Космічна наука і технологія. 1998. Т. 4. № 4. С. 49–54.

УДК 681.32:528.533

К проблеме использования клеточных автоматов в качестве космических бортовых вычислительных устройств

В. А. Вальковский, Д. Д. Зербино

Науково-дослідний інститут інформаційної інфраструктури Національної академії наук України, Львів

Надійшла до редакції 13.04.98

Обговорюється можливість застосування в космічних бортових системах обробки інформації та управління клітинних автоматів спеціалізованої архітектури. Особливості пристроїв, які використовують клітинні автомати — це регулярність структури, висока надійність, можливість нарощування, взаємозамінність універсальних модулів та нагромадження, масовий паралелізм в обробці інформації і висока швидкодія. Запропоновані системи володіють підвищеною надійністю, яка виражається в можливості їх функціонування в умовах відмови деяких комірок. Наведені приклади арифметичних клітинних автоматів для реалізації алгоритмів над комплексними числами. Розглядаються питання незалежності результатів обчислень від порядку застосування правил автомата.

ВВЕДЕНИЕ

При рассмотрении проблемы оснащения космических аппаратов бортовыми компьютерами с необходимостью приходим к следующим требованиям для них. Во-первых, они должны быть компактны по объему и иметь небольшой вес. Во-вторых, быть надежными относительно сбоев, особенно тех, которые присущи космической среде. В-третьих, быть «адекватными решаемым задачам». Последнее выражается в их узкой специализации как по архитектуре, так и по типам обрабатываемых данных. Например, при решении двумерных задач, представляющих большую часть всей нагрузки на бортовые компьютеры, необходимо быстро оперировать с комплексными числами.

Общепризнанно, что среди наиболее подходящих типов вычислителей, удовлетворяющих этим требованиям, важное место занимают клеточные автоматы [3]. Клеточные автоматы на плоскости представляют собой однородную ячеечную среду, преобразование информации в которой ведется по некоторым локальным правилам. Правила применяются параллельно и независимо. Таким образом, клеточный автомат осуществляет асинхронную распределенную обработку информации.

Клеточные автоматы подходят для решения как «внешних», так и «внутренних» задач. Под первыми подразумеваются те задачи, ради решения которых был запущен космический аппарат. Типичный пример — обработка космических снимков в реальном масштабе времени. «Внутренние» задачи — это те задачи, которые решаются в целях жизнеобеспечения и оптимизации использования оборудования самого космического аппарата. Среди них можно рассмотреть усовершенствование подсистемы ориентации солнечных батарей корабля на солнце. Очевидно, что как одна из важнейших подсистем она должна быть абсолютно надежной, а также компактной и не зависеть от маневров, выполняемых самим кораблем. Кроме того, в ее функции должны входить не только ориентация плоскостей батарей, но и коммутация отдельных их фрагментов в общую цепь с целью максимальной энергоотдачи, диагностика повреждений, прогноз старения и загрязненности элементов, наращиваемость. Клеточный автомат легко допускает расширение и замену фрагментов, а также остается работоспособным при отказе некоторых из его частей. Это структура, которая может быть «распределена» по всей плоскости солнечной батареи, а на борт корабля, параллельно с подачей электроэнергии, передавать сигналы для управления ориентацией батареи и данные в бортовой компьютер о состоянии каждого ее элемента.

Таким образом, клеточный автомат как специализированное вычислительное устройство удовлетворяет всем условиям, накладываемым на бортовой компьютер.

Однако, помимо специфических операций, обработка информации на борту космического устройства требует выполнения определенного количества традиционных арифметических действий. Это, например, нахождение взвешенной суммы при решении задачи цифровой фильтрации, которая является неотъемлемой частью решения проблемы анализа изображений, или расчет направления поворота антенны. Таким образом, возникает проблема выбора: *либо* «порции» необходимых арифметических вычислений отправлять на Землю, *либо* оснащать космическое устройство дополнительным спецпроцессором для выполнения в реальном времени необходимых арифметических операций, *либо* попытаться приспособить для их выполнения имеющийся на борту специализированный процессор, в нашем случае — клеточный автомат.

В настоящей работе предлагается подход, ориентированный на последнюю из перечисленных возможностей [2]. Более точно, в рамках достаточно традиционной модели клеточного автомата предлагается некоторая система правил и «стратегия» их применения, позволяющая реализовать арифметические операции средствами того же самого специализированного устройства, которое предназначено для решения типовых «внешних» и «внутренних» задач. При этом нами показана корректность использования этих правил, возможность их применения в поле комплексных чисел, а также надежность использования клеточных автоматов в условиях выхода из строя некоторых ячеек.

Предлагаемая модификация формализма клеточных автоматов, характеризуется включением в нее ряда простых правил глобального характера. Это приближает ее к понятию ассоциативной памяти, что позволяет унифицированными средствами реализовать более широкий класс вычислительных операций. При одном и том же наборе правил и типе кодирования, меняя области ввода и вывода информации, а также осуществляя небольшое число подготовительных и глобальных управляющих действий, удается добиться реализации принципиально различных арифметических операций.

В отличие, например, от [1], наш подход ориентирован на применение в космической среде. Мы используем несколько иные асинхронные правила, не зависящие от контекста. Особое внимание уделяется инвариантности получаемых результатов от хода поступления входной информации. Кроме того, полученные результаты не зависят от разрядности, а при вычислениях дается возможность сохранять все значащие цифры результата.

БАЗОВЫЙ ФОРМАЛИЗМ

В работе используется традиционное понятие однослойного клеточного автомата на плоскости с измерениями i и j. Пара (i, j) обозначает клетку с координатами i, j. Клетка может содержать значение 0 или 1. Запись $([k : l], [m : n])$ идентифицирует прямоугольник в вычислительном поле. Некоторые из значений k, l, m, n могут быть $-\infty$ или $+\infty$, т. е. минус бесконечность и бесконечность. Если $k = l$ или $m = n$, то пишется одно из этих значений, а двоеточие и квадратные скобки опускаются. Таким образом, $([-\infty : 1], 2)$ обозначает бесконечную влево линию ячеек, идущую на уровне 2 от ячейки $(1, 2)$. В целях повышения надежности все вычисления четырехкратно дублируются: выполняются в каждом из четырех квадрантов плоскости.

С помощью приведенных выше соглашений локализуются области ввода операндов ARG1 и ARG2 той или иной операции, область вывода результата RESULT, а также области действия вспомогательных и управляющих операций.

Нами используются два бесконтекстных правила обработки в вычислительном поле автомата (рис. 1).

Правила будем называть перекрывающимися в ячейке (i, j), когда (i, j) можно изменить одновременно по нескольким из них. Если какие-либо правила перекрываются, то в каждый момент времени можно выполнить только одно из них. Если правила не перекрываются, то их можно выполнять одновременно.

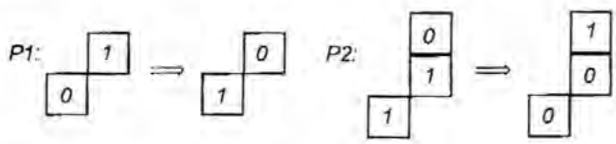

Рис. 1. Система правил для классической двоичной системы кодирования

Подготовительные и управляющие операции разделяются на следующие классы:
- акты настройки. Каждый такой акт применяется перед осуществлением той или иной операции и не зависит от операндов;
- настройка после ввода. Соответствующие действия применяются после того, как в области ввода аргументов внесены исходные данные. Они могут осуществляться одновременно с применением основных обрабатывающих правил $P1$, $P2$;
- предписывающие и ограничивающие правила. Последние либо предписывают строгий порядок выполнения $P1$, $P2$ с целью получить необходимый результат, либо каким-то образом ограничивают свободу их применения.

Действия этих правил для каждой конкретной операции будут пояснены дополнительно.

РЕАЛИЗАЦИЯ ЭЛЕМЕНТАРНЫХ ОПЕРАЦИЙ БОРТОВОГО СПЕЦПРОЦЕССОРА

Коснемся вначале проблемы реализации простых операций, процент которых при вычислениях в космических условиях весьма высок.

К простым операциям мы относим побитовые операции над словами логического характера, а также сложение и вычитание. Объясним работу автомата на примере логического отрицания.

а) Подготовительное действие — во все ячейки бесконечного влево слова $([-\infty : 1], 2)$ засылаются единицы:

$$(\ldots 1 \ldots 1) \to ([-\infty : 1], 2).$$

б) Единственный операнд вводится в область $([-\infty : 2], 1)$.

в) Правила $P1$, $P2$ применяются в произвольном порядке, но только на линии $([-\infty : 1], 2)$, для $P1$ на этой линии должна находиться единица, для $P2$ — средняя единица.

г) Область вывода результата — $([-\infty : 2], 1)$. Приведенный на рис. 2 пример иллюстрирует работу автомата на слове (1 0 0 1 0 1).

Выполнение других простых операций представим таблицей. В ней атрибут ARG указывает на области занесения исходных данных. Атрибут RESULT указывает на местоположение результата операции. Атрибут «Правила» указывает на разрешенные во время выполнения операции асинхронные правила. Операция считается законченной, если разрешенные правила не применимы. Запись $\{P1, P2\}$ означает асинхронное применение указанных правил до остановки автомата.

0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	0	1	0	1	0

$\Longrightarrow$

0	0	0	0	1	0	0	1	0	1
1	0	0	0	0	0	0	0	0	0
1	1	1	0	1	1	0	1	0	0

Рис. 2. Пример реализации операции инвертирования строки

Операция	ARG1	ARG2	RESULT	Правила
(A1)and(A2)	$([-\infty : 1],2)$	$([-\infty : 2],1)$	$([-\infty : 1], 3)$	$\{P2\}$
(A1)or(A2)	$([-\infty : 1],2)$	$([-\infty : 2],1)$	$([-\infty : 2], 1)$	$\{P1\}$
not(A2)	$([-\infty : 1],2)$	$(1,\ldots 1) \to ([-\infty : 2],1)$	$([-\infty : 2], 1)$	$\{P2\}$
A1 + A2	$([-\infty : 1],2)$	$([-\infty : 2],1)$	$([-\infty : 2], 1)$	$\{P1, P2\}$
Сдвиг вверх ARGUM 1	$([-\infty : 1],2)$	$(1,\ldots 1) \to ([-\infty : 2],1)$	$([-\infty : 1], 3)$	$\{P2\}$
Сдвиг вверх и вправо ARGUM 1	$([-\infty : 2],1)$	$(1,\ldots 1) \to ([-\infty : 1],2)$	$([-\infty : 1], 2)$	$\{P1\}$

Изменение знака (перевод числа в дополнительный код) требует следующих действий:

$$(\mathrm{ARG1}) \to ([-\infty : 1], 2);$$
$$(1,\ldots,1) \to ([-\infty : 2], 1); \{P2\};$$
$$(0,\ldots 0,0,\ldots 0) \to ([-\infty : 1], [2 : 3]);$$
$$1 \to (1, 2); \{P1, P2\};$$
$$([-\infty : 2], 1) \to \mathrm{RESULT}.$$

Вычитание выполняется аналогично сложению, за исключением того, что вычитаемое представляется в дополнительном коде:

$$(-\mathrm{ARG1}) \to ([-\infty : 2], 1);$$
$$(\mathrm{ARG2}) \to ([-\infty : 1], 2); \{P1, P2\};$$
$$([-\infty : 2], 1) \to \mathrm{RESULT}.$$

Умножение требует такой операции, как взятие декартова произведения D двоичных данных, расположенных на осях:

$$(\mathrm{ARG1}) \to (0, [1 : \infty]);$$
$$(\mathrm{ARG2}) \to ([-\infty : 1], 0);$$
$$\{D(([-\infty : 1], 0), (0, [1 : \infty])); P1, P2\};$$
$$([-\infty : 1], 1) \to \mathrm{RESULT}.$$

Деление требует асинхронного применения операции условного декартова произведения Dc с продвижением единиц в левый квадрант, и преобразований $\{P1, P2\}$ до полной остановки процесса:

$$(\mathrm{ARG1}) \to (0, [1 : \infty]);$$
$$(-\mathrm{ARG2}) \to ([1 : \infty], 0);$$
$$\{Dc((x, 0), (0, y), (x, y) = 0, x = [1{:}\infty]), P1, P2\};$$
$$([-\infty, 1], 1) \to \mathrm{RESULT}.$$

В приведенном примере деление реализуется методом разложения делимого на элементарные слагаемые, например: 1101 = 1000 + 100 + 1. Это единственная операция, которая нарушает четырехкратное дублирование всех процессов.

Процесс деления может быть остановлен при достижении необходимой точности или при превышении такта реального времени. Результат деления состоит из целой части, дробной и остаточной информации, характеризующей остановку процесса деления на каком-либо шаге.

В зависимости от вида выражения, которое следует вычислить, необходимо размещать на двумерном клеточном пространстве автомата области рабочих регистров и входных аргументов, а также область результатов вычислений. Для реализации разветвляющихся и циклических процессов необходимо использовать специальные управляющие шаблоны данных, которые активизируются при каждой операции освобождения поля автомата.

Технология «мелкозернистых» вычислений, которую нужно шире внедрять в бортовых спецпроцессорах, позволяет быстро и надежно обрабатывать информационные потоки данных, приходящие от различных датчиков состояния корабля, а также вести мониторинг на различных уровнях иерархии.

ОБОСНОВАНИЕ КОРРЕКТНОСТИ ВЫПОЛНЕНИЯ СЛОЖНЫХ ОПЕРАЦИЙ БОРТОВЫМ СПЕЦПРОЦЕССОРОМ

Поскольку клеточные автоматы одновременно играют роль памяти и процессора, то при расширении системы управления и введении ее новых звеньев не потребуется вносить изменения в устоявшуюся систему обработки информации. Ее необходимо будет обычным для клеточных автоматов образом «нарастить» из таких же блоков, привязав соответствующие шаблоны ввода-вывода к существующим данным.

В данном разделе остановимся на одном важном результате, который служит предпосылкой корректности выполнения сложных операций. Как обычно, автомат называется однозначным, если любой путь применения правил (см. рис. 1) из одного и того же состояния приводит к одному и тому же заключительному состоянию.

Если S — некоторое состояние автомата, то через $|S|$ обозначим число единиц в нем. Слово $(|-\infty : 0|, 0)$ в этом разделе назовем регистром. Будем считать, что начиная с него правила $P1$, $P2$ неприменимы. Заключительное состояние, как обычно, это то состояние, в котором правила при заданных ограничениях на выполнение неприменимы. Довольно просто показать, что для любого конечного состояния любой процесс применения правил, представленных на рис. 1, оканчивается, при этом для любого промежуточного и заключительного состояния S' выполняется $|S'| \leq |S|$.

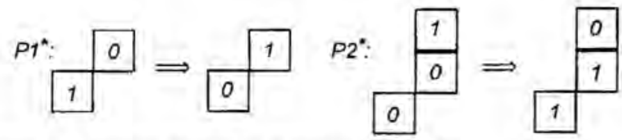

Рис. 3. Система обратных правил для $P1$, $P2$

Если P — некоторое правило, то P^* обозначает инверсное к нему правило, в котором левая и правая части меняются (рис. 3).

Пусть задан автомат с системой правил (см. рис. 1) и некоторое конечное состояние S. Пусть $N = \{0, 1, \dots, n, \dots\}$ — натуральный ряд, расширенный нулем. Характеристической функцией состояния S, обозначаемой $f(S, i)$ называется натуральное число, выражающее количество единиц на диагонали, идущей от точки на горизонтальной оси с координатой i вправо-вверх до вертикальной оси. Состояния $S1$ и $S2$ назовем сильно эквивалентными, если их характеристические функции одинаковы: $\forall i\ (f(S1, i) = f(S2, i))$. В дальнейшем эту запись будем обозначать как $f(S1) = f(S2)$. Ясно, что состояния $S1$ и $S2$ сильно эквивалентны, если они получаются одно из другого применением правил $P1$, $P1^*$.

Пусть $f : i \to N$ — произвольная целочисленная функция. Будем говорить, что функция $f' : i \to N$ получается из f применением акта элементарной редукции или, сокращенно, e-редукцией (обозначение $f' = r(f)$), если f' отличается от f лишь в двух значениях: i и $i + 1$, где $f(i + 1) \geq 2$ и $f'(i) = f(i) + 1$, $f'(i + 1) = f(i + 1) - 2$.

Неформально говоря, акт e-редукции смещает две единицы значения f в произвольной точке i на шаг влево, превращая их в одну единицу. Ясно, что математическое определение редукции отражает правило $P2$.

Функция f' называется нередуцируемой, если к ней неприменим акт e-редукции, что возможно только в том случае, когда для любого аргумента i значение f' не превышает 1. Функция f редуцируема к функции f', если существует цепочка элементарных редукций $r_1, r_2, \dots, r_m$, такая, что $f' = r_m(r_{m-1}(\dots r_1(f)\dots))$. Обозначим этот факт $f' = R(f)$.

Довольно просто показать, что если $R1$ и $R2$ — две цепочки элементарных редукций, приводящие f к нередуцируемым функциям, то $R1(f) = R2(f)$. Иначе говоря, каким бы путем ни проходило редуцирование функции, результат будет один и тот же.

Кроме того, если $S1$ — некоторое состояние и $S2$ получается из $S1$ путем однократного применения правила $P2$, то $f(S2) = r(f(S1))$. А если для некоторых состояний $S1$ и $S2$ имеет место $f(S2) = r(f(S1))$, то $S2$ может быть получено из $S1$ путем применения правил $P1$, $P1^*$ и однократного применения правила $P2$.

Очевидно, что используя систему на рис. 1 конечный результат вычислений автомата можно разместить в одной строке, поскольку нередуцируемая функция $f'(i)$ может принимать значения только 0 или 1.

Из приведенного выше видно, что любой процесс применения правил $P1$, $P2$, начинающийся в состоянии S, приводит в заключительное состояние. Применение правила $P1$ не изменяет характеристическую функцию, а применение $P2$ сопровождается актом ее e-редукции. Любой процесс обрывается в состоянии S', для которого $f(S')$, фактически совпадающая с S', — нередуцируема. Наконец, любой путь редуцирования характеристической функции состояния автомата приводит к единственной нередуцируемой функции. Все сказанное позволяет сделать заключение о корректности использования правил (см. рис. 1), т. е. рассматриваемый автомат является однозначным.

ВЫЧИСЛЕНИЯ НА АВТОМАТАХ С ДЕФЕКТАМИ И ОТКАЗАМИ В УСЛОВИЯХ КОСМИЧЕСКОЙ СРЕДЫ

Возвращаясь к вопросу о надежности вычислений в условиях космической среды и возможности отказа некоторых ячеек, упомянем о модели клеточного автомата с дефектами. В целях применения в космосе нами разработан формализм для организации вычислений в условиях, когда некоторые ячейки рабочего пространства автомата не могут правильно выполнять своих функций. В предлагаемой модели их необходимо отмечать специальным символом, запрещающим выполнение какого-либо локального правила, а также ввод-вывод или сохранение данных. При этом в систему локальных правил (см. рис. 1) добавилось еще одно, позволяющее единицам «перепрыгивать» через соседние ячейки, в том числе и дефектные, что позволяет быстрее получить результат и одновременно решить проблему одиночных дефектов рабочего пространства автомата.

Для представления чисел в модели с дефектами использован двоичный негабинарный код с весами разрядов $(-2)^n$. Перед началом вычислений необходимо зафиксировать координату нулевого разряда. Если i — вертикальная координата, j — горизонтальная, и (i_0, j_0) — координата нулевого разряда, то единица в ячейке (i, j) означает число $(-2)^{(i-i0)+(j-j0)}$. Таким образом, можно вводить числа на клеточную плоскость либо по вертикали, либо по горизонтали, придерживаясь указанной разрядной сетки. Пример такой записи: $1.5 = (110.1) = 4 + (-2) + (-0.5)$.

В том случае, если один из битов вводимого числа попадает на дефектную ячейку, то ввод нуля никак не отражается на изменении состояния автомата, а вводимую единицу необходимо перенести в первую свободную по диагонали от дефекта ячейку.

Выполнение локальных правил перемещения единиц можно совмещать с вводом данных в поле автомата. При этом в строке, к которой «движутся единицы» будет накапливаться результат операции. Очевидно, что при данной системе кодирования правила не зависят от знака чисел.

Организация взаимодействия вертикальных и горизонтальных регистров в условиях наличия дефектов потребовала дополнительной операции, которая при отсутствии возможности ввода данных в ячейку (дефект либо наложение) может осуществлять ввод единицы в соседнюю по диагонали ячейку. При наличии дефектных ячеек в области регистров средствами введенной модели удается перенести отдельные регистровые ячейки на соседние по диагонали, не изменяя «формы» вычисляемого выражения. Расширенная система правил неизбыточна и однозначна, т. е. обеспечивает получение одного и того же результата независимо от последовательности применения этих правил.

ВЫЧИСЛЕНИЯ НАД КОМПЛЕКСНЫМИ ЧИСЛАМИ

Для «адаптации» концепции клеточного вычисления к комплексным числам, не вводя новых значений, кроме {0, 1}, необходимо выяснить, что представляет собой «информационная единица». Ответ на этот вопрос может дать система кодирования. Нами была использована упомянутая выше негабинарная система кодирования, которая позволила компактно представить комплексные числа и создать универсальную знаконезависимую систему правил локальных преобразований, реализующих вычисления в комплексном поле. Таким образом, та же самая техника, которая была развита для преобразований действительных чисел, вполне применима для реализации арифметических операций над полем комплексных чисел. Причем речь идет не об «искусственном» декартовом произведении действительной и мнимой составляющих, а определенном механизме их «упаковки» и «распаковки». Опишем это более детально.

Если каждой ячейке с координатами (i, j) придать вес $(-2)^{(i+j)/2}$, то если сумма координат ячейки (x, y) четна, то ячейка имеет действительный вес, если нечетна — то мнимый. Если сумма координат делится на 4, то вес ячейки положителен, если нет — то отрицателен.

Если обозначить $c = \sqrt{-2}$, то любое комплексное число, независимо от знака действительной и мнимой части можно однозначно представить двоичным кодом с фиксированной точкой в следующей разрядной сетке:

$$\{..., 4, -2c, -2, c, 1, -0.5c, -0.5, 0.25c, ...\}.$$

Интересным оказалось то, что как и в предыдущем случае, для комплексных чисел существует система асинхронных знаконезависимых правил для клеточного параллельного вычисления арифметических выражений. Основные из них приведены на рис. 4.

Получена также обобщенная система правил локальных преобразований клеточного автомата для универсальных негабинарных систем $(-2)^{n/r}$. Это служит предпосылкой компактного представления любых составных значений, что особенно важно в условиях космоса. Однако длительность процесса деления чисел, при отсутствии приоритетов применения правил, на порядок превышает длительность процесса умножения. Эта разница увеличивается с увеличением r.

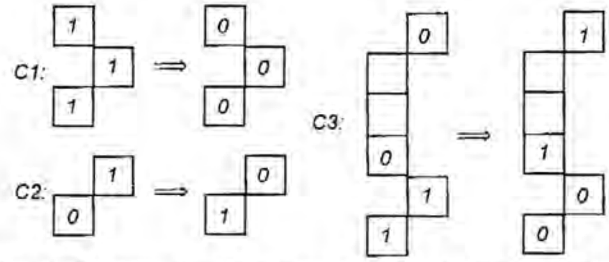

Рис. 4. Система вычислительных правил для комплексных чисел

И наконец, спецпроцессоры, созданные на базе клеточных автоматов, сохраняют свою работоспособность при некоторых дефектах рабочего поля и минимальной избыточности, что также немаловажно в условиях космического окружения. Все предложенные в работе методы тщательно проверены на разработанной нами системе моделирования клеточных автоматов.

ВЫВОДЫ

Анализируя все сказанное, можно заключить, что клеточные автоматы являются удобным средством для организации надежных параллельных вычислений в условиях космоса. Кроме того, они просты в технической реализации и на их основе можно создавать программируемые структуры для выполнения трудоемких расчетов в режиме реального времени. Бортовые системы управления космическими аппаратами, создаваемые на их основе, можно легко структурно и программно наращивать. Эти системы могут быть легко программно перестраиваемыми и простыми в эксплуатации. В них можно быстро найти и устранить неисправность, что немаловажно в экстремальных условиях.

1. Ачасова С. М., Бандман О. Л. Корректность параллельных вычислительных процессов. — Новосибирск: Наука, 1990. — 252 с.
2. Вальковский В. А., Зербино Д. Д. Реализация арифметических вычислений в знакопеременных кодах на клеточных автоматах // Пробл. упр. и автоматики. — 1997. — № 2. — С. 49—64.
3. Codd E. F. Cellular Automata: — New York: Acad. Press, 1968. — 198 p.

ON THE USE OF CELLULAR AUTOMATA AS SPACE-BORNE CALCULATION SYSTEMS

V. A. Val'kovskii and D. D. Zerbino

The possibility of using cellular automata of special structure in the space-borne systems of information processing and control is discussed. Such automata are distinctive in that they have a regular structure, are highly reliable, provide the interchangeability of their modules, extensibility, massive parallelism of information processing, and high productivity. Their high reliability allows them to operate in the case of failure of some cells. Some examples of arithmetic cellular automata designed for the realization of algorithms on complex numbers are given. We discuss the independence of computation results from the order of application of automation rules.