



(12) 发明专利

(10) 授权公告号 CN 109101218 B
(45) 授权公告日 2023. 04. 28

(21) 申请号 201810862197.8
(22) 申请日 2014.11.24
(65) 同一申请的已公布的文献号
申请公布号 CN 109101218 A
(43) 申请公布日 2018.12.28
(62) 分案原申请数据
201480074503.1 2014.11.24

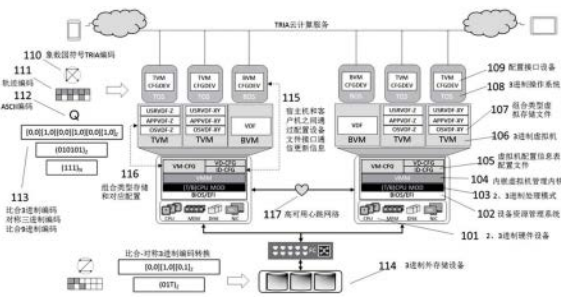
(51) Int.Cl.
G06F 7/38 (2006.01)
G06F 15/76 (2006.01)
(56) 对比文件
CN 102681756 A, 2012.09.19
US 2006221056 A1, 2006.10.05
审查员 邓丽婉

(73) 专利权人 蔡光贤
地址 100075 北京市东城区安乐林路景泰
西里西区4号楼3单元503室
(72) 发明人 蔡光贤
(74) 专利代理机构 北京铭硕知识产权代理有限公司 11286
专利代理师 曾世骁 苏银虹

权利要求书7页 说明书41页 附图10页

(54) 发明名称
输入法和使用该输入法的输入装置、输入系统

(57) 摘要
本发明公开一种输入法和使用该输入法的输入装置、输入系统。所述输入法包括：检测用户在输入装置上输入的输入轨迹，所述输入装置具有若干个有效输入点，所述输入轨迹是从所述若干个有效输入点之一一起始，并经过至少一个其他有效输入点而结束；将所述输入轨迹转换为轨迹码；经预定义或自定义的映射表，将所述轨迹码转换成对应的ASCII码、Unicode码或其他进制编码。



1. 一种输入法,其特征在于包括:

检测用户在输入装置上输入的输入轨迹,所述输入装置具有若干个有效输入点,所述输入轨迹是从所述若干个有效输入点之一一起始,并经过至少一个其他有效输入点而结束;

将所述输入轨迹转换为轨迹码;

经预定义或自定义的映射表,将所述轨迹码转换成对应的ASCII码、Unicode码或其他进制编码,

其中,所述轨迹码使用将一维二进制象数编码转换为二维二进制或八进制编码框架上的符号系统,所述符号系统预定义或自定义所述轨迹码和键码键值的映射关系,

其中,所述一维二进制象数编码是6位编码并具有64种象数格式的编码符号,

其中,所述二维二进制编码框架包括由不在一条直线上的四个顶点作为有效输入点以及各个顶点之间相互连接的六条线表示的64种二维象数图符号,所述64种二维象数图符号与所述一维二进制象数编码的64种象数格式的编码符号对应,所述二维二进制格式框架上的四个顶点中的任意一个顶点能够与其他三个顶点连接形成三元编码的几何特征,

其中,所述二维八进制编码框架包括两个八进制编码,所述两个八进制编码由分布在一个圆形的八个点和圆心或一条直线上的八个点定义的有效输入点形成输入轨迹的方向转折点表示,所述两个八进制编码组合形成64种二维象数图符号,所述64种二维象数图符号与所述一维二进制象数编码的64种象数格式的编码符号对应,所述二维八进制格式框架上的方向转折点具有两种自由度的二元编码的几何特征。

2. 如权利要求1所述的输入法,其特征在于,

所述轨迹码是基于矩形的四个顶点通过相互连接产生的几何图形的符号轨迹的编码,并且所述轨迹码包括起始位和轨迹描述位,

其中,通过一笔画完成一个文字、数字、标点符号或控制码的输入轨迹,所述一笔画经过1~4个点组成的矩阵拓扑结构,通过利用一笔画经过的点的三值编码来获得所述轨迹码,

所述输入轨迹是从矩形的四个顶点之一一起始,沿着矩形的四个边或对角线移动并且通过四个顶点中的至少两个顶点,由输入轨迹经过的顶点编码来顺序组成所述轨迹码。

3. 如权利要求1所述的输入法,其特征在于还包括:

基于所述轨迹码对应的ASCII码、Unicode码或其他进制编码,查找语言字库中的对应的文字图象信息,输出到标准输出设备或外设设备。

4. 如权利要求1所述的输入法,其中,

所述的输入装置提示输入轨迹对应的文字图象信息。

5. 如权利要求1所述的输入法,其中,

矩形顶点组成四点一画特征的输入轨迹符号“彩文”有四个边和两个对角线,下表列出彩文的输入轨迹符号的轨迹码和对应的其他关联码之间的关系表,所述轨迹码是田字格上下左右四个顶点 $\{[1,2], [3,4]\}$ 的编码顺序并且使用0表示断码连接:

彩文(Chai)	∴	∴	ℵ	ℵ	∴	∴	ℵ	ℵ
轨迹码	0	34	14	143	24	243	142	24143
易卦=盲文	⠠⠠⠠⠠	⠠⠠⠠⠠	⠠⠠⠠⠠	⠠⠠⠠⠠	⠠⠠⠠⠠	⠠⠠⠠⠠	⠠⠠⠠⠠	⠠⠠⠠⠠
ASCII=高丽音			l=]	j= ㄹ			y= ㅡ	
UniCode	4dc1	4dd7	4dc6	4dd2	4dce	4de3	4ded	4dca
彩文(Chai)	∴	ℵ	ℵ	ℵ	∴	∴	ℵ	ℵ
轨迹码	13	134	314	1341	13024	1342	3142	24314
ASCII=高丽音	‘	l= o	<	a= ㅏ		u= ㅓ	~	6
易卦=盲文	⠠⠠⠠⠠	⠠⠠⠠⠠	⠠⠠⠠⠠	⠠⠠⠠⠠	⠠⠠⠠⠠	⠠⠠⠠⠠	⠠⠠⠠⠠	⠠⠠⠠⠠
UniCode	4dcf	4df2	4de7	4df5	4dfd	4df6	4ddf	4de1
彩文(Chai)	ℵ	ℵ	ℵ	ℵ	ℵ	ℵ	ℵ	ℵ
轨迹码	23	234	14023	1432	324	2342	1423	14324

ASCII=高丽音	l	4		w=ㄴ	>	5	x=ㄴ	ia=â= ㅈ
易卦=盲文	䷌	䷍	䷎	䷏	䷐	䷑	䷒	䷓
UniCode	4dc7	4dc2	4ddc	4dfb	4de6	4dfe	4def	4dc4
彩文(Chai)	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ
轨迹码	132	13234	2314	23143	1324	13423	13241	314324
ASCII=高丽音	v=ㅈ		8		n=ㄴ	f=ㄴ	h=ㅈ	
易卦=盲文	䷌	䷍	䷎	䷏	䷐	䷑	䷒	䷓
UniCode	4ddc	4dd0	4dee	4df9	4dde	4df0	4ddb	4dea
彩文(Chai)	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ
轨迹码	12	12034	214	2143	124	1243	1241	34214
ASCII=高丽音	+		7	2	g=ㄴ	c=ㅈ	t=ㅈ	}
易卦=盲文	䷌	䷍	䷎	䷏	䷐	䷑	䷒	䷓
UniCode	4dd6	4dda	4dc3	4de8	4df3	4dd5	4dd1	4dd9
彩文(Chai)	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ
轨迹码	213	2134	21413	13412	3124	1243	12413	143124
ASCII=高丽音	7	3		ie=ê= ㅈ		o=ㅈ	p=ㅈ	r=ㅈ
易卦=盲文	䷌	䷍	䷎	䷏	䷐	䷑	䷒	䷓
UniCode	4de2	4dd4	4dff	4de5	4df7	4ddd	4df1	4dcd
彩文(Chai)	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ
轨迹码	123	1234	3214	12341	12324	12432	21423	214234
ASCII=高丽音	k=ㅈ	z=ㅈ		s=ㅈ		e=ㅈ	?	:
易卦=盲文	䷌	䷍	䷎	䷏	䷐	䷑	䷒	䷓
UniCode	4dd3	4de9	4dfa	4dfc	4df4	4de4	4df8	4dc8

彩文(Chai)	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ
轨迹码	1231	43123	12314	123143	21324	231243	132142	123142034
ASCII=高丽音	d=ㄷ	{	q=ㄱ	b=ㅁ	9	0	m=ㅇ	
易卦=盲文	䷎=⠠	䷏=⠠	䷐=⠠	䷑=⠠	䷒=⠠	䷓=⠠	䷔=⠠	䷕=⠠
UniCode	4dc6	4dd8	4dc5	4dc9	4de0	4dcc	4deb	4dc0
高丽音=彩文	ㅅ=ㄷ	ㅆ=ㄴ	ㅈ=ㄷ	ㅊ=ㄹ	ㅌ=ㄷ	ㅍ=ㅅ	ㅍ=ㅁ	ㅍ=ㅁ
UniCode	d7a4	d7a5	d7a6	d7a7	d7a8	d7a9	d7aa	d7ab
高丽音=彩文	ㅍ=ㄷ	ㅍ=ㄴ	ㅍ=ㄷ	ㅍ=ㄹ	ㅍ=ㄷ	ㅍ=ㅅ	ㅍ=ㅁ	ㅍ=ㅁ
UniCode	d7ac	d7ad	d7ae	d7af	afd3	ae67	af47	c9fc

默认的彩文轨迹码在田字格上下左右四个可识别触摸顶点的编码顺序{[0,1],[2,3]}和键码对应,以下表是默认的彩文轨迹码和对应的键码键值关系表:

轨迹码	键码	轨迹码	键码	轨迹码	键码
0320,0230	a	C+[a-z]轨迹编码	大写 A-Z	21	/
03213,03123	à	0	SHIFT	30	\
012032, 023012, 032012, 032102, 021032, 012302	b	1	space	20	
0132	c	120,121	backspace	202	!
0120,0210	d	130	DELETE	3203, 3023	@
01321,01231	e	2	Control	32013, 31023	#
02301,03201	è	3	Enter	2301,3210	\$
02312,02132	f	21301	ESCAPE	3213,3123	%
013	g	23	CURSOR_RIGHT	302	^
02130,03120	h	32	CURSOR_LEFT	32103,30123,3012	&
03	i	31	CURSOR_UP	3021	*
032	j	13	CURSOR_DOWN	3120	~
012	k	212	DUPLICATELINE	312	`
023	l	12	1	02	'
021031, 031021, 012031, 013021, 021301, 031201	m	1032	2	203,010	"
0213	n	1023	3	0202	;

01320,02310	o	123	4	1313	:
01302,03102	p	1231,1321	5	020	,
02103,01203	q	10230, 10320, 13203, 13023	6	131	.
013203, 031023, 032013, 023013, 023103, 013023,	r	102, 103	7	230	>
01230,03210	s	1203	8	321	<
0130,0310	t	10213, 12013,	9	320	(
0231	u	102312, 120132, 123102, 132012, 102132,10231, 13201,	0	231)
021	v	10	-	3201	[
0321	w	01	+	2310]
0312	x	323	-	32012,32102	{
031	y	232	=	21023,20123	}
0123	z	213	tab	310,3101	?

6. 如权利要求1所述的输入法,其特征在于:

所述输入装置具有包括分布在圆周上的8个检测区域和圆心上的检测区域的9个不同的检测区域,所述输入轨迹从圆周上的8个检测区域中的任意一个检测区域起始并且沿着半径方向经过圆心的检测区域,然后再次移动到圆周上的8个检测区域中的任意一个而结束,

其中,圆周上的8个检测区域由8个不同的数值表示,

所述编码由输入轨迹的起始区域位置和结束点位置的数值表示。

7. 如权利要求1所述的输入法,其特征在于:

所述输入装置具有包括分布在圆周上的8个不同的检测区域,所述输入轨迹从圆周上的8个检测区域中的任意一个检测区域起始并且沿着顺时针方向或逆时针方向移动至圆周上的一个检测区域,然后以相反的方向从该检测区域再次沿着圆周移动到圆周上的任意一个检测区域而结束,

其中,圆周上的8个检测区域由8个不同的数值表示,

其中,所述编码由输入轨迹的起始点位置和结束点位置的数值表示。

8. 如权利要求1所述的输入法,其特征在于:

所述输入装置具有包括分布在一条线段上的8个检测点,所述输入轨迹从线段上的8个检测点中的任意一个检测点起始并且沿着线段的一个方向移动至线段上的一个检测点,然后以相反的方向从该检测点再次沿着线段移动到线段上的任意一个检测点而结束,

其中,8个检测点由8个不同的数值表示,

其中,所述编码由输入轨迹的起始点位置和结束点位置的数值表示。

9. 一种计算机输入系统,其特征在于包括:

检测模块,检测用户在输入装置上输入的输入轨迹,所述输入装置具有若干个有效输入点,所述输入轨迹是从所述若干个有效输入点之一一起始,并经过至少一个其他有效输入点而结束;

轨迹码转换模块,将所述输入轨迹转换为轨迹码;以及

通用编码转换模块,经预定义或自定义的映射表,将所述轨迹码转换成对应的ASCII码、Unicode码或其他进制编码,

其中,所述轨迹码使用将一维二进制象数编码转换为二维二进制或八进制编码框架上的符号系统,所述符号系统预定义或自定义所述轨迹码和键码键值的映射关系,

其中,所述一维二进制象数编码是6位编码并具有64种象数格式的编码符号,

其中,所述二维二进制编码框架包括由不在一条直线上的四个顶点作为有效输入点以及各个顶点之间相互连接的六条线表示的64种二维象数图符号,所述64种二维象数图符号与一位二进制象数编码的64种象数格式的编码符号对应,所述二维二进制格式框架上的四个顶点中的任意一个顶点能够与其他三个顶点连接形成三元编码的几何特征,

其中,所述二维八进制编码框架包括两个八进制编码,所述两个八进制编码由分布在一个圆形的八个点和圆心或一条直线上的八个点定义的有效输入点形成输入轨迹的方向转折点表示,所述两个八进制编码组合形成64种二维象数图符号,所述64种二维象数图符号与一位二进制象数编码的64种象数格式的编码符号对应,所述二维八进制格式框架上的方向转折点具有两种自由度的二元编码的几何特征。

10. 如权利要求9所述的输入系统,其特征在于,

所述轨迹码是基于矩形的四个顶点通过相互连接产生的几何图形的符号轨迹的编码,并且所述轨迹码包括起始位和轨迹描述位,

其中,通过一笔画完成一个文字、数字、标点符号或控制码的输入轨迹,所述一笔画经过1~4个点组成的矩阵拓扑结构,通过利用一笔画经过的点的三值编码来获得所述轨迹码,

所述输入轨迹是从矩形的四个顶点之一一起始,沿着矩形的四个边或对角线移动并且通过四个顶点中的至少两个顶点,由输入轨迹经过的顶点编码来顺序组成所述轨迹码。

11. 如权利要求9所述的输入系统,其特征在于还包括:

输出模块,所述的输出模块基于所述轨迹码对应的ASCII码、Unicode码或其他进制编码,查找语言字库中的对应的文字图象信息,输出到标准输出设备或外设设备。

12. 如权利要求9所述的输入系统,其中,

所述的输入装置在检测输入轨迹时提示输入轨迹对应的文字图象信息。

13. 如权利要求9所述的输入系统,其中,所述的输入装置是六面体,六面体的每个面上设置有9个相似图案符号的9个按键。

14. 如权利要求9所述的输入系统,其中,所述的检测模块以接触方式或非接触方式检测输入轨迹。

15. 一种基于三进制编码的输入法,包括:

检测用户在输入装置上输入的输入轨迹;

基于输入轨迹确定与输入轨迹对应的图案符号的三进制编码;

基于三进制编码与符号文字的映射表确定和输出与所述三进制编码对应的符号文字,

所述三进制编码是基于矩形的四个顶点通过相互连接产生的几何图形的符号轨迹的编码,

其中,通过一笔画完成一个文字、数字、标点符号或控制码的输入轨迹,所述一笔画经过1~4个点组成的矩阵拓扑结构,每个点有三个方向的编码形式,通过利用一笔画经过的

点的三值编码来获得所述三进制编码，

其中，所述输入轨迹是从矩形的四个顶点之一一起始，沿着矩形的四个边或对角线移动并且通过四个顶点中的至少两个顶点的一笔画输入轨迹，

其中，在所述三进制编码中，由一个三进制数位表示起始点的位置，由三进制数的三个不同数位之一分别表示两个顶点之间的顺时针运动轨迹、对角线的运动轨迹和逆时针运动轨迹。

16. 如权利要求15所述的输入法，其中，三进制编码符号被存储在ASCII编码或Unicode编码的扩展编码中。

17. 如权利要求15所述的输入法，其中，除了一笔画输入轨迹之外的特定输入轨迹被定义为与特定的控制字符对应。

18. 如权利要求15所述的输入法，其中，所述输入装置具有四个检测点，所述四个检测点位于矩形的顶点，用户通过所述四个检测点来输入特定的输入轨迹。

19. 如权利要求15所述的输入法，其中，所述输入装置具有与多个输入轨迹的图案符号一一对应的实体或虚拟的多个按键，用户通过按压按键来输入对应的输入轨迹。

20. 如权利要求19所述的输入法，其中，所述多个按键包括一笔画输入轨迹形成的所有图案符号中的全部或至少一部分。

21. 如权利要求15所述的输入法，其中，所述输入装置是六面体，六面体的每个面上设置有9个相似图案符号的9个按键。

输入法和使用该输入法的输入装置、输入系统

[0001] 本申请是申请日为2014年11月24日,申请号为“201480074503.1”,标题为“三进制和二进制混合运算的计算机系统”的发明专利申请的分案申请。

技术领域

[0002] 本发明涉及计算机领域,更具体地说,涉及一种能够执行三进制和二进制混合运算的计算机系统,包括该混合运算的计算机系统的运算系统、内存系统、存储系统、网络系统、外设系统等。本发明的计算机系统适用于电子和光子计算。此外,根据本发明的外设系统的交互方式也适用于物联网和穿戴电子设备的物理和虚拟方式(VR/AR)的交互界面。

背景技术

[0003] 三进制是以3为基数的进制。与二进制相似,三进制的数位称为三进制位(trit)。每个单位信息量密度上三进制位比二进制位的信息量大。通常,三进制中使用0、1、2三个数字。但在对称三进制中,则使用-1(记作T)、0、1来表达三进制位。

[0004] 现今的计算机都使用二进制的数字系统,尽管它的计算规则非常简单,但其实二进制逻辑并不能完美地表达人类的真实想法。相比之下,三进制逻辑更接近人类大脑的思维方式。因为在一般情况下,人对于问题的看法不是只有“真”和“假”两种答案,还有一种“非常道”。在对称三进制逻辑学中,符号“1”代表“真”;符号“-1”代表“假”;符号“0”代表“不知道”。显然,这种逻辑表达方式更符合计算机在人工智能方面的发展趋势。它为计算机的模糊运算和自主学习提供了可能。三进制代码的一个特点是对称,即相反数的一致性,因此它就和二进制代码不同,不存在“无符号数”的概念。这样,三进制计算机的架构也要简单、稳定、经济得多。其指令系统也更便于阅读,而且非常高效,更加适宜用于非线性光计算处理。

[0005] 如上所述,基于对称三进制的计算机系统并非无法实现。但是,当前的现实是:计算机系统普遍采用了“二进制”数字系统,为此需要一种能够在二进制计算机环境下执行三进制计算的技术。

发明内容

[0006] 在下面的描述中将部分地阐明本发明另外的方面和/或优点,通过描述,其会变得更加清楚,或者通过实施本发明可以了解。

[0007] 本发明提供了一种能够在二进制和三进制之间自由切换的计算机系统,并提供了与该计算机系统相适应的能够基于二进制和三进制进行运算的处理器、存储系统、编码系统、输入系统和网络系统等。

[0008] 根据本发明的一方面,提供了一种输入法,包括:检测用户在输入装置上输入的输入轨迹,所述输入装置具有若干个有效输入点,所述输入轨迹是从所述若干个有效输入点之一一起始,并经过至少一个其他有效输入点而结束;将所述输入轨迹转换为轨迹码;经预定义或自定义的映射表,将所述轨迹码转换成对应的ASCII码、Unicode码或其他进制编码。

[0009] 根据本发明的一方面,提供了一种计算机输入系统,包括:检测模块,检测用户在输入装置上输入的输入轨迹,所述输入装置具有若干个有效输入点,所述输入轨迹是从所述若干个有效输入点之一一起始,并经过至少一个其他有效输入点而结束;轨迹码转换模块,将所述输入轨迹转换为轨迹码;以及通用编码转换模块,经预定义或自定义的映射表,将所述轨迹码转换成对应的ASCII码、Unicode码或其他进制编码。

[0010] 根据本发明的另一方面,提供了一种基于三进制编码的输入法,包括:检测用户在输入装置上输入的输入轨迹;基于输入轨迹确定与输入轨迹对应的图案符号的三进制编码;基于三进制编码与符号文字的映射表确定和输出与所述三进制编码对应的符号文字,其中,所述输入轨迹是从矩形的四个顶点之一一起始,沿着矩形的四个边或对角线移动并且通过四个顶点中的至少两个顶点的一笔画输入轨迹,其中,在所述三进制编码中,由一个三进制数位表示起始点的位置,由三进制数的三个不同数位之一分别表示两个顶点之间的顺时针运动轨迹、对角线的运动轨迹和逆时针运动轨迹。

[0011] 根据本发明的另一方面,还提供了一种输入法,包括:在输入装置上确定用户输入的输入轨迹;确定与输入轨迹的图案对应的编码;基于编码与符号文字的映射表,确定并输出与编码对应的符号文字,其中,所述输入装置具有包括分布在圆周上的8个检测区域和圆心上的检测区域的9个不同的检测区域,所述输入轨迹从圆周上的8个检测区域中的任意一个检测区域起始并且沿着半径方向经过圆心的检测区域,然后再次移动到圆周上的8个检测区域中的任意一个而结束,其中,圆周上的8个检测区域由8个不同的数值表示,所述编码由输入轨迹的起始区域位置和结束点位置的数值表示。

[0012] 根据本发明的另一方面,提供了一种输入法,包括:在输入装置上确定用户输入的输入轨迹;确定与输入轨迹的图案对应的编码;基于编码与符号文字的映射表,确定并输出与编码对应的符号文字,其中,所述输入装置具有包括分布在圆周上的8个不同的检测区域,所述输入轨迹从圆周上的8个检测区域中的任意一个检测区域起始并且沿着顺时针方向或逆时针方向移动至圆周上的一个检测区域,然后以相反的方向从该检测区域再次沿着圆周移动到圆周上的任意一个检测区域而结束,其中,圆周上的8个检测区域由8个不同的数值表示,其中,所述编码由输入轨迹的起始点位置和结束点位置的数值表示。

[0013] 根据本发明的另一方面,提供了一种输入法,包括:在输入装置上确定用户输入的输入轨迹;确定与输入轨迹的图案对应的编码;基于编码与符号文字的映射表,确定并输出与编码对应的符号文字,其中,所述输入装置具有包括分布在线段上的8个检测点,所述输入轨迹从线段上的8个检测点中的任意一个检测点起始并且沿着线段的一个方向移动至线段上的一个检测点,然后以相反的方向从该检测点再次沿着线段移动到线段上的任意一个检测点而结束,其中,8个检测点由8个不同的数值表示,其中,所述编码由输入轨迹的起始点位置和结束点位置的数值表示。

附图说明

[0014] 通过下面结合附图对实施例进行的描述,本发明的这些和/或其他方面和优点将会变得清楚和更易于理解,其中:

[0015] 附图1为根据本发明实施例的三进制计算环境的架构图;

[0016] 附图2为根据本发明实施例的“比合”三进制全加法器的逻辑结构图;

- [0017] 附图3为根据本发明实施例的“比合”三进制全加法器的逻辑电路图；
- [0018] 附图4为根据本发明实施例的基于光路和电路的逻辑运算元件的示意图；
- [0019] 附图5为根据本发明实施例的“比合”三进制的光路全加法器的各部分功能实现示意图；
- [0020] 附图6为根据本发明实施例的象数图符号轨迹编码原理和示意图；
- [0021] 附图7为根据本发明实施例的象数图输入法在移动终端上的应用示意图；
- [0022] 附图8为根据本发明实施例的象数图符号全键盘和穿戴设备示意图；
- [0023] 附图9为根据本发明实施例的象数图实体立体键盘和虚拟键盘输入方法示意图；
- [0024] 附图10为根据本发明实施例的象数图卡片输入装置的示意图。

具体实施方式

[0025] 下面将参照附图来详细描述本发明的实施例的各个方面。为了与传统计算机相区别，为根据本发明的计算机系统定义了一些概念以更好地理解本发明的理念。将在以下详细描述本发明的计算机系统所采用的计算机架构和理论。以下的说明中定义的术语将仅作为示例性的描述而不是限制本发明的范围。

[0026] 为了与传统计算机系统区别，根据本发明的基于二进制和三进制的混合运算的计算机系统被称为TRIA计算机系统。根据本发明的TRIA计算机系统是一种二维阵列的矩阵计算机系统。相应地，为了将应用于根据本发明的TRIA计算机系统各种概念和术语与现有技术中使用的概念和术语进行区别，在各个术语之前增加“TRIA”。例如，“TRIA计算机系统”指的是根据本发明的计算机系统，而将“现有的计算机系统”或“计算机系统”指的是现有的二进制计算机系统。

[0027] 如上所述，对称三进制(1,0,-1组成)属于特殊的三进制(常用的不对称三进制0,1,2组成)，每个基本信息单元在表示数据方面也需要三个符号，而现有的计算机系统则是基于二进制的计算系统，其只能通过“0”和“1”来表示两种“数”或两种“状态”。换言之，现有的基于二进制的计算机系统的基本单位的一个位只能表示“0”和“1”，其无法用一个位来表示对称三进制的“-1(以下，将用符号T表示)”、“0”和“1”三种状态。

[0028] 如上所述，对称三进制的每个基本单位是三值的状态变化。为了在二值的环境下实现三值的计算处理，本发明引入了TRIA“比合”三进制的概念来处理对称三进制运算。

[0029] 二进制、三进制、八进制、十六进制和十进制等，都是不同编码的组合形式的从一种状态到另一种状态之间变化的规则。量子计算中的量子比特是利用同一时刻两个状态的相干和叠加来表示一个量子重叠态的运行一个“昆比特位”(qubit)，其同时储存0和1。一个昆比特位能同时储存所有的2个二进制数的 $C1|0\rangle + C2|0\rangle$, $C1|0\rangle + C2|1\rangle$, $C1|1\rangle + C2|0\rangle$, $C1|1\rangle + C2|1\rangle$ 的四种叠加态。“比合”三进制也类似于量子“昆比特”(Qubit)的叠加态的表示方式，把对称三进制的一个单位“才特”(trit)通过2个二进制位组合成一个数组或阵列(array)方式，表示一位“比合”三进制单位“才亚”(tria, tri-array, 三元阵列数，又名“比合数”)，“比合”三进制的一位“才亚”与对称三进制的一位“才特”一一对应。与双螺旋DNA的遗传密码(A,T,C,G的双链上的组合)结构相似，“比合”三进制存储结构是双链的比特链条(“非”字形链条)，其中一个单链条上的一个比特对应于另一个单链条上的一个比特，二者相互结合成一位“才亚”信息单元，每链条上的每个“比特”值是三进制信息(正或反的1

或0信息)。通过6对比特位(12比特)的6位“才亚”信息单元组成的TRIA双链条,表示一个对称三进制的一个字节(“tryte”,和二进制的“byte”属于相同概念),因为“比合”三进制的双链条的特征,使用这种组合结构计算方式的“比合”三进制运算,既有二进制运算特征也有对称三进制的运算特征和优点,一个TRIA“比合”全加法器就能处理加法和减法,无需二进制计算的补码处理方式就能处理减法计算。

[0030] TRIA“比合”三进制编码规则和其他进制编码信息单元规则对照表,具体参照如下表1。

[0031] 表1:TRIA“比合”三进制信息单元编码和其他进制编码对比

[0032]	编码	二进制比特 B	十进制单位	量子比特 QB	DNA 编码	非对称三进制	对称三进制才特(trit)	比合三进制才亚(tria)
		1	1	01	A	1	1	TRIA[1, 0]
			2	10	C	2	-1 (T)	TRIA[0, 1]
		0	3	11	G	10	0	TRIA[0, 0]
			0	00	T	0		

[0033] TRIA“比合”三进制双链条组合结构和对称三进制字节(tryte)对照示例表,具体参照如下表2。

[0034] 表2:TRIA比合三进制和对称三进制一个字节各位之间比较组合方式

[0035]	三进制比较	组合形式	复数	3^5	3^4	3^3	3^2	3^1	3^0	排列形式
	“对称”三进制	混合链条	Z	1	T	0	1	T	0	1T01T0
	TRIA “比合”三进制	正数链条	X	1	0	0	1	0	0	([1, 0], [0, 1], [0, 0], [1, 0], [0, 1], [0, 0])
		负数链条	Y	0	1	0	0	1	0	

[0036] 即, $Z = 1T01T0 = (100100)_3 + (0T00T0)_3 = (100100)_3 - (010010)_3$;

[0037] $Z = T(X, Y) = ([1, 0], [0, 1], [0, 0], [1, 0], [0, 1], [0, 0])_Z$;

[0038] $Z = T(X, Y) = X(100100)_Z + Y(010010)_Z$; (X=+1和Y=-1的正反数据链条)

[0039] 为了更加清晰地理解传统对称三进制和“比合”三进制之间的区别,可参照表3所示的传统对称三进制和“比合”三进制之间的组合方式。

[0040] 表3:对称三进制和“比合”三进制组合方式上的区别

[0041]	a	3^1	T0	T0	T0	0	0	0	10	10	10	$+3^n$	00	00	01	0	0	1	10	10	11
	b	3^0	0T	00	01	T	0	1	0T	00	01	-3^n	TT	T0	T0	T	0	0	0T	00	00
	a+b	对称	TT	T0	T1	T	0	1	1T	10	11	比合	TT	T0	T1	T	0	1	1T	10	11

[0042] 从表3中可以看出,传统的对称三进制是按照T、0、1和T0、00、10的进位方式相加组合成一个对称三进制数,而“比合”三进制是正负区分叠加成一个对称三进制数,传统对称三进制是一种混比合数据链条而“比合”三进制是正反结合的两串数据链条。因为正反链条本身就有性质上的区别所以“反数”(负数)部分的T可替换成1表示,所以硬件能直接辨识和操作的数据类型和格式0、1两个编码的组合就能表示一个T、0、1三个编码组合的一个对称三进制数。

[0043] 为了简化对称三进制的长度,“对称”九进制是一种以-4(D)、-3(C)、-2(B)、-1(A)、0、1、2、3、4为基本字元的九进制,如 $(1T)_Z = (2)_N$ 或 $0Z1T = 0N2$ 、 $(TT)_Z = (D)_N$ 或 $0ZTT = 0ND$ 等编码表示,对称三进制(0Z开头)与“对称”九进制(0N开头)的关系如下表4。

[0044] 表4:对称三进制与“对称”九进制对应表

[0045]

对称三进制数	T	T	0	0	1	1	1	1	数独	4	A	B	洛数	A	4	C
十进制数	-4	-3	-2	-1	0	1	2	3		3	0	C		B	0	2
“对称”九进制表示编码	D	C	B	A	0	1	2	3		2	1	D		3	D	1

[0046] “对称”九进制两位数表示从-40(0Ndd)到+40(0N44)的81个数字,如下表5中 $(13)_{10} = (14)_N = 0N14$ 、 $(-14)_{10} = (B4)_N = 0Nb4$ 等编码表示,具体参照如下表。

[0047] 表5:“对称”九进制与十进制对应表

[0048]

十进制数		-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14	-15
“对称”九进制数		a	b	c	d	a4	a3	a2	a1	a0	aa	ab	ac	ad	b4	b3
十进制	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
“对称”九进制数	0	1	2	3	4	1d	1c	1b	1a	10	11	12	13	14	2d	2c

[0049] 如上所述,“比合”三进制是“对称”三进制的阵列形式的表现形式,实质是对三元逻辑和运算的不同组合编码方式方法,三元逻辑的“真”、“假”、“非常”态的三元特征跟二元逻辑运算上有所不同,具体参照如下表。

[0050] 表6:三元逻辑真值表,T逻辑假、0未知、1逻辑真

[0051]

三元逻辑真值列表	逻辑与				逻辑或				逻辑与非				逻辑或非				逻辑异或			
	T	T	0	1	T	T	0	1	T	T	0	1	T	T	0	1	T	T	0	1
	T	T	T	T	T	T	0	1	T	1	1	1	T	1	0	T	T	T	0	1
	0	T	0	0	0	0	0	1	0	1	0	0	0	0	0	T	0	0	0	0
	1	T	0	1	1	1	1	1	1	1	0	T	1	T	T	T	1	1	0	T
	逻辑合意				逻辑调和				逻辑非				备注: -1=T=T=False=假 0=0=R=None=无, 非真非假 1=1=I=True=真							
	T	T	0	1	T	T	0	1	T	T	0	1								
	T	T	0	0	T	T	T	0		1	0	T								
	0	0	0	0	0	T	0	1												
	1	0	0	1	1	0	1	1												

[0052] 具体来说,参考表6中三值逻辑真值和“比合”三元逻辑运算结果的比合真值运算,通过TRIA比合三元逻辑运算中更好的理解三元逻辑和二元逻辑之间的内在规律。

[0053] 比合三元逻辑的A、B、C为三元逻辑值是I、0和T的平衡三进制逻辑值相互对应的比合数。一种定义方式为 $I = [1, 0]$, $0 = [0, 0]$, $T = [0, 1]$ 。另一种定义方式为 $I = [0, 1]$, $0 = [0, 0]$, $T = [1, 0]$ 。这两种方式的比合三进制运算结果相反(通过比合逻辑非相互转换),其他编码方式 $I = [1, 1]$, $0 = [0, 0]$, $T = [1, 0] = [0, 1]$ 等编码方式,通过 $d = a \& a'$, $d' = a \wedge a'$ 等方法相互转换为默认形式,其他比合编码的变种 $T = [1, 1]$, $0 = [0, 0]$, $I = [1, 0] = [0, 1]$ 方式也是比合逻辑处理的保护范围。

[0054] 正反数用表1和2表示区分比合参数; $A = [a, a']$, $B = [b, b']$ 和三元运算运算结果值 $C = [c, c']$,其中a、b和c是正数位,a'、b'和c'反数位,可通过C语言或verilog HDL等来验证;使用分组逻辑和比合分式来标识正反比合间变化关系,如下:

[0055] 三元逻辑(非): $\text{NOT}(C) = [c', c]$;

[0056] 比合三元逻辑(非)分式表示法: $!C = ![\frac{c}{c'}] = [\frac{c'}{c}]$;

[0057] 三元逻辑(比差): $\text{ZON}(A) = [a \wedge (a \& a'), a' \wedge (a \& a')]$;

- [0058] 比合逻辑(比差)分式表达式: $\%A=\%[\frac{a}{a'}]=[\frac{a^{a \wedge (a \& a')}}{a' \wedge (a \& a')}] = [\frac{a \& (!a')}{a' \& (!a)}];$
- [0059] 三元逻辑(与): $A \& B = [a, a'] \& [b, b'] = [(a \& b), (a' | b')] = [c, c'] = C;$
- [0060] 比合三元逻辑(与)分式表示法: $A \& B = [\frac{a}{a'}] \& [\frac{b}{b'}] = [\frac{a \& b}{a' | b'}];$
- [0061] 三元逻辑(或): $A | B = [a, a'] | [b, b'] = [(a | b), (a' \& b')] = [c, c'] = C;$
- [0062] 比合三元逻辑(或)分式表示法: $A | B = [\frac{a}{a'}] | [\frac{b}{b'}] = [\frac{a | b}{a' \& b'}];$
- [0063] 三元逻辑(与非): $\text{NOT}(A \& B) = [(a' | b'), (a \& b)] = [c, c'] = C;$
- [0064] 比合三元逻辑(与非): $!(A \& B) = !([\frac{a}{a'}] \& [\frac{b}{b'}]) = ![\frac{a \& b}{a' | b'}] = [\frac{a' | b'}{a \& b}];$
- [0065] 三元逻辑(或非): $\text{NOT}(A | B) = [(a' \& b'), (a | b)] = [c, c'] = C;$
- [0066] 比合三元逻辑(或非): $!(A | B) = !([\frac{a}{a'}] | [\frac{b}{b'}]) = ![\frac{a | b}{a' \& b'}] = [\frac{a' \& b'}{a | b}];$
- [0067] 三元逻辑合意(同与): $A @ B = [a, a'] @ [b, b'] = [(a \& b), (a' \& b')] = [c, c'] = C;$
- [0068] 比合三元逻辑半加器进位(同与): $A @ B = [\frac{a}{a'}] @ [\frac{b}{b'}] = [\frac{a \& b}{a' \& b'}];$
- [0069] 三元逻辑调和(同或): $A \# B = [a, a'] \# [b, b'] = [(a | b), (a' | b')] = [c, c'] = C;$
- [0070] 比合三元逻辑(同或): $A \# B = [\frac{a}{a'}] \# [\frac{b}{b'}] = [\frac{a | b}{a' | b'}];$
- [0071] 三元逻辑(异或): $A \wedge B = [a, a'] \wedge [b, b'] = [(a | b) \& (a' | b'), (a | b') \& (a' | b)] = [c, c'] = C;$
- [0072] 比合三元逻辑(异或): $A \wedge B = [\frac{a}{a'}] \wedge [\frac{b}{b'}] = [\frac{(a | b) \& (a' | b')}{(a | b') \& (a' | b)}];$
- [0073] 三元逻辑异与(乘法):
- [0074] $A * B = [a, a'] * [b, b'] = [(a \& b) | (a' \& b'), (a \& b') | (a' \& b)] = [c, c'];$
- [0075] 比合三元逻辑(异与): $A * B = [\frac{a}{a'}] * [\frac{b}{b'}] = [\frac{(a \& b) | (a' \& b')}{(a \& b') | (a' \& b)}];$
- [0076] 三元逻辑加(累加位):
- [0077] $A + B = [a, a'] + [b, b'] = \text{ZON}[(a' \& b') | (\sim a \& b) | (\sim b \& a), (a \& b) | (\sim a' \& b') | (\sim b' \& a')];$
- [0078] 比合三元逻辑(半加器累加位): $[\frac{a}{a'}] + [\frac{b}{b'}] = \%[\frac{(a' \& b') | (\sim a \& b) | (\sim b \& a)}{(a \& b) | (\sim a' \& b') | (\sim b' \& a')}]。$
- [0079] 三元比合半加法器:
- [0080] 对称三进制的加法的进位值部分是三元逻辑合意(同与)操作结果值,累加值部分是三元逻辑(累加)操作结果值的组合。
- [0081] 三元逻辑(与)水平对称:
- [0082] $A \& \& B = [a, a'] \& \& [b, b'] = [(a \& b'), (a' | b)] = [c, c'] = C;$

[0083] 比合三元逻辑(与)分式表示法: $A \&\&B = [\frac{a}{a'}] \&\& [\frac{b}{b'}] = [\frac{a' \& b'}{a' | b}]$;

[0084] 三元逻辑合意(同与)平对:

[0085] $A@@B = [a, a'] @@ [b, b'] = [(a \& b'), (a' \& b)] = [c, c'] = C$;

[0086] 比合三元逻辑加法进位(同与): $A@@@B = [\frac{a}{a'}] @@@ [\frac{b}{b'}] = [\frac{a \& b'}{a' \& b}]$;

[0087] 三元逻辑加(累乘):

[0088] $A++B = [a, a'] ++ [b, b'] = [(a' | b') \& (\sim a | b) \& (\sim b | a), (a | b) \& (\sim a' | b') \& (\sim b' | a')]$;

[0089] 比合三元逻辑(累乘): $[\frac{a}{a'}] ++ [\frac{b}{b'}] = [\frac{(a' | b') \& (\sim a | b) \& (\sim b | a)}{(a | b) \& (\sim a' | b') \& (\sim b' | a')}]$;

[0090] 二元逻辑中包括AND, OR, XOR等3种可交换律逻辑在内一共有 $2^4=16$ 种不同逻辑结果, 三元逻辑的逻辑结果是 $3^9=19683$ 种不同逻辑结果, 其中可交换律逻辑包括以上列举的比合三元逻辑共729种, 其变化太多在此不一一列举。比合三元逻辑运算是通过一对二元逻辑组综合运算描述三元逻辑结果的运算方法, 使二元逻辑元件的组合来运算三元逻辑运算的方法, 通过表1的DNA或量子位和TRIA编码匹配方式来运算比合三进制的生物计算和比合三进制量子计算(量子单位以四元逻辑计算其变化值是 $4^{16}=4294967296$ 种), 通过比合逻辑把二元逻辑推算三元逻辑变化, 反过来三元逻辑运算中所组成的二元逻辑的变化规律, 二进制计算机模拟运输三进制计算机和对三进制计算机下运算二进制计算机模拟运算提供可操作的方法。使用低维度逻辑的比合方法研究更高维度的逻辑变化规律提供了一种方法和思路, 通过不同于传统代数逻辑运算的比合分数逻辑, 此比合方法可通过三元逻辑去研究四元逻辑变化规律, 比合逻辑对数理逻辑和人工智能等方面会有所帮助。

[0091] “对称”三进制的运算运算, 三元值的运算, 具体参照如下表。

[0092] 表7: 对称三进制“加+, 减-, 乘*, 除/”的行列计算表

[0093]	对称三进制运算表 行与列的真值	加法				减法				乘法				除法			
		+	T	0	1	-	T	0	1	*	T	0	1	/	T	0	1
		T	T1	T	0	T	0	T	T1	T	1	0	T	T	1	1/0	T
		0	T	1	1	0	1	0	T	0	0	0	0	0	0	0/0	1
		1	0	1	1T	1	1T	1	0	1	T	0	1	1	T	T/0	1

[0094] 如上所述, 为了解决对称三进制在二进制环境上的运算的方法, 本发明人引入了一个“比合”概念, 并通过“比合”概念的基础上引申和演算“阴阳比叁”的原理和对称三进制在二进制环境下的运算方法。“比合”是相同类型的不同形式的离散事物的相比相互配合的结合, 精确表现出另一个单位事物的表现处理方式。举个例子, 如我们的手臂两个相反功能的上肱二头肌和肱三头肌的收缩舒张相互配合来控制手臂的精确动作; 两只眼睛的相互配合视觉感受三维立体图像; 解释几何中的复数的实数部和虚数部(实际上相同实数的不同坐标轴上)的相比相合来精确描述二维空间上的一个点, 生活中到处有“比合”处理的踪影。通过此原理, 本发明处理运算对称三进制的“比合”三进制计算处理, “比合”文字符号, “比合”向量输入方法, “比合”角度输入方法, “比合”有限和无限等。

[0095] 如上所述, TRIA“比合”三进制数是一对数的组合(正反相结合)的相比应结合而成

描述一个对称三进制的三进制数,TRIA的一个对称数 $Z=T(x,y)$ 的代数形式 $Z=x+yj$,其中 x 是正数部(阳数部), y 是反数部(阴数部或负数部)而 y 的单位 j 是-1。通过两个二进位为变量值,采用类似复数(complex number)模式形式的实数部和虚数部相似的表示方式。在夹角为180度的坐标系上一个 Z 点的 X 轴和 Y 轴的解释几何形式表现跟夹角为90度的笛卡尔坐标系上的一个复数 Z 点的 X 轴和 Y 轴的几何表现形式相同,区别在于坐标系的夹角不同,这种正反相互结比合数是一种“比合数”(combine number),比合数是类复数 $f(x,y)$ 的特殊形式 $f(A,x,y)$,其中坐标轴 A 的夹角为180度(π)。

[0096] TRIA比合数包括二进制比合数和三进制比合数。其中一个TRIA三进制比合数是一个正反(阴阳结合)两部份的相互结合而成,表示一个对称三进制的一种比合数。其中 Z 是对称三进制数表示, $T(x,y)$ 是TRIA比合数表示,对称三进制数和TRIA“比合”三进制数是相互对应。以下为了区别于对称三进制,命名为TRIA“比合”三进制。TRIA比合三进制是一种复数形式的数,应理解一个TRIA比合数包含阴阳两部分,一个TRIA比合数表示一个对称三进制。通俗理解,一个铜币的正面反面结合成一枚硬币(或者,一张纸),相似的“比合”数以正数(正数)和反数(负数)的相互结合组成,一个“比合”单位值“政”(Z),如 $(1T)_Z=[10,01]_Z$ 编码表示。

[0097] 现有的计算机系统中,二进制运算单位(或,数据单位)通常是8位(一个字节)、16位(一个字)、32位或64位。因此,根据本发明的TRIA计算机系统的TRIA“比合”三进制运算单位可以是两个8位、两个16位、两个32位、两个64位,或6位、24位、48位、96位、192位、384位等不同位。

[0098] 根据本发明的TRIA计算机系统是能够执行二进制运算和对称三进制运算的计算机系统,因此若TRIA计算机系统中,二进制运算单位为8位,则执行对称三进制运算的对称三进制运算单位为两个8位。

[0099] 在下面的描述中,假设对称三进制的运算单位为两个8位,即,16位、两个字节或一个字。

[0100] 根据本发明的TRIA计算机系统中,通过二进制的两个运算单位来表示对称三进制的一个运算单位,具体地讲,两个二进制运算单位中的一半用于表示具有对称三进制中的“1”的位,而另一半用于表示具有对称三进制中的“T”的位,且对称三进制“T”、“0”和“1”分别对应于 $[0,1]$ 、 $[0,0]$ 和 $[1,0]$ 。在下面的描述,将表示对称三进制中的“1”的位的二进制单位的数据成为“X链数据”、“X链”或“正数链数据”,而将表示对称三进制中的“T”的位的二进制单位的数据成为“Y链数据”、“Y链”或“负数链数据”。

[0101] 正反叠合平行坐标系中的一个 Z 数是一对 X 轴和 Y 轴上的数的叠合表示,叠合点的表示跟单位距离的宽度无关,也就是说与具体使用的进制无关,可以是二进制,三进制,十进制等。现代计算机主流使用二进制系统计算和存储,通过类复数形式在 X 轴和 Y 轴上的数据以一对二进制表示相对应的对称三进制“叠合”表示方法和“比合”表示方法;

[0102] TRIA比合三进制是通过一对二进制的数据的“比合”表示一个对称三进制数据,其中应注意的是 X 和 Y 轴上的数据单位是三进制,而不是二进制,参数是以一对二进制数0和1表示的一个比合三进制数字。

[0103] 虽然在上面的实施例中,通过 $[0,0]$ 来表示对称三进制的“0”,但是也可以通过 $[1,1]$ 来表示对称三进制的“0”。在下面的描述中,若没有特殊声明,则通过 $[0,0]$ 来表示对称三

进制的“0”。

[0104] 下面结合附图,详细描述根据本发明的TRIA计算机系统。

[0105] 图1是示出根据本发明的计算机系统的框图。下面将参照图1描述其中的各个部分。

[0106] 如图1所示,根据本发明实施例的支持二进制和三进制的计算机处理环境(计算机系统)101中包括诸如CPU、内存、硬盘存储器、网络接口等硬件设备,在计算机处理环境101中内嵌的硬件设备管理层(BIOS/EFI固件)102中提供二进制与三进制开关模式103。也就是说,在BIOS/EFI固件中包括内嵌的虚拟机管理内核104和虚拟机客户机配置信息表文件105,虚拟机管理内核104和虚拟机客户机配置信息表文件105可被称为虚拟机管理模块VMM。当虚拟机106启动的时候,通过配置信息与对应的不同类型的虚拟存储介质文件107来组合成一个虚拟机,虚拟机启动后引导操作系统108。虚拟机配置信息109以一种特殊设备形式在客户机上配置相关应用服务。当用户通过手写板等输入终端设备写入不同的轨迹符号110时,根据本发明的三进制的输入系统可通过轨迹转换将轨迹符号110处理成“比合”轨迹编码111。随后通过TRIA计算机系统内部的转译成具体的ASCII码字符或Unicode编码字符112以提供给用户界面显示为文字符号。在TRIA三进制操作系统上通过“比合”三进制编码113和对称三进制之间的转译方式在TRIA计算机系统中进行处理计算,并且处理结果在三进制存储介质114上保存。TRIA虚拟机运行当中操作系统以设备形式实施变更配置信息给虚拟机管理层115,通过不同类型虚拟文件系统与配置信息快速组合116以部署新的虚拟机。TRIA主机之间可通过心跳方式117提供高可用性集群,通过配置信息的同步更新快速切换成灾备环境,提供稳定的云服务。具体的TRIA虚拟机启动操作步骤,参照如下表8。

[0107] 表8:TRIA计算机虚拟化操作步骤

[0108]

BIOS/EFI				
开启		TCPU	关闭=>	BCPU
TOS 三进制操作系统	启<=关闭	VMM	关闭=>启	BOS 二进制操作系统
开启 TVMM(TRIA VMM) 环境, (TCPU=1, VMM=1 时)				
Z(X,Y)	串联奇偶	VMCFG	并联正反	[X,Y]z
T=1 时, TCPU 三进制模式		CPUCFG 处理器配置	T=0 时, BCPU 二进制模式	
奇偶串链模式(Z 模式)		MEMCFG 内存配置	正反并链模式(XY 模式)	
操作系统虚拟类型	ZOSVDFS	DISKCFG 存储配置	XOSVDFS	YOSVDFS
应用系统虚拟类型	ZAPPVDFS	SYSCFG 虚拟环境和虚拟机配置接通信口模块 IP、ID 等	XAPPVDFS	YAPPVDFS
用户数据虚拟类型	ZUSRVDFS		XUSRVDFS	YUSRVDFS
串联式 TVM(TRIA VM) 虚拟机		NETWORK 网络配置	并联式 TVM(TRIA VM) 虚拟机	
		Other Devices 外设置		
T=1 时, TOS 三进制操作系统		系统启动时读取 VMCFGDEV	T=0 时, BOS 二进制操作系统	
提供三进制、二进制或混合模式的计算服务				

[0109] 根据本发明的实施例,TRIA计算机系统的硬件层提供对称三进制的运算环境,并且以硬件虚拟化的方式,TRIA运算系统(即,CPU)通过指令相互切换成二进制计算处理方式和对称三进制计算处理方式。

[0110] 当这种TRIA计算机系统被第一次启动时,例如,当TRIA计算机系统初次启动时,VMM中的程序将会启动。通过VMM的管理接口配置和生成各种二进制和对称三进制的虚拟机。当二进制或对称三进制虚拟机启动的时候,调用预定指令初始化一个CPU的标志位(“比合”三进制的标志位1,二进制标志位0),为此虚拟机上赋予一个切换成“比合”三进制,将对

称三进制计算的处理器提供硬件级的计算服务。通过在该对称三进制虚拟机上运行基于对称三进制的操作系统,可以在基于对称三进制的操作系统上运行应用程序。因此,当根据本发明的TRIA计算机系统用作基于云计算服务队服务器时,可以给云终端提供对称三进制的计算环境。

[0111] 如上所述,当根据本发明的TRIA计算机系统包括在多核多线程的处理器(CPU)时,可在VMM上同时运行不同模式的二进制计算环境和对称三进制计算环境。

[0112] 在现有技术中,云计算中基础设施即服务的实现方式是通过软件层进行资源整合(包括但不限于计算资源(例如,CPU和内存),网络资源,I/O资源,存储资源)和再次分配而完成。

[0113] 根据本发明的实施例TRIA计算机系统中,VMM环境中配置各种类型应用系统虚拟硬盘文件(VDF)和对应的配置信息文件,通过配置信息的调整方式调优虚拟机运行时的应用服务的最优化。通过一种特殊的虚拟配置设备形式运行于虚拟机上的客户环境上,通过读写此设备变更VMM上的配置信息。跟传统虚拟机做模板需要把操作系统和应用一起制作一个虚拟硬盘文件,TRIA虚拟机是把操作系统、各应用系统各自独立出各不同的虚拟系统文件上,通过配置信息文件与个虚拟系统文件的匹配方式,以搭积木形式快速组建虚拟环境。云计算环境部署可提供更加便捷的解决方案,并且双存储节点和计算节点的分离提高安全性。

[0114] 虽然,图1的TRIA计算机系统被示出为其包括了TRIA存储器、TRIA输入设备、TRIA输出设备、TRIA运算处理器和TRIA进制控制单元,但是其可被扩展为包括TRIA存储系统、TRIA输入系统、TRIA输出系统、TRIA运算系统等。

[0115] 下面将分别参照附图来介绍根据本发明实施例的TRIA计算机系统的各个系统组成,包括TRIA运算系统、TRIA内存系统、TRIA存储系统、TRIA输入系统。

[0116] <TRIA运算系统>

[0117] TRIA计算处理

[0118] 如本领域技术人员熟知的,不管计算机系统是二进制计算机系统还是三进制计算机系统,或者是其它进制的计算机系统,其核心运算是加、减、乘、除等基本数学运算以及与、或、非等基本逻辑运算,通过这些基本数学运算和基本逻辑运算的组合,可实现各种运算。因此,下面将介绍根据本发明的TRIA计算机系统的基本数学运算以及基本逻辑运算。

[0119] 如上所述,根据本发明的TRIA运算处理,包括TRIA全加法器、减法运算等四则运算和逻辑处理。其中,根据本发明实施例的TRIA全加法器可包括二进制和三进制切换控制电路、双加法器处理、双加法结果值的二维阵列的矩阵比较结合处理、比较结合二进制值的三进制转译处理和对称三进制转换处理等部分。将在稍后参照图2到图5进行详细描述。

[0120] 如上所述,根据本发明的TRIA数据存储结构中,通过X链和Y链数据的一对阵列值来表示三进制的1、0和T。因此,根据本发明实施例的采用TRIA比合三进制的全加法器可以在二进制运算和对称三进制运算之间切换。例如,针对加法运算,TRIA加法模块可被实现为TRIA双进制加法器,从而TRIA加法模块可根据进制T控制信号,使TRIA双进制加法器在二进制加法与对称三进制加法之间切换而进行加法运算。

[0121] 此外,针对TRIA CPU所包括的TRIA加法模块、TRIA减法模块、TRIA乘法模块、TRIA除法模块、TRIA逻辑与模块、TRIA逻辑或模块以及TRIA逻辑非模块,其中的一些模块可被实

现为二进制运算器 and 对称三进制运算器, 而另一些模块可被实现为双进制运算器。例如, TRIA 加法模块可被实现为如上所述的 TRIA 双进制加法器。换言之, 根据本发明的 TRIA CPU 可以上不同模块的组合来实现。

[0122] 此外, 针对加、减、乘、除、逻辑与、逻辑或以及逻辑非运算, 可通过一个硬件模块实现上述多个运算中的两个或多个运算。例如, 可通过加法器实现加法和减法运算。

[0123] 下面, 介绍根据本发明的 TRIA 三进制的加法、减法的实现方式。

[0124] TRIA “比合” 三进制全加法器运算

[0125] 加法是算术运算中最基本的运算, 因此如果想搭建一台计算机, 那么首先就要造出可以计算两个数的和的器件。加法计算就是计算机要做的唯一工作。如果可以造出加法器, 同样地, 就可以利用加法来实现减法、乘法和除法。

[0126] 根据本发明的 “比合” 三进制全加法器 (以下简称, T 全加法器) 可通过控制信号不同而切换成为二进制全加法器和三进制全加法器。当控制信号 0 时, 进行一对二进制全加法器处理; 当控制信号 1 时, 进行 “比合” 三进制全加法器处理。如图 2 所示, 根据本发明实施例的 T 全加法器 201 的输入信号可包括: 进制切换控制信号 202、一对正反信号 203 和 204, 即表示比合三进制数的正数部的 A、B 输入信号与 C 进位信号的 203, 表示比合三进制数的反数部 (负数部) 的 a、b 输入信号与 c 进位信号 204。T 全加法器 201 可包括: 二进制与三进制切换逻辑门电路 205; 一对二进制加法器, 包括正加法器 206 和反加法器 207; “比合” 处理逻辑门电路 208; 二进制转三进制逻辑门电路 209。T 全加法器 201 的输出是一对加法结果输出信号 210 和进制切换信号。

[0127] 当处理需要时, 通过 “比合” 转换 “对称” 三进制的处理单元, 可将 T 全加法器的输出 210 转换成对称三进制的值输出。 “比合” 和 “对称” 三进制之间, 可以通过逻辑门 ‘电路’ 或逻辑门 ‘光路’ 相互间转换。将在稍后参照图 3-图 5 介绍利用电路逻辑门元件和利用光路逻辑门元件实现 T 全加法器 201 的构思。

[0128] 下面介绍对称三进制加法和比合三进制加法之间的关系。

[0129] 首先, 本领域的技术人员知道, 对称三进制加法可表示为如下:

[0130] $0+0=0, T+T=T1, T+0=T, T+1=0, 0+T=T, 1+1=1T, 1+0=1, 1+T=0, 0+1=1$

[0131] 这样, 可以得出根据本发明实施例的比合三进制加法可表示为:

[0132] $Z = Z1 + Z2 = T(X1, Y1) + T(X2, Y2) = (X1 + Y1j) + (X2 + Y2j) = (X1 + X2) + (Y1 + Y2)j = T(X1 + X2, Y1 + Y2) = [X1 + X2, Y1 + Y2]$

[0133] 将 TRIA 比合三进制加法和对称三进制加法之间比较, 具体参照如下表 9。

[0134] 表 9: TRIA 比合三进制和对称三进制加法真值表

[0135]

加法	现有 “对称” 三进制			TRIA “比合” 三进制			
----	-------------	--	--	---------------	--	--	--

[0136]

+	T	0	1	+	[0, 1]	[0, 0]	[1, 0]
T	T1	T	0	[0, 1]	[01, 10]	[0, 1]	[0, 0]
0	T	0	1	[0, 0]	[0, 1]	[0, 0]	[1, 0]
1	0	1	1T	[1, 0]	[0, 0]	[1, 0]	[10, 01]

[0137] 根据上面的表 8 可以看出, 将对称三进制加法和比合三进制加法比较, TRIA 比合三

进制的加法特征是：同位相异时本位1；同位同时0时本位0；同位同时1时本位清零、进位1、平移到相反的位1（唯一进位变动情况）。这样，根据以上的规则可以将正数部累加值S（以下，S符号表示正数部累加值）、正数进位值G（以下，G符号表示正数部进位值）、负数部累加值s（以下，s符号表示负数部累加值）和负数部进位值g（以下，g符号表示负数部进位值）变化总结为如下，具体参照如下表10。

[0138] 表10:TRIA比合加法运算阵列方式，各位真值变化关系表

对称数	比合数	相对位置		正和位 S				负和位 s				正进位 G				负进位 g			
[0139] 对称三进制	正数	进位 G	本位 S	S	0	0	1	s	1	0	0	G	0	0	1	g	1	0	0
				0	1	0	0	1	0	1	0	0	0	0	0	1	1	0	0
	反数	反进位 g	反位 s	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
				1	0	1	0	0	0	0	1	1	0	0	1	0	0	0	0

[0140] 如上所述，根据本发明的对称三进制和TRIA“比合”三进制的加法运算对应关系：

[0141] $T+T=>T1 \Leftrightarrow [0,1]_Z+[0,1]_Z=>[0,1][1,0]_Z$ ；

[0142] $1+1=>1T \Leftrightarrow [1,0]_Z+[1,0]_Z=>[1,0][0,1]_Z$ ；

[0143] $0+T=>T \Leftrightarrow [0,0]_Z+[0,1]_Z=>[0,1]_Z$ ； $0+1=>1 \Leftrightarrow [0,0]_Z+[1,0]_Z=>[1,0]_Z$ ；

[0144] $0+0=>0 \Leftrightarrow [0,0]_Z+[0,0]_Z=>[0,0]_Z$ ； $1+T=>0 \Leftrightarrow [1,0]_Z+[0,1]_Z=>[0,0]_Z$ ；

[0145] 根据本发明的T全加法器201具有正数部和负数部共6个输入值，其中包括正数部3个输入值（A、B与进位C），负数部3个输入值（a、b与进位c）；4个输出值：其中包括，正数部2个输出值（进位G和累加值S）；负数部2个输出值（进位g和累加值s）。

[0146] T全加法器201和二进制全加法器相比，二进制全加法器有三个输入值的 $2^3=8$ 种不同变化信号输入值和2个输出值 $2^2=4$ 种不同结果信号，而T全加法器201的输入值是6个输入值的变化 $2^6=64$ 个不同输入信号和4个输出值 $2^4=16$ 种不同信号结果值。64种不同信息输入通过逻辑门判断，输出到16种不同信息变化的矩阵运算，结果计算判断输出7种不同（-3和+3各1种、-2和+2各6种、-1和1各15种，0位20种排列组合共64种输入编码和7种输出编码）阵列结果。实现这样的输入值和输出值的对应关系的逻辑电路是错综复杂的。为了解决T全加法器201与二进制全加法器相比几何数增加的信息输入的矩阵运算处理的逻辑门计算，本发明通过复用二进制全加法器的逻辑门处理部分、“比合”运算逻辑门的处理部分和二进制转译三进制逻辑门处理部分的相互配合与组合逻辑门的位操作运算，完成TRIA“比合”全加法器单元的计算处理，并且TRIA二进制和三进制切换逻辑门的处理部分完成2进制和3进制的全加法器计算的相互切换。

[0147] 下面将参照图3来详细说明根据本发明实施例的T全加法器201的具体电路结构和工作方式。附图3-301的正数部的3个输入值（A、B、C）和负数部3个输入值（a、b、c）各自并行计算处理，在附图3-302的正、反（负）二进制全加法器逻辑运算处理后，二进制矩阵比合逻辑运算电路303对二进制全加法器302输出的正数部的进位值G和负数部的进位值g之间进行相互“比合”运算，并对正数部的累加值S和负数部的累加值s之间进行相互“比合”运算。经过比合处理后的一对进位值（G、g）和一对累加值（S、s）的二进制“比合”数[G,S]和[g,s]通过附图3-304的二进制转译三进制处理输出成对称三进制值的三进制“比合”数[G,S]和[g,s]，完成T全加法器计算处理。

[0148] 其中“比合”处理方法是两个输入值相异时不变化输出两个输出值,两个输入值相同时同时清零处理后输出到两个输出值逻辑门处理方法。

[0149] 下面将分别介绍图3中的比合逻辑运算电路303以及三进制转译逻辑电路304的工作原理。

[0150] 比合运算处理(combine)

[0151] T全加法器201中比合处理的数学原理是0的分化处理 $0=A+(-A)$ 方式。“比合”计算中把[1,1]转换处理成[0,0]形式,如下公式推导。

[0152] $Z=T(X,Y)=X+Yj, T(A,A)=A+Aj=0; Z=Z-0=T(X,Y)-T(A,A)=T(X-A,Y-A);$

[0153] $Z=Z_1+Z_2-0=T(X_1,Y_1)+T(X_2,Y_2)-T(A,A)$

[0154] Z函数形式= $T(X_1,Y_1)+T(X_2-A,Y_2-A)$ 或= $T(X_1,Y_1)+T(X_2-A,Y_2-A)$

[0155] Z比合数形式= $[X_1+Y_1j]+[(X_2-A)+(Y_2-A)j];$

[0156] 或= $[(X_1-A)+(Y_1-A)j]+[X_2+Y_2j]$

[0157] TRIA比合二进制B的二进制“比合数”表示方法,单位是2。

[0158] $B_x=\sum X=x_n2^n+x_{n-1}2^{n-1}+\dots+x_22^1+x_02^0; (x\{1,0\});$

[0159] $B_y=\sum Y=y_n2^n+y_{n-1}2^{n-1}+\dots+y_22^1+y_02^0; (y\{1,0\});$

[0160] $B=(x_n-y_n)2^n+(x_{n-1}-y_{n-1})2^{n-1}+\dots+(x_0-y_0)2^0; (x,y=\{1,0\});$

[0161] $=B_nB_{n-1}\dots B_1B_0;$

[0162] 如下表格11是一对二进制全加法器计算结果的阵列的比合操作表。如表10所示,T全加法器201的一对正反输入值包括正数部输入值A、B和进位C以及反数部输入值a、b和进位c的二进制全加法器加法结果(0,0),(0,1),(1,0),(1,1)值,正数部(G,S)和反数部(g,s)的(进位值,累加值),通过一对比合数的进位值[G,g]之间G与g和累加值[S,s]之间S和s的阵列比合处理,完成二进制比合处理输出(G,S)和(g,s)的比合数,具体的“比合”处理真值表,参照如下表11。

[0163] 表11:双二进制计算输出的“比合”计算表,其中

双2进制全加法器			“比合”二进制输出表		(0,0)	(0,1)	(1,0)	(1,1)
(G,S)和(g,s)			双二进制 (G,S),(g,s) “比合”处理 [G,g][S,s]	(0,0)	(0,0),(0,0)	(0,0),(0,1)	(0,0),(1,0)	(0,0),(1,1)
B+	(0,0)	(1,0)		(0,1)	(0,1),(0,0)	(0,0),(0,0)	(0,1),(1,0)	(0,0),(1,0)
0,0	(0,0)	(1,0)		(1,0)	(1,0),(0,0)	(1,0),(0,1)	(0,0),(0,0)	(0,0),(0,1)
0,1	(0,1)	(1,1)		(1,1)	(1,1),(0,0)	(1,0),(0,0)	(0,1),(0,0)	(0,0),(0,0)

[0165] 图3的比合逻辑门处理电路单元

[0166] “比合”处理单元逻辑电路方法

[0167] 为了清晰表示,根据本发明的“比合”运算处理的方法,结合附图3-303中的虚框包含的“二进制矩阵比合逻辑运算电路”,具体描述。首先,正反二进制的结果输出正进位G、正累加S(二进制里的进位值输出C和累加值输出S)和反进位g、反累加s(相对于第一个二进制加法相比,另一个加法器输出值),进位值G与g和累加值S与s之间各自“比合”处理后输出结果值。

[0168] 结合附图4-402的“比合”处理是输入值A和a,通过输入值A与输入值a之间与操作(AND)后结果值,再次跟输入值A之间通过异或操作(XOR)输出为B结果值,以下表是双二进制全加法器输出的一对比合操作中的一个比合操作逻辑门电路的输出结果,逻辑表达式是

$B = (A \& a) \wedge A, b = (A \& a) \wedge a$, 具体参考如下表。

[0169] 比合处理分式表达式(比差): $\%A = \%[\frac{a}{a'}] = [\frac{a \wedge (a \& a')}{a' \wedge (a \& a')}]$ 。

[0170] 原“比合”处理和比合逻辑名称之间混淆而改为“比差”处理, 比合数 $A = [a, a']$ 的正反数相差 ($A = a - a'$) 结果等价, $[1, 0] = 1 - 0 = 1, [0, 0] = 0 - 0 = 0, [1, 1] = 1 - 1 = 0, [0, 1] = 0 - 1 = -1$; 比合逻辑中一对比合数 a 和 a' 的数组 $[a, a']$ 表示相比用分式表示 $A = \frac{a}{a'}$ 更加合理和简单易懂, 可以更容易比较比合逻辑中两个正反比合数之间的逻辑运算的变化关系, 更容易区分二进制逻辑和比合三进制逻辑之间的差异, 传统数理逻辑在形式上整数逻辑表达式, 比合三进制逻辑在形式上分式逻辑表达式, 如表12比差(比合)处理。

[0171] 表12: TRIA比合处理的逻辑电路处理真值表

二进制比合数	位	入	出	“比合”运算电路	入	出	入	出	入	出	入	出
[1, 0], [0, 0], [0, 1]	正	a	b	$b = a \wedge (a \& a') B = \text{XOR}(AND(A, a), A)$	0	0	1	1	0	0	1	0
对称“比合”值 1,0,-1	反	a' A	b' b	$b' = a' \wedge (a \& a') \text{XOR}(AND(a, A), a)$	0	0	0	0	1	1	1	0

[0173] “比合”处理单元光干涉等效处理方法

[0174] 下面将参照图4来说明根据本发明实施例的采用光学元件实现根据本发明的比合处理的示意图。

[0175] “比合”处理在不仅在T全加法器中, 在TRIA计算机处理中“比合”三进制信息的传输处理、正反并行结构双链位操作、串行结构奇偶位的存储上也使用, 为了更加有效的处理, 根据本发明的“比合”处理可通过光学方法等效处理来实现计算逻辑门电路的“比合”处理。

[0176] 若干个光波(成员波)相遇时产生的光强分布不等, 由各个成员波单独造成的光强分布之和, 而出现明暗相间的现象。光学中两列光波的频率相同, 相位差恒定, 振动方向一致的相干光源, 才能产生光的干涉。光波长 λ 通过双缝到屏上发生干涉, 如果路程差是半波长的偶数倍, 即 $\Delta X = 2n\lambda/2$; 那么产生的是亮条纹, 如果半波长的奇数倍, 那么产生的是暗条纹 $\Delta X = (2n+1)\lambda/2$ 。光电耦合器是以光为媒介传输电信号的一种电-光-电转换器件。它由发光源和受光器两部分组成。把发光源和受光器组装在同一密闭的壳体内, 彼此间用透明绝缘体隔离。在光电耦合器输入端加电信号使发光源发光, 光的强度取决于激励电流的大小, 光照射到封装在一起的受光器上后, 因光电效应而产生了光电流, 由受光器输出端引出, 这样就实现了电-光-电的转换。发光源的引脚为输入端, 受光器的引脚为输出端, 常见的发光源为发光二极管, 受光器为光敏二极管、光敏三极管等。

[0177] 根据本发明干涉光等效“比合”处理, 附图4-403中的比合逻辑门光路等效处理器单元可通过光学元器件, 如光电耦合器等来实现。至少以半个波长的五倍为直径的一个单位圆空间上, 圆周上布置投光器和受光器, 通过A和a上投光器两个入射光和B和b上受光器的光检测装置组成, 其中入射光单位长度为半个光波长 $\lambda/2$, A与a和B与b在一个同心圆上。A与b和a与B在同心圆的两个直径距离, 长度为 $5 * (\lambda/2)$, 入射光源A与a的距离为 $3 * (\lambda/2)$, A与B和a与b的投射距离为 $4 * (\lambda/2)$ 。入射光源A和a, 以同频率、同相位、同方向入射(输入信号A

$=1, a=1$)到密闭的壳体内,AB与aB和ab与aB的光路程相差半个波长,受光器B和b的上两个光波的叠加而显示暗光,使B和b的受光器同时为输出值信号0(输出信号 $B=0, b=0$)。入射光源A与a之中,只有一个入射时,没有光的干涉,当A入射光源信号1同时a入射光源信号0时($A=1, a=0$),B受光器输出信号1、b受光器无光信号检测而输出信号0($B=1, b=0$);同理,当a入射光源信号1时同时A入射光信号0时($A=0, a=1$),b受光器输出信号1、B受光器无光信号检测而输出信号0($B=0, b=1$);入射光源A与a同时无光信号时(输入信号 $A=a=0$),受光器B与b同时无光信号检测而输出信号为0(输出信号 $B=b=0$)。其中入射光源距离A与a用勾股数3的半个光波长相差倍数,A与B和a与b的垂直投射距离用勾股数4的半个光波长相差倍数,按照勾股定理干涉光源A与b和a与B之间的距离为勾股数5的半个光波长相差倍数,两个干涉光路程相差 $5-4=1$ 一个半个光波长的倍数,使B和b受光器在光干涉时,同时光波峰和光波谷相互抵消而产生暗光。本发明通过光的干涉原理和勾股定理,在最小的空间上实现“比合”逻辑门等效处理,此等效“比合”干涉光处理方法,不仅可用在光电耦合器上,也可以适用于全光器元器件(全光器,光控制光的光学元器件)上。以下是干涉光“比合”处理的表(表中入A与a,出B与b信号值)。应理解,这里的3、4、5倍半波长仅是勾股数的一个示例,只要满足 $2n+1, 2n^2+2n, 2n^2+2n+1$ 的勾股数关系即可,勾股数的目的是为了两个干涉光之相差达到半个波长相差,使两束干涉光同时输入信号相互抵消作用比合计算目的,不影响本发明的实现目的的情况下,应该理解,其他长度的调整的变动、修改及其等同物全部包括在本发明的范围内。

[0178] 综上所述,“比合”干涉光处理是一对入射光源与一对投射输出光之间的半个光波长相差距离,通过光的衍射和干涉现象,改变投射部分的明暗值的变化规律,计算处理“比合”操作,具体的处理“入射值”和“输出值”变化关系,参考如下“比合”等效光路表13。

[0179] 表13:TRIA比合三进制处理“比合”逻辑门等效光路

[0180]	比合操作 光干涉等 效处理	勾股定理在光干涉效应的“比合”处理			入	出	入	出	入	出	入	出
		A 入	AB 与 aB 光路程半个光波相差	B 出	0	0	1	1	0	0	1	0
		a 入	ab 与 Ab 光路程半个光波相差	b 出	0	0	0	0	1	1	1	0

[0181] 通过“比合”光元器件替代“比合”电元器件,提高处理效率和处理信息密度。“比合”干涉光等效处理单元替代电路的效益是,第一,光器件允许通过的光频率高、范围大,也就是所谓的带宽非常大,传输和处理的信息量极大。两束光要发生干涉,必须频率相同,振动方向一致和有不变的初始相位差。第二,信息传输中畸变和失真小,信息运算速度高。第三,光传输和转换时,能量消耗极低。使用光波而不是电流来处理数据和信息对于计算机的发展而言是非常重要的一步;摩尔定理也接近临界值,在将来,光计算机将为我们带来更强劲的运算能力和处理速度。

[0182] 附图5的502示出二进制光路全加法器,其中包括示出二进制半加器光路元件,通过两个光路半加器的进位光的聚焦作为进位光接收器的二进制进位值C,两个光路半加器的串联结果值S的相互结合一个二进制全加法器的光路逻辑运算,这和二进制全加法器的逻辑结构相似。

[0183] 以上介绍了根据本发明的光干涉“比合”处理等效勾股数光学元器件。下面将介绍“比合”三进制“才亚”转对称三进制“才特”(trito)的光学元器件在“比合”光学元器件的基础上改进,详细内容在“才亚”转“才特”部分中描述。

[0184] “比合”二进制转译“比合”三进制处理(tria2to3)

[0185] 在本申请中,TRIA比合三进制Z是类复数 $f(x,y)$ 的三进制“比合数”,其单位是3。

[0186] $Z_x = \sum X = x_n 3^n + x_{n-1} 3^{n-1} \cdots + x_3 3^2 + x_2 3^1 + x_0 3^0; (x \in \{1, 0\})$;

[0187] $Z_y = \sum Y = y_n 3^n + y_{n-1} 3^{n-1} \cdots + y_3 3^2 + y_2 3^1 + y_0 3^0; (y \in \{1, 0\})$;

[0188] $Z_t = |\sum X| - |\sum Y| = x_n 3^n + x_{n-1} 3^{n-1} \cdots + x_2 3^1 + x_0 3^0 - y_{n-1} 3^{n-1} \cdots - y_1 3^1 - y_0 3^0 = (x_n - y_n) 3^n + (x_{n-1} - y_{n-1}) 3^{n-1} + \cdots + (x_1 - y_1) 3^1 + (x_0 - y_0) 3^0; (x, y = \{1, 0\})$;

[0189] $= (Z_n) 3^n + (Z_{n-1}) 3^{n-1} \cdots + (Z_1) 3^1 + (Z_0) 3^0; (Z = \{1, 0, T\})$;

[0190] $= Z_n Z_{n-1} \cdots Z_1 Z_0$;

[0191] 其中对称三进制1表示,即: $Z_n = 1 = x_n - y_n; (x = 1, y = 0)$;

[0192] 表示对称三进制0,即: $Z_n = 0 = x_n - y_n; (x = 1, y = 1 \text{ 或 } x = 0, y = 0)$;

[0193] 表示对称三进制T(T为-1),即: $Z_n = T = -1 = x_n - y_n; (x = 0, y = 1)$;

[0194] 二进制比合数和三进制比合数之间不同编码方式,-3、-2、2、3的二进制对称数BB、B0、10、11的编码,在三进制对称数T0、T1、1T、10编码表示;二进制对称数-1和1各有两种编码方式0B、B1和01、1B编码,用三进制对称数T和1编码表示。T全加法器201中,经过tria2to3逻辑门处理的位操作转译,使“比合”二进制与“比合”三进制的阵列之间转译处理,输出三进制进位的对称三进制数,以下表14是比合二进制和比合三进制转译比较表。

[0195] 表14:T全加法器处理,比合二进制与比合三进制转译表(符号B=-1,T=-1)

[0196]

十进制	-3	-2	-1	-1	0	1	1	2	3
2进制比合	0-3=-3	0-2=-2	0-1=-1	1-2=-1	0-0=0	1-0=1	2-1=1	2-0=2	3+0=3
B(正)(反)	(0,0)(1,1)	(0,0)(1,0)	(0,0)(0,1)	(0,1)(1,0)	(0,0)(0,0)	(0,1)(0,0)	(1,0)(0,1)	(1,0)(0,0)	(1,1)(0,0)
比合2进制	[0,1][0,1]	[0,1][0,0]	[0,0][0,1]	[0,1][1,0]	[0,0][0,0]	[0,0][1,0]	[1,0][0,1]	[1,0][0,0]	[1,0][1,0]
对成2进制	BB	B0	0B	B1	00	01	1B	10	11
比合3进制	[0,1][0,0]	[0,1][1,0]	[0,0][0,1]	[0,0][0,1]	[0,0][0,0]	[0,0][1,0]	[0,0][1,0]	[1,0][0,1]	[1,0][0,0]
对称3进制	T0	T1	T	T	0	1	1	1T	10

[0197] 如上所述,为了解决T全加法器201的“比合”二进制阵列运算和“比合”三进制运算结果之间的矛盾,根据本发明的实施例T全加法器201中的tria2to3逻辑门处理单元304在完成双二进制加法处理后,可进行“比合”二进制输出值的“比合”三进制的转译处理。

[0198] 为了清晰表示,结合附图3-304中的虚框包含的TRIA比合三进制转译逻辑电路来说明根据本发明的比合二进制转译三进制的方法。

[0199] 其中,二进制和三进制之间转译处理方法是:当正部(正数链)和反部(负数链)的两个进位值(G,g)逻辑异或(XOR)判断结果为0时,即两个输入值(G,g)相同时直接输出各自的相应的“比合”处理后的结果值。当两个进位值(G,g)逻辑异或(XOR)判断结果为1时,即两个输入值相异时,继续进行下一步的判断,其中,当正进位值G是1时,正进位值G和负累加值s相互逻辑异或(XOR)处理后的结果输出到正进位G,正进位值G和负累加值s之间与操作(AND)处理后结果输出到正累加值S,反进位值g直接输出到反进位值g,正进位值G、正累加值S和反累加值s之间的异或操作(XOR)处理结果输出到反累加值s;当反进位值g为1时,正进位值G直接输出到正进位值G,反进位值g、反累加值s和正累加值S之间(XOR)异或操作后结果输出到正累加值S上,反进位值g和正累加值S之间(XOR)异或操作后结果输出到反进位值g上,反进位值g和正累加值S之间(AND)与操作后结果输出到反累加值s上,这样,完成二进制和三进制“比合”值之间转译处理。

[0200] 为了清晰表示根据本发明的T全加法器201的整个处理过程,具体参照如下表15。

[0201] 表15: TRI-10A“比合”三进制处理过程(正与反的真值处理过程)

输入值	二进制计算处理	二进制比合处理 2^n	二、三进制的比合转换后输出处理 3^n	正进位 G	正累加 S
				反进位 g	反累加 s
[0202] 正部 A, B 和进位 C	正二进制全加法器	进位比合处理 ($2^n G/2^n g$)	两进位值的异或值 XOR (G, g)=0 时, 直接返回各值 (G, S, g, \underline{s})	G	S
			进位值 $G=1$ 时, 2 和 3 比合转换处理后, 返回结果 (G, S, g, \underline{s}),	\underline{g}	\underline{s}
反部 a, b 和进位 c	反二进制全加法器	累加比合处理 ($2^n S/2^n \underline{s}$)	进位值 $g=1$ 时, 2 和 3 比合转换处理后, 返回结果 ($G, S, \underline{g}, \underline{s}$),	$G=G \wedge \underline{s}$	$S=G \& \underline{s}$
			进位值 $g=1$ 时, 2 和 3 比合转换处理后, 返回结果 ($G, S, \underline{g}, \underline{s}$),	$\underline{g}=\underline{g}$	$\underline{s}=\underline{g} \wedge \underline{s}$

[0203] TRIA“比合”全加法器 TRI-10B 的方法2: 首先正数部的3个输入值 (A, B, C) 和反数部3个输入值 (a, b, c), 经过“比合”和转译等, 逻辑门处理后结果值输出到两个二进制全加法器上, 模拟二进制全加法器结果输出为跟“比合”三进制计算处理结果相同的计算输出结果 [G, S] 和 [g, s] 的“比合”三进制处理方法。

[0204] TRIA“比合”全加法器 TRI-10C 的方法3: 首先正数部的3个输入值 (A, B, C) 和反数部3个输入值 (a, b, c), 先正反“比合”处理后, 整数部 (A, B, C) 二进制加法处理结果 ($G1, S1$), 反数部 (a, b, c) 二进制加法处理结果 ($g1, s1$), 在经过二进制转三进制比合转译模块处理, 结果输出 [G, g] [S, s]_z 比合三进制数。

[0205] “比合”三进制“才亚”转对称三进制“才特” (trito)

[0206] 如上所述, 根据本发明的 TRIA“比合”三进制的一个信息单位“才亚”通过逻辑门处理转换成“对称”三进制的一个信息单位“才特”。通过“才特”的一个信息单元的三元值, 在可存储一个信息单元由三值状态的存储信息单元上存储信息, 使存储的密度提高。这样, 可以为云计算和大数据等所带来的大容量存储需求提供存储资源。跟相同位数的二进制存储相比三进制存储在单位存储上提供更高的信息密度 ($3^{16}/2^{16}=656$ 倍, $3^{32}/2^{32}=431439$ 倍的信息量)。

[0207] “比-对”三进制处理单元逻辑电路方法

[0208] 如上所述, 根据本发明的 TRIA“比合”三进制的“才亚” (tria) 信息阵列 [0, 1]、[0, 0]、[1, 0] 转换成“对称”三进制的“才特” (trit) 的 T、0、1 三值, 可根据如图3所示的“比-对”转换逻辑门电路处理单元完成。为了更加清晰解释, 参照附图4-405的“比合-对称”三进制三进制电路部分, 处理单元包括“才亚”信息 A 与 a 的信息输入和 I、R、T 三个信息输出。具体信息处理方式是 A 与 a 的阵列组合信号为 [A, a] = [1, 0] 时, 直接输出到 I 电路输出; 同理, A 与 a 的阵列组合信号为 [A, a] = [0, 1] 时, 直接输出到 I 电路输出; A 与 a 的阵列组合信号为 [A, a] = [0, 0] 时, 首先, 输入值 A 与 a 逻辑或 (OR) 位操作后, 结果值再次逻辑非 (NOT) 位操作结果值成为 1, 在电路 R 输出。通过此处理方法, “才亚”信息转换为“才特”信息, 通过“才特”信息的三值, 在不同的三元存储介质上写入三进制“才特”值。

[0209] 综上所述, 通过两个输入值的逻辑门操作, 输出三个不同状态的真值, 将一对阵列信息的变化转变成不同的三个状态值, 输出三值状态, 改变阵列信息转换成三值信息, 完成对称三进制输出的方法, 参照如下表格16的比-对信息转真值表。

[0210] 表16: TRIA比合三进制转对称三进制处理逻辑门电路真值表

[0211]	比合三进制转对称三进制逻辑电路	比-对处理电路			比	对	比	对	比	对	输出信号	转换值
		A	A	I	1	1	0		0		输出信号 1	1
			Not (OR (A, a))	R	0		0		1	1	输出信号 1	0
		a	a	T	0		1	T	0		输出信号 1	T

[0212] “比-对”三进制处理单元等效干涉光路方法

[0213] 根据本发明干涉光等效“比-对”三进制处理可由附图4-406的“比合”三进制-对称三进制转换逻辑门光路等效处理器单元实现,也就是说,可通过光学元器件光电耦合器或全光干涉元件来实现比合三进制-对称三进制转换。跟“比合”光处理单元相同,在以半个波长的至少五倍为直径的一个单位圆空间的圆周上布置投光器和受光器,其中,在A和a所示的位置处输入投光器的两个入射光,在I、R、T所示的位置处布置受光器的光检测装置作为输出。A、a、I、R、T在一个同心圆上。A到T和a到I之间的距离是同心圆的两个直径,AR与aR的路程相同,长度为 $5 * (\lambda/2)$,入射光源的位置A与a的距离为 $3 * (\lambda/2)$,A与I和a与T的投射距离为 $4 * (\lambda/2)$ 。同频率、同相位、同方向的入射光源A和a入射(这时,输入信号 $A=1, a=1$)到密闭的壳体内,AI与aI和aT与aI的光路程相差半个波长,受光器I和T的上两个光波的叠加而显示暗光,使I和T的受光器同时为输出值信号0(输出信号 $I=0, T=0$),而R受光器因为A与a的干涉光的叠加而感应强光,通过光电效应转换为电信号1。入射光源A与a之中,当只有一个入射时,由于无光的干涉,所以当A入射光源信号而a没有入射光源信号时($A=1, a=0$),I受光器输出信号1,T与R受光器无光信号检测而输出信号0($I=1, T=0, R=0$);同理,当a入射光源信号1同时A入射光信号0时($A=0, a=1$)时,T受光器输出信号1,I与R受光器无光信号检测而输出信号0($I=0, R=0, T=1$);入射光源A与a同时无光信号时(输入信号 $A=a=0$),受光器I、R、T同时无光信号检测,而输出信号为0(输出信号 $T=R=I=0$)。其中,入射光源A与a之间的距离为相差3勾股数倍数的半个光波长,A与I以及a与T的垂直投射距离为相差倍数4勾股数的半个光波长,这样,按照勾股定理,干涉光源A与受光器T以及干涉光源a与受光器I之间的距离为相差倍数5的半个光波长,这样,两个干涉光路程相差 $5-4=1$ 一个半个光波长,使I和T受光器在光干涉时,同时光波峰和光波谷相互抵消而接收到暗光,而R因干涉光的叠加而接收到强光。

[0214] 综上所述,“比-对”干涉光转换处理利用一对入射光源、一对投射输出光之间相差半个光波长距离和入射光相同距离的点,产生三个投射光。通过光的衍射和干涉现象,改变投射部分的明暗的值变化规律,计算处理“比-对”干涉光转换处理操作,具体的处理“入射值”和“输出值”变化关系,参考如下“比合”等效光路表17。

[0215] 表17:TRIA比合三进制转对称三进制处理逻辑门等效光路真值表

[0216]	比合-对称转换光干涉等效处理转换光路	比-对处理光路干涉			比	对	比	对	比	对	输出信号	转换值
		A	只有 A 通道发光 1	I	1	1	0		0		输出信号 1	1
			A 和 a 光相互干涉 1	R	0		0		1	1	输出信号 1	0
		a	只有 a 通道发光 1T	T	0		1	T	0		输出信号 1	T

[0217] 此外,还可以按照类似的方式来实现光学的各种逻辑运算元件。

[0218] 参见图4的407,在“比-对”干涉光转换处理的光学元件的输出I和T受光器屏蔽,只有R的受光器检测。入射光之间的干涉效应将此“比-对”干涉光转换处理光学元件成为一个AND与操作的光学元件。继续以表16的数据解释,当入射光A和a的同时输入光信号时,一个

受光器R能接受到光信号 ($A \& a = \rightarrow R, R = 1 \& 1 = \rightarrow 1$), 而当入射光A或a的光信号为不同时入射或没有入射的时候, R受光器不能收到光信号, 这等效于逻辑门电路操作的AND ‘与’ 位操作。这里, A和a与R之间的距离相等; 入射光A=1和a=1时, 两束光相互干涉从而在R上接收到一个亮点结果值R=1; 入射光A=1和a=0时, 两束光之间无干涉现象, 在R上无干涉光产生因此结果值为R=0; 同样地, A=0和a=1时, 两束光之间无干涉现象, 在R上无干涉光现象, 产生结果值为R=0; 两个同时无入射光时A=0和a=0, 结果在R上无光干涉现象, 结果值为R=0。

[0219] 如图4的408所示, 输入端A或a的光信号被直接对准发射在R受光器上。这里, 在元件内部形成光路使得光信号被直接发射向R受光器。A和a与R之间的距离相等。这样, 可实现处理OR光路逻辑门操作。具体地, 当A或a的无光信号输入 ($A=0, a=0$) 时, 聚焦点R受光器无光信号 ($R=0$); 当A或a中的至少一个有光信号输入时, 聚焦点R受光器就能接受到光信号 ($R=1$)。通过此方法处理, 可实现光学的等效OR ‘或’ 位操作光学逻辑运算元器件, 另外, 采用两束光的凸镜的汇集方法也能实现相同的光逻辑或计算。

[0220] 如图4的409所示, 输入端A是将被执行逻辑非运算的光信号, 而a是保持恒定的光信号, 即, $a=1$ 。这样, 可实现NOT逻辑非门操作。具体地, 当A无光信号输入 ($A=0$) 时, 输出端T的受光器可接收到来自a的光信号而输出1; 当A有光信号输入 ($A=1$) 时, 输出端T的受光器接收到A和a的光信号的干涉形成的暗光信号, 因此输出T=0。附图4-409示出的是通过光的衍射的逻辑非门实现方式; 附图4-403示出的是通过两束光的干涉也能实现相同功能。设a为恒定光, A为信号光, 输出结果信号b。当A=0无信号的时候a的恒定光直接投射到b上, 逻辑计算结果为b=1; 当A=1有光信号的时候A和a的光的相互干涉, 使b上暗光逻辑计算结果为b=0。

[0221] 如图4的410所示, 根据本发明的光学等效的异或逻辑运算可由如406所示的比合-对称转换处理单元和或门逻辑运算单元组成。当入射光A=1和a=1时, 经过406的光比合处理得到结果为A=0和a=0, 经过408所示的光或门得到结果为0; 当入射光A=1, a=0时, 406光比合处理结果为A=1, a=0, 经过408的光或门汇集结果为1; 当入射光A=0, a=1时, 经过406光比合处理结果为A=0, a=1, 经过408的光或门汇集结果为1; 当无入射光A=0, a=0时, 406光比合处理结果为A=0, a=0, 经过408的光或门汇集结果为0。也就是说, 通过光比合单元406和光或门408的结合410来处理光异或门光逻辑处理单元。

[0222] 综上所述, “比合”等效逻辑门光路元器件, 利用光的干涉原理, 在半个波长的“5倍勾股数”为直径的单位圆空间上形成。根据本发明的“比合”光路元件通过对一对入射光的光干涉采用不同处理输出而得到不同的光信号, 包括: 二入三出 (2to3, 图4-406)、二入二出 (2to2, 图4-403) 和二入一出 (2to1, 图4-407, 408, 409)。其中, 二入二出 (2to2) “比合”单位圆光路元件用于平行输出两个光信号, 执行根据本发明的“比合”位操作; 二入三出 (2to3) “比合”单位圆光路元件输出三个光信号, 实现根据本发明的“比合三进制-对称三进制”转换位操作; 二入一出 (2to1) “比合”单位圆光路元件, 输出一个光信号, 分别实现等效的“与”操作光学元器件 (图4-407)、“或”操作光学元器件 (图4-408)、“非”操作光学元器件 (图4-409)。

[0223] 可以在纳米板上印刷此“比合”单位圆光学处理元器件的光路。通过此方法, 可以通过在纳米基板上印刷光路的形式来实现等效处理逻辑门光路。半个波长5倍勾股数直径的

单位圆等效逻辑门光路元件比电路元器件结构简单。也就是说,可以通过印刷方式蚀刻纳米材料来制作光路,从而通过处理光信号来完成逻辑门和位操作。

[0224] “二进制”切换“三进制”逻辑电路

[0225] 根据本发明的一实施例,根据本发明的T全加法器201还可通过一个输入信号来改变T全加法器201的计算模式。当输入信号值T为0时,通过切换控制逻辑电路205,T全加法器201改变为一对二进制加法器计算处理;当输入信号值T为1时,通过切换控制逻辑电路205,T全加法器改变为三进制加法器计算处理。

[0226] 如上所述,切换逻辑的电路包括(NOT)反相器和(AND)与逻辑门阵列。输入信号分支成一对正反分支的逻辑电路,通过(NOT)反相器区别正反分支;通过正反分支输出的控制信号,与其他的输入信号(AND)与操作控制其他输入信号的开启和关闭,达到切换逻辑门电路的处理。

[0227] 如上所述,根据本发明的T全加法器201上,参照附图3的二进制与三进制切换逻辑电路单元301。在控制信号输入T被输入后,该信号T分支成一个反相器和直通的电路分支,然后分别跟各自的输入信号组(A,B,C,a,b,c)进行与操作,从而开启和关闭输入信号的不同流向。当T控制信号为0时,反相器通过输入值0转换为信号值1,使输入信号值之间进行与操作。因控制信号的值为1结果值取决于控制信号的输入信号的1和0的状态开启了输入信号,通过电路输入信号分别输入到两个二进制加法器上,并且切换控制电路开启二进制输出的电路,计算处理二进制计算。当T控制信号为0时,反相器的结果值为1,屏蔽了当切换信号T为1时的电路信号处理,反相后的控制信号1和三进制输入信号之间与位操作,开启三进制输入信息。通过三进制逻辑门电路的处理,包括双二进制加法处理、“比合”计算、二与三“比合”进制之间的转译的进位值(G/g)和累加值(S/s)计算处理结果输出,再次控制信号跟二进制的逻辑电路的输出结果信号屏蔽和开启三进制计算处理结果值的信息输出。

[0228] 参照回到图2,根据本发明的T全加法器201(以下,T全加法器来表示)可以将8个T全加法器单元连接,每个T全加法器单元的进位输出都作为下一个T全加法器单元的进位输入,这样就构成一个8位的T全加法器,并且还通过T全加法器的T控制信号切换各T全加法器的计算处理模式。应理解,T全加法器的串联个数不局限于8位,可以是不同位的T全加法器。

[0229] 综上所述,根据本发明的切换控制逻辑电路205将控制信号T通过反相器的分支电路后,和输入输出信息之间进行与位操作来开启和屏蔽输入输出信息,从而实现二进制和三进制的相互切换计算处理。详细的控制过程,参照如下表18的信息变化。

[0230] 表18:切换不同进制的切换控制电路的真值表

[0231]	T 控制信号	反相器	控制信号分支值	输入信号	AND (控制,输入)	与操作结果值
	1	1	1	1	AND (1,1)	1
		NOT (1)	0	1	AND (0,1)	0
	0	0	0	1	AND (0,1)	0
		NOT (0)	1	1	AND (1,1)	1

[0232] 另外,根据本发明的TRIA处理器不仅可将“比合”三进制转换为“对称”三进制,相反的逆转换方式也简便。如附图4的401所示,“对称”三进制转换“比合”三进制的光路,在T、R、I三个对称三进制信号输入后,利用输出端 $A=I+R$ 和 $a=R+T$ 转换成比合三进制逻辑输出。在401中,形成I直通A、T直通a和R分别同时通向A和a的光路。这样,当对称三进制的值为1(I=1)时,I的发光器投射光射向A受光器使输出端(A,a)转换成[1,0];当对称三进制值为0(R

=1)时,R的发光器投射光同时分别射向A受光器和a受光器,使输出端(A,a)转换成[1,1];当对称三进制值为-1(T=1)时,T的发光器投射光射向a受光器使输出端(A,a)转换成[0,1],这样,通过光路元器件来完成“对称”三进制到“比合”三进制的转换。

[0233] “比合”三进制模拟器引擎

[0234] 根据本发明的对称三进制的TRIA“比合”三进制的全加法器运算处理过程,可以在二进制计算环境下通过Python语言模拟“比合”三进制计算处理方法。此运算处理过程方法不仅限于Python语言,也可通过其他汇编、C语言等来进行进一步优化。用此方法在二进制环境下编译成TRIA“比合”三进制加法器操作引擎提供API接口给操作系统,从而可在二进制计算机操作系统上虚拟化“比合”三进制计算处理的对称三进制计算的模拟操作系统环境。

[0235] 如上所述,根据本发明的T全加法器模拟器可参照附图2的逻辑门电路的模拟程序,包括“比合”操作功能模块(combine函数),二进制全加法器功能模块(bin_adder函数),二和三“比合”转译模块(tria2to3函数)和“才亚”转换“才特”三进制(trito函数)。通过T全加法器模拟器功能模块(tria_adder函数)调用计算,其中,输入参数分别为二进制和三进制切换输入值(以下,变量T表示),正数部输入值(A,B和进位C),反数部输入值(a,b和进位c)。当输入值T为0时,切换到二进制计算模式各自调用bin_adder处理后输出;当输入值T为1时,首先并行调用bin_adder计算处理后,调用combine函数“比合”处理,并判断正、负进位值异或值(XOR)等于0时直接返回结果值,当异或值(XOR)等于1时调用tria2to3函数转译操作后返回结果值完成,“比合”三进制全加法器处理。“比合”三进制结果通过调用trito,按位转换处理输出“对称”三进制。

[0236] T全加法器模拟程序的各模块和处理步骤,具体参照如下表19。

[0237] 表19:二进制环境下“比合”三进制模拟引擎,Python语言程序例

[0238]	处理模块和处理步骤	Python 语言模拟 TRIA 逻辑电路程序
--------	-----------	-------------------------

[0239]	功能模块	1) 上下比合操作逻辑运算	<code>def combine(h, l):</code>
			<code>return (h&l)^h, (h&l)^l</code>
		2) 二进制全加法器逻辑运算	<code>def bin_adder(aIn, bIn, cIn):</code>
			<code>return (aIn+bIn+cIn)/2, (aIn+bIn+cIn)%2</code>
		3) 二进制转换三进制逻辑运算	<code>def tria2to3(ha, hb, la, lb):</code>
		上进位 (正进位) 1 时逻辑运算	<code>if ha == 1:</code> <code>return ha^lb, ha&lb, la, ha^hb^lb</code>
		下进位 (负进位) 1 时逻辑运算	<code>if la == 1:</code> <code>return ha, la^lb^hb, la^hb, la&hb</code>
		4) "比合"转换"对称"三进制处理 [1, 0]=1, [0, 0]=0, [0, 1]=-1 (-1, 符号 T 表示)	<code>def trito(A, a):</code> <code>if not(A a): return 0</code> <code>if A == 1: return 1</code> <code>if a == 1: return -1</code>
	TRIA 比合 三进制 运算	T 全加法器	<code>def tria_adder(T, A, B, C, a, b, c):</code>
		(1) 二进制加法操作	<code>G, S = bin_adder(A, B, C)</code> <code>g, s = bin_adder(a, b, c)</code>
		当 TRIA 控制信号 1, 处理比合操作	<code>if T == 1:</code>
		(1) 二进制结果上下进位和累加值 比合处理	<code>G, g = combine(G, g)</code> <code>S, s = combine(S, s)</code>
		(2) 比合二进制值转换 TRIA 比合 三进制操作	<code>if G^g==0:</code> <code>return "Tri:", (G, S, g, s)</code>
			<code>else:</code>
			<code>return "T:", tria2to3(G, S, g, s)</code>
	二进制 运算	当 TRIA 控制信号 0, 处理二进制 输出二进制各自的结果值	<code>else:</code> <code>return "B:", (G, S, g, s)</code>

[0240] 如上所述,根据本发明“比合”三进制双链数据结构特征和“比合”三进制模拟器引擎,编制处理对称三进制虚拟机双字节码的面向对象TRIA计算机语言,跟JAVA计算机语言的虚拟机字节码相似。

[0241] TRIA运算处理,通过T全加法器运算,处理输出“比合”三进制结果,并且三进制转二进制的方式(此说明中未示出),转换成二进制的结果的输出、再次转入到二进制全加法器上继续运算处理的二进制三进制混合计算。三进制T全加法器的减法运算结果,判断处理负数时转译为二进制的补码形式。

[0242] 减法运算和乘法运算

[0243] 如上所述,根据本发明的TRIA数据存储结构,“比合”三进制的正反双链的信息组合方式,减法运算的时候,被减数不变,只需互换减数的正反双链的信息阵列后,通过T全加法器加法计算,就能计算处理TRIA计算机减法运算。相对于二进制的减法运算,需要减数反码处理和补码处理等操作处理省略,“比合”三进制计算机的优点,通过一个T全加法器完成加法和减法的处理。

[0244] 以下内容是为了清晰说明此操作方法的详细内容。

[0245] 对称三进制减法: $T-T=0$, $T-0=T$, $T-1=T1$, $0-T=1$, $0-0=0$, $0-1=T$, $1-T=1T$, $1-0=1$, $1-1=0$

[0246] “比合”三进制减法转加法,是将减数的X2和Y2互换后的加法处理。

[0247] 比合数方法表示: $Z=Z1-Z2=(X1-Y1)-(X2-Y2)=(X1-Y1)+(Y2-X2)$;

[0248] 函数方法表示: $Z=T(X1, Y1)-T(X2, Y2)=T(X1, Y1)+T(Y2, X2)$;

[0249] 其中, $Z=T(X, Y)$; $Z1=T(X1, Y1)=X1-Y1$; $Z2=T(X2, Y2)=X2-Y2$;

[0250] 如上所述,根据本发明的TRIA减法处理,只需要减数的比合数中(正,反)的参数,互换成比合数(反,正)的参数值后,T全加法器中被减数和减数互换后的比合数的减数带入后“比合”加法运算就能将减法转换为加法处理。

[0251] 根据本发明的TRIA数据存储结构,按照“比合”三进制的正反双链的信息组合方式处理对称三进制乘法运算,相对于全加法器,乘法器无进位。

[0252] 对称三进制乘法: $T * T = 1, T * 1 = T, T * 0 = 0, 0 * T = 0, 0 * 0 = 0, 1 * 1 = 1, 1 * T = T, 0 * 1 = 0, 1 * 0 = 0$

[0253] “比合”三进制乘法: $Z1 * Z2 = \rangle [A, a] * [B, b] = [D, d] = \rangle Z$

[0254] $T * T = [0, 1] * [0, 1] = [1, 0] = 1, T * 1 = [0, 1] * [1, 0] = [0, 1] = T, 0 * T = [0, 0] * [0, 1] = [0, 0] = 0$

[0255] $1 * 1 = [1, 0] * [1, 0] = [1, 0] = 1, 0 * 0 = [0, 0] * [0, 0] = [0, 0] = 0, 0 * 1 = [0, 0] * [1, 0] = [0, 0] = 0$

[0256] $Z1 * Z2 = [((a \& b) | (a' \& b')), ((a \& b') | (a' \& b))] = [d, d'] = \rangle Z$

[0257] “比合”三进制的正反“比合”数[A, a]和[B, b]输入和结果[D, d],具体逻辑门处理表达式可用于电路和光路处理上。

[0258] 也就是说,根据本发明实施例比合三进制的乘法器的运算模块实现的输入和输出的逻辑运算表达为如下:

[0259] $d = \text{OR}(\text{AND}(a, b), \text{AND}(a', b')), d' = \text{OR}(\text{AND}(a, b'), \text{AND}(a', b))$

[0260] 比合三元逻辑(异与): $\left[\frac{a}{a'} \right] * \left[\frac{b}{b'} \right] = \left[\frac{(a \& b) | (a' \& b')}{(a \& b') | (a' \& b)} \right] = \left[\frac{d}{d'} \right]$ 。

[0261] 比合三进制的乘法是表6中三元逻辑异或的两个比合参数 $A = (a, a')$ 和 $B = (b, b')$ 的比合三进制异或结果值 $A \wedge B = \left[\frac{a}{a'} \right] \wedge \left[\frac{b}{b'} \right] = \left[\frac{(a|b) \& (a'|b')}{(a|b') \& (a'|b)} \right]$ 的比合逻辑非的关系。

[0262] 比合逻辑 $(A \wedge B) = ! (A * B)$ 或 $(A * B) = ! (A \wedge B)$,比合逻辑乘法器和比合逻辑异与是同一个比合三元处理运算,比合逻辑异或和比合逻辑非处理等价。

[0263] 比合运算 $(A * B) = ! (A \wedge B)$ 的分式表达 $\left[\frac{(a \& b) | (a' \& b')}{(a \& b') | (a' \& b)} \right] = ! \left[\frac{(a|b) \& (a'|b')}{(a|b') \& (a'|b)} \right]$ 。

[0264] 其中,d和d'是乘数结果值正、反位数,具体参照如下表20。

[0265] 表20:“比合”三进制乘法逻辑门处理方法

乘法	Z1	Z2	Z	逻辑表达式	例	Z1	Z2	Z	对	Z1	Z2	Z	对	Z1	Z2	Z	对
[0266] 对称 $Z1 * Z2 = Z$	a	b	d	$(a \& b) (a \& b')$	子	1	1	1	1	0	0	0	0	0	1	0	T
	a'	b'	d'	$(a \& b') (a' \& b)$		0	0	0		0	0	0		1	0	1	

[0267] “比合”三进制的其他运算,包括乘法、除法等运算处理是真值运算的一位操作结果值,因此跟T全加法器的进位相比,逻辑门运算结构简单。以乘法运算为例,通过乘数的各位中的顺序提出一位信息,与被乘数的全部各位信息按照规定的逻辑门电路逐步顺序乘法处理后,累加到T全加法器上,从而实现处理乘法运算。除法运算和乘法运算相似没有进位,因此不再赘述。

[0268] 光路全加法器

[0269] 根据本发明,还可采用光学器件来实现单位圆光路设计的全加法器。结合附图5的

500的单位圆勾股数光电等效干涉光元件原理,其中,501虚框部分表示根据本发明实施例的比合三进制光路全加法器,其包括:双“比合”光路的二进制全加法器(+Bin,-Bin)503、双“比合”运算处理器(Combine)504、二进制三进制转译器(Tri-Translator)505以及二进制与三进制转换部分(省略,参考图3的T全加法器电路二进制计算输出部分T信号切换方法)。其中,光路二进制全加法器的具体光路图如虚线框502所示(一对正反二进制全加器503中之一),二进制三进制转译器505的具体光路图包括图5中的507、508和509。

[0270] 首先,一对正和反“比合”光路二进制全加法器(OI-Bin Adder)503分别包括比合三进制数的正数部的二进制加法器和反数部的二进制加法器。以输入为(A,B,C)的正数二进制加法器为例,其累加输出 $S=XOR(XOR(A,B),C)$,而进位输出 $G=OR(AND(A,B),AND(XOR(A,B),C))$ 。这里使用的光学逻辑运算元件在图4中已经参照407-410进行了描述,因此不再详细描述。同样地,可以得到反数二进制加法器的累加输出 $s=XOR(XOR(a,b),c)$,进位输出 $g=OR(AND(a,b),AND(XOR(a,b),c))$ 。

[0271] 分别将进位光信号(G与g)和累加值信号(S与s)输出到正和反光路“比合”处理器(OI-Combiner)504,然后进行比合处理后输出[G,g][S,s]二进制“比合”光信号。

[0272] 三进制转译器(OI-Translator)505包括三种分支,其输入为一对“比合”处理完的光信号[G,g][S,s]。其中一对正反进位光信号(G与g)的光路XOR异或门处理,如果异或信号处理值为0时“ $XOR(G,g)=0$ ”,通过光路反相器NOT把光信号转变成1,从而输入切换开启506所示的光路与门组,使光信号直接输出[G,g][S,s];一对正反进位光信号异或值为1时(即,“ $XOR(G,g)=1$ ”),当光路判断 $g=1$ 时,开启507所示的光路与门组,使得光信号参数改变为(g,G,s,S)输入并通过509所示的Tria4outer光信号处理单元,然后输出(g,G,s,S);当光路判断 $G=1$ 时,开启508所示的光路与门组将光信号参数改变为(G,g,S,s)输入并通过509所示的Tria4outer光信号处理,然后输出(G,g,S,s)。Tria4outer 509的具体光信号处理方法是:以光信号参数(G,g,S,s)为例,输入光信号G和s的光路XOR异或门处理结果作为正数进位输出G,输入光信号g直接输出作为反数进位输出g,输入光信号G和s的光路AND与门处理结果作为输出正数位累加输出S,输入光信号G、S的XOR异或处理结果与输入光信号s的光路XOR异或处理结果作为反数位累加输出s。

[0273] 另外,虽然在图5中没有示出,但是对称三进制输入的三值信号通过401的光(3to2)的元件处理,可实现对称三进制与“比合”和三进制的切换。

[0274] 另外,根据本发明的T全加法器还可如图5的510所示,先对输入的比合三进制数进行比合运算,然后进行加法运算,最后进行三进制转译。

[0275] 具体地,图5种的510是501的不同变化形式,首先Combine比合处理正和反的输入数字信号(A,a),(B,b),(C,c),每一对正反信号通过403的比合逻辑元件处理,比合处理元件可使用非线性光学晶体和介质达到光的比合计算处理的目的。再次(A,B,C)比合处理后的输出信号为正数部分二进制全加器的输入信号计算后输出(G,S)计算结果输出;(a,b,c)比合处理后的输出信号为反数部分二进制全加器处理输入信号计算后输出(g,s),502的两个双二进制全加器计算处理。计算结果值的进位(G,g),累加值(S,s)通过正反进位值之间判断,把比合二进制转译成比合三进制。510和501的全加器不同点在于,510是先比合处理后二进制全加器计算,501是先二进制全加器计算后比合处理,最终以相同方式把二进制比合转译成三进制比合数。

[0276] 在制作光路板时,可使用3D打印光路板(加法),激光蚀刻光路板(减法)的方法开凿光道,制造集成光路板。

[0277] <TRIA输入系统>

[0278] 从编码(CODE)的角度分析,计算机系统是一种编码处理系统,通过不同物理介质的状态和不同环境条件下变化关系,使用编码形式转换后,经过这些编码的变化特点和规律来处理信息的系统,通过编码的输入、计算处理、输出、存储和传递。

[0279] 计算机由运算器、控制器、存储器、输入设备和输出设备等五大部件组成计算机硬件环境。

[0280] 根据本发明的输入方法跟传统的输入方法不同之处是附图7的700所示,通过最适比合数字处理的科学方法最优简化的TRIA“比合”文字符号的输入达到输入自然语言文字的输入,简化了输入设备和输入方式。TRIA符号语言是针对于计算机设计的语言文字,而传统文字符号书写方便而自然地随文化演化的语言。利用TRIA语言符号编码和传统语言符号编码的对应关系,可以通过输入TRIA符号经过计算机输入系统转译处理方法来输入传统语言文字符号。因为TRIA“比合”编码的象数图符号是几何形状,与传统文字符号相比更加简易和自然。输入装置通过不同的输入方式生成TRIA“比合”符号编码,经过计算机输入系统的TRIA“比合”编码与传统语言文字符号的ASCII码或UNICODE码比对,输入文字符号和存储。TRIA象数图符号本身也是一种语言符号,可以在扩展ASCII码和UNICODE中做语言文字编码,通过输入方法输入和保持TRIA符号语言文字。

[0281] TRIA符号编码系统

[0282] 根据本发明的实施例TRIA“比合”符号编码系统,通过一对阵列信息编码编制而成。包括TRIA“比合”三进制象数图符号编码系统和TRIA“比合”二进制象数图符号编码系统,以象和数之间相互对应的一对编码“比合”的符号系统。TRIA“比合”符号编码系统,通过此原理符号(象)和编码(数字)相互匹配的象数图符号系统。

[0283] TRIA象数图符号是象和数之间相互关联的一种文字符号系统,TRIA符号是二维符号系统。TRIA二元符号系统是一对3位二进制编码的二维几何中的表现形式。TRIA三元符号系统是玄画的三元符号的四重变化的二维几何中的编码方式;在正方形的四个顶点中,每个顶点有三种线段联系方式,运用这个变化生成的二维符号系统。TRIA三元符号系统,更加适合于现代的拼音文字、数字、标点符号和控制符号的统一描述,更易于书写、阅读和识别。基于TRIA三元符号文字语言,配合一个符号发音一个音符,单词的发音和拼写完全一致,变化简单有规律的一种辅助语。

[0284] 根据本发明的TRIA三进制符号系统是基于“口”字形的拓扑点之间的矢量变化关系的编码,附图6的601的端点A、B、C、D四点之间变化产生的符号编码(如附图603所示的象数图三元符号编码)。

[0285] 根据本发明的TRIA二进制符号系统是基于“田”字形的几何拓扑点之间的矢量变化关系的编码,如附图6的601和602的端点1、2、3、4和A、B、C、D八点和中心点之间变化产生的象数图符号编码。下面将对两套符号系统进行详细说明(如附图604、605、606的象数图二元符号编码)。

[0286] TRIA三进制符号编码

[0287] 根据本发明的实施例TRIA三进制符号系统利用矩形几何图的几何特征。如图6的

601所示,按顺时针A、B、C、D的四点组成的一个矩形。相对于每个端点与其它3个端点之间,只有3种不同方向连接方式。“才”字形相似的三种不同方向的笔画,如A点为起始点,包括矢量AB顺时针方向连接、矢量AD的逆时针方向连接和矢量AC对角线方向连接的三种方向的轨迹变化,分别对称三进制表示1、T和0,以“比合”三进制编码表示 $[1,0][0,1][0,0]$ 。

[0288] 为了说明四点一画轨迹编码的矢量值与比合三进制编码关系,以符号Z为例。在书写Z时,轨迹顺序经过ABDC端点,轨迹顺序包括顺时针(AB)、对角线(BD)、逆时针(DC)的轨迹,这样,可用对称三进制编码10T来表示符号Z,相应的比合三进制编码表示为 $[1,0][0,0][0,1]_Z$ 或 $[0,1][0,0][1,0]_Z$ 。

[0289] 这样,在根据本发明的计算机的存储单元中,TRIA符号的TRIA对称三进制存储格式的可包括起始位和轨迹描述位,这样,可通过一位“才特”(TRIT)的起始位置描述和5位“才特”的轨迹描述,总共6个“才特”来描述一个TRIA三元符号。

[0290] 四点一画的轨迹编码,因几何特征和轨迹移动方向的准确性,容易在触摸板等输入装置上识别出来,识别率也比其他语言符号高,易于计算机输入设备使用,TRIA三进制符号(以下,三元符号表示)轨迹编码关系,具体参照如下表21。

[0291] 表21:TRIA三元符号轨迹编码对照表

[0292]	TRIA 三进制符号轨迹编码	四点		矢量	顺时针 AB	逆时针 AD	对角线 AC	符号 Z	ABDC
		A	B	对称三进制	1	T	0	对称值	10T
		D	C	比合三进制	[1, 0]	[0, 1]	[0, 0]	比合值	[1, 0] [0, 0] [0, 1]

[0293] 根据本发明的实施例的TRIA三元符号轨迹编码系统是基于矩形的四个顶点通过相互连接产生的几何形态符号轨迹的编码系统。TRIA三元符号包含76个不同形状的符号，包括53个一笔画构成的符号(/, \, □, ∠, Δ, ▽, Γ, X, N, Ø, V, X, X, \,

N, U, V, W, X, Y, Z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z,

␣, ␤, ␥, ␦, ␧, ␨, ␩, ␪, ␫, ␬, ␭, ␮, ␯, ␰, ␱, ␲, ␳, ␴, ␵, ␶, ␷, ␸, ␹, ␺, ␻, ␼, ␽, ␾, ␿,

1, 2, 3, 4 的TRIA一笔画53个符号)、8个两笔画构成的符号(5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 8

↖, ↗, ↘, ↙, ⇄, ↕的TRIA二笔画符号)、15个矩形顶点中的一个至四个顶点组成的符号等(.点4个、...点6个、...点4个、...点1个)。TRIA三元符号的轨迹编码理论上存在 $3^6=729$ 个编码容量。TRIA三元轨迹编码符号(“象数图”符号)有简易(原理简单)、变易(不同变化)、不易(规则统一)的特征,其中TRIACode的一笔画的轨迹编码量是120个,跟4个角点组合可产生 $120*(1+4)=600$ 个不同编码,可以满足不同要求的交互编码,因为编码形式简单,可以应用不同尺寸的设备 and 手势识别上。与传统输入设备相比,采用TRIA三元符号语言的输入设备结构简单,功能丰富,在不同操作交互环境下,保持一致的三元符号手势语言,在特殊领域可以替代QWERT全键盘的位置语言。只需掌握此三元符号语言的输入方法,就可以控制和交互不同形态三元符号支持的设备。

[0294] 根据本发明的TRIA三元符号,可分为四种类型:文字符号、数字符号、标点符号、控制符号。类型的编码规则的一个例子如下:以起始点从左上角(A点)开始生成的TRIA三元符号为字母符号,起始点从右上角(B点)开始的生成的TRIA符号为数字符号,起始点从右下角(C点)开始生成的符号为标点符号,起始点从左下角(D)开始生成的符号主要为控制符号和

一些标点符号。

[0295] 根据本发明的实施例中,英文字母Tria可使用TRIA三元符号 $\overline{7}\overline{0}\overline{1}$ 表示;英文 Hello world,由TRIA三元符号 $\overline{H}\overline{Z}\overline{L}\overline{L}\overline{O}\overline{X}\overline{O}\overline{O}\overline{L}\overline{W}$ 表示;汉字的“爱”用拼音方式“ $\overline{A}\overline{I}$ ”日语的“ $\overline{に}\overline{ほ}\overline{ん}\overline{ご}$ ”用“ $\overline{N}\overline{H}\overline{O}\overline{L}\overline{G}\overline{O}$ ”韩语的“ $\overline{한}\overline{글}$ ”用“ $\overline{M}\overline{A}\overline{N}\overline{G}\overline{L}$ ”书写和发音。TRIA语言是形中抽象与运数演绎形成的“象数图”符号作为文字符号以一象一音的拼写方式的一种辅助象数音义结合的语言。

[0296] 根据本发明的TRIA三元符号,计算机上输入、处理、保存和输出,需要TRIA三元符号计算机字符编码和字库。计算机上常用的字符编码ASCII码有7位码和8位码两种形式,其中扩充ASCII字符集分别可以扩充128个字符。TRIA三元符号字符编制在扩充128个高位编码当中。以TRIA编码方法为 $ASCII(z)+128=TRIA(z)$,此编码容易计算英文字母之间的换算关系。

[0297] 根据本发明的TRIA三元符号,可以应用于在目前计算机中用得最广泛的字符集及其编码Unicode(统一码)上的字符编码方式上。Unicode的2800至28FF是盲文字符集,2900至2B5F是各种符号,2B60至2BFF编码区域是空的保留区。TRIA字符集编码可使用所述保留区(2B60-2BFF)。字母A的对应的TRIA码 \overline{A} 的Unicode编码可以是0x2bc1,TRIA三元符号的Unicode码与TRIA三元符号ASCII扩展码之间相差0x2b00(十进制11008),这样易于计算机上编码转换时计算。

[0298] 根据本发明的TRIA三元符号的符号类型和各种编码可参照下面的表22-25。

[0299] 作为一种编码方式,TRIA三元符号的数字是从左上角为起始点演绎的轨迹编码。为了易于记忆,在组合方式上跟传统QWERT键盘上的数字键相互匹配。在TRIA三元符号中,配合SHIFT上档键“|”(从左下角为起始点,逆时针一个线段单位的轨迹编码)组合成传统键盘的每个数字(1,2,3,4,5,6,7,8,9和0)相互匹配的特殊符号“!、@、#、\$、%、^、&、*、(和)”表示,具体参照表22所示。

[0300] 表22:TRIA数字和标点符号和其他编码比较表。

[0301]	数字	0	1	2	3	4	5	6	7	8	9
	TRIA 数字	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$	$\overline{8}$	$\overline{9}$
	扩展 ASCII	0xb0	0xb1	0xb2	0xb3	0xb4	0xb5	0xb6	0xb7	0xb8	0xb9
	Unicode 码	2bb0	2bb1	2bb2	2bb3	2bb4	2bb5	2bb6	2bb7	2bb8	2bb9
	轨迹编码	101111	10	1T01	1TTT	10T	1110	1TTT0	1TT	10T0	11110
	标点符号	!	@	#	\$	%	^	&	*	()
	TRIA 标点	$\overline{7}$	$\overline{5}$	\overline{E}	\overline{Z}	\overline{A}	\overline{S}	\overline{F}	\overline{K}	\overline{N}	$\overline{0}$

[0302]	扩展 ASCII	0xa1	0xc0	0xa3	0xa4	0xa5	0xde	0xa6	0xaa	0xa8	0xa9
	Unicode 码	2ba1	2bc0	2ba3	2ba4	2ba5	2bde	2ba6	2baa	2ba8	2ba9

[0303] TRIA三元小写字母是从左上角为起始点,轨迹通过“比合”数字演绎一笔画构成的。

[0304] 表23:TRIA三元符号的小写字母符号参照表,包括字母韩语发音

[0305]

小写	a	b	c	d	e	f	g	h	i	j	k
TRIA	ᄁ	ᄂ	ᄃ	ᄄ	ᄅ	ᄆ	ᄇ	ᄈ	ᄉ	ᄊ	ᄋ
韩语	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅝ	ㅟ	ㅛ	ㅝ	ㅟ
ASCIIX	0xe1	0xe2	0xe3	0xe4	0xe5	0xe6	0xe7	0xe8	0xe9	0xea	0xeb
Unicode	2be1	2be2	2be3	2be4	2be5	2be6	2be7	2be8	2be9	2bea	2beb
轨迹值	0TT0	010101	0111	0101	01110	0TTT0	011	0T010	00	001	010
小写	l	m	n	o	p	q	r	s	t	u	v
TRIA	ᄌ	ᄍ	ᄎ	ᄏ	ᄐ	ᄑ	ᄒ	ᄓ	ᄔ	ᄕ	ᄖ
韩语	ㅓ	ㅕ	ㅗ	ㅛ	ㅝ	ㅟ	ㅛ	ㅝ	ㅟ	ㅛ	ㅝ
ASCIIX	0xec	0xed	0xee	0xef	0xf0	0xf1	0xf2	0xf3	0xf4	0xf5	0xf6
Unicode	2bec	2bed	2bee	2bef	2bf0	2bf1	2bf2	2bf3	2bf4	2bf5	2bf6
轨迹值	0TT	0T0T0T	0T01	0TTTT	0110T	0T0T0	001111	010T0	0110	0TTT	0T0
小写	w	x	y	z	sp	,	.	;			
TRIA	ᄘ	ᄙ	ᄚ	ᄛ	ᄜ	ᄝ	ᄞ	ᄟ			
韩语	ㅓ	ㅕ	ㅗ	ㅛ							
ASCIIX	0xf7	0xf8	0xf9	0xfa	0xa0	0xac	0xae	0xbb			
Unicode	2bf7	2bf8	2bf9	2bfa	2b20	2b2c	2b2e	2b3b			
轨迹值	010	00T0	00T	010T		00101	10101	1T0T0T			

[0306] TRIA三元符号的大写字母是采用上档轨迹符号和小写轨迹符号的结合而成的编码规则。为了区别大写和小写字母形式,在TRIA三元小写字母符号之上或左侧用点或横表示TRIA大写字母,如下表23中以小写字母之上一横表示大写,实际类似于传统键盘的大写字母的“Shift+小写字母”的轨迹组合编码形式。大写A的轨迹码(Tricode)为Shift的轨迹码T1和小写a轨迹码0TT0结合,如:大写A的轨迹码是T1-0TT0。为了大写字母便于区别于小写字母,在小写字母基础上左侧或之上加一横或一点表示大写。

[0307] 表24:TRIA三元符号的大写字母参照表,Shift+小写字母(T1+tricode)

[0308]

大写	A	B	C	D	E	F	G	H	I	J	K
TRIA	𐀀	𐀁	𐀂	𐀃	𐀄	𐀅	𐀆	𐀇	𐀈	𐀉	𐀊
ASCIIH	0xc1	0xc2	0xc3	0xc4	0xc5	0xc6	0xc7	0xc8	0xc9	0xca	0xcb
Unicode	2bc1	2bc2	2bc3	2bc4	2bc5	2bc6	2bc7	2bc8	2bc9	2bca	2bcb
大写	L	M	N	O	P	Q	R	S	T	U	V
TRIA	𐀌	𐀍	𐀎	𐀏	𐀐	𐀑	𐀒	𐀓	𐀔	𐀕	𐀖
ASCIIH	0xcc	0xcd	0xce	0xcf	0xd0	0xd1	0xd2	0xd3	0xd4	0xd5	0xd6
Unicode	2bcc	2bcd	2bce	2bcf	2bd0	2bd1	2bd2	2bd3	2bd4	2bd5	2bd6
大写	W	X	Y	Z		<	>	:			
TRIA	𐀘	𐀙	𐀚	𐀛		𐀜	𐀝	𐀞			
ASCIIH	0xd7	0xd8	0xd9	0xda		0xbc	0xbe	0xba			
Unicode	2bd7	2bd8	2bd9	2bda		2bbc	2bbe	2bba			

[0309] 为了跟传统键盘的控制键相互对应易于学习和记忆,TRIA三元符号中的Shift键“|”、Enter键“𐀛”、空格键“-”等控制符号跟传统键盘上的控制符号形状相似,如表25所示。

[0310] 表25:TRIA三元符号控制码之间参照表

[0311]

控制符号	Fn	Ctrl	Alt	Enter	Shift	Space	Del	Esc
TRIA 控制	𐀀	𐀁	𐀂	𐀛	𐀜	-	𐀞	𐀟
Tricode(轨迹码)	0T	T111	T10	TTT	T1	01	T01	T0T0T
标点符号	`	‘	-	=	/	\	[]
TRIA 标点	𐀠	𐀡	𐀢	𐀣	𐀤	𐀥	𐀦	𐀧
Tricode(轨迹码)	T10T	T10101	TT	T0T0	10111	1T0	T0TTT	TTTT0
标点符号	~	“	_	+	?		{	}
TRIA 标点	𐀨	𐀩	𐀪	𐀫	𐀬	𐀭	𐀮	𐀯

[0312] 下面介绍,根据本发明的实施例的TRIA符号系统的字库。

[0313] 根据本发明的TRIA符号系统的字库包括了点阵字库和矢量字库。

[0314] 在现有技术中,英文字母是8×8的点阵二进制,因此需要8个字节才能够描述,而根据本发明的TRIA三元符号系统的符号几何特征,采用至少5×5=25位就能够识别不同符号之间的差别。因此,可以在更少的点阵下展现字形,节省了字库存储空间,从而适合于微型计算设备上的显示和使用。

[0315] 根据本发明的TRIA三元符号自身带有的几何特征,使矢量字库比英文或其他文字的矢量字库更加简单。在HTML5中,函数形式定义各TRIA三元文字符号的矢量字符的描述定义。如TRIA三元字母a,定义函数function tria_a(){context.moveTo(10,10);context.lineTo(10,30);context.lineTo(30,30);context.lineTo(10,10);context.stroke();}之后,在网页上调用此函数tria_a(),就能在HTML5中显示TRIA三元符号。因TRIA三元符号结构简单,在HTML5之中,可以不需要字库而用函数调用方式来显示

TRIA数字、文字和标点符号。

[0316] 根据本发明的另一实施例的TRIA三元符号系统中,可基于TRIA对称三进制表示方法执行矢量字库显示。

[0317] 具体地讲,根据本发明的实施例的TRIA三元字体生成引擎定义字体模板的框架,包括字的大小、线的粗细、线的颜色和边角处理方式等。当显示和打印字体时,通过TRIA三元字体生成引擎,基于上述字体信息在画布上画出TRIA三元符号以显示字体。

[0318] TRIA二进制符号编码

[0319] 根据本发明的TRIA二元符号编码,包括‘钟表式’(附图6-604)、“方向盘式”(附图6-605)、“往来式”(附图6-606)轨迹编码方法。通过此轨迹编码方法便捷输入盲文。一个盲文字符有左(1-2-3)右(4-5-6)两列自上而下的编码,其中左右两列的编码在TRIA二元符号编码对应一个轨迹编码。‘钟表式’编码,可分为‘对称形’轨迹编码方式和‘循环形’轨迹编码方式。只要记住盲文字母或新编制的编码排列规则中对应的一对8进制数和字母符号,就能轻易输入“比合”二元轨迹编码方式输入盲文。

[0320] 表26:对称方式和循环方式方位编码与各种轨迹输入编码关系

[0321] TRIA “比合”二进制编码	对称			循环			盲文字母 R 八进制 (7, 2) [111][010]	编码方式	R 轨迹[7][2]	易卦	盲文
	3	7	6	7	0	1		钟表式(圆)	7S2		
	2	S	5	6	Q	2		方向盘式(圆)	[01234567][76]		
	1	0	4	5	4	3		往来式(直线)	[01234567][76]		

[0322] 布莱尔盲文左三右三排列6点组成的63个点字符相似(与“无点状”合计64种符号编码),使用分组的8进制的2进制组合编码方式,一个是左右排列组合(盲文),另一个是上下排列组合。TRIA二元符号编码是利用一对8种不同的矢量变化的轨迹编码,表示一个文字符号。

[0323] 下面的表示出了根据本发明的钟表式和圆周边式字符系统与现有的编码规则之间的关系在九宫格上S形对称式排列编码,只要记住一方就能推敲出另一方。

[0324] 表27:TRIA二元象数图符号的不同形式与盲文、易卦对照

[0325]	八进制	0	1	2	3	4	5	6	7
	八卦编码	☰	☷	☱	☶	☴	☵	☲	☳
	二进制编码	000	001	010	011	100	101	110	111
	盲文列编码	○ ○ ○	○ ○ ●	○ ● ○	○ ● ●	● ○ ○	● ○ ●	● ● ○	● ● ●
	方位编码	下	左下	左	左上	右下	右	右上	上
	钟表方位	6	7:30	9	10:30	4:30	3	1:30	12
	向心方向	↑	↗	→	↘	↖	←	↙	↓
	离心方向	↓	↙	←	↖	↘	→	↗	↑
	顺时针角度	0°	+45°	+90°	+135°	+180°	+225°	+270°	+315°
	逆时针角度	0°	-45°	-90°	-135°	-180°	-225°	-270°	-315°

[0326] 九宫格的不同方位的角度和周边长度的不同变化的组合方式输入盲文。在触摸板上如图602的④-①-②依次经过感应区滑动,就能输入盲文r字母,比传统的盲文输入相比简单和方便,也可应用于触摸屏移动设备上。

[0327] 下面的表27示出了根据本发明的TRIA“比合”二进制符号的盲文的钟表式和圆周边式编码以角度变化的输入方法步骤。

[0328] 表28:TRIA二元输入法使用步骤,(参考图604,图605,图606)

字母(r)例	盲文左列编码(7)	盲文左列编码(2)	盲文全码(左右列)
盲文 R, [7, 2]	● ● ●	○ ● ○	●○ ●○ ●○
钟表式, R[7, 2] 附图 6-604	↓	←	↙
周边式, R[7, 2] 附图 6-605	+315° 第一步, 顺时针移动	-90° 第二步, 逆时针移动	+++++++ 和 --
标尺式, R[7, 2] 附图 6-606	经过 7 个标尺点 第一步, 从左往右移动	经过 2 个标尺点 第二步, 从右往左移动	----->和<--
			两种左右互逆长度变化

[0330] TRIA方向变化编码输入法(三元输入法)

[0331] 下面将参照图6-图10来说明根据本发明的用于TRIA计算机系统的输入法和输入设备。根据本发明的TRIA输入设备可包括现有计算机系统的输入设备以及根据本发明TRIA输入器二者。或者, 可仅包含根据本发明TRIA输入器。这里, 所述根据本发明TRIA输入器指的是基于以作为矩形的顶点的四个点中的一个点为起始点并通过一笔画方式完成的轨迹的用户输入轨迹的输入装置。这里, 由于用户的一笔画是基于矩形的顶点的一笔画, 因此其移动轨迹, 也即移动方向将包括顺时针、逆时针移动和对角线移动等三种移动方向。

[0332] TRIA方向变化编码输入方法的包括点式输入方法和形状式输入方法, 其特征在于一笔画完成一个文字、数字、标点符号和控制码的符号编码输入, 其中一笔画可经过1~4点组成的矩阵拓扑结构上, 每个点有三种方向的编码组织形式, 包括一笔画经过的点之间的三值编码规则, 判断输入字符方式和一笔画经过的几何图形状结构来识别判断输入字符方式。

[0333] 根据本发明的TRIA一笔画经过的点之间的编码规则判断输入字符方式, 通过设置特定识别区域, 并检测这些区域之间的一笔画经过的不同方向包括顺时针方向、逆时针方向、对角方向(对角点或对角区域)的编码值的组合来判断输入符号, 优点是识别率高, 抗干扰能力强, 输入设备简单, 较低计算处理方式实现高识别率。

[0334] TRIA一笔画经过的几何图形状结构识别方法中, 包括结构识别法和统计识别方法。其中结构识别方法是通过建立手写模板库和单字特征数据库, 对用户的输入的字进行模板匹配, 如果不能确定则给出类似字。结构识别法的优点是区分相似字的能力强, 缺点是抗干扰能力差。统计识别方法是将输入字看为一个整体, 其所有的特征是从这个整体上经过大量的统计而得到的, 然后按照一定准则所确定的决策函数进行分类判决识别输入符号。统计识别的特点是抗干扰性强, 缺点是细分能力较弱。相对于传统的英文和其它符号相比, TRIA的四点轨迹生成的几何图形状, 更容易被计算机识别。

[0335] 因此, 用户所输入的一笔画与根据本发明的TRIA数据(存储)格式的X链和Y链的映射关系如下: 根据本发明的TRIA数据(存储)格式的X链和Y链的特定位(例如, 最低位或最高位)的值的组合表示轨迹的开始点, 剩余位的值表示一笔画的移动方向。

[0336] 根据上述映射规则, 还可将无法通过一笔画或无需通过一笔画输入的轨迹限定为特定控制字符。


[0337] 因此, 可通过将用户的输入轨迹与字符的映射来实现TRIA输入法。图6为示出根据本发明的轨迹、字符与TRIA对称三进制数据(编码)三者的关系示图。在下面的描述中, 基于

本发明的一笔画轨迹称作TRIA符号,且基于这种TRIA符号的系统将被称作TRIA符号系统。

[0338] 根据本发明的TRIA输入设备可由现有计算机系统的输入设备或输入系统实现。按照方向变化和按照角度变化进行编码输入。此符号语言的对称特点容易理解其符号生成原理和书写和识别,TRIA语言符号和几何图形相似,因此可独立于民族和地域文化存在。TRIA文字符号是以QWERT计算机键盘的符号为基础一一对应设置其文字、数字、标点符号和控制符,更加容易跟人机交互。

[0339] 下面将介绍根据本发明实施例的基于TRIA文字符号编码的各种输入法和输入设备。

[0340] TRIA角度变化编码输入法(二元输入法)

[0341] 根据本发明的TRIA文字符号编码的二元输入法是一种根据角度变化编码的输入方法,通过()九宫格的中心点与其它周边的各8个点之间角度或边缘轨迹的不同一笔画完成数字和文字符号的输入方法,以钟表指针的变化输入文字符号方法,解决盲文对盲人阅读容易触摸屏设备上,输入难的问题。下面将参照图6介绍根据本发明的二元输入法(盲文八进制输入方法)的几个实施例。

[0342] 钟表式输入法

[0343] 根据本发明的一个实施例的“钟表式”的角度输入方法具体的如下:首先,如图6的604所示,在圆形或“田”字形的周边上形成8个不同的检测区域,以其中的任意一个检测区域为起始点向心方向手势滑动经过中心点后,再次从中心点离心方向手势滑动移动到周边8个不同方向的区域,完成一个符号文字的输入方法。也就是说,根据本发明的钟表式输入法从周边起始、经过中心、最后在周边结束的顺序滑动完成一个盲文符号的输入,通过一次向心和一次离心滑动组合来完成成一个符号文字的输入方法。手势输入的编码容量一共为64种不同的编码组合方法,其中周边8个点的编码方式顺序有0形或S形的分布方式。

[0344] S形的编码形式是以8个点中的一个起始点以此顺时针(逆时针)排列编码到一半(0、1、2、3),下一半(4、5、6、7)的编码顺序以逆时针(顺时针)排列编码方式。盲文字母r为例,左侧编码1-2-3为111以八进制表示7,右侧编码4-5-6为010以八进制表示2;按照角度输入方法对照表,从九宫格上部为7,左部为2。输入设备切换成角度输入方法,触控面板上手指按下后,从周边上部分(7)移动到圆心部分,再次移动到周边左部分(2)后手指离开触控板,输入设备采集到触控板的不同部位的坐标数据,通过角度输入法处理触控事件数据的比对判断出用户输入的是八进制的72,按照对应表查询对应的字母(r)输入到系统,完成一个字母的角度输入。

[0345] 方向盘式输入法

[0346] 如图605,以“周边式”长度输入方法具体如下,每45°周边长度为一个单位把一个圆分成8个部分。以此移动一个单位分别表示(1、2、3、4、5、6、7、0)顺时针方向移动为盲文的左列1-2-3编码和逆时针方法移动为盲文的4-5-6编码对应。圆形手表为例,一个闭合两段有电压差的触控线圈布置在手表的周边边框上,当手移动边框的上移动变化而改变改变线圈的电流值,通过采集这些连续变化的一对增大和减小的电流值的一组输入数据进行计算处理,判断用户输入的文字符号,周边输入方法。盲文字母r为例,左侧编码(1-2-3)为111以八进制表示7,右侧编码(4-5-6)为010以八进制表示2;按照周边输入方法对照表,首先手指接触到触控线圈上,从表盘周边顺时针移动315°左右的弧度移动,不离开触控线圈的情况

下逆时针方向移动90°完成盲字母的r的输入。

[0347] 0形的编码形式是以8个点的其中一个为起始点,一次顺时针或逆时针以此排列方式,操作方式跟S形编码形式相同,S形的编码形式相对0形编码形式更容易记忆和操作方面,0形编码具体实现方式省略。

[0348] 标尺式输入法

[0349] 如图6-606,以“标尺式”长度输入方法具体如下,在一个线段分成8个感应节点,左起始点排列依次为(0、1、2、3、4、5、6、7)感应点滑动经过至顶点,此路径上感应节点的个数;再次从右起始点开始依次滑动通过指定感应节点,输入系统通过电流感应判断相应编码,左起始点滑动长度单位对应盲文左列的8进制编码,右起始点滑动长度对应盲文右列8进制编码,通过采集的左右两次编码值和盲文左右列的编码值匹配查询对应的盲文字母。满文字母r为例,从左开始滑动至右端第7个节点(1-7单位),再次从右端点开始滑动2个单位(7,6单位)完成输入盲文r字符。

[0350] TRIA输入装置

[0351] 如上所述,根据本发明的TRIA符号编码系统和二元输入方法法和三元输入方法结合,介绍基于TRIA编码的输入装置,包括TRIA移动设备、TRIA字母全键盘与手写输入板、TRIA表键盘、TRIA立体键盘、TRIA虚拟立体键盘和TRIA卡片键盘。

[0352] TRIA移动设备

[0353] 如图7的700所示,根据本发明的输入装置和输入法是通过输入根据本发明的TRIA亚语言字元,然后将该字元与现有的语言体系的文字字母进行对应之后输出我们熟知的文字字母。

[0354] 结合附图7的移动设备,跟传统触摸屏移动设备上的全键盘相比,虚拟键盘覆盖显示屏上的一部分信息覆盖,而特别是表格等信息的录入上不便,为了解决此问题,如附图7-701所示,TRIA虚拟键盘是通过透明的四点显示在屏幕上,通过本发明的TRIA三元输入方法,在虚拟键盘上录入信息。

[0355] 在触摸屏移动设备上,解决盲文的输入困难问题,在移动设备的屏幕下方,不影响美观的条件下,如附图7-702所示TRIA盲文键盘,设置8个突起点的感应线,当手指移动而改变线两段的电流变化感应处理,根据本发明的标尺式TRIA二元输入法输入信息。

[0356] 在移动设备的指纹系统,指纹被盗用会引起安全问题。为此,附图7-703所示的TRIA指纹键盘,在采用指纹系统识别的同时,通过手指在指纹键盘上采用如上所述的基于四个点的滑动方式录入,生成基于TRIA三元符号字母的密码字符串轨迹符号来输入字符密码。这样当两道密码都通过时,可以登录到移动设备上。

[0357] 因为TRIA二元或三元编码的简单易用性,可通过附图7-404上的一对摄像头检测隔空方式,用手指、眼睛等运动轨迹片段来输入信息。

[0358] 与图7的右侧所示的传统输入法相比,本发明的输入法不用在屏幕上显示键盘的虚拟按键,从而可以显示更多的内容。

[0359] TRIA字母全键盘和手写输入板

[0360] 附图8-801示出的是TRIA字三元符号全键盘,跟传统全键盘方式相同,可通过击打每个按键来按照不同的输入法在计算机上输入TRIA符号语言或英语等其他语言。

[0361] 附图8-802示出了TRIA触控板键盘。在传统的触控板基本上只使用于鼠标的定位。


根据本发明的触控板键盘,在触控板上设置田字形状设置9个触控检测区域,这样可通过如上所述的本发明的输入“钟表式”符号语言来输入盲文或其他自定义控制快捷键,或四点一画TRIA三元输入法。

[0362] 附图8-803示出的是TRIA盲文线性键盘,在不影响传统键盘录入和美观的条件下,键盘左下方或右下方设置如上所述的TRIA盲文线性键盘,解决盲文输入的问题,具体操作参照附图6-606和相关说明。

[0363] TRIA表键盘




[0364] 附图8-804示出了TRIA的表键盘。这是TRIA符号编码在穿戴式设备上的应用。根据本发明的TRIA表键盘,外观上跟传统手表美观和样式不变的情况下,在表盘的边缘周围配置8个如附图8-805TRIA表感应点,并且这些感应点和表盘表面上涂上感应材质。根据本发明的TRIA三元输入法或TRIA二元钟表式输入方法来输入不同类型的控制信息,表冠的旋转配合与按下确认等方式,把指令信息无线传输到智能家居或智能车上,从而可以控制外部设备。在表镜圈扣上配置,结合附图8-806的TRIA表感应圈,可根据本发明的TRIA二元方向盘式输入方法来录入信息和控制设备。

[0365] TRIA立体键盘

[0366] 附图9-901示出了根据本发明实施例的TRIA立体键盘。立体键盘的表面使用触摸屏,形成三阶魔方的54个表面方块(6个面9个格,总共54个表方块)上布置有TRIA三元符号的一笔画符号,这样正好是53个特征点加上“”符号的54个TRIA符号。在三阶魔方体表面布置TRIA的符号,通过这些符号的触击方式录入信息。

[0367] TRIA虚拟立体键盘

[0368] 附图9-902示出了TRIA虚拟立体键盘。在TRIA立体键盘的各TRIA符号的布置内容的基础上,立体键盘的6个面上的中心方块部分的6个符号在一个触摸屏上显示。如附图9-903的TRIA虚拟显示层,当需要的文字符号录入时,首先选择符号所在的6个面上的中心方块上的符号,此时触摸屏所选符号正上方上弹出覆盖在中心方块上的符号框(图9-902),显示符号框是如附图9-904的各立体键盘关联的TRIA虚拟覆盖层的8个符号,通过手指再次点击选择或滑动到符号的区域上一画完成录入此符号。虚拟立体键盘的输入方法,只用点击和滑动8个方位的方法,录入文字、数字、标点符号和控制命令。该方法通过TRIA符号的对称性和结合特性,跟TRIA三元输入法需要各轨迹都要移动才能完成相比,具有输入速度快的特点。

[0369] 如附图9-901和9-904所示,先定义6个面上的每个面的中心方块和关联的8个边角方块上的TRIA符号,按照TRIA符号的各类特征分类安排,用户需要记忆每个面上的符号特征。将6个面上的中心方块的TRIA符号显示在用户输入的输入界面上,如903所示,六个中心按键部署在9宫格或6宫格上。要输入字母b时,先选择b所在的面的中心方格的符号“”,此时输入系统界面的选择符号正上方,弹出902的9宫格符号叠加显示在输入界面“”中心方块和8个边角方块上的TRIA符号,用户如902所示再次点击选择或手势滑动到要输入的符号“”的方格上完成字母b的输入,输入系统通过判断用户输入的中心方格和边角方格的趋势或两次点击,一画趋势输入英文字母或其他各国文字,提高输入速度。

[0370] TRIA安全卡片键盘

[0371] 结合附图10的TRIA安全卡键盘,包括附图10-1001TRIA安全芯片、键盘录入感应面

板,通过手指在感应面板上移动产生的电流变化和按下所产生的压力感应,根据本发明的TRIA三元符号编码输入方法可产生相应的密码,该密码在TRIA安全芯片上的易失性记忆存储介质暂时保存。当读取了身份验证设备上通过无线或接触性交换的录入的密码并且安全芯片上的身份密钥相互匹配后确认通过并清零此卡片上的暂存的密钥信息,或一段时间后易失性记忆存储介质上信息自动清零,保证卡片的身份认证的安全。

[0372] 附图10-1002示出了TRIA四点滑动式键盘,其根据本发明的四点一画的TRIA三元编码模式输入信息。附图10-1005示出了TRIA左右式键盘,其根据本发明的TRIA符号编码组合完成输入信息。附图10-1003示出了TRIA拨号式键盘,其根据本发明的TRIA符号中的一些基本几何符号,辅助配合1004的旋转,按照对称的转换方式录入信息。附图10-1006的TRIA钢琴式键盘将根据本发明的53个一笔画符号以钢琴形式分类排定作为一种键盘模式来录入信息。

[0373] TRIA存储系统

[0374] 如上所述,根据本发明的TRIA计算机系统,通过两个二进制数据单位来表示一个对称三进制数据单位。具体地讲,两个二进制运算单位中的一半用于表示具有对称三进制中的“1”的位,而另一半用于表示具有对称三进制中的“T”的位,即,两个数据单位中的一个数据单位表示对称三进制数据中的“正数”部分,并将这部分称作“X链数据”、“X链”或“正数链数据”,而另一个数据单位表示对称三进制数据中的“反数”(负数)部分,并将这部分称作“Y链数据”、“Y链”或“反数链数据”。

[0375] 串行和并行“比合”三进制数据相互转换

[0376] 如上所述,根据本发明的TRIA计算机系统,通过两个二进制数据单位来表示一个对称三进制数据单位,因此根据本发明的TRIA存储系统可采用现有的存储器结构存储对称三进制数据,即,根据本发明的TRIA存储系统可存储二进制数据和比合三进制,或比合三进制转换为对称三进制数据,最小存储单位是“才特”或“才亚”。

[0377] 附图4-404示出了根据本发明的并联和串联“比合”互换电路。一对“正、反”并联的并行输入信号(A、B、C)和(a、b、c),经过中合层的转换电路进行“奇、偶”一对的“比合”处理,输出到串行的[1,2],[3,4],[5,6],即,输出奇偶“比合”三进制串联(Zipper)转换方式。与此同理另一端[1,2],[3,4],[5,6]串行输入后,输出信号(A、B、C)和(a、b、c)的并联(Un-Zipper)转换方式。

[0378] 并联与串联“比合”互换电路在硬件上不仅可用传统的单面印刷电路方式,根据TRIA数据结构的特点还可在三层电路板上实现,其中,正数部元器件布置在上层电路板上、反数部元器件布置在下层电路板上、正反相互结合的中间层部分,通过奇偶相互交替方式把上层和下层的一对信号转换电路布置在中间层电路上。

[0379] TRIA“比合”计算处理过程中包括两种计算机存储方式。一种是“比合”三进制的正反一对数据以相互交替方式串联(Zipper)操作的存储方式。另一种是正、反两个阵列的一对数据并联(Un-Zipper)操作的存储方式。TRIA“比合”三进制存储结构的串联布置方式和并联布置方式之间,可以相互转换。例如,计算机的寄存器AX中高位(AH)放置正数部数据,(AL)低位放置反数,或相反方式,低位放置正数、高位放置反数。另一种是在计算机存储结构中,满二叉树(B-Tree)存储结构方式,左叶节点上存储正数(X链)部数据,右叶节点上存储反数部数据(Y链)存储方式。

[0380] 表29串联布置至并联布置的转换

二叉树			满二叉树左叶节点组								满二叉树右叶节点组							
寄存器	AX		AH 高位(X 链)								AL 低位(Y 链)							
[0381] 节点编号	位操作	位	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
左右分叶	Un-Zipper	并	1	0	0	1	1	0	1	0	0	1	0	0	1	1	0	1
满二叉树/AX	Zipper	串	1	0	0	1	0	0	1	0	1	1	0	1	1	0	0	1

[0382] 如上所述,根据本发明的TRIA存储系统中,利用二进制的两个位表示对称三进制的一个位。例如,若两个对应(或配对的)二进制位是(1,0),则表示对称三进制的“1”;若两个位是(0,0),则表示对称三进制的“0”;若两个位是(0,1),则表示对称三进制的“T”。由于不存在与(1,1)对应的对称三进制的“数”,因此二进制对(1,1)可用来表示控制符或用于特殊用途,(1,1)比合操作(0,0)相同结果。

[0383] 现有的存储器结构中,可以以16比特、32比特、64比特等单位存储数据。下面的描述中,为了描述的方便,假设根据本发明的TRIA存储系统以16比特为单位存储数据,即,二进制数据 and 对称三进制数据都是以16比特位单位存储数据。需要注意的是,虽然下面将以二进制和对称三进制均以16比特为单位存储数据的假设进行描述,但是根据本发明的TRIA存储系统,二进制和对称三进制的存储数据的单位可以不同,也可以相同,且存储数据的单位可以包括但不限于16比特、32比特、64比特。

[0384] 根据本发明的TRIA存储系统以16比特为单位存储数据,即,一个字或两个字节。根据本发明的TRIA存储系统,将16比特的存储单位划分为正数链(以下,简称“X链”)和反数链(以下,简称“Y链”)。

[0385] 根据本发明的TRIA存储系统中,X链中与对称三进制数据的“1”对应的位为“1”,其余位为“0”,而Y链中与对称三进制数据的“T”对应的位为“1”,其余位为“0”。例如,针对对称三进制数据A=(0010TT01),其等于十进制 $1 \times 3^5 + (-1) \times 3^3 + (-1) \times 3^2 + 1 \times 3^0 = 243 - 27 - 9 + 1 = 208$ 。X链的(00100001),在十进制可以是 $1 \times 3^5 + 1 \times 3^0 = 244$,而Y链的(00001100),在十进制可以是 $(-1) \times 3^3 + (-1) \times 3^2 = 36$ 。因此,A=X链值-Y链值。

[0386] 如上所述,由于X链中与对称三进制数据的“1”对应的位为“1”,其余位为“0”,而Y链中与对称三进制数据的“T”对应的位为“1”,其余位为“0”,因此不会出现情况,而这种X链和Y链的对应的位均为“1”的数据可用作信息传递或存储时的控制或校验信号。

[0387] 根据本发明的TRIA存储系统中,X链的数据和Y链的数据分别存储于不同的存储节点上,当TRIA计算机系统使用的时候,并行读入此两个节点上的数据块,“比合”处理成一个完整的数据信息,在“云计算”中把数据拆分成两个部分存储于不同节点,即使一个节点的数据泄露也不影响数据的安全性,只有通过安全机制在一个计算节点上,“比合”两个节点的数据才能解密此数据,二进制计算机存储方式相比提高了“云计算”中的数据安全性。

[0388] 双层记录光盘“比合”三进制存储

[0389] 根据本发明的记录光盘主要包括,基板、记录层和反射层,当此光盘在进行烧录时,激光就会对在基板上涂的有机染料,进行烧录,直接烧录成一个接一个的“坑”,这样有“坑”和没有“坑”的状态就形成了‘0’和‘1’的信号。通过激光头写在光盘记录层,可使用单层不同角度或双层记录的两个反光点方式,TRIA“比合”三进制数的正数部和负数部的数据,通过光信号写入光盘上存储。

[0390] 表30:

[0391]	双层光 存储	对称三进制	1	0	0	T	1	T	T	0	记录 层	/	x	x	\	/	\	\	x	双反射 读取
		比合三 进制	正	1	1	1	0	1	0	0	1	上层	/	/	/		/		/	上层
		反	0	1	1	1	0	1	1	1	1	下层		\	\	\	\	\	\	下层

[0392] 偏振光单层光盘“比合”三进制存储

[0393] 如上所述,根据本发明的TRIA计算机系统,通过一对相互垂直方向的刻录“比合”三进制的正反一对二元数据在写在光盘的一个相互垂直的一个点的交叉区域上,合成写入一个对称三进制信息,当读取信息时两个感光器通过刻录的凸凹垂直反射光分离出来,转换成一对TRIA“比合”三进制数。

[0394] 通过一对垂直交叉的反射光信号值,分别采集和转换成“比合”三进制数据读取双反射光的一位对称三进制数,提高单位存储单元的存储密度。

[0395] 表31:

[0396]	偏振光存储 和传递	对称三进制	1	0	0	T	1	T	T	0	偏振 光		+	+	-		-	-	+	双反射 读取
		比合三 进制	正	1	1	1	0	1	0	0	1	纵波								纵向写
		反	0	1	1	1	0	1	1	1	1	横波		-	-	-		-	-	横向写

[0397] 全息技术存储三进制数据

[0398] 全息存储与全息照相完全相同,同样是利用了光的干涉原理。与其它存储技术不同,全息存储技术并不仅仅利用介质表面,它通过在整个存储介质内记录干涉图案来存储数据,这些干涉图案是由两束激光在某种晶体上相交来改变材料的光学特性所形成。激光器产生的激光束被分光镜一分为二,其中一束被命名为“物光束”,直接照射到被拍摄的物体,另一束则被称为“参考光束”,直接照射到感光胶片上。根据本发明的TRIA计算机系统,TRIA比合三进制信号通过调制解调的两束激光的在感光胶片上产生的干涉条纹记录存储TRIA比合三进制数据信息,再次通过一套高效率的全息照相系统读取感光胶片上的TRIA三进制记录的干涉条纹,提取全息感光胶片上的TRIA三进制的数据。参考表30,两束光的比合三进制信号在光的衍射和干涉下不同的明暗条纹在感光胶片上,记录正反比合三进制的数据信息,并且逆向操作读取比合三进制数据。

[0399] “比合”三进制转“对称”三进制存储

[0400] 如上所述,根据本发明的TRIA计算机系统,正反位的“比合”三进制和一位“对称”三进制存在一对一的关系,因此参照表16的转换方式,可将“比合”三进制数据输出成一个三值状态的存储介质上,提高存储密度,再次,存储介质上的三值状态的信号转换成“比合”三进制数据,在TRIA计算机系统中运算处理,“比合”和“对称”三进制存储信息相互转换。

[0401] TRIA网络系统

[0402] TRIA计算系统的数据结构方式跟二进制计算机的数据结构方式不同,使网络协议上比合式IPv4或IPv6形式表示,IPv4是2的32位表示一个网络地址,网络节点数是 $2^{32}=4.294967296E+9$ 个,基于TRIA三进制的网络节点数是 $3^{32}=1.85302E+15$ 个,IPv6是2的128位表示一个网络地址,网络节点数是 $2^{128}=3.40282E+38$ 个,基于TRIA三进制的网络节点数是 $3^{128}=1.17902E+61$ 个。

[0403] 根据本发明的TRIA计算机网络系统,使用实体IP地址和虚拟IP地址的“比合”方式

组合编制成一个对称三进制地址。如实体IPv4地址为172.16.30.1主机上虚拟机的IPv4地址为192.168.30.2的网络环境下,对称三进制表示 $(0T101100.T0T1T000.0.T0)_Z$ 和对称九进制表示 $(A340.CBC0.0.B)_N$ 的TRIA网络编码。TRIA网络编码IPv4方式的一个子网编码数是 $3^8=6561$ 个,二进制的子网编码数 $2^8=256$ 个相比更多的网络编码容量,解决虚拟机带来的IP数的增加,参照如下表所示。

[0404] 表32:

[0405]	TRIA IPv4 对称三进制表示	$(0T101100)_Z$	$(T0T1T000)_Z$	$(0)_Z$	$(T1)_Z$
	TRIA IPv4 平衡九进制表示	$(A340)_N$	$(CBC0)_N$	$(0)_N$	$(B)_N$
	实体 IPv4	172	16	30	1
		$(10101100)_B$	$(10000)_B$	$(11110)_B$	$(1)_B$
	虚拟 IPv4	192	168	30	2
		$(11000000)_B$	$(10101000)_B$	$(11110)_B$	$(10)_B$

[0406] 如上所述,根据本发明的TRIA网络IP通过实体和虚拟IP地址的正反比合叠加而成的对称三进制或平衡九进制表示一个地址。

[0407] 如上所述,不仅比合方式表示网络地址,可用按照此原理通过比合方式表示内存寻址地址编码标记,如正和反的内存地址的比合地址表示。

[0408] <TRIA计算机系统的软件系统>

[0409] 如上所述,根据本发明的TRIA计算机系统包括TRIA运算处理系统、TRIA输入系统、TRIA输出系统、TRIA存储系统等。TRIA计算机系统可根据进制选择信号,二进制运算模式 and 对称三进制运算模式之间切换。并且可以两者混合计算,比如二进制运算模式处理结果转换为三进制模式处理的输入,相反三进制模式处理结果转换为二进制模式处理的输入。当TRIA计算机系统执行“比合”三进制运算时,TRIA计算机系统切换成“比合”三进制内存模式跟TRIA处理器交换和处理数据。TRIA内存数据,当保存或网络传递时,双链并行方式或奇偶串联方式,把数据存储或传递“比合”数据结构,并且“比合”三进制数据转换为对称三进制方式存储或传递数据信息,TRIA硬件层给软件层提供基于“比合”三进制的对称三进制运算和处理的接口。

[0410] 计算机系统的指令的作用在于,一条指令或指令的序列命令计算机中的特定部分或多个部件执行特定操作。TRIA运算处理系统(即,CPU)根据指令或指令序列的功能,产生相应的操作控制信号,发送给相应的部件,从而控制这些部件按指令的工作功能执行相应动作。TRIA处理器处理指令时,从存储器或双高速缓冲存储器中取出指令程序,放入指令寄存器,并对指令译码。处理器将指令分解成一系列的微操作,然后发出各种控制命令,执行微操作系列,从而完成一条指令的执行。通过指令控制器使多核处理器中的一部分内核切换成对称三进制的计算处理模式,并通过特定的对称三进制指令集提供给软件层,从硬件级别支持对称三进制计算处理,包括基于三进制的操作系统(TOS,即TRIA OS的简写)和三进制的虚拟机(TVM,即TRIA VM)。

[0411] 基于TRIA计算机系统的虚拟机

[0412] 虚拟化或虚拟机(Virtual Machine)指通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统。通过虚拟机,用户可以在一台物理计算机上模拟出另一台或多台虚拟的计算机,这些虚拟机完全就像真正的计算机那样进行工作,例如,用户可以安装操作系统、安装应用程序、访问网络资源等等。

[0413] TRIA计算机系统,从硬件上提供二进制和三进制的平行处理或混合处理。使用硬件计算资源和硬件虚拟化技术可提供基于芯片的功能,借助兼容虚拟机管理器(VMM)软件。

[0414] 如上所述,虚拟机的工作原理是在计算机硬件层和操作系统上面插入一个精简的软件层,该软件层包含一个以动态和透明方式分配资源的虚拟机监视器(VMM或Hypervisor称“管理程序”),多个操作系统可以运行在单台物理计算机上,彼此之间共享硬件资源。由于将整台计算机(包括CPU、内存、操作系统和网络设备)封装起来,由此虚拟机可与所有标准的操作系统、应用程序和设备驱动完全兼容。可以同时单台物理计算机上安全运行多个操作系统和应用程序,每个操作系统和应用程序都可以在需要的时间访问其所需要的资源。

[0415] 下面,介绍根据本发明实施例的利用了虚拟机技术的TRIA计算机系统。

[0416] 如上所述,根据本发明的TRIA计算机系统包括TRIA运算系统、TRIA TRIA输入系统、TRIA输出系统、存储系统等。TRIA计算机系统可根据进制选择信号在二进制运算模式和比合(对称)三进制运算模式之间切换。当TRIA计算机系统执行“比合”三进制运算时,TRIA计算机系统切换成“比合”三进制内存模式并使用TRIA处理器交换数据。当保存数据或网络传递时,采用双链并行方式或奇偶串联方式存储或传递“比合”数据,并且可将“比合”三进制数据转换为对称三进制方式以存储或传递数据信息,同时给软件层提供对称三进制运算的处理接口。

[0417] 在根据本发明实施例的TRIA计算机系统中,在计算机系统的固件(例如,主板)中内置优化的虚拟机管理模块(VMM)和对称三进制计算指令,以将其提供给虚拟机上基于对称三进制计算处理的计算环境,内嵌固化的VMM是通过BIOS/EFI的管理接口而被关闭和开启。默认情况下,内置的VMM的开启是裸机情况下直接接通电源,远程网络就能部署云计算环境。三进制或对称三进制计算中的信息以0或1的形式可分别存放在两个二进制存储节点,计算的时候在计算节点上合并处理成“比合”三进制方式处理对称三进制计算。

[0418] 计算机集成度的提高,在处理器内部完全整合包括南北桥芯片在内的主板芯片组、网络控制器、接口控制器,VMM的软件层的存储也包含在一个芯片当中,简单来说就是完整的SoC(System on a Chip)方式设计,根据本发明的TRIA计算机中VMM也可以内置在一个芯片之中,并且提供API接口给上层的操作系统和应用程序调用三进制驱动程序。

[0419] 在TRIA计算机系统中,比合三进制是基于双二进制数的比合操作原理计算,可以在二进制计算机的操作系统底层,通过软件方式模拟比合三进制计算,从底层包装API接口给上层应用调用。这样,根据本发明的比合三进制的特征不仅在基于比合三进制模式的硬件结构上运行,同时也可以二进制计算环境下通过软件方式模拟比合三进制计算。

[0420] 根据本发明的TRIA计算机系统中,主机开启后,本地或远程登陆(如HP的ILO等远程服务)裸机系统(BIOS/EFI)在操作界面或命令行方式操作,首先TCPU模式关闭时,主机关闭三进制模式正常使用BCPU(二进制CPU)模式运行;开启时,使用“比合”三进制计算处理模式。在(BIOS/EFI)上关闭VMM模式时,需要安装配置操作系统,如果开启了TCPU模式可运行基于三进制的操作系统(TOS),关闭了TCPU模式运行二进制系统(BOS)。开启了VMM模式后,TRIA计算机的系统由VMM核心模块接管,并且内置的安全用户管理模块,提供基于二进制和三进制的混合运行的虚拟机环境服务。登录到VMM上,通过图形方式或命令行方式安装和配置虚拟机,创建一个虚拟机VM时,创建一个配置文件(VMCFG),在配置文件上包括处理器配

置 (CPUCFG)、内存配置 (MEMCFG)、存储配置 (DISKCFG) 和网络配置 (NETCFG)、外设配置 (DEVCFG, 显卡、声卡等) 和系统配置 (SYSCFG)。当虚拟机启动的时候, 宿主机上的配置信息编译成机器编码后, 通过API接口在客户机上的BIOS/EFI上以特殊设备形式 (配置设备), 客户机自检时读取此配置设备信息, 同时为该设备分配中断、通道和I/O端口等资源。客户系统BIOS读取主引导记录, 读取配置设备中的存储配置, 选择操作系统的镜像文件读入到客户机分配的内存中, 从而启动客户机的操作系统。并且从配置设备读取的配置信息找到用户镜像文件挂在到系统中, 增量写入客户机定制的优化配置信息、用户环境配置信息和用户数据到用户镜像中。

[0421] 处理器配置上包括处理器内核使用数量、处理器使用模式 (TCPU)、寄存器叠合处理模式等配置。虚拟机使用的处理器 (CPU) 模式; 如 $T=1$ 时, 调用硬件上的三进制虚拟化计算处理资源, 双寄存器之间的叠合处理等; $T=0$ 时, 虚拟机调用硬件上的二进制虚拟化计算处理资源。

[0422] 内存配置上包括内存大小、内存组织方式、位存储数据结构等。在内存组织方式上, “比合” 计算的特殊性使用双堆栈方式处理“比合” 数据, 在三进制计算机内存的数据结构方式上位存储数据结构上, $ZXY=1$ 时, 使用串联奇偶的数据结构处理“比合” 数据和保存数据; $ZXY=0$ 时使用并联正反的数据结构处理“比合” 数据和保存数据, 在“比合三进制” 计算处理上, 系统判断串联奇偶、并联正反和混合使用模式。

[0423] 网络配置上包括主机的网卡配置、虚拟机网卡物理地址 (MAC) 和网络通信方式, 根据本发明的TRIA虚拟机的网络通过, 主机 (Host) 和客户机 (Guest) 的IP地址的“比合” 方式的三进制IP地址寻址不同主机上的客户机。

[0424] 存储配置上包括存储的方式、存储文件系统格式、存储文件配置类型。存储的方式中内置 (DAS) 和外挂 (SAN, NAS) 等配置, 存储文件系统格式中操作系统的文件系统格式的配置。根据本发明的TRIA虚拟机系统中按照应用类型分类成不同的虚拟存储文件类型类型上包括: 操作系统虚拟文件系统类型 (OSVDFS)、应用服务虚拟文件系统类型 (APPVDFS)、用户数据虚拟文件系统类型 (USRDFS); 存储数据结构上串联奇偶 (Z开头) 和并联正反形式 (X或Y开头), 串联 (并联) 的ZOSVDFS (XOSVDFS, YOSVDFS)、ZAPPVDFS (XAPPVDFS, YAPPVDFS) 和ZUSRDFS (XUSRDFS, YUSRDFS) 形式。例如, 外挂存储方式时, 采用并联正反形式把X和Y虚拟系统文件分别挂接在不同的存储节点上提高读写效率, 在读取时X和Y数据块文件读入到一对堆栈上分别装入并且进行“比合” 计算, 从而简化和提高串联形式的正反分类的操作。根据本发明的TRIA虚拟计算机系统中每个类型虚拟系统文件和配置文件是成对的, 如, 操作OSVDFS文件系统的配置文件OSVDFS.CFG上配置跟具体硬件资源状态下的系统优化配置参数, APPVDFS.CFG上配置具体应用提供服务的最优配置参数, USRDFS.CFG上配置用户数据的存储方式和安全编码方式等配置参数等, 在VMM环境下配置而不是虚拟机启动后在客户机 (GUEST) 上配置, 这种虚拟系统文件类型和配置文件关联方式, 在部署云计算环境时快速分发部署, 配置和类型虚拟文件两个文件部署到各个宿主机上, 在虚拟机管理服务器端只更新宿主机上的配置文件, 就能把虚拟上的应用最优化状态下运行, 而不是在每个客户机上登录后更改优化应用程序的配置文件。比如ORACLE数据库为例, 把数据库系统制作成一个APPVDFS文件并且提供一个VMM上可调优的配置文件, 当虚拟机启动的时候首先读入配置文件到内存中, 并且把APPVDFS虚拟磁盘文件挂载到此配置文件指定的挂载目录, 启动数据

库的时候系统参照从APPVDFS.CFG的配置文件上的参数,在内存中开辟空间给数据库运行,在客户机上更新了数据库的配置就通过VMM提供接口更新VMM环境中外挂配置文件的参数。

[0425] 根据本发明的TRIA虚拟机的虚拟机系统配置上包括客户机的名称、客户机的网络地址、客户机的用户、服务端口等配置参数,当虚拟机启动的时候配置参数首先读入到内存表上,并且更新客户机操作系统的相关配置,在客户机中TOS中发现为一种特殊设备(TVMCFGDEV),运行中通过客户机系统用户权限读写客户机的设备更改系统配置,VM系统通过TVMCFGDEV接口程序实时更新VMM上的各虚拟机的配置表参数。

[0426] 虚拟管理系统只通过OSVDFS的模板的内存镜像,快速克隆给新定制用户环境并配置上参数信息装载和配置产生新虚拟机,用户信息保存在USRDFS用户配置文件系统上,使一个通用模板如OS和多个定制用户组合成不同虚拟机,节省系统模板的每次重复克隆给用户造成的磁盘空间资源浪费。

[0427] 尽管参照其典型实施例的表示和描述了本发明的实施例,以举例说明本发明的原理,但本发明并不限于表示和描述的实施例。应该理解,在不脱离由所附权利要求限定的本发明的精神和范围的情况下,本领域的技术人员可进行各种变动和修改。因此,应该理解,这样的变动、修改及其等同物全部包括在本发明的范围内。

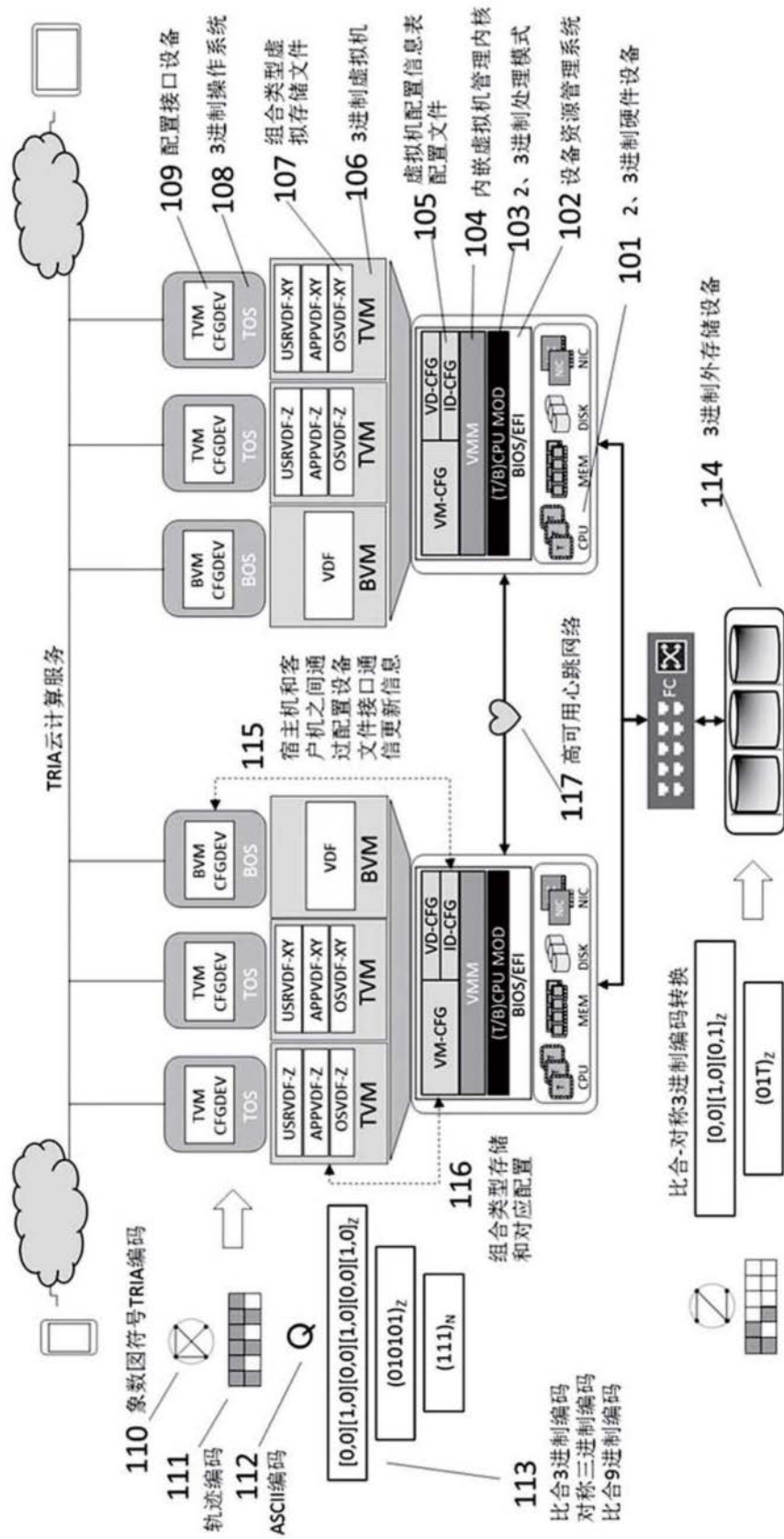


图1

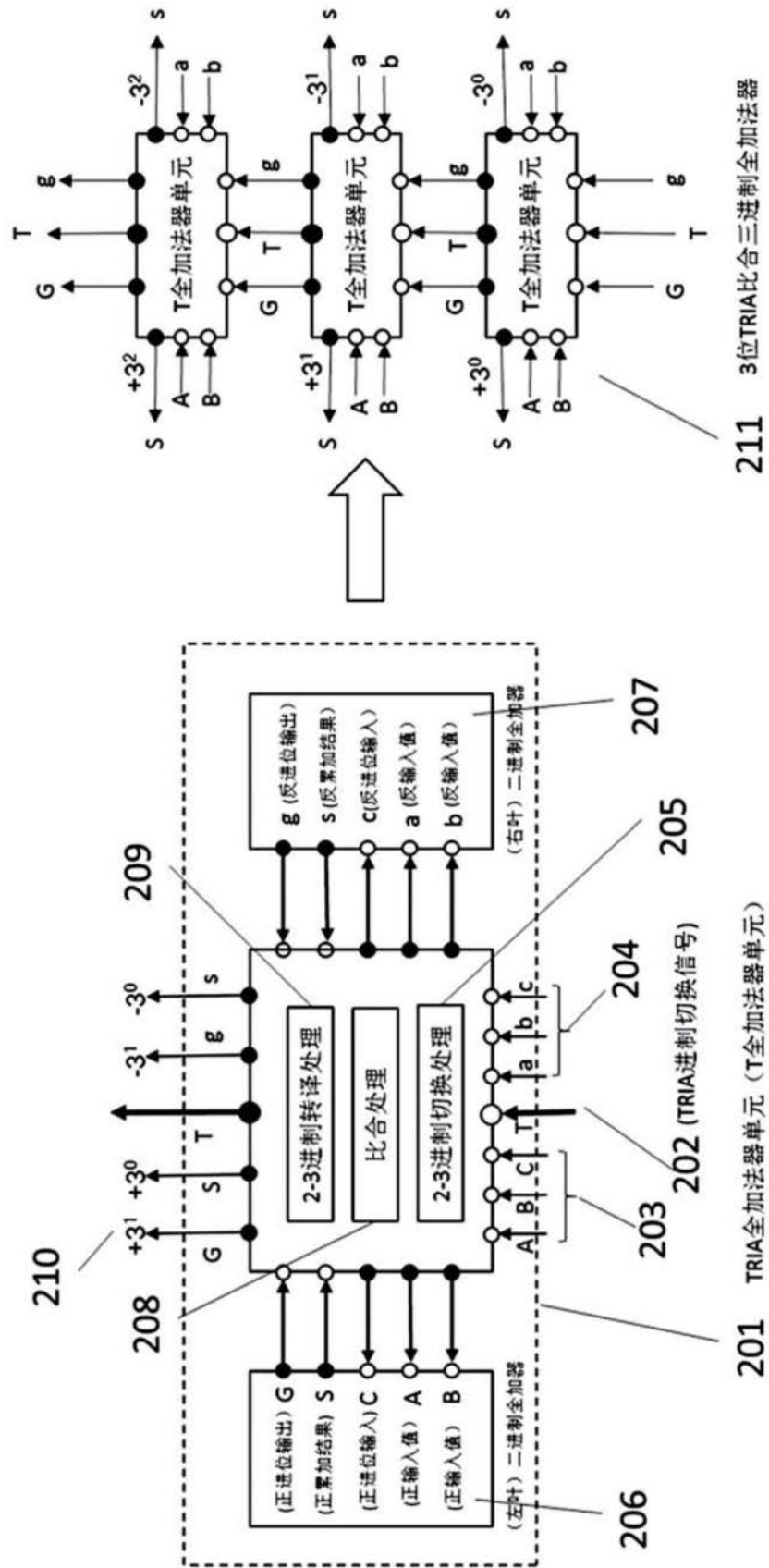


图2

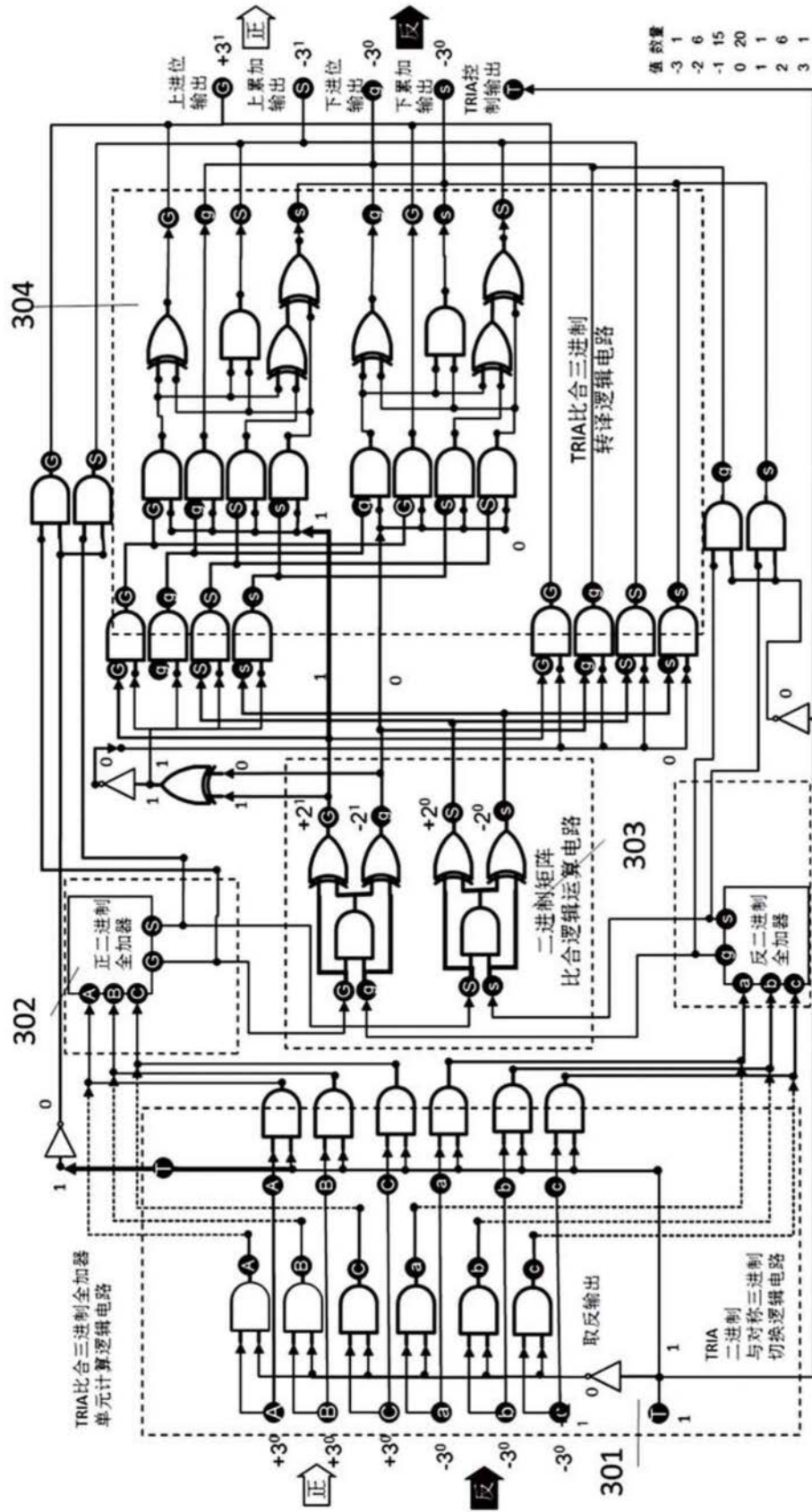


图3

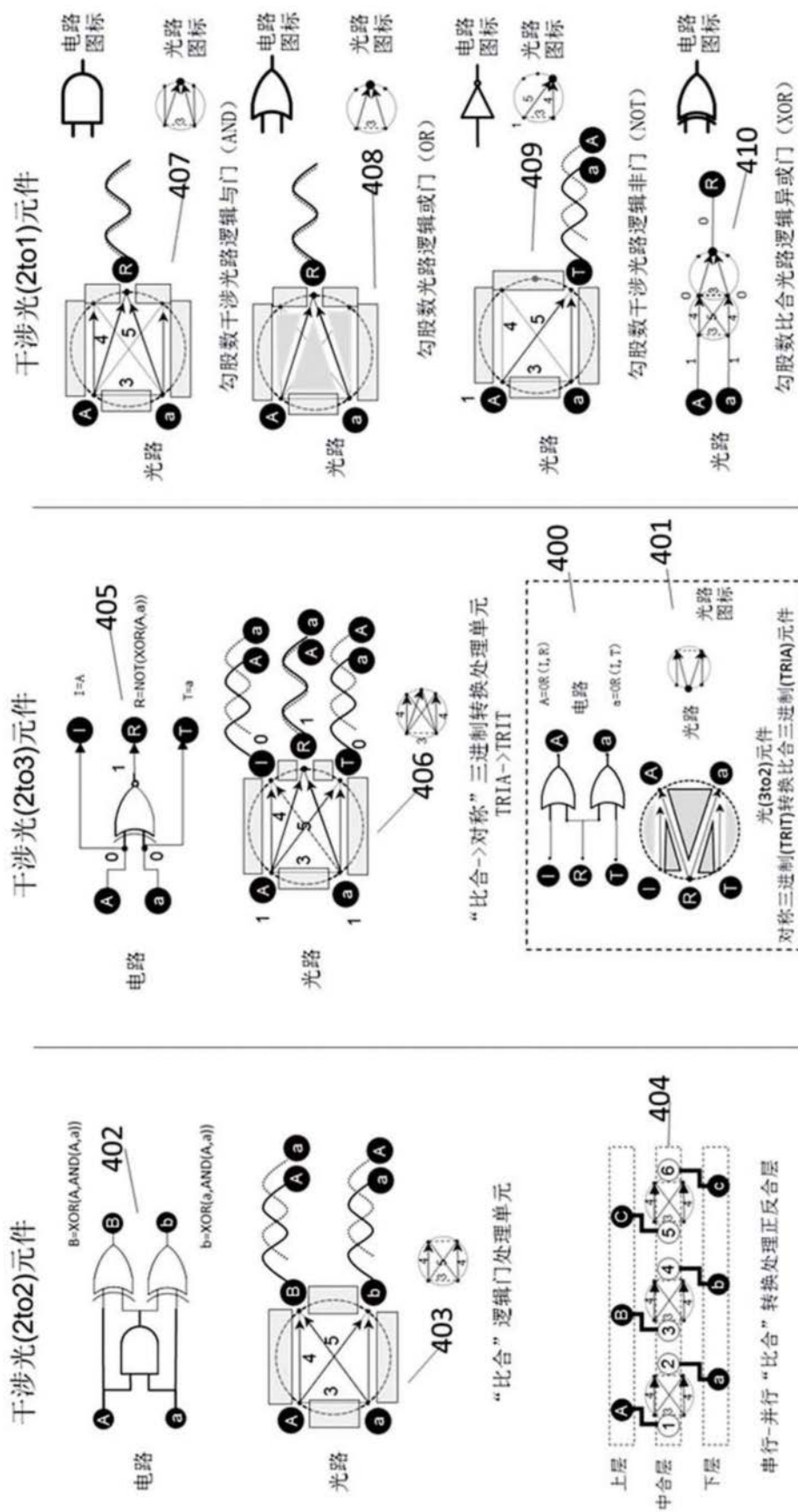


图4

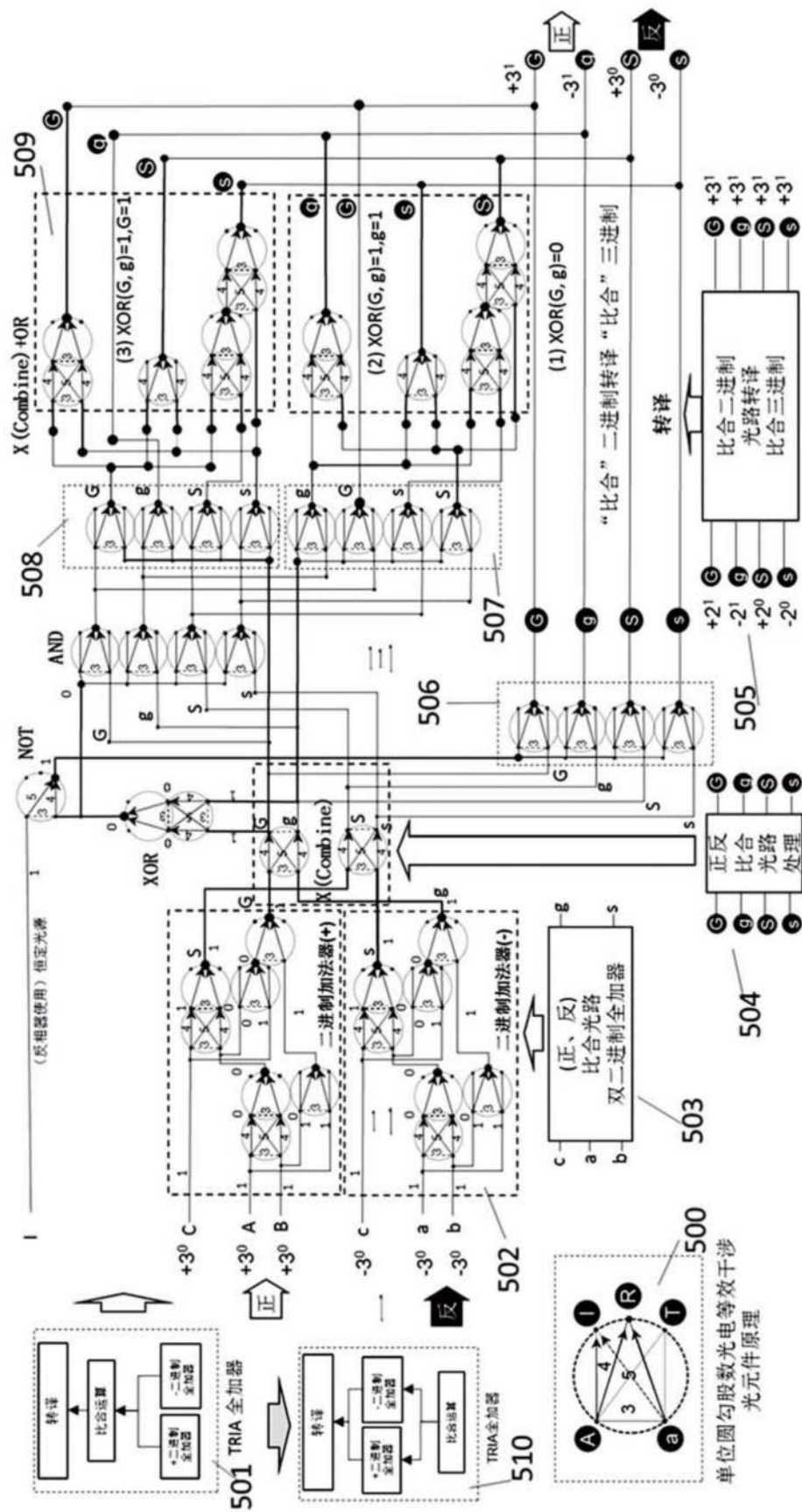


图5

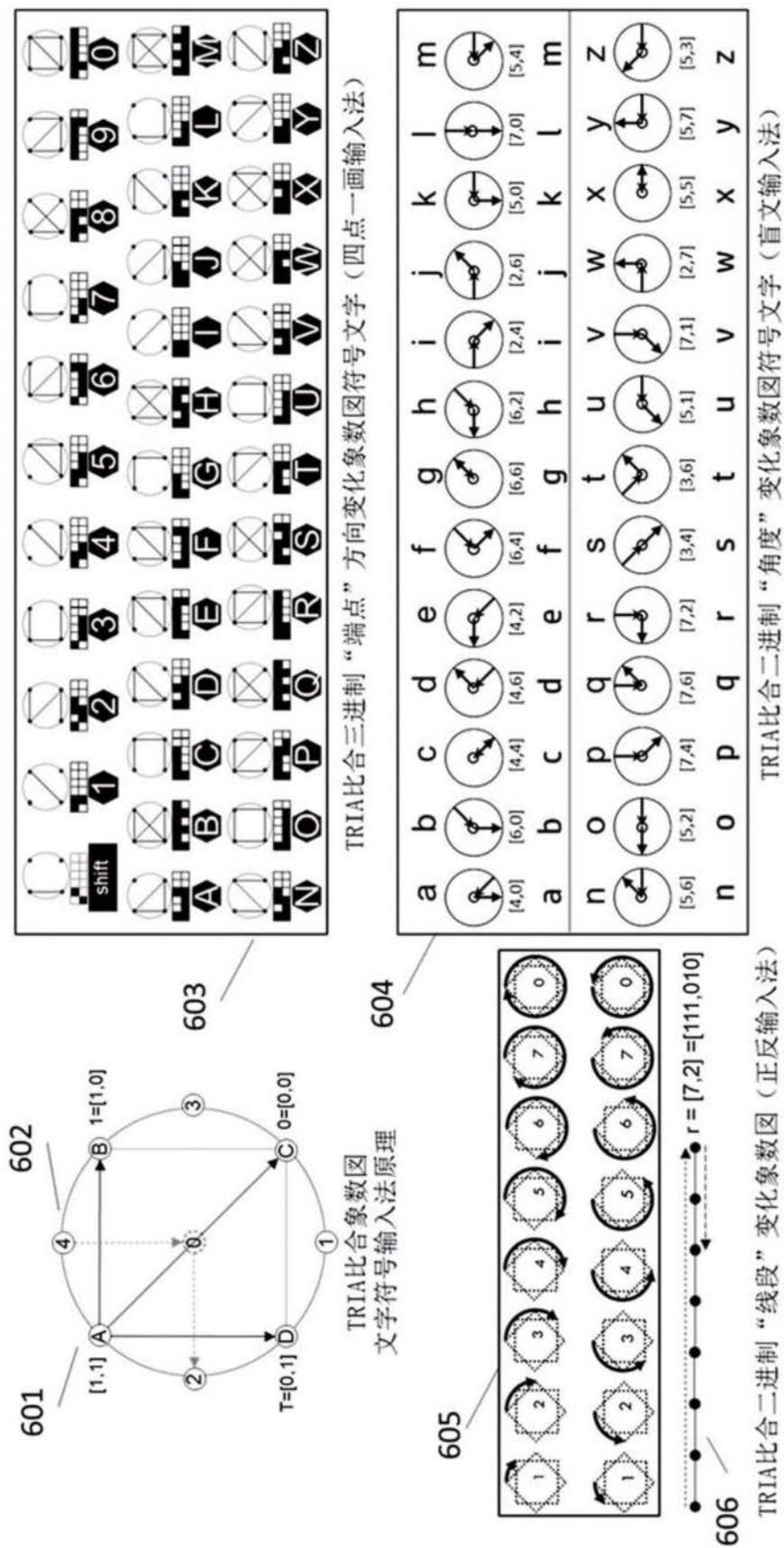


图6

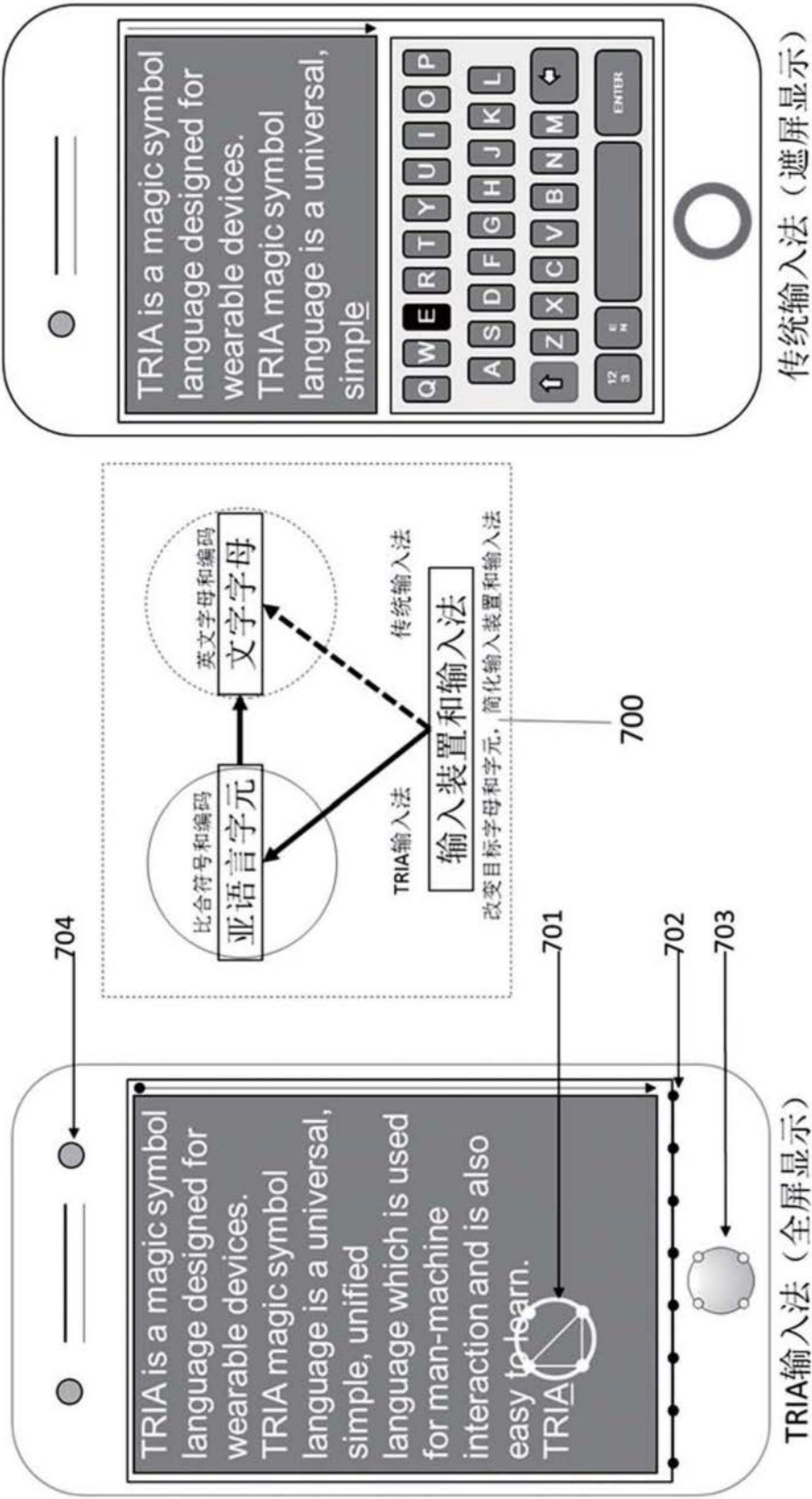


图7

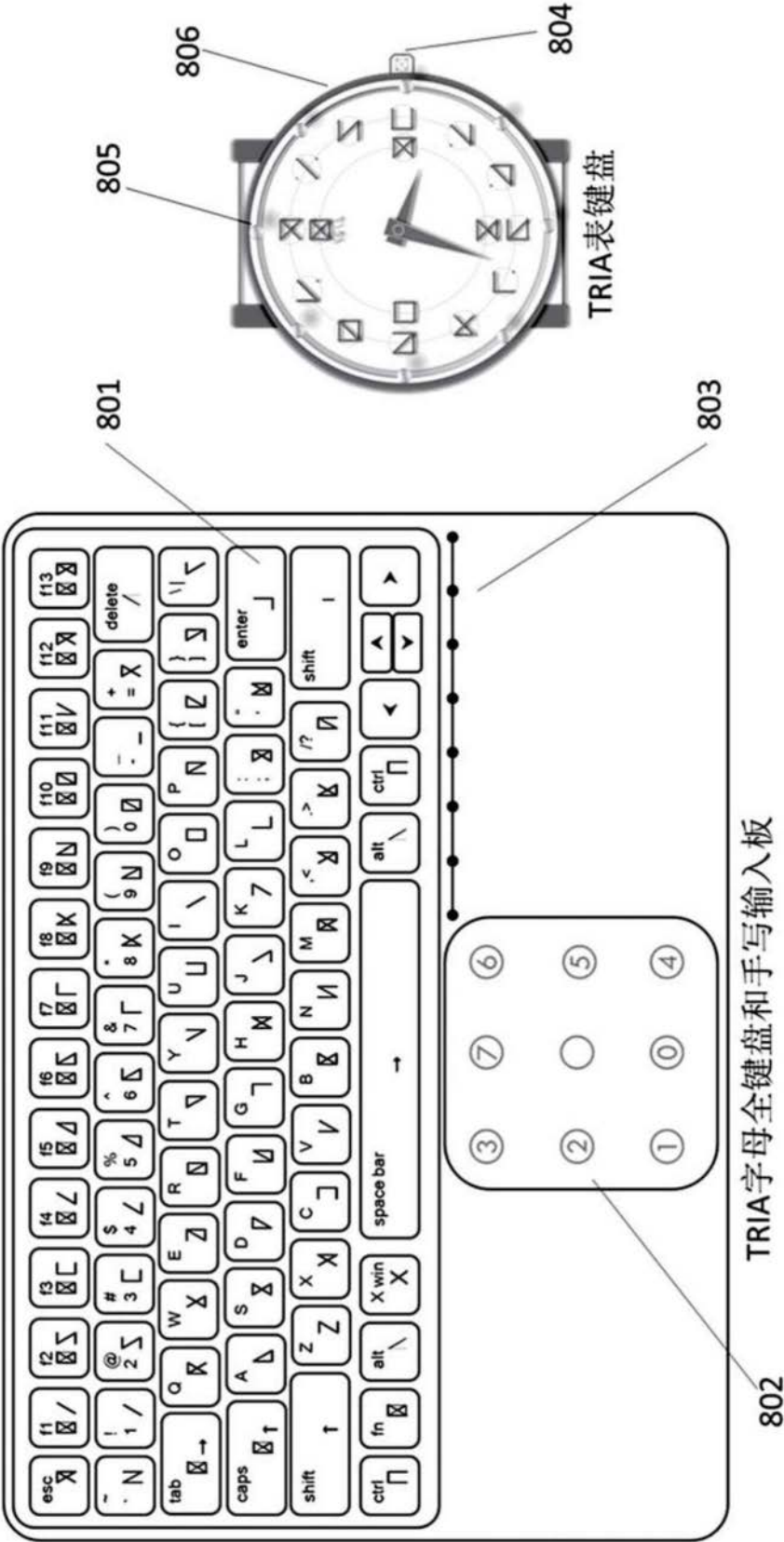


图8

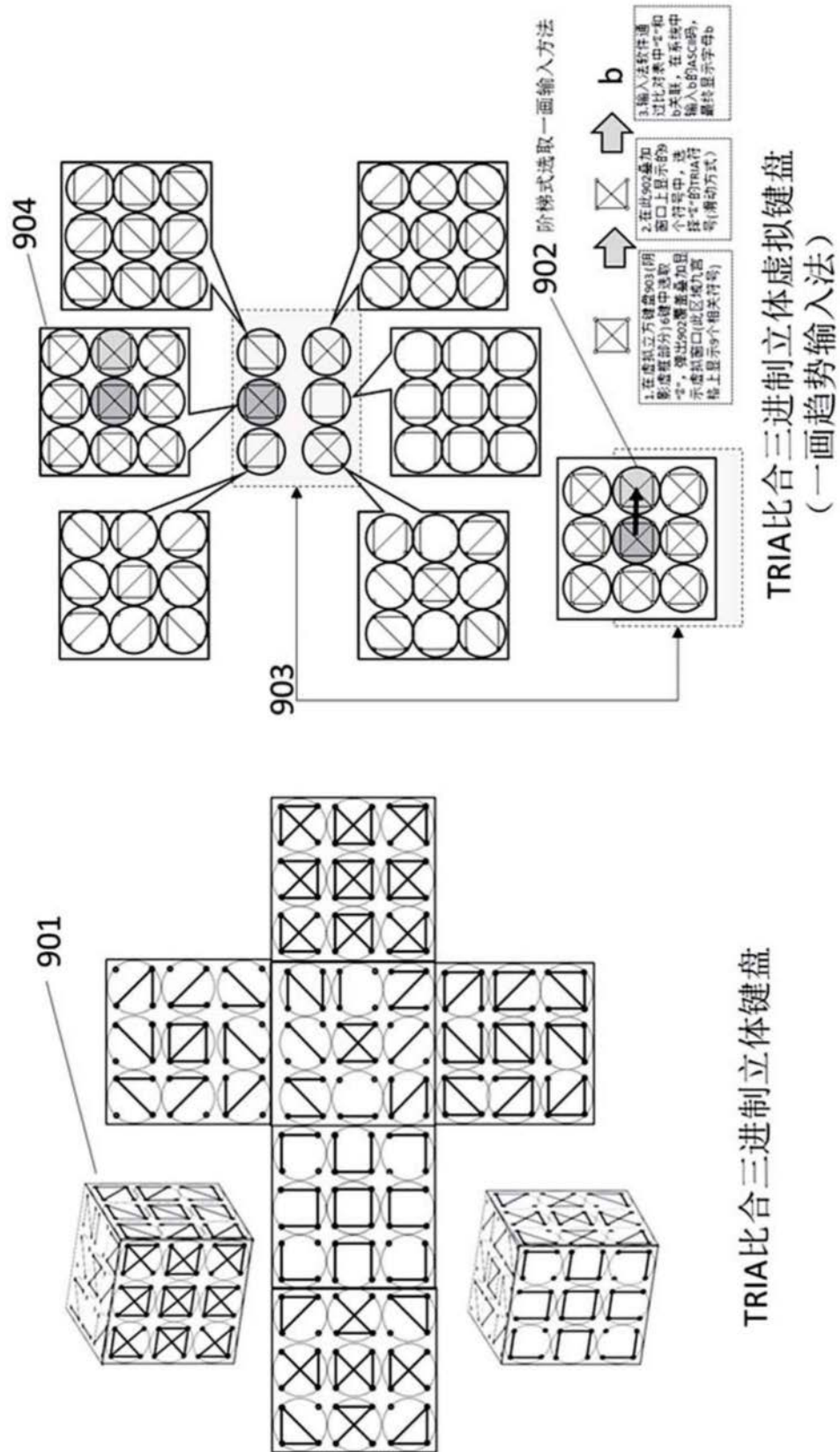


图9

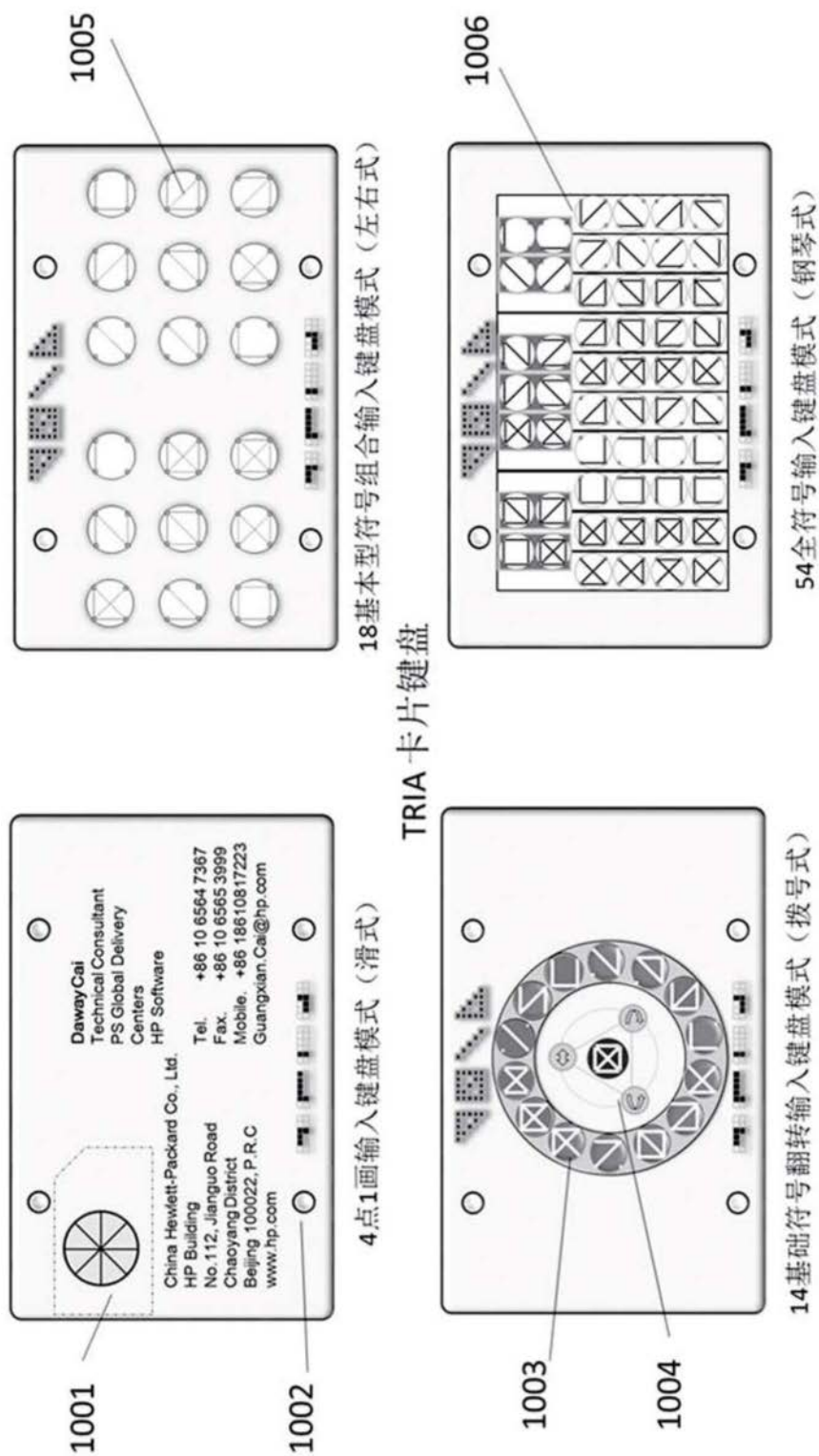


图10