134
设置
126
130
寄存器号N
132
值
121
134
设置
126
变量1
变量m
127

图 5

135 136

引脚号	信道0	信道1	信道2	信道3
62	唤醒	UARTB_RX	PWM1	唤醒
64	STREAM_ACK	UARTB_TX	PWM2	GPIO25
65	STREAM_STS	GPIO21	UARTB_RX	GPIO27
66	STREAM_REQ	GPIO22	UARTB_TX	GPIO12

图 6

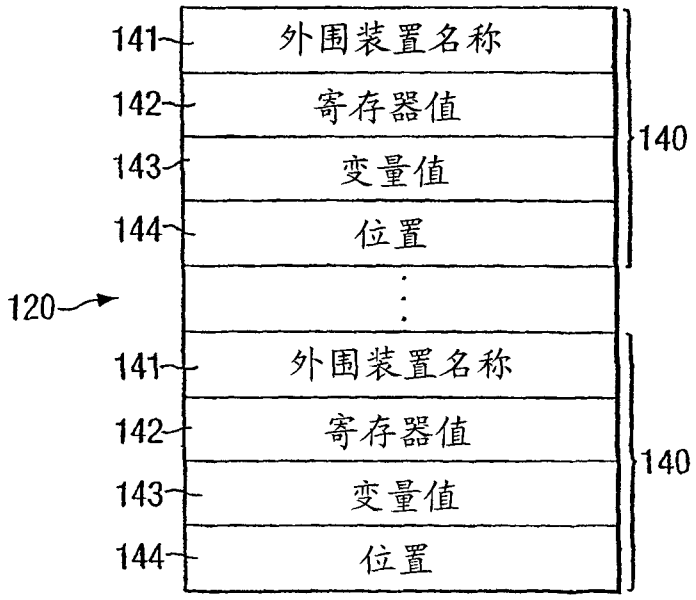


图 7

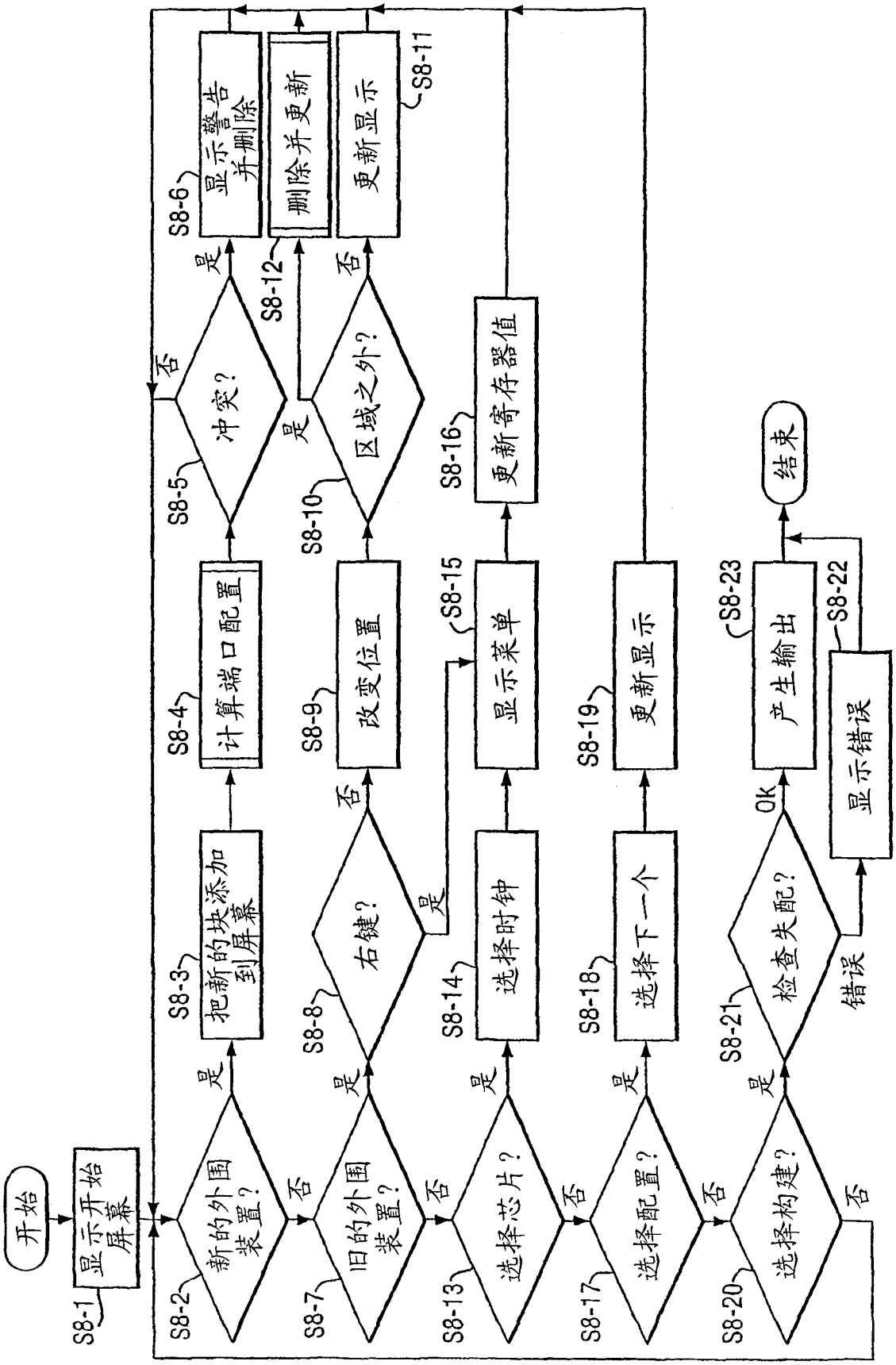

开始
显示开始屏幕
S8-1
新的外围装置?
S8-2
旧的外围装置?
S8-7
选择芯片?
S8-13
选择配置?
S8-17
选择构建?
S8-20
检查失配?
S8-21
产生输出
S8-23
结束
把新的块添加到屏幕
S8-3
计算端口配置
S8-4
冲突?
S8-5
显示警告并删除
S8-6
右键?
S8-8
改变位置
S8-9
区域之外?
S8-10
删除并更新
S8-12
更新显示
S8-11
选择时钟
S8-14
显示菜单
S8-15
更新寄存器值
S8-16
选择下一个
S8-18
更新显示
S8-19
显示错误
S8-22
是
否
Ok
错误

图 8

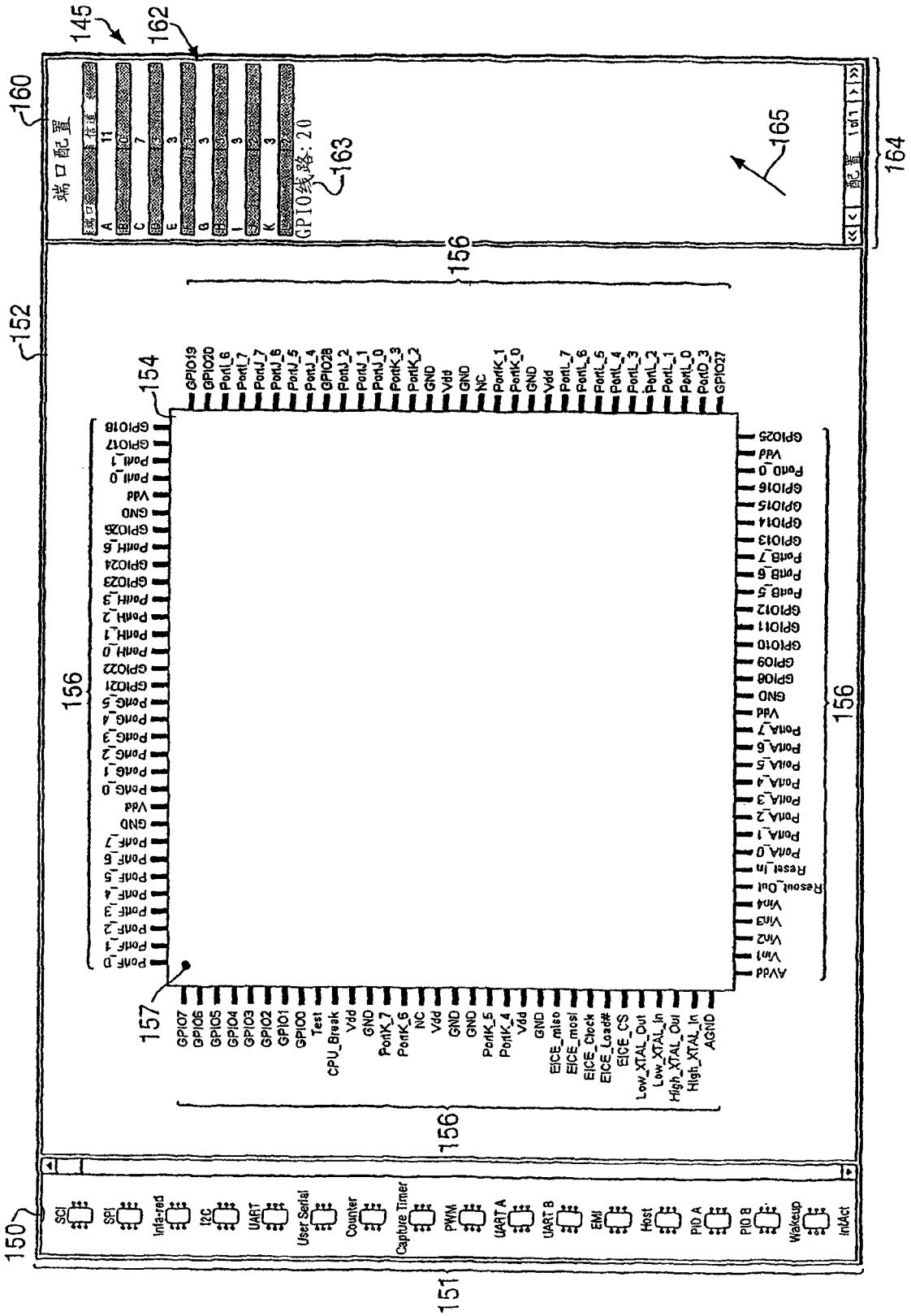

145
162
160
端口配置
A
C
E
G
I
K
GPIO线路: 20
163
165
164
156
152
154
GPIO19
GPIO20
PortL_6
PortL_7
PortJ_7
PortJ_6
PortJ_5
PortJ_4
GPIO28
PortJ_2
PortJ_1
PortJ_0
PortK_3
PortK_2
GND
Vdd
GND
NC
PortK_1
PortK_0
GND
Vdd
PortL_7
PortL_6
PortL_5
PortL_4
PortL_3
PortL_2
PortL_1
PortL_0
PortD_3
GPIO27
GPIO18
GPIO17
PortI_1
PortI_0
Vdd
GND
GPIO26
PortH_5
GPIO24
GPIO23
PortH_3
PortH_2
PortH_1
PortH_0
GPIO22
GPIO21
PortG_5
PortG_4
PortG_3
PortG_2
PortG_1
PortG_0
Vdd
GND
PortF_7
PortF_6
PortF_5
PortF_4
PortF_3
PortF_2
PortF_1
PortF_0
GPIO25
Vdd
PortD_0
GPIO16
GPIO15
GPIO14
GPIO13
PortB_7
PortB_6
PortB_5
GPIO12
GPIO11
GPIO10
GPIO9
GPIO8
GND
Vdd
PortA_7
PortA_6
PortA_5
PortA_4
PortA_3
PortA_2
PortA_1
PortA_0
Reset_In
Reset_Out
Vin4
Vin3
Vin2
Vin1
AVdd
157
GPIO7
GPIO6
GPIO5
GPIO4
GPIO3
GPIO2
GPIO1
GPIO0
Test
CPU_Break
Vdd
GND
PortK_7
PortK_6
NC
Vdd
GND
GND
PortK_5
PortK_4
Vdd
GND
EICE_miso
EICE_mosi
EICE_Clock
EICE_Load#
EICE_CS
Low_XTAL_Out
Low_XTAL_In
High_XTAL_Out
High_XTAL_In
AGND
156
156
156
150
151
SCI
SPI
Infrared
I2C
UART
User Serial
Counter
Capture Timer
PWM
UART A
UART B
EMI
Host
PIO A
PIO B
Wakeup
intAct

图 9

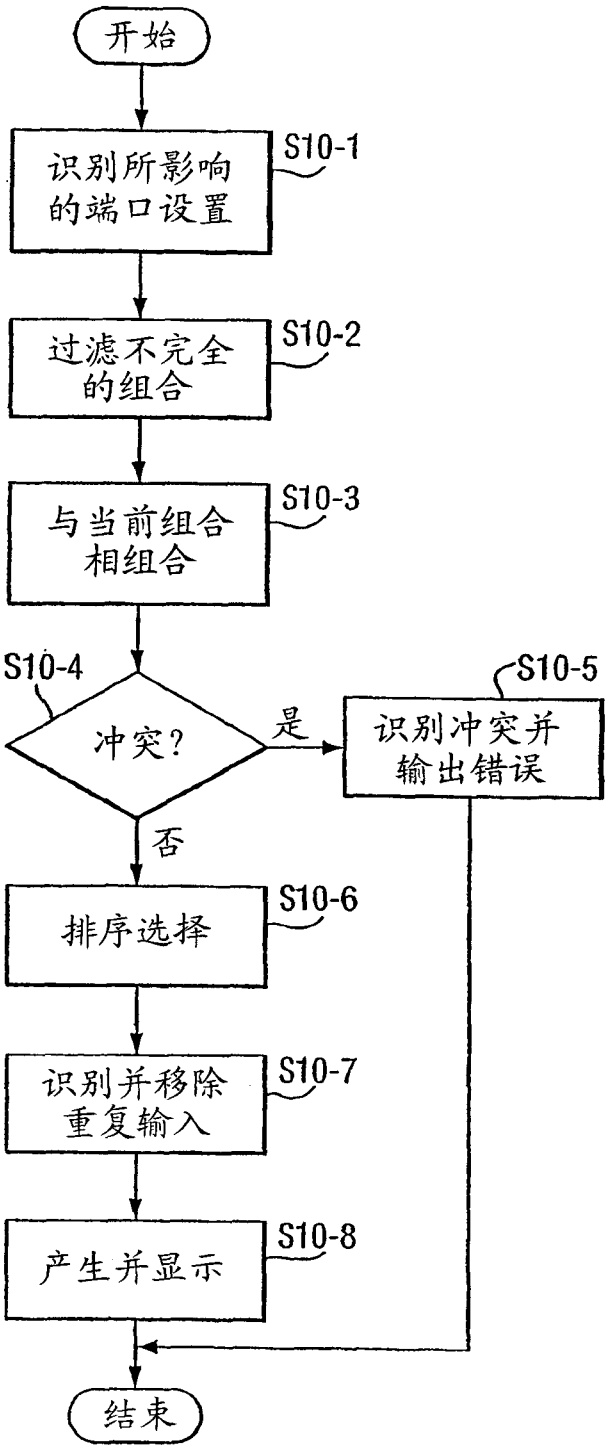

开始
识别所影响的端口设置
S10-1
过滤不完全的组合
S10-2
与当前组合相组合
S10-3
S10-4
冲突?
是
S10-5
识别冲突并输出错误
否
排序选择
S10-6
识别并移除重复输入
S10-7
产生并显示
S10-8
结束

图 10

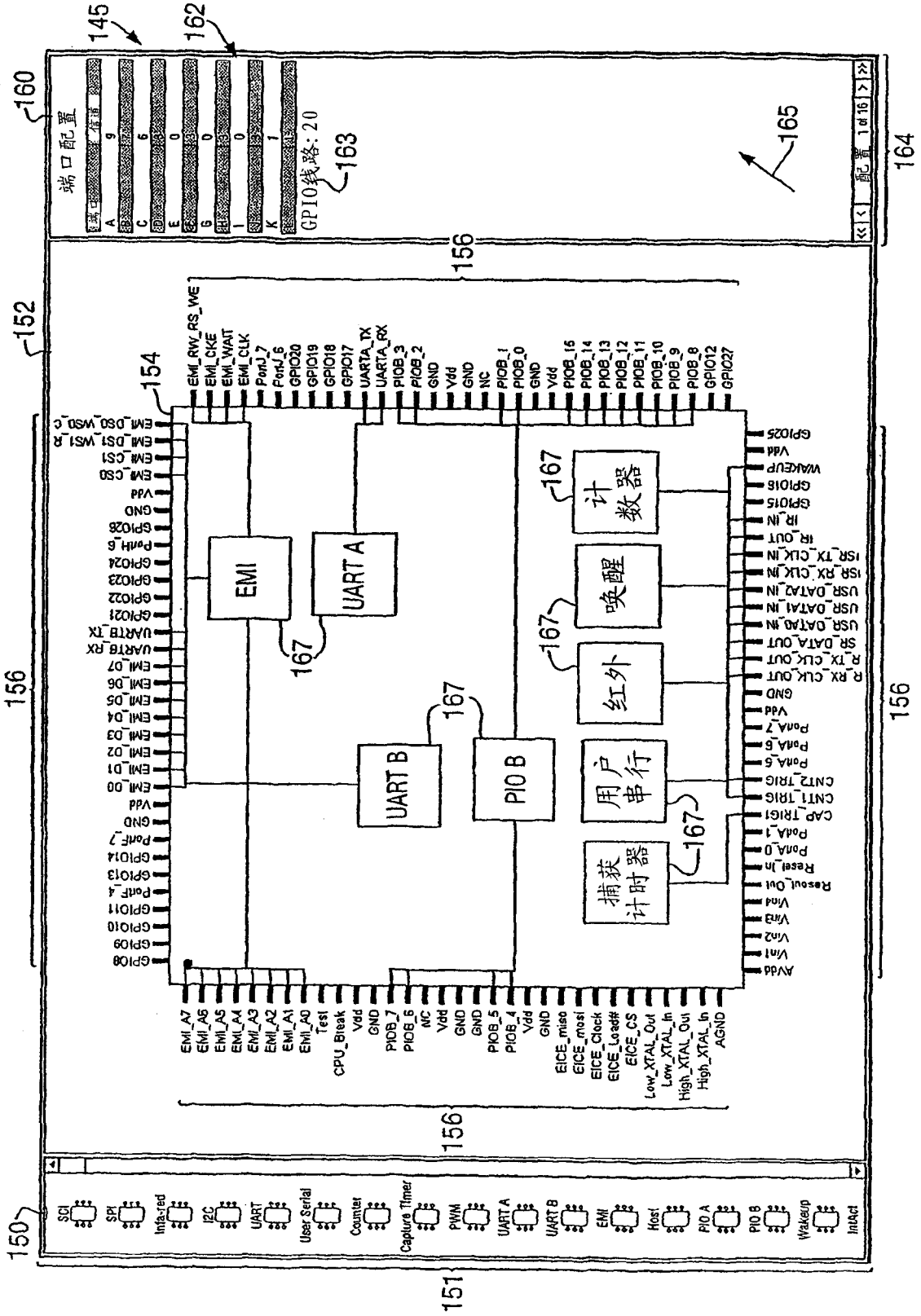

145
162
160
端口配置
GPIO线路: 20
163
165
164
156
152
154
EMI
UART A
UART B
PIO B
167
计数器
唤醒
红外
用户串行
捕获计时器
150
151
SCI
SPI
Infra-red
I2C
UART
User Serial
Counter
Capture Timer
PWM
UART A
UART B
EMI
Host
PIO A
PIO B
Wakeup
IntAct

图 11

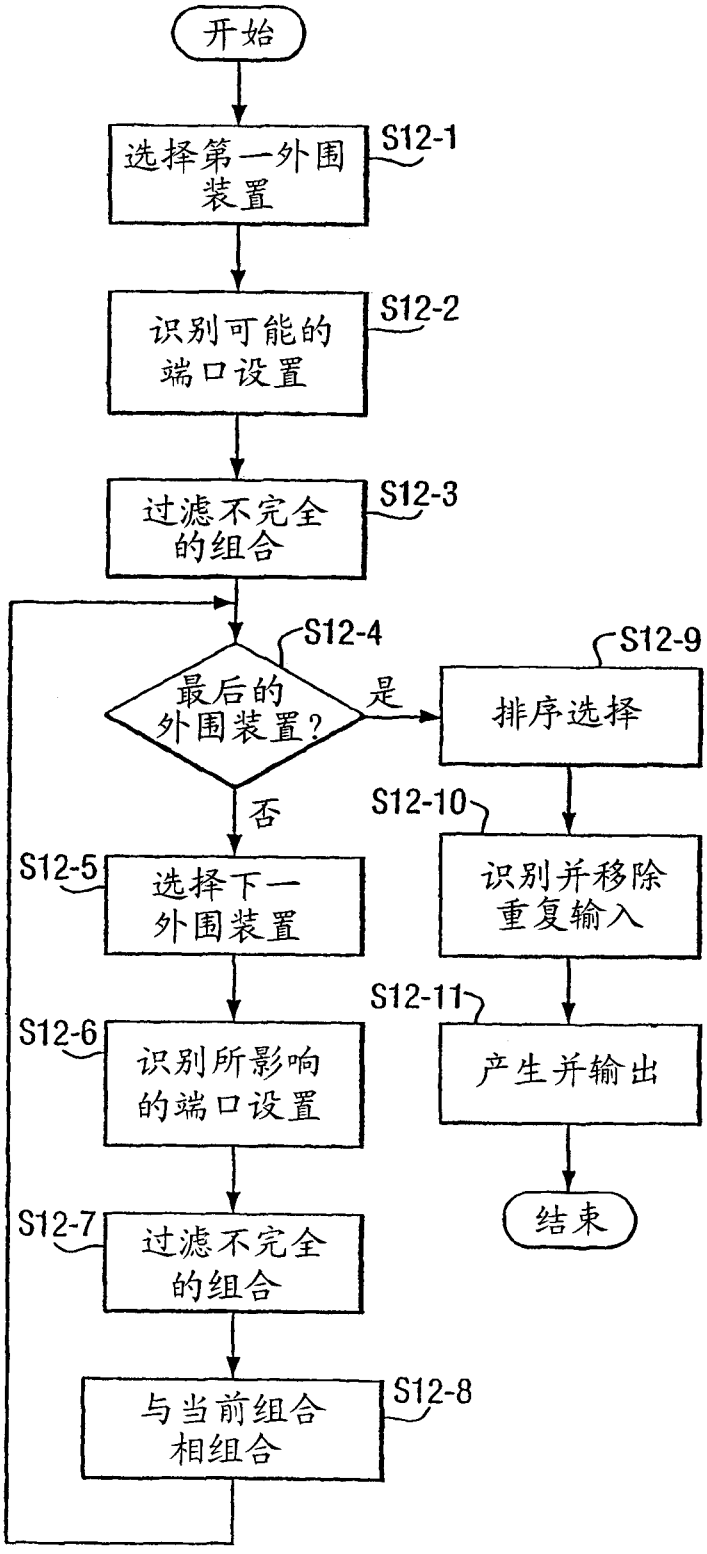

开始
选择第一外围装置
S12-1
识别可能的端口设置
S12-2
过滤不完全的组合
S12-3
S12-4
最后的外围装置?
是
S12-9
排序选择
否
S12-10
S12-5
选择下一外围装置
识别并移除重复输入
S12-11
S12-6
识别所影响的端口设置
产生并输出
结束
S12-7
过滤不完全的组合
S12-8
与当前组合相组合

图 12

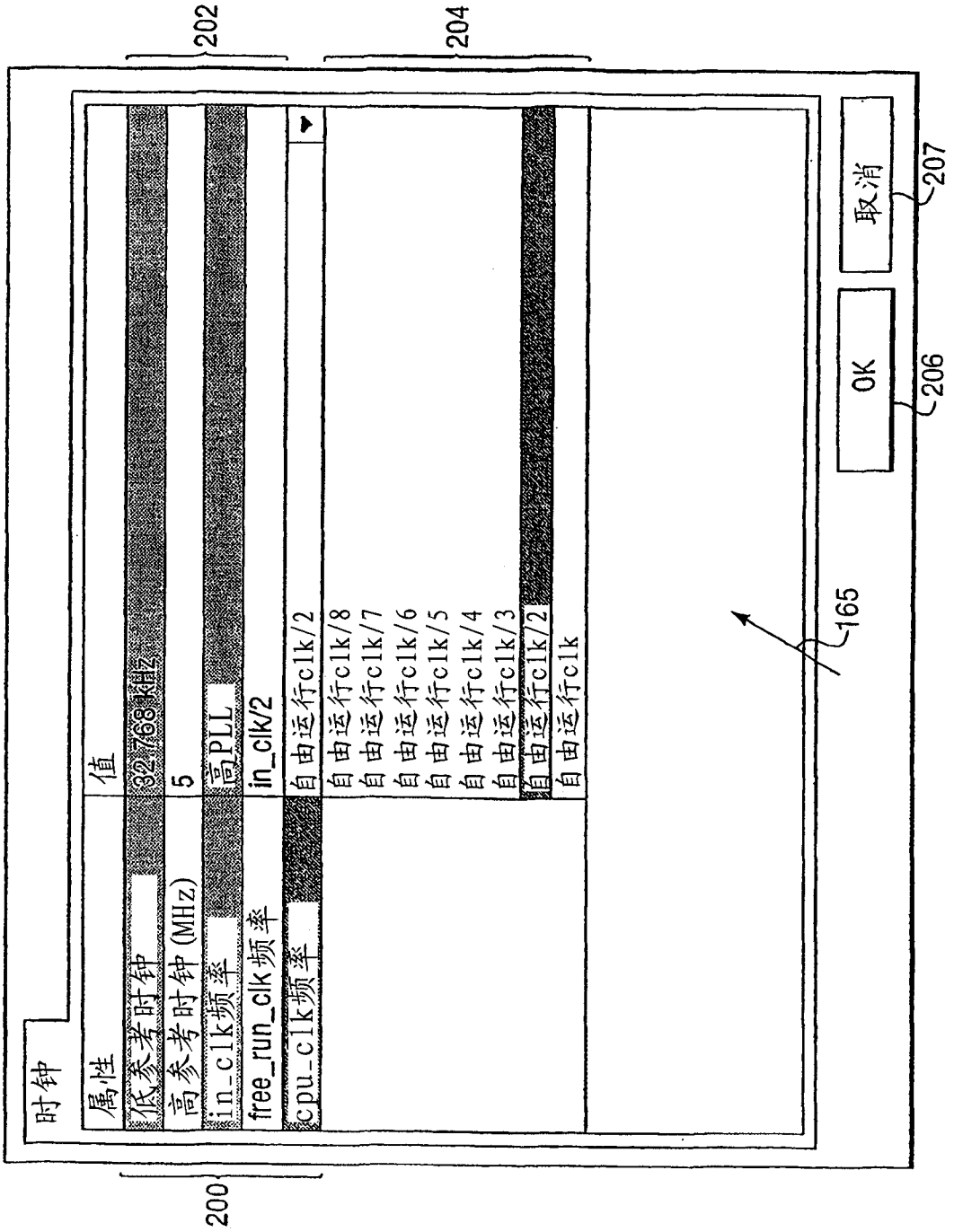

时钟
属性
值
低参考时钟
32768赫兹
高参考时钟 (MHz)
5
in_clk频率
高PLL
free_run_clk 频率
in_clk/2
cpu_clk频率
自由运行clk/2
自由运行clk/8
自由运行clk/7
自由运行clk/6
自由运行clk/5
自由运行clk/4
自由运行clk/3
自由运行clk/2
自由运行clk
OK
取消
200
202
204
206
207
165

图 13

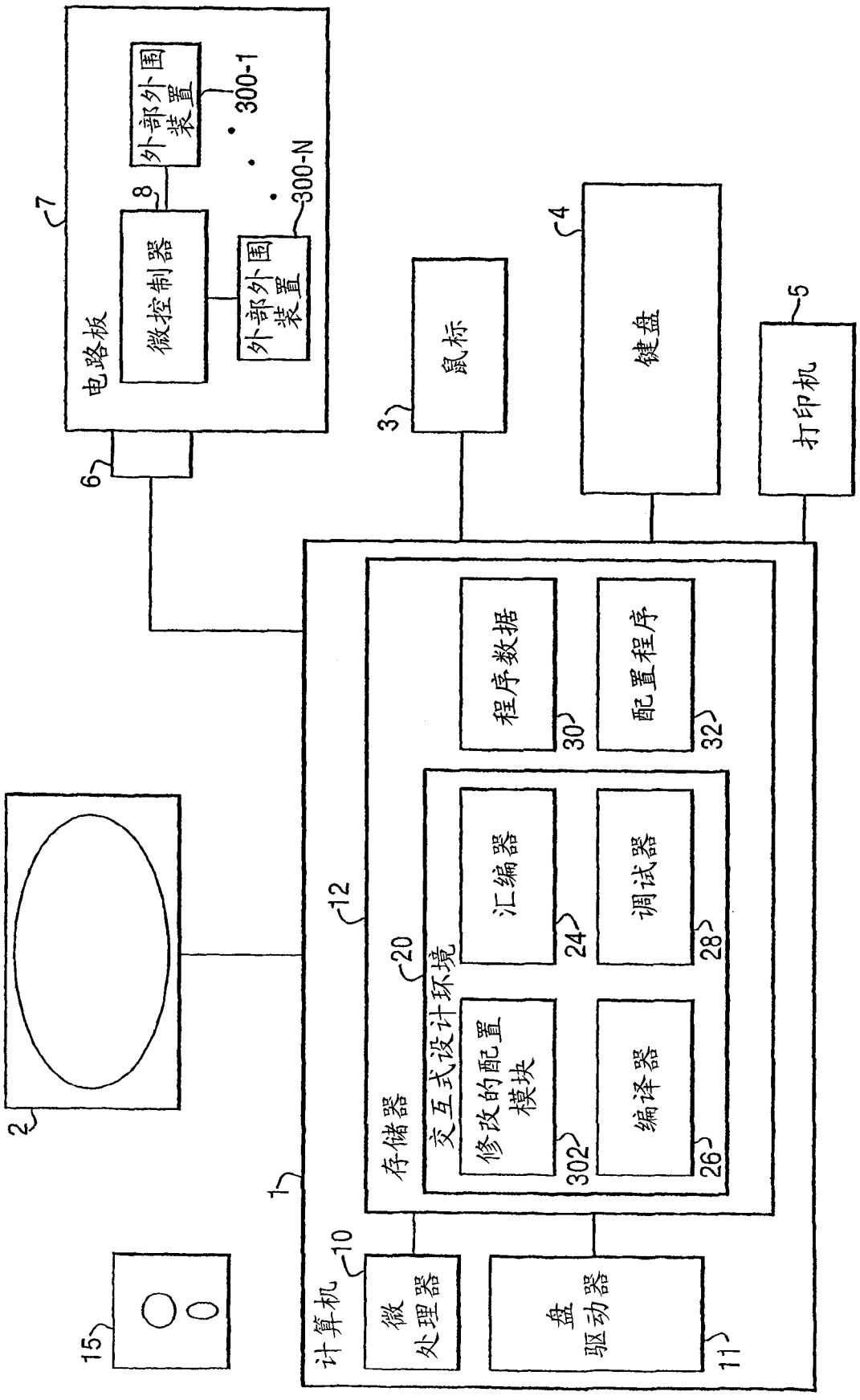

计算机
1
微处理器
10
盘驱动器
11
存储器
12
交互式设计环境
20
修改的配置模块
302
编译器
26
汇编器
24
调试器
28
程序数据
30
配置程序
32
显示器
2
鼠标
3
键盘
4
打印机
5
电路板
6
微控制器
7
外部外围装置
8
外部外围装置
300-1
300-N
15

图 14

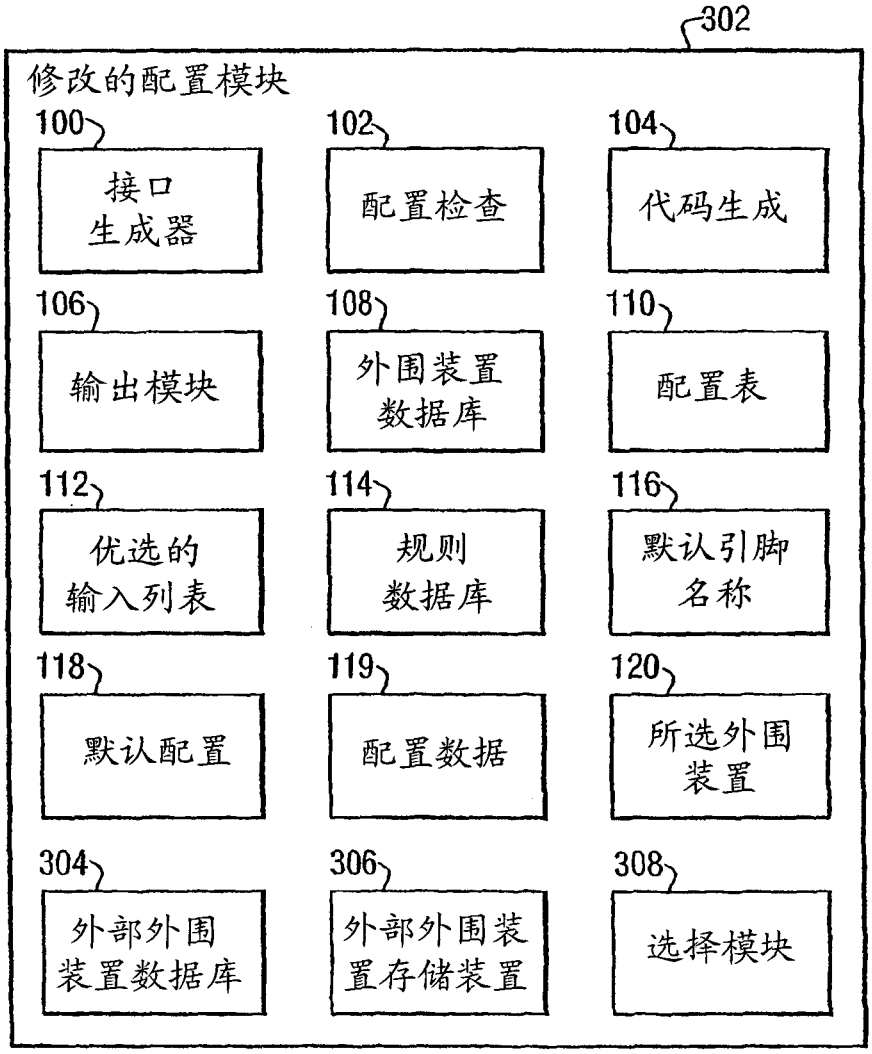

302
修改的配置模块
100
接口
生成器
102
配置检查
104
代码生成
106
输出模块
108
外围装置
数据库
110
配置表
112
优选的
输入列表
114
规则
数据库
116
默认引脚
名称
118
默认配置
119
配置数据
120
所选外围
装置
304
外部外围
装置数据库
306
外部外围装
置存储装置
308
选择模块

图 15

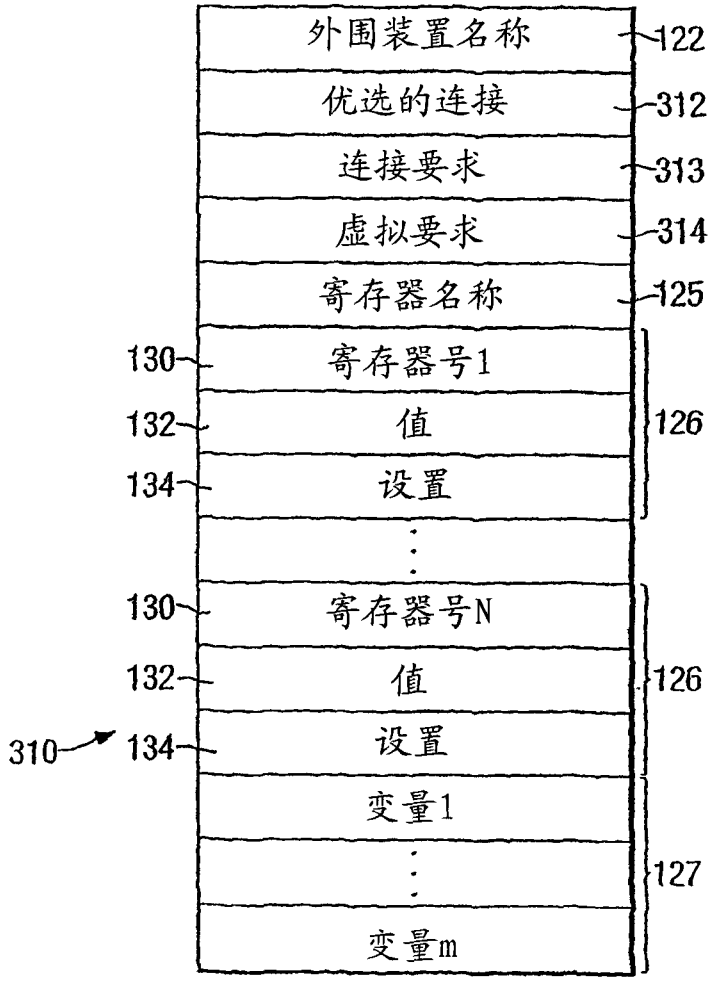

外围装置名称
122
优选的连接
312
连接要求
313
虚拟要求
314
寄存器名称
125
130
寄存器号1
132
值
134
设置
126
130
寄存器号N
132
值
134
设置
126
310
变量1
变量m
127

图 16

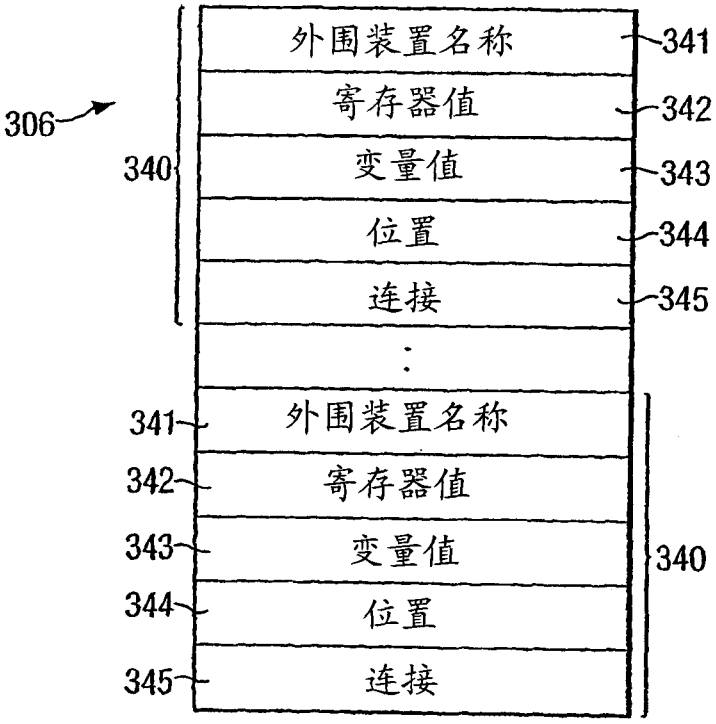

306
340
外围装置名称
341
寄存器值
342
变量值
343
位置
344
连接
345
外围装置名称
341
寄存器值
342
变量值
343
位置
344
连接
345
340

图 17

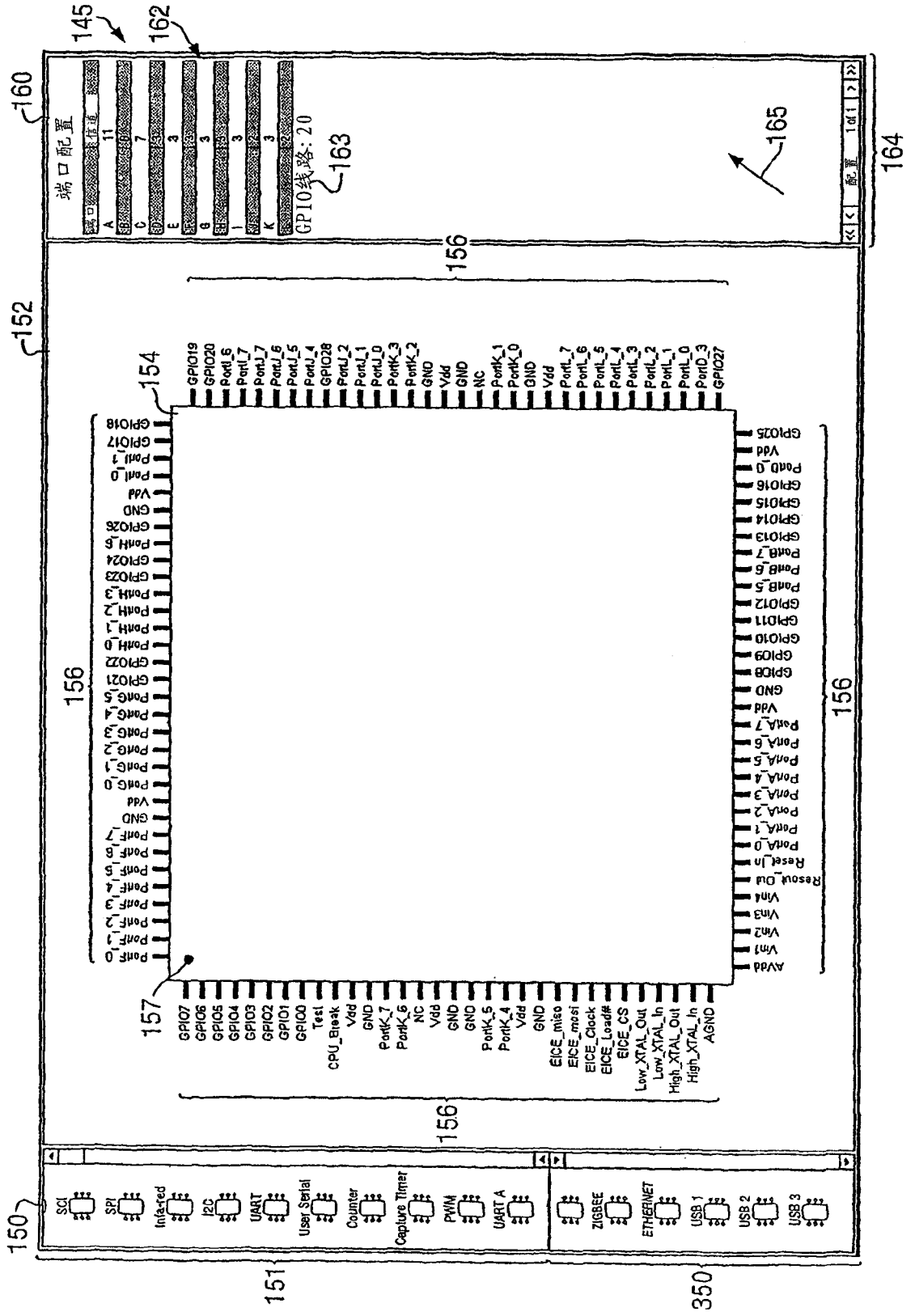

145
162
160
端口配置
GPIO线路: 20
163
165
164
152
156
154
157
150
151
350
SCI
SPI
Infared
I2C
UART
User Serial
Counter
Capture Timer
PWM
UART A
ZIGBEE
ETHERNET
USB 1
USB 2
USB 3

图 18

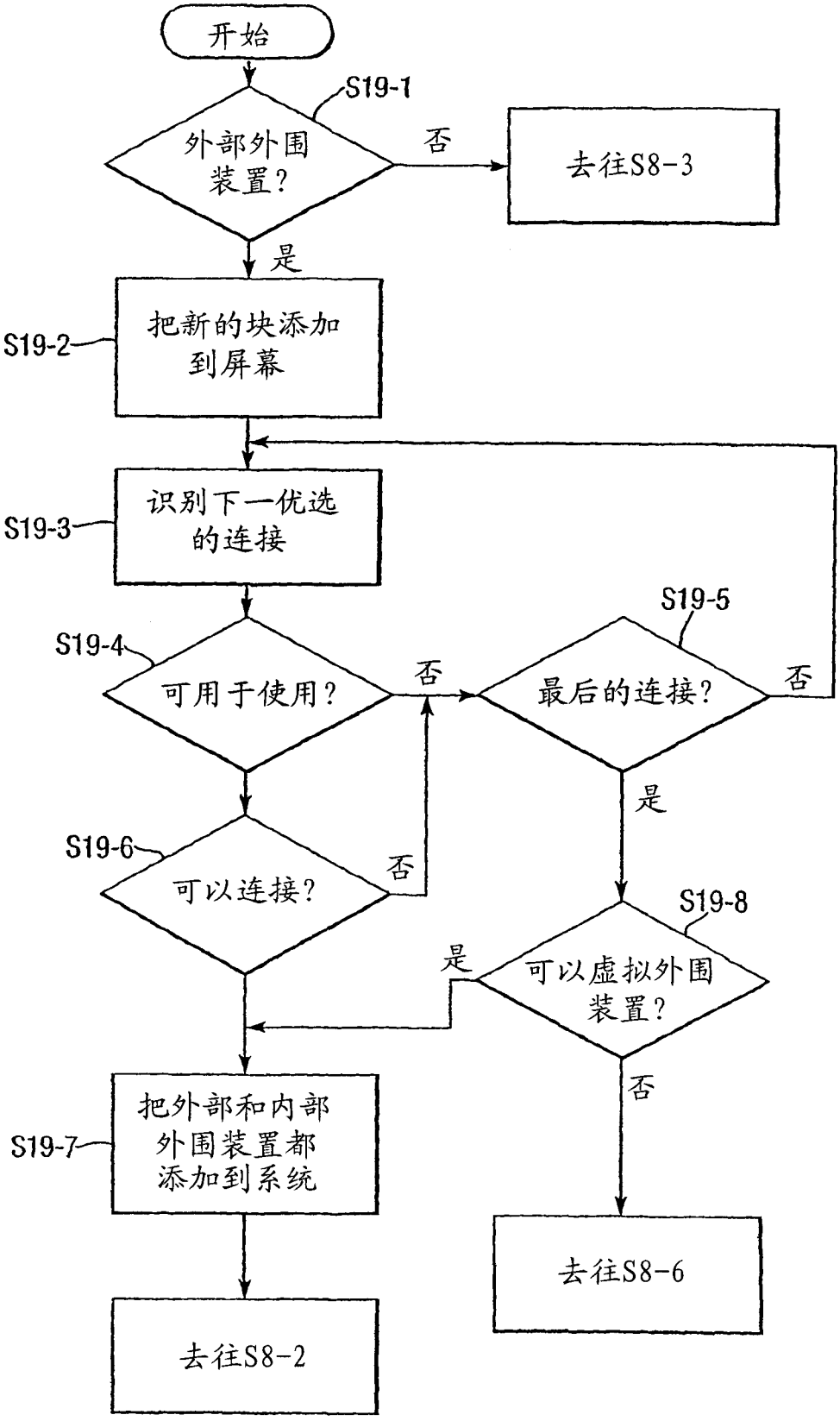

开始
S19-1
外部外围装置?
否
去往S8-3
是
S19-2
把新的块添加到屏幕
S19-3
识别下一优选的连接
S19-4
可用于使用?
否
S19-5
最后的连接?
否
是
S19-6
可以连接?
否
S19-8
可以虚拟外围装置?
是
否
S19-7
把外部和内部外围装置都添加到系统
去往S8-6
去往S8-2

图 19

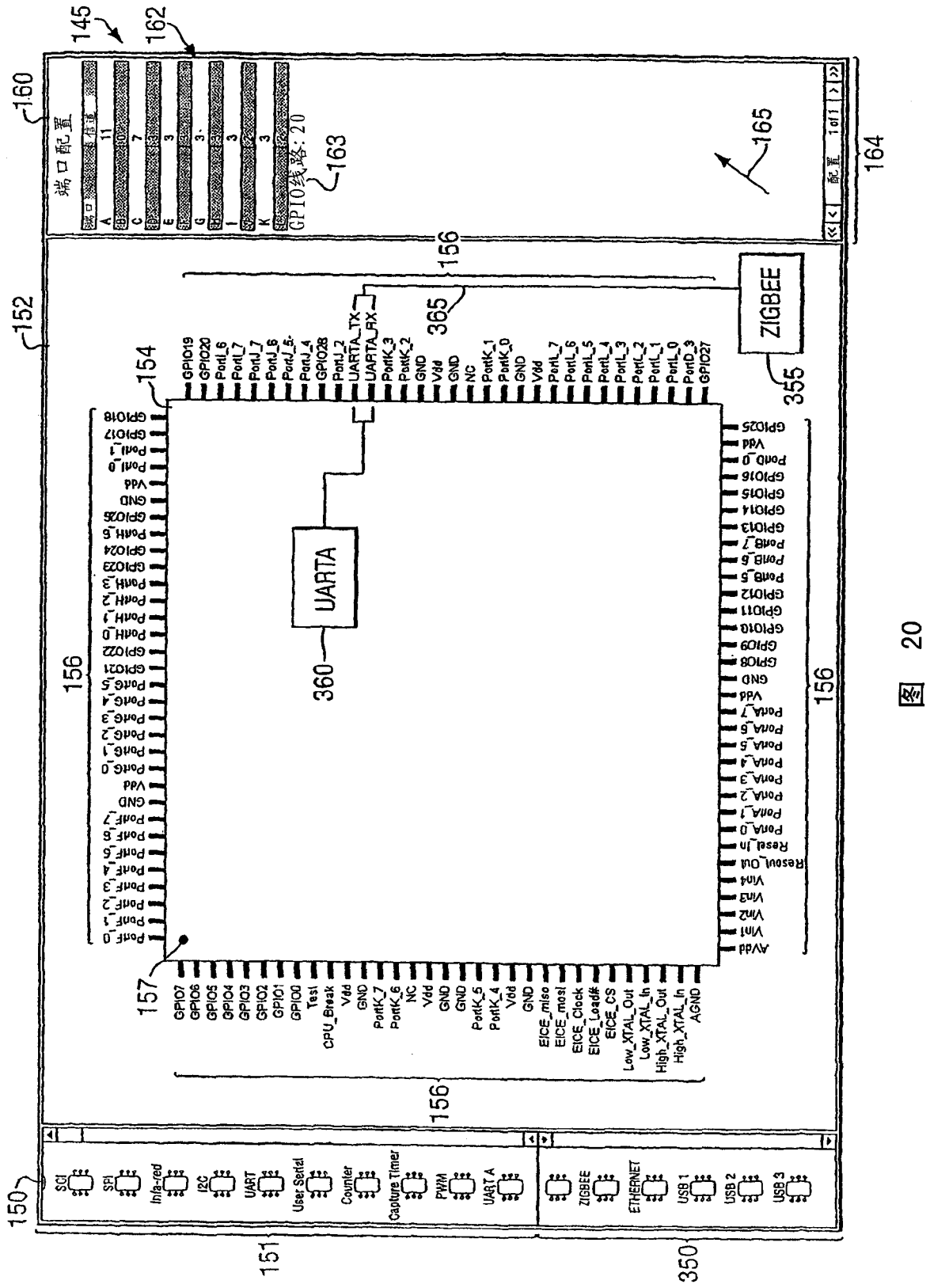

端口配置
GPIO线路: 20
ZIGBEE
UARTA
145
162
160
163
164
165
152
154
156
157
150
151
350
355
360
365

图 20

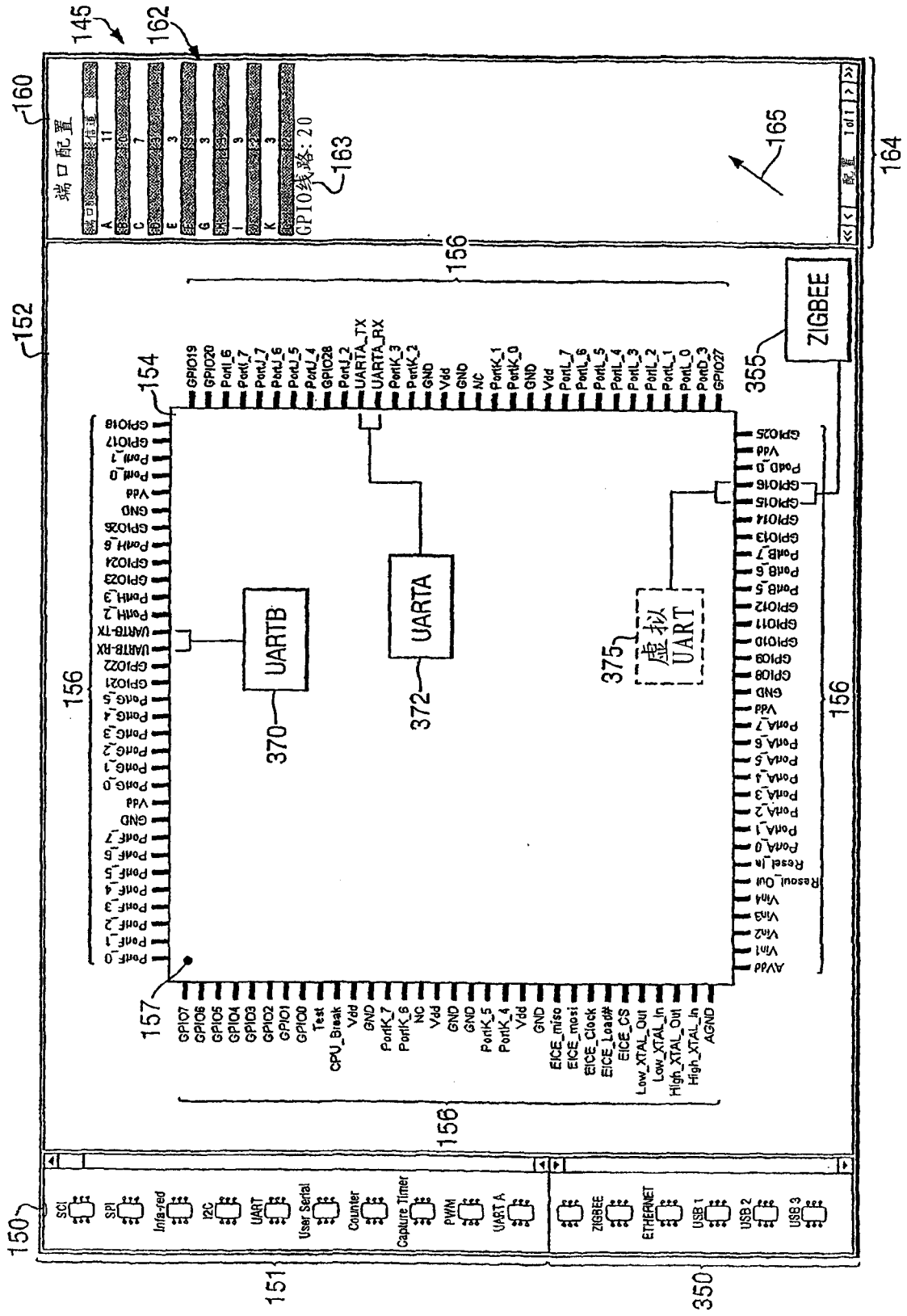

145
162
160
端口配置
GPIO线路: 20
163
165
164
152
156
154
ZIGBEE
355
UARTB
370
UARTA
372
虚拟
UART
375
157
150
151
350
SCI
SPI
Infa-red
I2C
UART
User Serial
Counter
Capture Timer
PWM
UART A
ZIGBEE
ETHERNET
USB 1
USB 2
USB 3

图 21