

УДК 519. 713.1

Е.Е. Федоров, д-р. техн. наук, доц.,
Х.. Альрабаба, аспирант
Донецкий национальный технический университет
fedorovee75@mail.ru

Разработка архитектуры многослойного клеточного автомата и способов записи и чтения из его ячеек

В статье изложен подход к построению высокопроизводительных клеточных автоматов. Для увеличения производительности аппаратной реализации клеточного автомата предлагается использовать многослойную структурную организацию. Предложена архитектура ячейки многослойного клеточного автомата, отличающаяся повышенной производительностью и расширенной функциональностью по сравнению с известными клеточными автоматами. Для повышения быстродействия аппаратной реализации клеточного автомата предложены способы реализации записи начальных значений и считывания результатов работы. Предложенный метод прогноза производительности может использоваться в интеллектуальных компьютерных системах для решения задач, связанных с моделированием сложных динамических процессов.

Ключевые слова: архитектура многослойного клеточного автомата, клеточный тетра-автомат, тетралогика и тетракоды, запись и чтение из ячеек клеточного автомата, непосредственная и косвенная адресация.

Введение

На сегодняшний день аппаратная реализация алгоритмов и вычислительных процессов является способом достижения максимальной производительности. В общем случае при аппаратной реализации по сравнению с программной достигается ускорение на 1-2 порядка. Однако клеточные автоматы являются довольно специфическим видом вычислительных устройств, в связи с чем известно относительно мало успешных попыток их аппаратной реализации.

В работах [1-3] определены постбинарные клеточные автоматы (ПКА), в который, в отличие от традиционных клеточных автоматов (КА), число состояний ячеек больше двух. В цикле работ, например [4-6] в качестве постбинарной логики и постбинарного кодирования выделены такие направления, как тетралогика и тетракодирование. Теория КА обязана своим появлением Джону фон Нейману. В конце 40-х годов, накопив огромный практический опыт в создании быстродействующих вычислительных машин, фон Нейман приступил к созданию общей математической теории автоматов [7]. После фон Неймана исследования были продолжены его учеником Станиславом Уламом и Джоном Холландом – сотрудниками университета Лос-Аламос в университете Мичигана [8–10]. Практически параллельно и независимо от исследований Улама и фон Неймана работу над КА начал Конрад Цузе, который в 1969 г. опубликовал книгу «Rechnender Raum» («Вычислительное

пространство») – «Calculating Space» [11]. В 1984 г. Кристофер Лэнгтон создал самовоспроизводящийся автомат, клетка которого может находиться в одном из восьми возможных состояний, но при этом для самовоспроизведения требуется гораздо меньшее количество клеток, чем сотни и тысячи ячеек, необходимых для самовоспроизведения в автомате Неймана [12]. После Лэнгтона теория самовоспроизводящихся автоматов стала двигаться в сторону упрощения. Среди последователей Лэнгтона широко известны работы: Эдгара Кодда, Эдвина Роджера Бэнкса, и др. исследователей, результатами работы которых стало появление множества КА с количеством состояний от 8 до 64 [13-16]. Самым известным последователем, продолжившим исследования в области упрощения КА, стал английский математик Джон Хортон Конуэй – создатель математической игры «Жизнь» [1, 17-18].

Согласно [1-18] одной из главных проблем ПКА являются недостаточная производительность. В данной статье предлагается аппаратно-программное решение этой проблемы.

Описание

Целью работы является разработка архитектуры многослойного клеточного автомата и способов записи и чтения из его ячеек.

Для достижения поставленной цели необходимо:

1. Разработать архитектуру многослойного клеточного автомата.

2. Создать архитектуру ячейки многослойного клеточного автомата.
3. Разработать способы реализации записи начальных значений в ячейки клеточного автомата.
4. Создать способа реализации считывания результатов работы из ячейки клеточного автомата.

1. Создание архитектуры многослойного клеточного автомата

Для ускорения работы клеточного автомата, а именно этапов записи начальных значений и процесса считывания результатов предлагается использовать многослойную структурную организацию клеточного автомата.

Предложенная структурная организация может быть использована как для обычных клеточных автоматов, так и для клеточных автоматов на основе тетралогики и тетракодов (тетраавтоматов).

Архитектура многослойного клеточного автомата приведена на рис. 1.

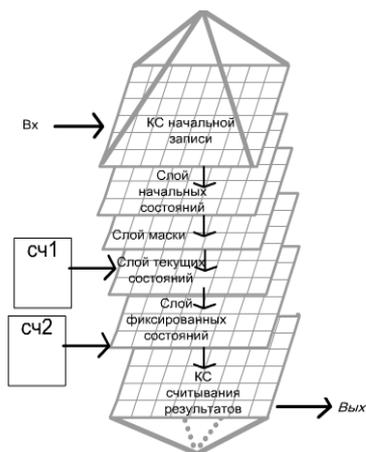


Рисунок 1 – Архитектура многослойного клеточного автомата

На заполнение слоя текущих состояний начальными значениями кроме слоя начальных состояний, может повлиять слой маски.

Обобщенная структурная схема ячейки многослойного клеточного автомата изображена на рис. 2.

Следует отметить, что блок записи начальных значений (ЗНЗ) и блок считывания результатов (СР), приведенные на рис. 2, не входят в структуру ячейки клеточного автомата.

Шина данных (MD) подключена к компьютеру и через специальный программный интерфейс пользователь может управлять работой клеточного автомата – загружать таблицу правил переходов и значение маски.

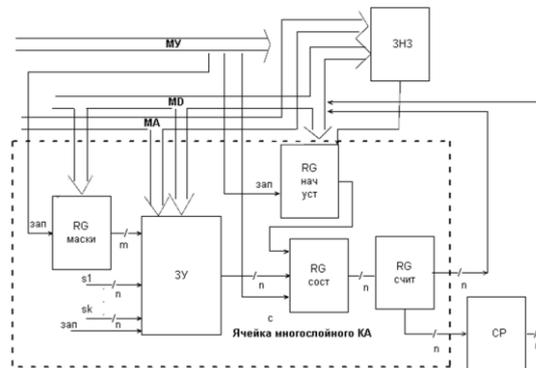


Рисунок 2 – Архитектура ячейки многослойного клеточного автомата

Комбинационного запоминающее устройство (ЗУ) хранит таблицу правил переходов клеточного автомата, в которой в зависимости от комбинации состояний соседних клеток явным образом определено, в какое состояние переключается текущая клетка.

Когда на вход ЗУ с шины управления (МУ) подается управляющий сигнал «запись» (зап), то в ЗУ с шины данных (MD) загружается таблица правил переходов клеточного автомата, а через адресную шину (МА) передаются адреса элементов, в которые осуществляется запись правил.

Когда на вход регистра маски (RG маски) с шины управления (МУ) подается управляющий сигнал «запись» (зап), то в регистр маски с шины данных (MD) загружается значение маски клеточного автомата.

Когда на вход синхронизированного регистра состояний (RG состояний) с шины управления (МУ) подается управляющий сигнал «сброс» (с), то регистр состояний сбрасывается в 0.

Когда на вход регистра начальных значений (RG нач. уст.) с шины управления (МУ) подается управляющий сигнал «запись» (зап), то в регистр начальных значений с шины данных (MD) загружается начальное состояние ячейки клеточного автомата.

При подаче соответствующих состояний соседних ячеек ($s_1 \dots s_k$) на адресные входы ЗУ на основе загруженной таблицы правил переходов клеточного автомата формируется новое состояние, которое записывается в регистр состояний.

Новое состояние поступает с выхода регистра состояний (RG состояний) на вход регистра считывания (RG счит.).

Запись новых состояний может осуществляться либо с регистра начальных значений (RG нач. уст), либо с ЗУ.

2. Разработка способа реализации записи начальных значений в ячейки многослойного клеточного автомата

Для архитектуры многослойного клеточного автомата, приведенной на рис.1, предлагаются два способа реализации записи начальных значений в ячейки клеточного автомата:

- запись в ячейку через непосредственную адресацию (без использования тетракодов);
- запись в ячейку через каскадную адресацию (на основе тетракодов).

Оба способа предусматривают разработку соответствующей архитектуры и принципов работы, связанных с этой архитектурой.

2.1. Запись начальных значений через непосредственную адресацию

На рис. 3 приведена обобщенная архитектура, используемая для записи начальных значений в ячейку клеточного автомата через непосредственную адресацию.

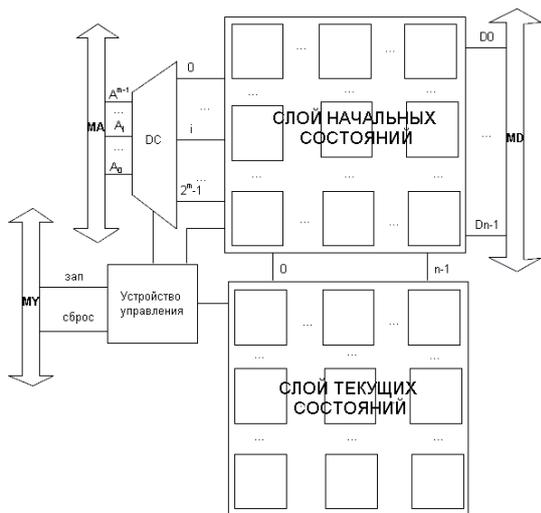


Рисунок 3 – Обобщенная архитектура, используемая для записи начальных значений в ячейку клеточного автомата через непосредственную адресацию

Слой начальных состояний клеточного автомата представляет собой блок регистров состояний, размерностью nxn, при этом размерность слоя начальных состояний должна быть равна слою текущих состояний.

Принцип работы для архитектуры рис.3 заключается в следующем.

Когда на вход дешифратора (DC) с шины управления (МУ) через устройство управления подается управляющий сигнал «запись» (зап), то в дешифратор через адресную шину (МА) передаются адреса регистров состояний слоя начальных состояний, в которые с шины данных (MD) осуществляется запись данных.

Когда на вход дешифратора (DC) с шины управления (МУ) через устройство управления подается управляющий сигнал «сброс» (с), то все регистры состояний слоя начальных состояний сбрасываются в 0.

Дешифратор на рис. 3 состоит из 2-х частей, дешифратора, предназначенного для адресации столбцов (или координаты X), DCX, и дешифратора для адресации строк (или координаты Y), DCY.

Логические уравнения для соответствующих выходных разрядов дешифратора имеют вид:

$$Qx_0 = \overline{Ax_0} \& \overline{Ay_0} \& \dots \& \overline{Ax_{n-1}} \& \overline{Ay_{n-1}} ,$$

$$Qy_0 = \overline{Ax_0} \& Ay_0 \& \dots \& \overline{Ax_{n-1}} \& Ay_{n-1} ,$$

$$Qx_{2^n-1} = Ax_0 \& Ay_0 \& \dots \& Ax_{n-1} \& \overline{Ay_{n-1}} ,$$

$$Qy_{2^n-1} = Ax_0 \& Ay_0 \& \dots \& \overline{Ax_{n-1}} \& Ay_{n-1} .$$

На вход записи всех регистров слоя начальной записи будет подаваться соответственно сигнал $зп_i = зап \& Qx_i \& Qy_i$.

На вход сброса всех регистров слоя начальной записи подается сигнал сброса $R = сброс$.

На вход данных всех регистров слоя начальной записи значение с шины данных (MD) подаются данные $D = MD_i$.

На вход D регистра всех регистров слоя текущих состояний будет подаваться соответствующий выход регистров со слоя начальных данных.

Рассмотрим пример записи начальных значений в ячейку клеточного автомата, размерностью 2x2, через непосредственную адресацию (рис. 4).

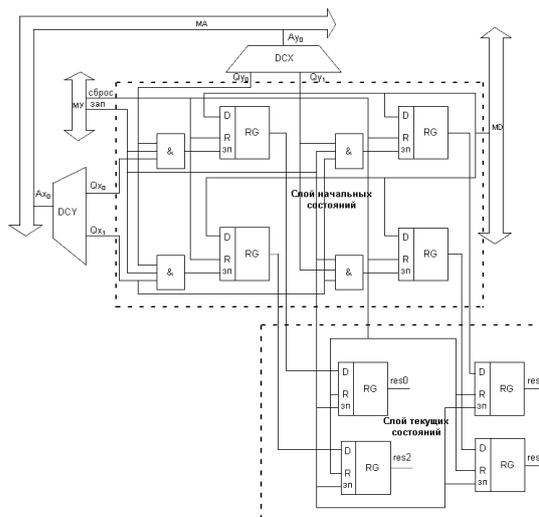


Рисунок 4 – Пример архитектуры, используемой для записи начальных значений в ячейку клеточного автомата размерностью 2x2 через непосредственную адресацию

Тогда логические уравнения для соответствующих выходных разрядов дешифратора имеют вид:

$$Qx_0 = \overline{Ax_0} \& \overline{Ay_0} ,$$

$$Qy_0 = Ax_0 \& \overline{Ay_0} ,$$

$$Qx_1 = Ax_0 \& \overline{Ay_1} ,$$

$$Qy_1 = Ax_0 \& Ay_0 .$$

При такой организации записи начальных значений время записи будет напрямую зависеть от размерности поля, поскольку за каждый такт в ячейку клеточного автомата будет записываться лишь одно начальное значение.

Аналогично может быть организована блочная запись в регистры начальных состояний. При этом в зависимости от возможного загружаемого блока будут уменьшены разрядности дешифратора, поскольку теперь функцией дешифраторов будет адресация блоков, а не каждого отдельного регистра. При этом количество тактов при осуществлении записи начальных значений при такой загрузке начальных значений будет уменьшаться с увеличением блока входных данных.

2.2. Запись начальных значений через каскадную адресацию

Каскадная адресация и запись начальных значений является наиболее эффективным по быстродействию способом задания начальных значений конечного автомата.

На рис. 5 представлен пример архитектуры, используемой для записи начальных значений в ячейку клеточного автомата размерностью 16x16 через каскадную адресацию.

Принцип работы для архитектуры рис.5 заключается в следующем.

1. Когда на вход регистра данных (RG) с шины управления (МУ) подается управляющий сигнал «запись» (зап), то в регистр данных с шины данных (MD) загружается значение, которое в общем случае может представлять собой одно из значений тетракода {1, M}. Когда на вход регистра данных (RG) с шины управления (МУ) подается управляющий сигнал «сброс» (с), то значение регистра сбрасывается в 0. Возможность задания значения A тетракода на данной структурной схеме не предусмотрено.

2. С адресной шины (МА) на входы дешифраторов: DCX₀, DCY₀, DCX₁, DCY₁, DCX₂, DCY₂ подаются адресные разряды, которые представляют собой одно из значений тетракода {0, 1, M}.

3. На рис. 5 функциями комбинационных схем: КС упр 1, КС упр 2, КС упр 3 – является формирование сигналов записи, которые подается на соответствующие входы «зп» всех реги-

стров каждого уровня. При этом КС упр 1 отвечает за формирование сигнала записи на 1 уровне, а КС упр 2 и КС упр 3, соответственно, – на 2 и на 3 уровне. Формирование сигнала зависит от выходов дешифраторов (для 1 уровня – DCX₀, DCY₀, для 2 уровня – DCX₁, DCY₁, для 3 – DCX₂, DCY₂) и от соответствующих выходов регистров предыдущего уровня, также участвующих в формировании адреса соответствующих регистров.

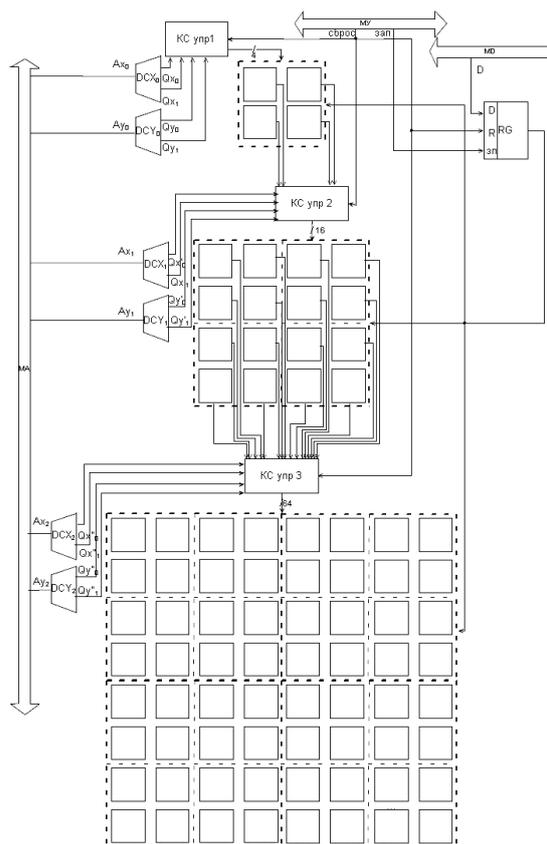


Рисунок 5 – Пример архитектуры, используемой для записи начальных значений в ячейку клеточного автомата размерностью 16x16 через каскадную адресацию

4. Верхний уровень каскада представляет собой 4 регистра, на входы данных которых подаются значения из регистра данных (RG). На вход R – сигнал «сброс». На вход записи (зп) – подается сигнал с КС упр 1. И в зависимости от этого сигнала в соответствующий регистр может быть записано, подаваемое с регистра данных значение, или регистр останется в предыдущем состоянии. Выходы каждого регистра подаются на КС упр 2 следующего уровня для формирования сигнала записи регистров следующего уровня. Второй и третий уровень каскада являются блоками регистров, работающих по тому же принципу, что и регистры 1 уровня, за исключением того, что адресация для подачи сигнала

записи на входи «зп» будет формироваться в схемах КС упр2 и КС упр 3 соответственно, в зависимости от выходов дешифраторов уровня 2 или 3 и выходов регистров предыдущего уровня. Размерность блоков регистров каждого последующего уровня больше в 4 раза блока регистров предыдущего уровня.

При подаче на адресные входы дешифраторов DCX₀, DCY₀ двоичных значений логические уравнения для соответствующих выходных разрядов дешифратора имеют вид:

$$\begin{aligned} Qx_0 &= \overline{Ax_0} \& Ay_0, \\ Qy_0 &= \overline{Ax_0} \& Ay_0, \\ Qx_1 &= Ax_0 \& \overline{Ay_0}, \\ Qy_1 &= Ax_0 \& Ay_0. \end{aligned}$$

При подаче на адресные входы дешифраторов значений множественности логические уравнения для соответствующих выходных разрядов дешифратора имеют вид:

$$\begin{aligned} Qx_0 \& Qx_1 &= \overline{Ax_0} \& Ay^M_0, \\ Qx_0 \& Qy_0 &= Ax^M_0 \& \overline{Ay_0}, \\ Qy_0 \& Qy_1 &= Ax_0 \& A^M y_0, \\ Qx_1 \& Qy_1 &= Ax^M_0 \& Ay_0, \\ Qx_0 \& Qx_1 \& Qy_0 \& Qy_1 &= Ax^M_0 \& Ay^M_0. \end{aligned}$$

Индекс М в формулах обозначает множественный разряд.

На последующих уровнях адресация будет зависеть от соответствующих выходных значений регистров предыдущего уровня.

На рис. 6 представлен пример архитектуры, используемой для записи начальных значений в ячейку клеточного автомата размерностью 4x4 через каскадную адресацию

Как формируются адреса верхнего уровня каскада рассмотрено выше.

Рассмотрим формирование адресов 2-го уровня (рис. 6). Значения, формируемые в регистрах: RG'00, RG'01, RG'10, RG'11, будут зависеть от: Ax₁, Ay₁, RG₀₀. Если RG₀₀=0, то в этих регистрах значения сформированы не будут.

$$\begin{aligned} Qx1_0 &= \overline{Ax_1} \& \overline{Ay_1} \& (RG_{00} | RG^M_{00}), \\ Qy1_0 &= \overline{Ax_1} \& Ay_1 \& (RG_{00} | RG^M_{00}), \\ Qx1_1 &= Ax_1 \& \overline{Ay_1} \& (RG_{00} | RG^M_{00}), \\ Qy1_1 &= Ax_1 \& Ay_1 \& (RG_{00} | RG^M_{00}), \\ Qx1_0 \& Qx1_1 &= \overline{Ax_1} \& Ay^M_1 \& (RG_{00} | RG^M_{00}), \\ Qx1_0 \& Qy1_0 &= Ax^M_1 \& \overline{Ay_1} \& (RG_{00} | RG^M_{00}), \\ Qy1_0 \& Qy1_1 &= Ax_1 \& A^M y_1 \& (RG_{00} | RG^M_{00}), \\ Qx1_1 \& Qy1_1 &= Ax^M_1 \& Ay_1 \& (RG_{00} | RG^M_{00}), \\ Qx1_0 \& Qx1_1 \& Qy1_0 \& Qy1_1 &= . \end{aligned}$$

$$= Ax^M_1 \& Ay^M_1 \& (RG_{00} | RG^M_{00}).$$

Значения, формируемые в регистрах: RG'02, RG'03, RG'12, RG'13, будут зависеть от: Ax₁, Ay₁, RG₀₁. Если RG₀₁=0, то в этих регистрах значения сформированы не будут.

Значения, формируемые в регистрах: RG'20, RG'30, RG'21, RG'31, будут зависеть от: Ax₁, Ay₁, RG₁₀. Если RG₁₀=0, то в этих регистрах значения сформированы не будут.

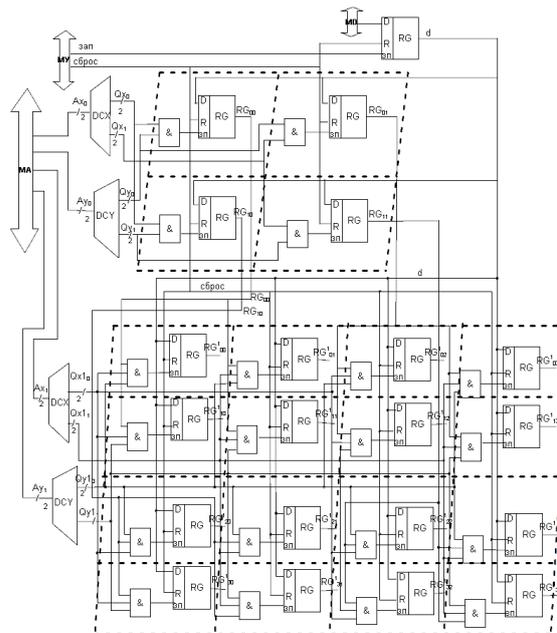


Рисунок 6 – Пример архитектуры, используемой для записи начальных значений в ячейку клеточного автомата размерностью 4x4 через каскадную адресацию

Значения, формируемые в регистрах: RG'22, RG'23, RG'32, RG'33, будут зависеть от: Ax₁, Ay₁, RG₁₁. Если RG₁₀=0, то в этих регистрах значения сформированы не будут.

Так, например, чтобы заполнить все регистры 2-го уровня одним значением за один такт, на вход данных всех регистров подается нужное значение, а на адресные входы Ax₀, Ay₀, Ax₁, Ay₁ значение «М».

Для обеспечения возможности задания неопределенных (вероятностных) значений А тетракода в схеме могут быть задействованы генераторы случайных чисел. Однако, в рамках данной работы аппаратная поддержка такой возможности детально не рассматривалась.

3. Создание способа реализации считывания результатов работы из ячеек многослойного клеточного автомата

Считывание результатов может осуществляться на каждом такте работы автомата, поэтому этот этап является наиболее критичным для

быстродействия конечного автомата. Скорость считывания будет существенно снижаться с увеличением размерности поля конечного автомата.

Для архитектуры многослойного клеточного автомата, приведенной на рис.1, предлагаются два способа реализации считывания результатов работы из ячейки клеточного автомата:

- непосредственное считывание из ячейки (без использования тетракодов);
- каскадное считывание из ячейки (на основе тетракодов).

Оба способа предусматривают разработку соответствующей архитектуры и принципов работы, связанных с этой архитектурой.

3.1. Непосредственное считывание результатов работы

Для ускорения считывания предлагается использовать слой считывания результатов, состоящий из такого же количества ячеек, что и слой текущих состояний.

Значения из слоя текущих состояний будут переписываться в регистре считывания через заданное количество шагов.

Для этого потребуются комбинационная схема, которая будет содержать 2 счетчика (Сч1 и Сч2, приведенные на рис.4).

Принцип работы непосредственного считывания результатов заключается в следующем.

Содержимое первого счетчика декрементаруется после каждой итерации в слое текущих состояний,

Содержимое второго счетчика декрементаруется после каждого считанного на шину данных значения.

Пока значения двух счетчиков не обнулятся, текущий слой будет находиться в состоянии ожидания. Как только оба счетчика станут равны 0, в слой результатов запишутся новые значения со слоя текущих состояний, на вход синхронизации все регистров слоя текущего состояния будет подан активный сигнал, и все регистры будут запущены снова на выполнение.

Так считывание результатов будет производиться частично параллельно с работой клеточного автомата (в зависимости от заданного пользователем значения счетчика шагов текущих состояний), таким образом можно существенно повысить быстродействие работы всей общей схемы клеточного автомата.

3.2. Каскадное считывание результатов работы

На слой считывания результатов работы может быть настроен каскад считывания результатов (рис. 7).

Принцип работы каскадного считывания результатов заключается в следующем.

Блоки по n регистров слоя считывания результатов подаются на соответствующие комбинационные схемы (КС). Функция каждой такой схемы, проверить, если в каждом таком блоке ненулевые элементы. Так, если все регистры блока равны 0, на выходе КС – 0, если есть несколько ненулевых значений – на выходе А, если 1 ненулевое значение – на выходе 1, если все значения в блоке ненулевые – на выходе КС значение М. На следующем уровне каскада считывания выходы КС предшествующего уровня группируются по n и подаются на соответствующие КС этого уровня. Все КС работают аналогичным образом параллельно.

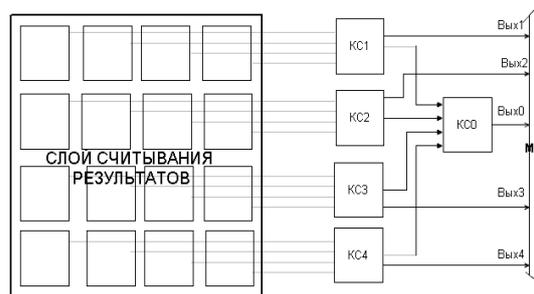


Рисунок 7 – Пример архитектуры, используемой для каскадного считывания результатов работы из ячейки клеточного автомата

На выходе самой верхней КС будет общее значение в формате тетракода, подаваемое в качестве результата на шину данных. Если верхняя КС (КС0) содержит 0, все клетки на поле пусты, нужно остановить автомат. Если верхняя КС содержит 1, есть лишь одна живая клетка, нужно остановить автомат, и можно не считывать значения. Если КС = А или М – на поле есть живые клетки. Тогда можно проверять КС более низкого уровня. Если в каких-то из них значения равны 0, то из соответствующего квадранта слоя считывания результатов, значения считывать необязательно, поскольку они = 0.

Заключение

В работе для увеличения производительности аппаратной реализации клеточного автомата предлагается использовать многослойную структурную организацию. Данная архитектура может быть использована для клеточного автомата на основе тетралогии и тетракодов. Предложена архитектура ячейки многослойного клеточного автомата, отличающаяся повышенной производительностью и расширенной функциональностью по сравнению с известными клеточными автоматами. Данная архитектура может быть использована для клеточного автомата на основе тетралогии и тетракодов. Для повышения быстродействия аппаратной реализации кле-

точного автомата предложены способы реализации записи начальных значений и считывания результатов работы. Данные способы могут быть использованы для клеточного автомата на основе тетралогии и тетракодов.

Созданные архитектуры и способы могут использоваться в интеллектуальных компьютерных системах для решения задач, связанных с моделированием сложных динамических процессов.

Список использованной литературы

1. Коноплева, А. П. Игра «Жизнь» Дж. Конвея на базе гиперкодов [Текст] / А. П. Коноплева, А. Я. Аноприенко // Материалы III международной научно-технической конференции «Информатика и компьютерные технологии — 2007», 11-13 декабря 2007 года, Донецк, ДонНТУ, 2007. С. 254–257.
2. Аноприенко, А. Я. Развитие идеи применения гиперкодов в моделировании клеточных автоматов [Текст] / А. Я. Аноприенко, А. П. Коноплева // Наукові праці Донецького національного технічного університету, серія «Інформатика, кібернетика та обчислювальна техніка», вып. 9 (132), Донецк, ДонНТУ, 2008. — С.115–118.
3. Аноприенко, А. Я. Управляемый постбинарный клеточный автомат / А. Я. Аноприенко, А. П. Коноплева [Текст] // Материалы II всеукраинской научно-технической конференции «Информационные управляющие системы и компьютерный мониторинг (ИУС и КМ 2011)» — 12–13 апреля 2011 г., Донецк, ДонНТУ, 2011. Т. 2.
4. Аноприенко, А. Я. Постбинарный компьютеринг и интервальные вычисления в контексте кодо-логической эволюции [Текст] / А. Я. Аноприенко, С. В. Иваница — Донецк, ДонНТУ, УНИТЕХ, 2011. — 248 с.
5. Аноприенко, А. Я. Особенности реализации постбинарных логических операций / А. Я. Аноприенко, С. В. Иваница // Научно-теоретический журнал «Искусственный интеллект», № 2, 2011. С. 110–121.
6. Аноприенко А. Я. Тетралогика, тетравычисления и ноокомпьютеринг [Текст] / А. Я. Аноприенко, С. В. Иваница — Донецк, ДонНТУ, УНИТЕХ, 2012. — 308 с.
7. Piwonska A., Sereczynski F., Szaban M. Searching Cellular Automata Rules for Solving Two-Dimensional Binary Classification Problem / Anna Piwonska, Franciszek Sereczynski, Mirosław Szaban — Cellular Automata. Lecture Notes in Computer Science Volume 7495, 2012, pp 121–130.
8. Martínez J. Localization Dynamics in a Binary Two-Dimensional Cellular Automaton: The Diffusion Rule / Genaro J. Martínez, Andrew Adamatzky, Harold V. McIntosh — Game of Life Cellular Automata. 2010, pp 291–315.
9. Schiff J. L. Cellular automata: a discrete view of the world. A John Wiley&Sons inc, Publication. University of Auckland.— 2008.— 279 p.
10. Cenek M., Mitchell M. Evolving Cellular Automata / Martin Cenek, Melanie Mitchell — Encyclopedia of Complexity and Systems Science. 2009, pp. 3233–3242.
11. Sutner K. Linear Cellular Automata and de Bruijn Automata / K. Sutner — Cellular Automata. Mathematics and Its Applications Volume 460, 1999, pp. 303–319.
12. Pivato M. Ergodic Theory of Cellular Automata / Marcus Pivato — Encyclopedia of Complexity and Systems Science. 2009, pp 2980–3015.
13. Breukelaar R., Back T. HW. Using a genetic algorithm to evolve behavior in multi dimensional cellular automata: emergence of behavior / R. Breukelaar, Thomas HW Back. — Gecco '05 Proceedings of the 2005 conference on Genetic and evolutionary computation. New York, NY, USA 2005.
14. Schiff J. L. Cellular automata: a discrete view of the world. A John Wiley&Sons inc, Publication. University of Auckland. — 2008. — 279 p.
15. Voccara N. Phase Transitions in Cellular Automata / Nino Voccara — Encyclopedia of Complexity and Systems Science. 2009, pp. 6771–6782.
16. Tempesti G. Self-Replication and Cellular Automata / Dr. Gianluca Tempesti, Dr. Daniel Mange, Dr. André Stauffer — Encyclopedia of Complexity and Systems Science. 2014, pp. 1–24.
17. Weisstein E.W. Elementary Cellular Automaton. [Электронный ресурс] / Weisstein Eric W. MathWorld — A Wolfram Web Resource. Режим доступа — <http://mathworld.wolfram.com/ElementaryCellularAutomaton.html>.
18. Cook M. Universality in Elementary Cellular Automata / Matthew Cook — Department of Computation and Neural Systems, Caltech, Mail Stop 136-93, Pasadena, California 91125, USA.
19. Степанцов М. Е. Клеточные автоматы как модели нелинейных явлений [Текст] / М. Е. Степанцов. Математические методы и приложения. Труды девятого математических чтений — МГСУ. Москва, 2002, с. 141–142.

Надійшла до редколегії 19.01.2015

Є.С. ФЕДОРОВ, Х. АЛЬРАБАБА

Донецький національний технічний університету

РОЗРОБКА АРХІТЕКТУРИ БАГАТОШАРОВОГО КЛІТИННОГО АВТОМАТА ТА СПОСОБІВ ЗАПИСУ ТА ЧИТАННЯ З ЙОГО ОСЕРЕДКІВ

У статті викладено підхід до побудови високопродуктивних клітинних автоматів. Для збільшення продуктивності апаратної реалізації клітинного автомата пропонується використовувати багат шарову структурну організацію. Запропонована архітектура осередку багат шарового клітинного автомата, що відрізняється підвищеною продуктивністю і розширеною функціональністю в порівнянні з відомими клітинними автоматами. Для підвищення швидкодії апаратної реалізації клітинного автомата запропоновані способи реалізації записи початкових значень і зчитування результатів роботи. Запропонований метод прогнозу продуктивності може використовуватися в інтелектуальних комп'ютерних системах для вирішення завдань, пов'язаних з моделюванням складних динамічних процесів.

Ключові слова: *архітектура багат шарового клітинного автомата, клітинний тетраавтомат, тетралогіка і тетракоди, запис і читання з осередків клітинного автомата, безпосередня і непряма адресація.*

E. FEDOROV, H. ALRABABA

Donetsk National Technical University

DEVELOPMENT OF A MULTI-LAYER ARCHITECTURE AND CELLULAR AUTOMATA WAYS OF WRITING AND READING OF THE CELLS

To date, the hardware implementation of algorithms and computational processes is a way to achieve maximum performance. In general, when a hardware implementation versus a software acceleration is achieved at about 1-2. However, cellular automata are quite specific type of computing device in connection with what little is known about their successful attempts hardware implementation. Currently widespread postbinary cellular automata (PCA), which, unlike traditional cellular automata (CA), the number of states of cells greater than two. A number of papers as postbinary logic and postbinary coding isolated areas such as tetralogic and tetracoding. CA theory owes its existence to John von Neumann. At the end of the 40s, having accumulated a wealth of practical experience in the creation of high-speed computers, von Neumann began to establish a general mathematical theory of automata. After von Neumann's studies were continued by his disciple Stanislaw Ulam and John Holland - the staff of the University of Los Alamos at the University of Michigan. Almost in parallel and independently of research Ulam and von Neumann's work on the spacecraft began Konrad Zuse, who in 1969 published a book «Rechnender Raum» - «Calculating Space». In 1984, Christopher Langton has created self-replicating machine, a cell which can be in one of eight possible states, but for self-reproduction requires far fewer cells than hundreds or thousands of cells necessary for self-reproduction in the machine Neumann. After Langton self-reproducing automata theory began to move in the direction of simplification. Among the followers of Langton's widely known work: Edgar Codd, Edwin Roger Banks, et al. Research, which resulted in the emergence of multiple spacecraft to the number of states from 8 to 64. The most famous follower, to continue research in the field of simplification of the spacecraft was the English mathematician John Horton Conway - creator of mathematical game "Life". One of the main problems is the lack of PCA performance. In this paper we propose a hardware-software solution to this problem. The work to increase the performance of hardware implementation of cellular automata is proposed to use a multi-layer structural organization. This architecture can be used for cellular automata based tetralogic and tetracode. An architecture of multilayer cell cellular automaton that improves performance and extended functionality in comparison with known cellular automata. This architecture can be used for cellular automata based tetralogic and tetracode. To improve performance, the hardware implementation of a cellular automaton provides methods for the implementation of the initial values of the recording and reading of the results. These methods can be used for the cellular automaton based tetralogic and tetracode. By architecture and methods can be used in intelligent computer systems for solving problems related to the simulation of complex dynamic processes.

Keywords: *architecture multilayered cellular automata, cellular tetraavtomatf, tetralogic and tetracode, writing and reading of the cells of a cellular automaton, the direct and indirect addressing.*