

文章编号:2095-7386(2019)04-0053-08

DOI:10.3969/j.issn.2095-7386.2019.04.011

基于脉冲神经膜系统的有符号整数补码器设计

王防修

(武汉轻工大学 数学与计算机学院,湖北 武汉 430023)

摘 要:为了用脉冲神经膜系统实现有符号加法运算,需要解决有符号整数的补码表示问题。设计了求有符号整数补码的脉冲神经膜系统。首先,设计了将负整数的原码转换为补码的脉冲神经膜系统;其次,设计了将负整数和非负整数分离的脉冲神经膜系统;最后,在负整数和非负整数分离的脉冲神经膜系统的基础上,设计了求任意整数补码的脉冲神经膜系统。系统仿真表明,所设计的脉冲神经膜系统能准确地求任意有符号整数的补码。当前问题的解决,将有助于脉冲神经膜系统的有符号加法器的设计。

关键词:脉冲神经膜系统;有符号整数;二进制补码;分离器

中图分类号:TP 301.6

文献标识码:A

Design of signed integer complement based on spiking neural membrane system

WANG Fang-xiu

(School of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan 430023, China)

Abstract: In order to realize the signed addition operation with the spiking neural membrane system, it is necessary to solve the problem of complement representation of signed integers. In this paper, a spiking neural membrane system is designed for symbolic integer complement. Firstly, a spiking neural membrane system is designed to convert the negative integer code into complement code. Secondly, a spiking neural membrane system is designed to separate negative integers from non-negative integers. Finally, on the basis of the spiking neural membrane system with the separation of negative integer and non-negative integer, the spiking neural membrane system is designed for arbitrary integer complement. System simulation shows that the designed system can accurately calculate the complement of any signed integer. The solution of the current problem will be helpful to the design of the symbolic adder for the spiking neural membrane system.

Key words: spiking neural p system; signed integer; binary complement; separator

收稿日期:2019-05-23.

作者简介:王防修(1973-),男,副教授,E-mail:wfx323@126.com.

基金项目:中国食品公益专项研究基金(201513004-3).

1 引言

当前,针对脉冲神经膜系统的研究^[1-5],理论成果^[6-9]较多,而应用研究相对不足。在为数不多的应用研究中,算术运算是脉冲神经膜系统研究的热点。目前,这类研究已经取得一些成果^[10-13],为生物型 CPU 的设计打下了一定基础。作为计算装置中重要的加法运算,脉冲神经膜系统只能执行无符号整数的加法运算,而有符号整数的加法运算则设计的不充分,需要更多的人工干预才能实现有符号的加法运算。然而,要设计有符号加法器,必须首先设计求有符号整数补码的脉冲神经膜系统。

2 求负整数的二进制补码的脉冲神经膜系统

定理 1 图 1 所示的脉冲神经膜系统能将长度为 m 的二进制原码所表示的负数转化为长度为 m 的二进制补码,而整个转换过程在 $t=3$ 到 $t=m+2$ 步内完成。

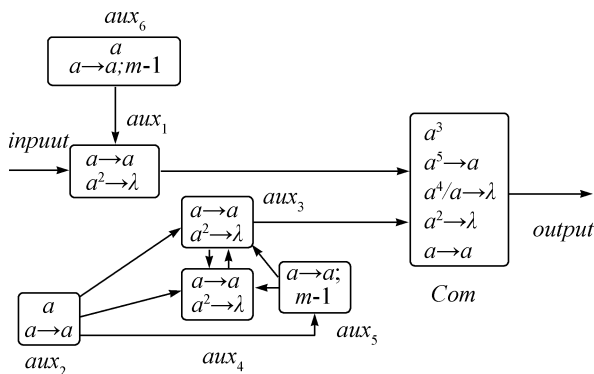


图 1 求长度为 m 的负二进制原码的补码 SNP 系统

证明:图 1 所示的 SNP 系统包括 7 个神经元,它可以实现将负二进制原码转换为补码。该系统定义如下:

$\Pi_1 = (O, \sigma_{aux_1}, \sigma_{aux_2}, \dots, \sigma_{aux_6}, \sigma_{Com}, syn, in, out), with O = \{a\},$

$\sigma_{aux_1} = (0, \{a \rightarrow a, a^2 \rightarrow \lambda\}), \sigma_{aux_2} = (1, \{a \rightarrow a\}),$

$\sigma_{aux_3} = \sigma_{aux_4} = (0, \{a \rightarrow a, a^2 \rightarrow \lambda\}), \sigma_{aux_5}$

$= (0, \{a \rightarrow a; m-1\}),$

$\sigma_{aux_6} = (1, \{a \rightarrow a; m-1\}), \sigma_{Com} = (3, \{a^5 \rightarrow a,$

$a^4/a \rightarrow \lambda, a^2 \rightarrow \lambda, a \rightarrow a\}),$

$syn = \{(aux_6, aux_1), (aux_6, Com), (aux_2,$

$aux_3), (aux_2, aux_4), (aux_2, aux_5), (aux_3, aux_4),$

$(aux_4, aux_3), (aux_5, aux_4), (aux_5, aux_3), (aux_3,$

$Com)\},$

$in = \sigma_{aux_1}, out = \sigma_{Com}.$

图 1 中的输入神经元是 aux_1 , 输出神经元是 Com 。在辅助神经元 aux_2, aux_3, aux_4 和 aux_5 的共同作用下,能保证在 $t=2$ 和 $t=m+1$ 步内向输出神经元 Com 连续输入 m 个脉冲。与此同时,长度为 m 位的负二进制原码从 $t=1$ 到 $t=m$ 步内从低位到高位被逐步输入到输入神经元。辅助神经元 aux_5 在系统中起着延时计数器的作用,即在延时激发规则 $a \rightarrow a; m-1$ 作用下,在第 1 步时被激发,经过 $m-1$ 步延时后,在 m 步时分别向神经元 aux_3 和 aux_4 发送一个脉冲,神经元 aux_3 和 aux_4 在第 m 步时激发规则 $a^2 \rightarrow \lambda$,故神经元从第 $m+1$ 步开始停止向输出神经元 Com 发送脉冲,因此,辅助神经元 aux_2, aux_3, aux_4 和 aux_5 在 $m+1$ 步后都停止工作。输出神经元 Com 在初始状态预置了 3 个脉冲,从第 3 步到第 $m+2$ 步会陆续从神经元 aux_1 和 aux_3 收到脉冲。在第 3 步到第 $m+2$ 步内,如果输出神经元 Com 收到一个脉冲,则该脉冲一定来自于辅助神经元 aux_3 ,故立即激发规则 $a^4 a \rightarrow \lambda$,在下一步会消耗一个脉冲而不向外部环境发送脉冲。反之,如果输出神经元 Com 在第 3 步到第 $m+2$ 步内收到两个脉冲,则一定是一个脉冲来自输入神经元 aux_1 ,而另一个脉冲来自辅助神经元 aux_3 ,故输出神经元 Com 激发规则 $a^5 \rightarrow a$,从而在下一个时刻消耗 5 个脉冲而向外部环境中发生一个脉冲。输出神经元 Com 一旦使用规则 $a^5 \rightarrow a$,将不再拥有脉冲,由于其后所接收的脉冲不会超过 2 个,故输出神经元 Com 所能使用的规则不外乎是 $a^2 \rightarrow \lambda$ 或 $a \rightarrow a$,而这正好就是原码对应位的取反过程。对于一个负整数而言,其二进制原码除符号位是 1 以外,其 $m-1$ 个数值位中至少有一个 1,这意味着输出神经元 Com 一定会执行规则 $a^5 \rightarrow a$ 一次且仅一次,而这就相当于反码的最低位加 1 的效果。当神经元 aux_6 延时 $m-1$ 步后,在第 m 步向输入神经元 aux_1 发送一个脉冲,故在 m 步输入神经元 aux_1 从外界环境中接收到负数原码的符号位 1 个脉冲和辅助神经元 aux_6 接收一个脉冲,从而激发规则 $a^2 \rightarrow \lambda$ 。所以,输出神经元 Com 在 $m-1$ 时刻仅从辅助神经元接收到一个脉冲,故此时激发规则 $a \rightarrow a$,这意味着补码的符号位是 1。因此,图 1 所示的神经膜系统能够准确地求出负整数的补码。

已知 -33 的二进制原码为 1100001,则通过图 1 的脉冲神经膜系统处理过程中各神经元所包含的脉冲数如表 1 所示。

表 1 系统中各神经元在不同时刻所有的脉冲数

Step	input	aux ₁	aux ₂	aux ₃	aux ₄	aux ₅	aux ₆	com	output
0	1	0	1	0	0	0	1	3	0
1	0	1	0	1	1	1	1	3	0
2	0	0	0	1	1	1	1	5	0
3	0	0	0	1	1	1	1	1	1
4	0	0	0	1	1	1	1	1	1
5	1	0	0	1	1	1	1	1	1
6	1	1	0	1	1	1	1	1	1
7	0	2	0	1	1	1	0	2	1
8	0	0	0	2	2	0	0	1	0
9	0	0	0	0	0	0	0	0	1

由于此时的负整数的二进制位数 $m=7$, 从表 1 所示的计算结果可以看出, 在 $t=3$ 到 $t=m+2=9$ 内得到补码为 1011111. 从测试结果以及表 1 所示的不同时刻的各神经元包含的脉冲数可以看出, 图 1 所设计的系统能够准确地计算出负整数的补码。

3 负整数和非付整数分离的脉冲神经膜系统

图 1 所设计的神经膜系统只能计算负整数的补码, 而无法求非负整数的补码。虽然非负整数的原码、反码和补码都是相同的, 但负整数和非负整数的判断应该交给神经膜系统来进行判断。如何从同一个输入神经元进入系统, 然后通过系统的判断, 将负整数和非负整数通过不同的神经元输出, 是本设计需要解决的关键问题。

定理 2 实现长度为 m 的二进制整数的负整数和非负整数分离的脉冲神经膜系统需要 $2m+9$ 个神经元, 而分离过程将在 $t=m+4$ 到 $t=2m+3$ 内完成。

证明: 设计如图 2 所示的脉冲神经膜系统, 它由 $2m+9$ 个神经元组成。系统定义如下:

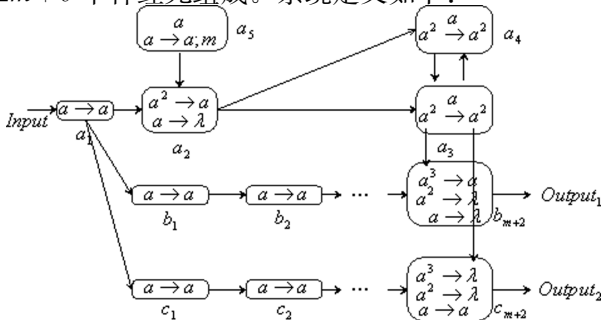


图 2 实现负整数和非负整数分离的脉冲神经膜系统

$\Pi_2 = (O, \sigma_{a_1}, \sigma_{a_2}, \dots, \sigma_{a_5}, \sigma_{b_1}, \sigma_{b_2}, \dots, \sigma_{b_{m+2}}, \sigma_{c_1}, \sigma_{c_2}, \dots, \sigma_{c_{m+2}}, syn, in, out), with,$

$O = \{a\}, \sigma_{a_1} = (0, \{a \rightarrow a\}), \sigma_{a_2} = (0, \{a^2 \rightarrow a, a \rightarrow \lambda\}), \sigma_{b_i} = (0, \{a \rightarrow a\}) \mid i=1, 2, \dots, m+1, \sigma_{c_i} = (0, \{a \rightarrow a\}) \mid i=1, 2, \dots, m+1, \sigma_{a_3} = \sigma_{a_4} = (1, \{a^2 \rightarrow a^2\}), \sigma_{a_5} = (1, \{a \rightarrow a; m\}), \sigma_{b_{m+2}} = \sigma_{c_{m+2}} = (0, \{a^3 \rightarrow \lambda, a^2 \rightarrow \lambda, a \rightarrow a\}),$

$syn = \{(a_1, a_2), (a_1, b_1), (c_1, c_1), (b_i, b_{i+1}) \mid i=1, 2, \dots, m, (c_i, c_{i+1}) \mid i=1, 2, \dots, m, (b_{m+1}, b_{m+2}), (c_{m+1}, c_{m+2}), (a_2, a_3), (a_2, a_4), (a_3, a_4), (a_3, b_{m+2}), (a_3, c_{m+2}), (a_4, a_3), (a_5, a_2)\},$

$in = \sigma_{a_1}, out = \{\sigma_{b_{m+2}}, \sigma_{c_{m+2}}\}.$

图 2 中的输入神经元是 a_1 , 神经元和 b_{m+2} 是两个输出神经元。辅助神经元 b_1 和 $c_i (i=1, 2, \dots, m+1)$ 起着传递脉冲的作用。神经元 a_5 已事先预置了 1 个脉冲, 在延时了 m 步后, 通过激发规则 $a \rightarrow a$, 在 $m+1$ 步向神经元 a_2 发送了一个脉冲。同时, 输入神经元在 $m+1$ 步也可能向神经元 a_2 发送一个脉冲。如果二进制原码是负整数, 则表示符号位的第 m 位是 1 码, 故此时输入神经元 a_1 在 $m+1$ 步向神经元 a_2 发送一个脉冲。此时, 辅助神经元 a_2 由于在 $m+1$ 步接收到两个脉冲, 故通过激发规则 $a^2 \rightarrow a$ 分别向神经元 a_3 和 a_4 发送一个脉冲。在 $t=m+2$, 神经元 a_3 和 a_4 由于都从神经元 a_2 接收到一个脉冲, 而自身本来就有 1 个脉冲, 故都会激发规则 $a^2 \rightarrow a^2$ 。因此, 神经元 a_3 和 a_4 在 $m+2$ 步会彼此向对方发送两个脉冲, 故从 $m+2$ 步开始, 神经元 a_3 和 a_4 会永不停息地向对方发送两个脉冲, 使得这两个神经元在此后的时刻总是有两个脉冲。同时, 从 $m+2$ 步开始, 神经元 a_3 会永不停息地分别向输出

神经元 b_{m+2} 和 c_{m+2} 发送两个脉冲。因此,从 $m+3$ 步开始,输出神经元 b_{m+2} 和 c_{m+2} 在随后的每个时刻都能从神经元 a_3 接收到两个脉冲。在 $m+3$ 步,输出神经元 b_{m+2} 开始接收来自神经元 b_{m+1} 发送的脉冲,而输出神经元 c_{m+2} 开始接收来自神经元 c_{m+1} 发送的脉冲。由于神经元 b_{m+1} 和 c_{m+1} 只是起着传递脉冲的作用,故在 $m+3$ 步,输出神经元 b_{m+2} 和 c_{m+2} 都是接收到输入神经元 a_1 传递过来的脉冲。如果输入神经元 a_1 传递过来一个脉冲,则输出神经元 b_{m+2} 由于总共有 3 个脉冲,故激发规则 $a^3 \rightarrow a$ 而向外部环境中发送一个脉冲。因此,外部环境 $Output_1$ 在步 $m+4$ 接收到一个脉冲。虽然此时输出神经元 c_{m+2} 也有 3 个脉冲,但由于激发规则 $a^3 \rightarrow \lambda$ 而无法向外部环境 $Output_2$ 发送脉冲。如果输入神经元在 $t=m+3$ 以后的某一步没有传递过来脉冲,则输出神经元 b_{m+2} 和 c_{m+2} 由于都只有两个脉冲,故均激发规则 $a^2 \rightarrow \lambda$ 而不向外部环境发送脉冲。从 $t=m+4$ 开始,这样过程需要进行 m 步,直到 $t=2m+3$,该分离过程结束。从该过程可以发现,如果输入神经元传递的是负整数,则输出神经元 b_{m+2} 向外部环境 $Output_1$ 发送的脉冲实质是输入神经元 a_1 发送脉冲的复制,而输出神经元 c_{m+2} 在任何时刻都不会向外部环境 $Output_2$ 发送脉冲,相当于输出神经元 c_{m+2} 被屏蔽。

反之,如果二进制原码是非负整数,则表示符号位的第 m 位是 0 码,故此时输入神经元 a_1 在 $m+1$ 步向神经元 a_2 不发送脉冲。因此,辅助神经元 a_2 由于在 $m+1$ 步接收到一个来自神经元 a_5 的脉冲,

故通过遗忘规则 $a \rightarrow \lambda$ 将该脉冲消耗掉。神经元 a_3 和 a_4 由于未能从神经元 a_2 接收到脉冲,而自身只有一个脉冲,故永远不会激发规则 $a^2 \rightarrow a^2$ 。因此,神经元 a_3 和 a_4 在任何时刻都没有机会向其他神经元发送脉冲。从 $m+3$ 步开始,输出神经元 b_{m+2} 和 c_{m+2} 在随后的 m 个时刻都只能接收从输入神经元 a_1 传递过来的脉冲。如果输入神经元 a_1 传递过来一个脉冲,则输出神经元 b_{m+2} 由于总只有 1 个脉冲,故通过遗忘规则 $a \rightarrow \lambda$ 将该脉冲消耗掉,故它不会向外部环境中发送脉冲。因此,外部环境 $Output_1$ 在 $m+4$ 步没有接收到脉冲。同时,输出神经元 c_{m+2} 也只有 1 个脉冲,但由于激发规则 $a \rightarrow a$ 而向外部环境 $Output_2$ 发送一个脉冲。如果输入神经元在 $t=m+3$ 以后的某一步没有传递过来脉冲,则输出神经元 b_{m+2} 和 c_{m+2} 由于都没有脉冲,故都不会向外部环境发送脉冲。从 $t=m+4$ 开始,这样过程需要进行 m 步,直到 $t=2m+3$,该分离过程结束。从该过程可以发现,如果输入神经元传递的是非负整数,则输出神经元 b_{m+2} 无法向外部环境 $Output_1$ 发送脉冲,而输出神经元 c_{m+2} 在这 m 步向外部环境 $Output_2$ 发送的脉冲相当于输入神经元的复制,而输出神经元 b_{m+2} 由于无法向外部环境发送脉冲而相当于被屏蔽。

因此,如图 2 所设计的脉冲神经膜系统能将负整数和非负整数准确地分离出来。

例如:二进制原码 10110 表示十进制数 -6,通过图 2 的脉冲神经膜处理后,表 2 所示的系统中各个不同神经元在各个时刻所包含的脉冲数。

表 2 脉冲神经膜分离系统中各神经元在不同时刻所拥有的脉冲数

Step	0	1	2	3	4	5	6	7	8	9	10	11	12	13
input	0	1	1	0	1	0	0	0	0	0	0	0	0	0
a_1	0	0	1	1	0	1	0	0	0	0	0	0	0	0
a_2	0	0	0	1	1	0	2	0	0	0	0	0	0	0
a_3	1	1	1	1	1	1	1	2	2	2	2	2	2	2
a_4	1	1	1	1	1	1	1	2	2	2	2	2	2	2
a_5	1	1	1	1	1	1	0	0	0	0	0	0	0	0
b_1	0	0	0	1	1	0	1	0	0	0	0	0	0	0
b_2	0	0	0	0	1	1	0	1	0	0	0	0	0	0
b_3	0	0	0	0	0	1	1	0	1	0	0	0	0	0
b_4	0	0	0	0	0	0	1	1	0	1	0	0	0	0
b_5	0	0	0	0	0	0	0	1	1	0	1	0	0	0
b_6	0	0	0	0	0	0	0	0	1	1	0	1	0	0

续表

<i>Step</i>	0	1	2	3	4	5	6	7	8	9	10	11	12	13
<i>b₇</i>	0	0	0	0	0	0	0	0	2	3	3	2	3	2
<i>c₁</i>	0	0	0	1	1	0	1	0	0	0	0	0	0	0
<i>c₂</i>	0	0	0	0	1	1	0	1	0	0	0	0	0	0
<i>c₃</i>	0	0	0	0	0	1	1	0	1	0	0	0	0	0
<i>c₄</i>	0	0	0	0	0	0	1	1	0	1	0	0	0	0
<i>c₅</i>	0	0	0	0	0	0	0	1	1	0	1	0	0	0
<i>c₆</i>	0	0	0	0	0	0	0	0	1	1	0	1	0	0
<i>c₇</i>	0	0	0	0	0	0	0	0	2	3	3	2	3	2
<i>out put₁</i>	0	0	0	0	0	0	0	0	0	0	1	1	0	1
<i>out put₂</i>	0	0	0	0	0	0	0	0	0	0	0	0	0	0

例如:十进制数 5 的二进制原码是 0101,通过各个不同神经元在各个时刻所包含的脉冲数。图 2 的脉冲神经膜处理后,得到表 3 所示的系统中

表 3 脉冲神经膜分离系统在分离 0101 时各神经元在不同时刻所拥有的脉冲数

<i>Step</i>	0	1	2	3	4	5	6	7	8	9	10	11
<i>input</i>	1	0	1	0	0	0	0	0	0	0	0	0
<i>a₁</i>	0	1	0	1	0	0	0	0	0	0	0	0
<i>a₂</i>	0	0	1	0	1	1	0	0	0	0	0	0
<i>a₃</i>	1	1	1	1	1	1	1	1	1	1	1	1
<i>a₄</i>	1	1	1	1	1	1	1	1	1	1	1	1
<i>a₅</i>	1	1	1	1	1	0	0	0	0	0	0	0
<i>b₁</i>	0	0	1	0	1	0	0	0	0	0	0	0
<i>b₂</i>	0	0	0	1	0	1	0	0	0	0	0	0
<i>b₃</i>	0	0	0	0	1	0	1	0	0	0	0	0
<i>b₄</i>	0	0	0	0	0	1	0	1	0	0	0	0
<i>b₅</i>	0	0	0	0	0	0	1	0	1	0	0	0
<i>b₆</i>	0	0	0	0	0	0	0	1	0	1	0	0
<i>c₁</i>	0	0	1	0	1	0	0	0	0	0	0	0
<i>c₂</i>	0	0	0	1	0	1	0	0	0	0	0	0
<i>c₃</i>	0	0	0	0	1	0	1	0	0	0	0	0
<i>c₄</i>	0	0	0	0	0	1	0	1	0	0	0	0
<i>c₅</i>	0	0	0	0	0	0	1	0	1	0	0	0
<i>c₆</i>	0	0	0	0	0	0	0	1	0	1	0	0
<i>out put₁</i>	0	0	0	0	0	0	0	0	0	0	0	0
<i>out put₂</i>	0	0	0	0	0	0	0	0	1	0	1	0

从表2和表3可以看出,如图2所设计的神经膜系统能够准确地将负整数和非负整数进行分离。

4 求任意符号整数补码的脉冲神经网络系统

图 1 所示的脉冲神经网络系统只能求出负整数的补码,而无法求出非负整数的补码。因此,需要设计求任意符号整数补码的脉冲神经网络系统。

定理 3 求长度为 m 的任意符号二进制整数的补码的脉冲神经网络系统需要个神经元,而求补过程将在到内完成。

证明:设计如图 7 所示的脉冲神经网络系统,它由个神经元组成。系统定义如下:

$\Pi_3 = (O, \sigma_{a_1}, \sigma_{a_2}, \dots, \sigma_{a_8}, \sigma_{b_1}, \sigma_{b_2}, \dots, \sigma_{b_{m+2}}, \sigma_{c_1}, \sigma_{c_2}, \dots, \sigma_{c_{m+2}}, \sigma_{aux_1}, \sigma_{aux_2}, \dots, \sigma_{aux_9}, \sigma_{Com}, syn, in, out)$,
with,

$O = \{a\}, \sigma_{a_1} = (0, \{a \rightarrow a\}), \sigma_{a_2} = (0, \{a^2 \rightarrow a, a \rightarrow \lambda\}), \sigma_{b_i} = (0, \{a \rightarrow a\}) \mid i = 1, 2, \dots, m+1, \sigma_{c_i} = (0, \{a \rightarrow a\}) \mid i = 1, 2, \dots, m+1, \sigma_{a_3} = \sigma_{a_4} = (1, \{a^2 \rightarrow a^2\}),$

$\sigma_{a_5} = (1, \{a \rightarrow a; m\}), \sigma_{b_{m+2}} = \sigma_{c_{m+2}} = (0, \{s^3 \rightarrow \lambda, a^2 \rightarrow \lambda, a \rightarrow a\}),$

$\sigma_{b_{m+2}} = \sigma_{c_{m+2}} = (0, \{a^3 \rightarrow \lambda, a^2 \rightarrow \lambda, a \rightarrow a\}), \sigma_{a_6} = \sigma_{a_7} = (0, \{a \rightarrow a\}),$

$\sigma_{aux_2} = \sigma_{aux_7} = \sigma_{aux_8} = (0, \{a \rightarrow a\}), \sigma_{aux_6} = (1, \{a^2 \rightarrow a^2; m\}),$

$\sigma_{a_8} = (0, \{a^2 \rightarrow a^2; m+1\}), \sigma_{aux_3} = \sigma_{aux_4} = (0, \{a \rightarrow a, a^2 \rightarrow \lambda\}),$

$\sigma_{aux_5} = (1, \{a \rightarrow a; m-1\}), \sigma_{aux_9} = (1, \{a \rightarrow a, a^2 \rightarrow \lambda, a^4 \rightarrow a\}),$

$syn = \{(a_1, a_2), (a_1, b_1), (a_1, c_1), (b_i, b_{i+1}) \mid i = 1, 2, \dots, m, (c_i, c_{i+1}) \mid i = 1, 2, \dots, m, (b_{m+1}, b_{m+2}), (c_{m+1}, c_{m+2}), (a_2, a_3), (a_2, a_4), (a_3, a_4), (a_3, b_{m+2}), (a_3, c_{m+2}), (a_4, a_3), (a_5, a_2), (a_5, a_6), (a_6, a_7), (a_7, aux_2), (aux_2, aux_6), (c_{m+2}, aux_7), (aux_7, aux_8), (aux_8, aux_9), (aux_6, aux_9), (a_3, a_8), (a_8, aux_9), (b_{m+2}, aux_1), (aux_1, Com), (Com, aux_9), (aux_2, aux_3), (aux_2, aux_4), (aux_2, aux_5), (a_5, a_3), (a_5, aux_4), (aux_3, aux_4), (aux_4, aux_3), (aux_2, aux_4), (aux_3, Com), a_3\},$

$in = \sigma_{a_1}, out = \sigma_{aux_2}.$

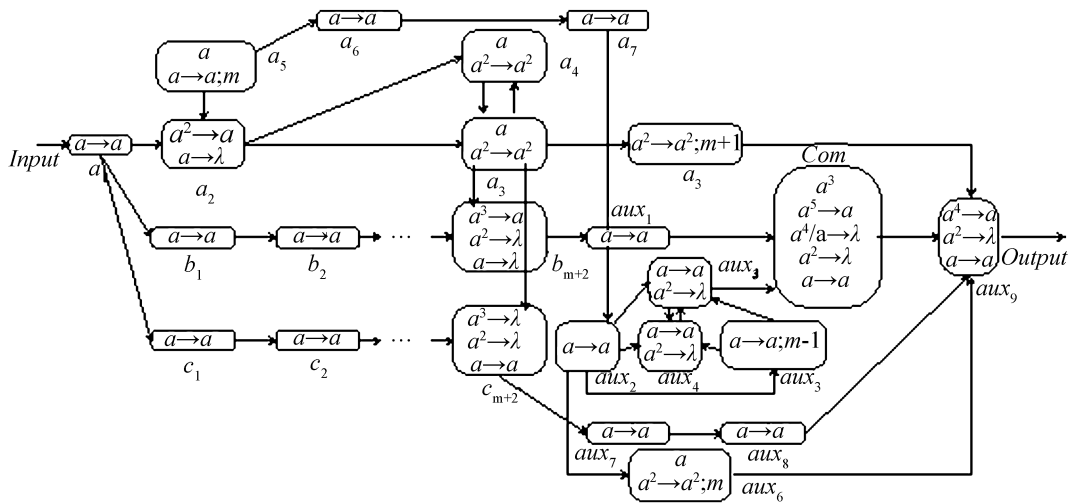


图 7 求任意符号整数补码的脉冲神经网络系统

当发送到输入神经元 σ_{a_1} 的二进制原码是负整数时,由定理 2 可知,从 $t=m+3$ 开始,神经元 $\sigma_{b_{m+2}}$ 开始向神经元 σ_{aux_1} 发送脉冲,而神经元 $\sigma_{c_{m+2}}$ 在任何时刻都不会向神经元 σ_{aux_7} 发送脉冲。由定理可知,从 $t=m+5$ 开始,神经元 σ_{Com} 开始向 σ_{aux_9} 发送脉冲。因此,从 $t=m+6$ 开始,输出神经元 σ_{aux_9} 接收到神经元 σ_{Com} 发送过来的补码脉冲。从图 7 中可以看出,从 $t=m+7$ 到 $t=2m+5$,输出神经元 σ_{aux_9} 向环境中

发送了数值位的补码脉冲。当 $t=2m+5$,输出神经元 σ_{aux_9} 从神经元 σ_{a_8} 接收到两个脉冲,从神经元 σ_{aux_6} 接收到两个脉冲,故在 $t=2m+5$ 时通过激发规则 $a^4 \rightarrow a$ 向环境发送一个脉冲。因此,在 $t=2m+6$ 步时,外界环境从输出 σ_{aux_9} 接收到一个脉冲。由此可见,该系统可以将负整数的原码转化为负整数的补码。

反之,如果发送输入神经元的是非负整数的补

码,则由定理 3 可知,神经元 $\sigma_{b_{m+2}}$ 被屏蔽,再由定理 1 可知,求负整数的所有神经元的工作都没有启动。因此,输出神经向外部环境发送的脉冲实质上就是对二进制原码的复制。

在求-1 补码过程中,系统中各神经元在不同时刻所包含的脉冲数如表 4 所示。

表 4 系统在求-1 补码时各个神经元包含的脉冲数

Step	0	1	2	3	4	5	6	7	8	9	10	11	12
input	1	0	1	0	0	0	0	0	0	0	0	0	0
a_1	0	1	0	1	0	0	0	0	0	0	0	0	0
a_2	0	0	1	0	2	0	0	0	0	0	0	0	0
a_3	1	1	1	1	1	2	2	2	2	2	2	2	2
a_4	1	1	1	1	1	2	2	2	2	2	2	2	2
a_5	1	1	1	1	0	0	0	0	0	0	0	0	0
a_6	0	0	0	0	1	0	0	0	0	0	0	0	0
a_7	0	0	0	0	0	1	0	0	0	0	0	0	0
a_8	0	0	0	0	0	0	2	2	2	2	2	2	2
b_1	0	0	1	0	1	0	0	0	0	0	0	0	0
b_2	0	0	0	1	0	1	0	0	0	0	0	0	0
b_3	0	0	0	0	1	0	1	0	0	0	0	0	0
b_4	0	0	0	0	0	1	0	1	0	0	0	0	0
b_5	0	0	0	0	0	0	3	2	3	2	2	2	2
c_1	0	0	1	0	1	0	0	0	0	0	0	0	0
c_2	0	0	0	1	0	1	0	0	0	0	0	0	0
c_3	0	0	0	0	1	0	1	0	0	0	0	0	0
c_4	0	0	0	0	0	1	0	1	0	0	0	0	0
c_5	0	0	0	0	0	0	3	2	3	2	2	2	2
aux ₁	0	0	0	0	0	0	0	1	0	1	0	0	0
aux ₂	0	0	0	0	0	0	1	0	0	0	0	0	0
aux ₃	0	0	0	0	0	0	0	1	1	1	2	0	0
aux ₄	0	0	0	0	0	0	0	1	1	1	2	0	0
aux ₅	0	0	0	0	0	0	0	1	1	1	0	0	0
aux ₆	1	1	1	1	1	1	1	2	2	2	2	0	0
aux ₇	0	0	0	0	0	0	0	0	0	0	0	0	0
aux ₈	0	0	0	0	0	0	0	0	0	0	0	0	0
aux ₉	0	0	0	0	0	0	0	0	0	1	1	4	0
com	3	3	3	3	3	3	3	3	5	1	2	0	0
output	0	0	0	0	0	0	0	0	0	0	1	1	1

5 结束语

作者设计了基于脉冲神经膜系统的有符号整数补码器,该补码器使有符号加法器的设计成为可能。为了准确地搭建该系统,需要先构造求负整数补码

的脉冲神经膜系统。由于非负整数的补码是其自身,故需要构建将负整数和非负整数分离的脉冲神经膜系统。最后,构造出求任意有符号整数补码的脉冲神经膜系统。在接下来的工作中,将是设计一个有符号加器的脉冲神经膜系统。

参考文献:

- [1] Gutiérrez-Naranjo M A, Leporati A. First steps towards a CPU made of Spiking Neural P systems[J]. *Int. J. of Computers, Communications and Control*. 2009, 4 (3), 244-252.
- [2] Zhang X Y, Zeng X X, Pan L Q. A spiking neural P system for performing multiplication of two arbitrary natural numbers[J]. *Chinese Journal of Computer*. 2009, 32(12): 2362-2372.
- [3] Song T, Pan L. Asynchronous spiking neural P systems with local synchronization[J]. *Inf Sci*, 2013, 219(1):197-207.
- [4] Wu B, Pan L, Pérez-Jiménez M J. Tissue P systems with protein on cells[J]. *Fundam Inform*, 2016, 144(1):77-107.
- [5] Wu T F, Zhang Z Q, Pan L Q. Cell-like spiking neural P systems[J]. *Theor Comput Sci*, 2016(623):180-189.
- [6] Pan L, Wang J, Hoogeboom H J. Asynchronous extended spiking neural P systems with astrocytes[C]. *International Conference on Membrane Computing*, berlin Heidelberg: soringer-verlag, 2012:243-256.
- [7] Zhang X, Zhang X, Song T, et al. Spiking neural P systems with thresholds[J]. *Neural Comput*. 2014, 26(7):1340-1361.
- [8] Zhang G, Rong H, Neri F, et al. An optimization spiking neural P system for approximately solving combinatorial optimization problems[J]. *Int J Neural Syst*, 2014, 24 (5):1-16.
- [9] Pan L Q, Zeng X, Zhang X. Time-free spiking neural P systems[J]. *Neural Comput*, 2011, 23(5):1320-1342.
- [10] Ramanujan A, Krithivasan K. Control words of transition P systems[J]. *Advances in Intelligent Systems and Computing*, 2013 (201):145-155.
- [11] Verlan S, Quiros J. Fast hardware implementation of P systems[C]. *International Conference on Membrane Computing*, berlin Heidelberg: soringer-verlag, 2013:404-423.
- [12] Pan L Q, Perez-Jimenez M J. Spiking neural P systems with neuron division and budding[J]. *Science China Information Sciences*, 2011, 54(8):1596-1607.
- [13] 王防修. 基于脉冲神经膜系统的有符号并行乘法器设计[J]. *武汉轻工大学学报*, 2019, 38 (2):44-49.