ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 143 за 2019 г.

ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Малинецкий Г.Г., Щадинский Д.М.

Конструирование
вычислительного устройства
на основе игры «Жизнь»

Рекомендуемая форма библиографической ссылки: Малинецкий Г.Г., Щадинский Д.М. Конструирование вычислительного устройства на основе игры «Жизнь» // Препринты ИПМ им. М.В.Келдыша. 2019. № 143. 14 с. <http://doi.org/10.20948/prepr-2019-143>
URL: <http://library.keldysh.ru/preprint.asp?id=2019-143>

О р д е н а Л е н и н а
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Р о с с и й с к о й а к а д е м и и н а у к

Г.Г. Малинецкий, Д.М. Щадинский

**Конструирование вычислительного
устройства на основе игры «Жизнь»**

Москва — 2019

Малинецкий Г.Г., Щадинский Д.М.

Конструирование вычислительного устройства на основе игры «Жизнь»

В работе исследуется проблема организации вычислений на клеточном автомате Джона Конвея «Жизнь». Получены конфигурации клеток, реализующие логические операции, необходимые для суммирования чисел в двоичной системе счисления. Предложена начальная конфигурация состояний клеток для полного одноразрядного сумматора, а также конфигурация для приема результирующего сигнала. Описана структура организации вычисления суммы двух многоразрядных двоичных чисел. В случае двухразрядных входных данных представлена конкретная конфигурация начального состояния клеток.

Ключевые слова: клеточный автомат, игра «Жизнь», полный одноразрядный сумматор.

Georgy Gennadiyevich Malinetskiy, Dmitry Mikhailovich Shchadinskiy

Design of the computing device based on the game of “Life”

The present paper deals with problem of computing on John Conway cellular automata "Life". Cells configurations which realizing required for adding two-based numbers logical elements are obtained. Initial cell state configuration for full binary adder and configuration for receiving signal are presented. Structure of computing two two-based multibit numbers sum is described. In case of two-bit input data initial cell state configuration is presented.

Key words: cellular automata, the game of “Life”, full adder.

Работа выполнена при поддержке РФФИ (проекты 19-01-00602-а, 18-511-00008-Бел_а).

Оглавление

Введение	3
Основные структурные элементы	4
Логические элементы.....	7
Полный одноразрядный сумматор	8
Полный многоразрядный сумматор	11
Заключение.....	14

Введение

Проблема конструирования вычислительных устройств всегда была крайне актуальной для человечества. С появлением первых арифмометров Вильгельма Шиккарда 1623 г., Блеза Паскаля 1642 г. стало ясно, что нужны принципы, позволяющие создавать более эффективные и совершенные машины. Например, Лейбниц впервые обнаружил достоинства двоичной системы счисления, использование которой в вычислительной технике сегодня является основным принципом.

В XX веке происходило бурное развитие компьютеров. Сначала машины с логическими цепями, основанными на электронных лампах, а потом и на полупроводниковых элементах, позволили решать огромное число задач. Но и современные вычислительные устройства имеют технологический предел, связанный с проблемой соотношения плотности элементов и рассеиваемой мощностью [1].

Одним из перспективных решений этой проблемы в будущем может быть использование клеточных автоматов, которые представляют собой конечную структуру, состоящую из решетки ячеек, каждая из которых может принимать конечное число состояний. При этом клетки некоторым образом (программно или физически) влияют друг на друга, меняя состояния клеточного автомата, по неизменным правилам. Одним из главных достоинств такого подхода для организации вычислений является высокая возможность параллелизма [2]. Стоит отметить, что использование клеточных автоматов может быть достойной альтернативой квантовым компьютерам.

Наиболее известным примером клеточного автомата является игра «Жизнь» Джона Конвея [3]. В этой игре клетки могут находиться в двух состояниях — «живом» (включенном), тогда клетка закрашена, и «мертвом» (выключенном), в этом случае клетка не закрашена. Распределение клеток в начале игры называется первым поколением, каждое следующее поколение рассчитывается по правилам:

- в мертвой клетке, рядом с которой ровно три живые клетки, зарождается «жизнь»;
- если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить;
- если соседей больше двух или трех, клетка умирает («от одиночества» или «от перенаселения»).

Под соседями клетки подразумеваются все восемь контактирующих с ней через ребра или вершины клеток. Стоит отметить, что мы имеем право задавать исключительно начальную конфигурацию, все последующее развитие происходит строго по указанным правилам.

В 1982 г. Джон Конвей доказал, что игра «Жизнь» эквивалентна универсальной вычислительной машине, показав комбинации клеток реализующие логические операции И, ИЛИ, НЕ [3]. С этого времени данный

клеточный автомат привлекает к себе внимание многих энтузиастов. В книгах [4], [5] представлено разнообразие структур, реализующих работу различных элементов вычислительных устройств. Заметим, что для построения этих структур предложенных операций недостаточно, требуются не только аналоги «проводов» для передачи сигнала, но и возможности для пересечения этих «проводов», множители сигналов и некоторые другие возможности.

В настоящей работе предлагается вариант организации вычисления суммы двух чисел на основе клеточного автомата «Жизнь». Рассматривается ряд структурных элементов, позволяющих кодировать сигнал, передавать его, менять направление распространения сигнала, а также позволяющих создавать несколько его копий. Предлагается структура для принятия результирующего сигнала. Описываются относительно компактные конфигурации, реализующие необходимые логические элементы. И, наконец, предлагается конфигурация полного одноразрядного сумматора, а также схема вычисления суммы чисел с произвольным количеством разрядов. Целью настоящей работы является демонстрация принципиальной возможности использования клеточных автоматов в качестве вычислительного устройства.

Основные структурные элементы

Для начала определим, как в нашем компьютере будет распространяться сигнал. Рассмотрим конфигурацию «планер» (англ. *glider*). Она способна двигаться по игровому пространству в диагональных направлениях, повторяя себя каждые 4 поколения, см. рис. 1. Здесь и далее числами над рисунками указан номер поколения. Наличие планера в заданном месте игрового пространства будет кодировать единицу, отсутствие — ноль. Стоит отметить важное свойство планеров: они способны взаимно уничтожаться при столкновениях. В книге [3] представлены возможные случаи подобного взаимодействия. Это свойство крайне важно для построения логических элементов нашего компьютера.

Рис. 1. Конфигурация планер

Следующее, что нам нужно — ликвидировать лишние одиночные планеры. Для этого нам понадобятся способные поглощать их устойчивые комбинации, две из которых представлены на рис. 2. Такие структуры называются поглотителями (англ. *eaters*). Представленные же на рис. 2 комбинации будем называть соответственно поглотитель 1 и поглотитель 2.

Рис. 2. Конфигурации, способные поглощать планеры

Поглотитель 2 можно модифицировать, добавив контрольный квадрат сверху, для того чтобы определять, произошло ли поглощение или нет (см. рис. 3). Такая конфигурация, будем называть ее получателем (англ. *getter*), нужна для того, чтобы иметь возможность принимать сигнал. Если квадрат уничтожился, значит, получатель принял 1, иначе — 0.

Рис. 3. Конфигурация «получатель»

При передаче сигнала по проводам крайне легко менять его направление, а также существует очевидная возможность дублирования сигнала. В нашем случае нужны инструменты, позволяющие сделать подобное. Для смены направления распространения планеров существует целый ряд конфигураций, которые называются отражатели на 90° (англ. *90 degree reflectors*) [6]. В нашем вычислительном устройстве мы будем использовать два отражателя. Первый — под названием ковбой (англ. *buckaroo*) с периодом 30, см. рис. 4, и второй — под названием бампер с периодом 4 (англ. *p4bumper*), см. рис. 5.

Рис. 4. Конфигурация отражателя «ковбой», стрелками указаны направления распространения планера до и после отражения

Крайне важным элементом нашего вычислительного устройства будет дупликатор (англ. *duplicator*), см. рис. 6, который при поглощении одного

планера порождает два, летящих во взаимно перпендикулярных направлениях. При этом помимо его основной функции дубликатор можно использовать как отражатель на 90° , если с помощью поглотителя остановить планер, который будет распространяться параллельно исходному, и как ускоритель, если остановить планер, распространяющийся перпендикулярно исходному.

Рис. 5. Конфигурация отражателя «бампер с периодом 4», стрелками указаны направления распространения планера до и после отражения

Под ускорителем в данном случае подразумевается конфигурация, через прохождение которой планер переместится на некоторое расстояние дальше, по сравнению с аналогичным планером, не прошедшим через нее.

Рис. 6. Сравнение движений планера после взаимодействия с дубликатором и свободного планера, стрелками указаны направления движений

Все это разнообразие конфигураций необходимо для того, чтобы мы могли синхронизировать нужным образом потоки планеров, ведь в нашем вычислительном устройстве нужна как возможность пролета планеров по пересекающимся траекториям без взаимодействия, так и возможность столкновения планеров со взаимным поглощением.

Логические элементы

Остановимся на базовой операции любого вычислительного устройства — сложении двух натуральных чисел. Для реализации данной функции необходим узел, выполняющий операцию сложения сигналов, т.е. полный одноразрядный сумматор. На вход сумматору подается три сигнала A , B и P_1 , которые можно интерпретировать как цифры из текущего разряда суммируемых двоичных чисел и сдвиг порядка от суммирования на предыдущем разряде (см. рис. 7).

Рис. 7. Схема сложения чисел и полного одноразрядного сумматора

Выходы S и P_2 определяются по формулам

$$S = A \oplus B \oplus P_1;$$

$$P_2 = (A \wedge B) \vee (A \oplus B) \wedge P_1;$$

где знаками $\wedge$, $\vee$ и $\oplus$ обозначены операции логического умножения, сложения и сложения по модулю 2 соответственно. Говоря простым языком, выход S равен единице, если количество единиц, поданных на вход, есть число нечетное, а P_2 равен единице, если это количество больше одного. При невыполнении соответствующих условий выходы S и P_2 равны нулю.

Для реализации этих операций были придуманы следующие логические схемы. На рис. 8 изображено устройство элемента для вычисления операции $A \oplus B$. Здесь и далее закрашенными кружками G_1 и G_2 отмечено начальное положение двух планеров. В зависимости от того, чему равны входы A и B , нулю или единице, соответствующими кружками обозначены места, где будут или не будут присутствовать планеры, отвечающие этим сигналам. E_1 и E_2 — обозначения поглотителей, *Getter* — приемник результирующего сигнала. Наконец, стрелками обозначены возможные траектории движения планеров, а окружностями отмечены места их потенциального столкновения. Видно, что в случае одинаковых входов A и B все планеры уничтожат друг друга и результат будет 0. В случае же когда один из входных сигналов равен единице, он остановит планер G_2 , который не сможет остановить планер G_1 , и результирующим сигналом будет 1.

Рис. 8. Схема реализации операции сложение по модулю 2

Для реализации всей логической операции $(A \wedge B) \vee (A \oplus B) \wedge P$ с тремя входами была придумана конфигурация, представленная на рис. 9. Можно заметить, что если количество входов, равных единице, больше одного, то они останавливают планеры G_2 и G_3 , которые в свою очередь не останавливают G_1 , таким образом, выходной сигнал равен 1. Иначе G_3 останавливает G_1 , и на выходе получается 0.

Рис. 9. Схема реализации операции $(A \wedge B) \vee (A \oplus B) \wedge P$

Стоит отметить, что в [3] представлены конфигурации, реализующие базовые логические операции И, ИЛИ, НЕ. Мы могли бы использовать и их, но в этом случае полученные схемы стали бы более громоздкими и требовали бы большего числа копий входных данных. Мы же, основываясь на принципах построения этих базовых операций, получили компактные конфигурации, реализующие необходимые логические соотношения.

Полный одноразрядный сумматор

С использованием вышеуказанных логических и основных структурных элементов была построена конфигурация, реализующая полный одноразрядный сумматор (см. рис. 10).

Рис. 10. Конфигурация клеток для полного одноразрядного сумматора

На рис. 11 схематично указан принцип работы сумматора. На вход сумматору в зависимости от значений A , B и P_1 подаются или не подаются в соответствующие места планеры. Также дополнительно независимо от входных данных задаются еще три планера. Все указанные конфигурации начинают свое

Рис. 11. Устройство полного одноразрядного сумматора

движение. Сначала планеры проходят через дубликаторы. Те, которые движутся в одну сторону, участвуют в расчете выхода P_2 , другие же — в расчете выхода S . Далее планеры взаимодействуют друг с другом, как в полученных ранее логических схемах. Основной сложностью при реализации данных схем была необходимость правильной синхронизации движения

планеров. Используя различные структурные элементы (отражатели на 90° , ускорители и т.д.), удалось добиться того, что в нужных местах (на рис. 11 они отмечены малым кружком) происходило столкновение планеров со взаимным уничтожением, при этом в других местах пересечений траекторий движения они двигались свободно и без взаимодействия.

Отметим некоторые свойства полученной конфигурации. Она занимает размер в 465×560 клеток, при отсутствии входных планеров повторяет сама себя за 60 поколений. Расчет выхода P_2 происходит менее чем за 1000 поколений, а выхода S — менее чем за 1700. Далее будет видно, что это свойство позволяет начинать расчет суммы и сдвига порядка для следующего разряда в то время, пока идет счет выхода S текущего, а это позволяет сэкономить число поколений при расчете суммы многоразрядных чисел.

Полный многоразрядный сумматор

Имея в своем распоряжении одноразрядный сумматор, легко построить конфигурацию, результатом работы которой будет вычисление суммы двух натуральных чисел, см. рис. 12. Для этого необходимо расположить одноразрядные сумматоры последовательно и на вход каждому последующему подавать значение выхода сдвига предыдущего.

Рис. 12. Схематичное устройство полного многоразрядного сумматора

Главной особенностью, которую нужно учесть при последовательном расположении конфигураций полных одноразрядных сумматоров, является то, что нужно своевременно «подгружать» планеры, соответствующие различным разрядам. На рис. 13 схематично изображено устройство предлагаемого нами полного многоразрядного сумматора. Параллелограмм «Full adder» обозначает всю конфигурацию полного одноразрядного сумматора. Вход P_0 в этом случае равен 0, а выходы S_0, S_1 и т.д. вычисляются последовательно новым сумматором.

Рис. 13. Схематичное устройство полного многоразрядного сумматора

Таким образом, в случае $(n+1)$ -разрядных суммируемых чисел планеры соответствующие значениям входных данных $A_0, A_1, \dots, A_n$ и $B_0, B_1, \dots, B_n$, располагаются попарно в первой строке на таком расстоянии друг от друга, чтобы при расчете они своевременно «подгружались» в сумматор, вычисляющий нужный разряд. После вычисления в местах $S_0, S_1, \dots, S_n, P_n$ располагаются приемники результирующих сигналов от каждого разряда. По полученным этими приемниками значениям можно определить результат сложения двоичных чисел.

Стоит отметить, что полученные по указанной схеме начальные состояния клеток для сумматора в случае произвольного числа разрядов при отсутствии входных планеров также являются полностью периодичными.

На рис. 14 представлена конфигурация, реализующая вычисление суммы двухразрядных чисел в двоичной системе счисления по схеме, представленной выше. Входные данные для сумматора 2 вначале движутся по направлениям, перпендикулярным направлениям входных данных для сумматора 1, а затем поворачивают на 90° и вместе с выходным сигналом сдвига порядка P_1 входят во второй сумматор. Вся конфигурация помещается в область размером 770×1000 клеток, а вычисления производятся менее чем за 3000 поколений.

Рис. 14. Конфигурация для вычисления суммы двухразрядных чисел в двоичной системе счисления

Заключение

В работе рассмотрен способ организации вычисления суммы двух двоичных чисел на основе клеточного автомата "Жизнь". Для этого были рассмотрены некоторые конфигурации клеток, реализующие передачу, дублирование, прием сигналов, а также возможность смены направления их распространения.

Получена конфигурация клеток, представляющая полный одноразрядный сумматор, а также конфигурация, реализующая вычисление суммы двух двухразрядных чисел в двоичной системе счисления.

Также была описана схема построения сумматора в случае произвольного числа разрядов, а именно расположение одноразрядных сумматоров, принцип расположения входных и приема выходных данных.

Список литературы

- [1] Suhas K. Fundamental Limits to Moore's Law. 2015. 3 p. URL: <http://arxiv.org/abs/1511.05956>
- [2] Dascalu M. Cellular Automata Hardware Implementations - an Overview // Romanian journal of information science and technology. 2016. Vol. 19. p. 360-368.
- [3] Berlekamp E.R., Conway J.H., Guy R.K. Winning Ways for Your Mathematical Plays, second edition. Vol. 4. Wellesley, Massachusetts: A K Peters, 2004. 214 p.
- [4] Adamatzky A. Game of Life Cellular Automata. Springer-Verlag London. 2010. 576 p. URL: <http://dx.doi.org/10.1007/978-1-84996-217-9>
- [5] Rendell P. Turing Machine Universality of the Game of Life. Springer International Publishing. 2016. 182 p. URL: <http://dx.doi.org/10.1007/978-3-319-19842-2>
- [6] Category: 90 degree reflectors // LifeWiki. Свободная энциклопедия. URL: https://www.conwaylife.com/wiki/Category:90_degree_reflectors (дата обращения: 03.12.2019).