УДК 004.2

*ЖАБИН В.И.,*
*ЖАБИНА В.В.,*
*БЕЗГИНСКИЙ М.А.*

# ЭФФЕКТИВНОСТЬ РЕАЛИЗАЦИИ ПОТОКОВЫХ ВЫЧИСЛЕНИЙ В СИСТЕМАХ С НЕПОСРЕДСТВЕННЫМИ СВЯЗЯМИ НА ПЛИС

Рассматривается возможность уменьшения необходимого ресурса ПЛИС для реализации потоковых вычислительных систем с непосредственными связями между вычислительными модулями. Показано, что применение вычислительных модулей, обрабатывающих операнды поразрядно с использованием избыточных систем счисления, позволяет совмещать выполнение зависимых по данным операции и сократить число связей между модулями. По сравнению с применением параллельных устройств уменьшается требуемое число функциональных ячеек и выводов ПЛИС при сохранении высокой скорости обработки данных. Показаны примеры погружения в ПЛИС вычислителей полиномов, исследованы их параметры.

Possibility of reduction of the necessary FPGA resource for realization of data flow computing systems with direct connections between computing modules is considered. It is shown that the use of the computing modules, which process operands digit-by-digit with the use of redundant numerical system, enables to perform data flow calculations as well as to reduce the number of connections between computing modules. In comparison with the use of parallel devices the demanded number of FPGA functional cells and pins can be reduced with data processing high speed maintenance. There are shown examples of polynomial calculator immersion into FPGA and investigated its parameters.

## Введение

Эффективность параллельных вычислений во многом зависит от реализуемого уровня параллелизма, что определяется зернистостью представления графа задачи. В системах реального времени многие алгоритмы обработки данных имеют мелкозернистую структуру. Например, при решении траекторных задач необходимо обеспечить высокую скорость интерполяции функций различными методами (полиномиальной аппроксимации, разложением в цепную дробь, табличко-алгоритмическими, итерационными и т.д.). Возникает необходимость решения систем алгебраических и дифференциальных уравнений [1]. Наиболее высокая степень распараллеливания может быть достигнута, когда вершинам графа соответствуют отдельные операции. При этом максимально увеличивается число параллельных ветвей, что дает потенциальную возможность использовать большее число параллельно работающих вычислительных модулей (ВМ).

Как известно, скорость обработки данных связана не только с длительностью выполнения операций, но и с затратами времени на обмен информацией между параллельными ветвями. Увеличение степени распараллеливания вычислений сопряжено с ростом интенсивности обмена данными между ВМ. Этот фактор является весьма важным и должен учитываться при выборе архитектуры систем и организации вычислений.

В работе рассматривается возможность уменьшения затрат времени на обмен данными за счет использования потоковых систем с непосредственными связями (ПСНС) между ВМ. В ПСНС выходы одних ВМ подключаются к входам других ВМ в соответствии с графом потока данных (ГПД). В процессе вычислений данные пересылаются от одних ВМ к другим, преобразуясь на каждом шаге в соответствии с операцией, заданной ГПД. При такой организации вычислительного процесса не требуются сложные процедуры пересылки данных между ВМ, что создает предпосылки к уменьшению непроизводительных затрат времени в процессе обработки информации.

При использовании ПЛИС для построения систем, кроме ускорения вычислений, важной задачей является сокращение требуемого ресурса микросхем. Это позволяет улучшить ряд характеристик систем, в том числе, повысить надежность и уменьшить энергопотребление. В работе исследуется возможность решения данной задачи путем сокращения количества связей между ВМ за счет поразрядной передачи данных.

## Организация систем с непосредственными связями

Обобщенную функциональную модель систем типа ПСНС можно представить в виде

$$S = \langle DI, DO, M, F, C, T, R, G \rangle,$$

где $DI$ – множество входных данных; $DO$ – множество результатов; $M$ – множество вычислительных модулей; $F$ – система функций преобразования данных; $C$ – характеристика средств коммутации; $T$ – требуемые характеристики системы; $R$ – ограничения, накладываемые на возможности реализации аппаратных средств; $G$ – ограничения, накладываемые на форму представления данных.

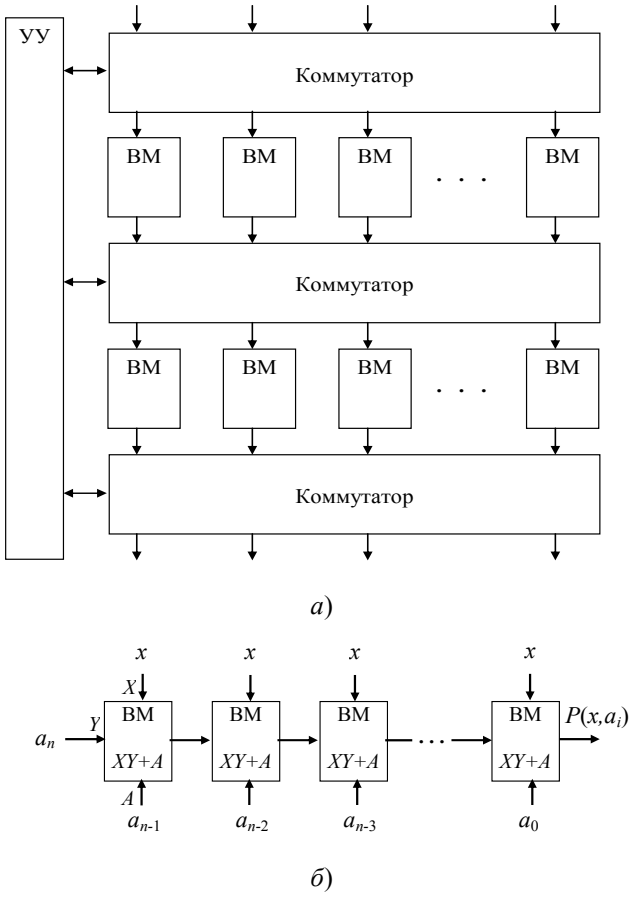


***Рис. 1. Системы с непосредственными связями между ВМ: а – с перестраиваемыми связями; б – с жесткими связями (для вычисления полиномов $P(x,a_i)$ по схеме Горнера); УУ – устройство управления***

Система функций $F$ описывается в виде алгоритмов выполнения операций в вычислительных модулях. Характеристика $C$ определяет возможности пространственного взаимодействия элементов системы. Требования $T$ к характеристикам системы определяются внешними факторами. Обычно они связаны с целевой функцией системы. Ограничения $R$ в основном определяются требованиями и возможностями элементной базы. Ограничения $G$ в основном связаны с диапазоном представления данных, точностью получения результата и т.д.

В реконфигурируемых ПСНС (рис. 1*а*) необходимое соединение между ВМ обеспечивает коммутационная среда, которая настраивается в соответствии с ГПД. Проблемам реконфигурации систем посвящено много научных работ, в том числе, монографий, например, [2, 3].

## Выполнение операций в неавтономном режиме

Использование современной технологии проектирования SoC (System on Chip – система на кристалле) позволяет создавать сложные системы на основе ПЛИС. Однако со стороны элементной базы накладывается ряд ограничений, связанных с числом выводов микросхем, наличием встроенных функциональных узлов и устройств.

При использовании ПЛИС ее ресурсов может не хватить для погружения всей системы. При этом возникает необходимость применения нескольких корпусов микросхем, связанных между собою внешними линиями связи. При параллельной передаче информации между микросхемами возникают проблемы, связанные с возможной нехваткой выводов ПЛИС. Кроме того, снижается надежность системы, так как контактные соединения между микросхемами относятся к ненадежным элементам системы. Возрастает также энергопотребление системы и увеличиваются габаритные размеры.

Учитывая важность указанных проблем, компания Virtual Machine Works разработала технологию под названием VirtualWire (виртуальные соединения), представленную как технология производства больших устройств, для реализации которых приходится использовать несколько микросхем [5]. Поскольку определенное количество внутренних ресурсов не используется в связи с отсутствием для дополнительной аппаратуры выводов микросхем, этот ресурс можно использовать для реализации последовательной передачи разрядов данных с помощью мультиплексирования. Данная технология, хотя и может в определенной степени решить проблему нехватки

выводов микросхем, но создает задержки продвижения потоков данных, что противоречит самой идее потоковой модели вычислений.

Одним из эффективных подходов к решению проблемы сокращения связей между компонентами системы является применение неавтономных методов выполнения операций, основанных на поразрядной передаче информации между ВМ. При таком режиме обработки данных, кроме сокращения числа связей, появляется возможность выполнять зависимые по данным операции в режиме частичного совмещения. При определенных условиях такой режим вычислений создает предпосылки к сокращению времени выполнения зависимых по данным операций.

В ГПД с последовательной и последовательно-параллельной структурой операции, лежащие на одной ветви, распараллелить нельзя, поскольку они зависят по данным (результат предыдущей операции является операндом для следующей). Распараллеливание вычислений на уровне обработки машинных слов в данном случае не представляется возможным.

При любом числе ВМ, работающих с параллельными кодами операндов, время вычислений будет составлять $T = \sum_j t_j$, где $j$ – индекс операции, лежащей на критическом пути в графе алгоритма, а $t_j$ – время выполнения $j$-й операции. В общем случае $T$ не определяет полное время реализации алгоритма, так как здесь не учитывается время обмена информацией между устройствами, которое существенно зависит от топологии системы.

Таким образом, если в вычислительных системах для выполнения операций используются методы параллельной машинной арифметики, то реализацию последовательности операций нельзя осуществить за время, меньшее суммарного времени выполнения всех операций.

Частичное совмещение во времени выполнения зависимых операций может быть достигнуто за счет применения методов машинной арифметики, позволяющих выполнять операции в неавтономном режиме с использованием избыточных систем счисления с естественным порядком весов. Способы построения таких вычислительных систем известны [4, 6, 7]. В их состав входят квазипараллельные ВМ, позволяющие совмещать выполнение зависимых операций на уровне обработки разрядов слов.

На каждом шаге в ВМ вводится по одному разряду операндов в системе счисления с основанием $k = 2^s$ $(s = 1,2,3...)$ и формируется один разряд результата. При этом разряд промежуточного результата, полученный на $i$-м шаге в одном ВМ при выполнении $j$-й операции, может быть использован на $(i+1)$-м шаге в другом ВМ при выполнении $(j+1)$-й операции. При таком режиме вычислений выполнение следующей операции будет начинаться не после завершения выполнения предыдущей операции, а сразу же после получения первого разряда результата этой операции. Режим работы таких ВМ называют неавтономным, так как для выполнения последовательности операций необходимо несколько ВМ, которые совместно выполняют цепочку операций, обмениваясь информацией в процессе работы.

Такие ВМ по структуре ближе к параллельным, а не последовательным устройствам, что определило их название «квазипараллельные». С использованием квазипараллельных ВМ реализуется параллелизм на уровне обработки разрядов операндов.

Разряды результата выдаются со старших разрядов, причем, первый разряд формируются с задержкой на $p$ шагов. Следовательно, число шагов, необходимое для получения $n$ старших разрядов окончательного результата при выполнении цепочки из $K$ операций, составляет

$$N = n - 1 + \sum_j^K \left(p_j + 1\right),$$

где $j$ – индекс операции, лежащей на критическом пути в графе алгоритма, а $p_j$.- задержка формирования результата при выполнении

$j$-й операции. Длительность цикла (шага) вычислений в синхронном режиме должна соответствовать условию

$$T_{\text{ц}} \geq \max_j T_{\text{ц}j}$$

где $T_{\text{ц}j}$ – длительность цикла формирования разряда результата при выполнении $j$-й операции. Тогда время выполнения цепочки операций будет определяться как

$$T = \left[ n - 1 + \sum_{j=1}^{K} \left(p_j + 1\right) \right] T_{\text{ц}}.$$

Для сравнения времени выполнения операций в системах с параллельными и квазипараллельными ВМ необходимо определить значения рассмотренных выше параметров $t_j$,

$p_j$ и $T_{\text{ц}}$ для конкретной структурной организации модулей. Заметим, что эффективным способом ускорения неавтономных вычислений является увеличение величины основания системы счисления. Например, переход от основания $k$=2 к основанию $k$=4 сокращает число шагов вычислений примерно в 2 раза, а к основанию $k$=8 – в 4 раза.

## Обоснование метода неавтономных вычислений

Методы неавтономной арифметики в основном рассмотрены для симметричных избыточных систем счисления, которые позволяют реализовать функции, как с положительными, так и с отрицательными частными производными [8, 9]. Однако на практике применяются вычислительные методы, основанные на реализации функций с положительными частными производными. Это имеет место, например, при численном интегрировании, цифровой обработки сигналов, вычислении полиномов. В случае положительных аргументов вычисления могут производиться в смещенных системах счисления, что позволяет упростить квазипараллельные ВМ [7].

Ниже рассматривается реализация на ПЛИС устройств для вычисления полиномов, которые можно рассматривать как фрагменты ПСНС. Рассмотрены варианты построения устройств на квазипараллельных и параллельных ВМ.

Для вычисления полиномов воспользуемся методом Горнера первого порядка. Вычислитель представляет собой цепочку ВМ, каждый из которых выполняет промежуточную операцию $F = XY + A$ (рис. 1б).

Получим алгоритм выполнения операции в неавтономном режиме с использованием смещенной позиционной избыточной системы счисления с естественным порядком весов и основанием $k$.

Будем считать, что операнды являются дробными $n$-разрядными числами и вводятся в ВМ со старших разрядов. Результат также формируется со старших разрядов с запаздыванием на $p$ шагов, то есть в ВМ выполняется операция $Z = 2^p(XY + A)$. Операнды $X, Y, A$ можно записать в виде:

$$X = \sum_{i-1}^{n} x_i k^{-i},\ Y = \sum_{i-1}^{n} y_i k^{-i},\ A = \sum_{i-1}^{n} a_i k^{-i}, \quad (1)$$

где $x_i, y_i, a_i \in \{\overline{0,q}\}$ - цифры операндов.

Естественно, что для получения $n$ разрядов функции $F$ необходимо сформировать $m = n + p$ разрядов $Z$ в виде

$$Z = \sum_{i+1}^{m} z_i k^{-i}, \quad (2)$$

где $z_i \in \{\overline{0,q}\}$ - цифры результата.

Коды, содержащие только $i$ старших разрядов, обозначим через $Z_i, X_i, Y_i, A_i$.

После выполнения $m$ шагов можно получить результат $Z_m = Z$ с погрешностью, не превышающей $k^{-m}$, если на каждом $i$-м шаге цифру $z_i$ выбирать таким образом, чтобы выполнялось условие

$$Z_i \le k^{-p}(X_i Y_i + A_i) < Z_i + k^{-i}. \quad (3)$$

Используя методику [7] и формулы (1), (2), (3), можно получить выражение для промежуточной переменной на $i$-м шаге выполнения операции в виде

$$H_i = kR_{i-1} + k^{-p} X_{i-1} y_i + k^{-p} Y_{i-1} x_i + k^{-p-i} y_i x_i + k^{-p} a_i,$$

на основании которой формируется значение цифры результата $z_i$ и очередной остаток $R_i$ для следующего шага в виде:

$$z_i = \text{ent}\, H_i,\ R_i = \text{rest}\, H_i. \quad (4)$$

Начальными являются значения

$$X_0 = Y_0 = R_0 = 0.$$

Переменная $H_i$ вычисляется за один такт. В отличие от симметричных систем счисления в данном случае не требуется выполнять анализ диапазона изменения значения $H_i$ с помощью логической схемы [8]. Цифра результата формируется автоматически в соответствии с (4) как целая часть $H_i$. За счет этого уменьшается оборудование ВМ и сокращается время формирования результата.

## Реализация вычислителей на ПЛИС

Проведено моделирование устройства для вычисления полинома пятой степени на базе ПЛИС EP3SL340F1760C2 семейства Stratix III фирмы Altera.

Для разработки системы использовалась среда проектирования Quartus II фирмы Altera. Результаты погружения в ПЛИС разработанного ВМ для обработки 64-разрядных операн-

дов (в двоичном эквиваленте) представлены на рис. 2 и в табл. 1.

Описание ВМ выполнено средствами среды Quartus II. В табл. 1 через разделитель показан соответственно используемый для построения вычислителя и общий ресурс ПЛИС определенного вида.

ВМ работают в смещенной системе счисления с основанием $k = 4$, что требует три проводника для передачи одной цифры между модулями. Повышение основания системы счисления позволило в два раза сократить число тактов обработки операндов по сравнению с двоичной системой. В состав ВМ входят блоки для формирования кодов переменных, входящих в правую часть выражения (3) и блок для суммирования этих кодов.

***Табл. 1. Используемые ресурсы ПЛИС для одного ВМ***

| Логические ячейки | Регистры | Выводы |
|---|---|---|
| 1 214/270 400 (<1%) | 235/270 400 (<1%) | 15/1120 (1%) |

Для вычисления полиномов пятой степени по схеме Горнера первого порядка построено вычислительное устройство на основе пяти ВМ (рис. 2). Вычислитель включает пять блоков умножения с накоплением MULADD64 и четыре блока задержки DELAY3CLK. Наличие блоков задержки связано с тем, что каждый ВМ формирует результат с задержкой. Необходимо чтобы значение $X$ поступало на каждый ВМ одновременно с результатом из предыдущего ВМ. Схема также содержит вход синхросигнала CLK, вход сброса RESET и вход разрешения вычислений ENABLE. Для обеспечения начала работы каждого ВМ в нужный момент сигнал ENABLE также проходит через блоки задержки.

***Табл. 2. Используемые ресурсы ПЛИС для схемы вычисления полиномов***

| Логические ячейки | Регистры | Выводы |
|---|---|---|
| 6 070/270400 (2%) | 1 223/270 400 (<1%) | 27/1120 (2,4%) |

Результаты погружения разработанного вычислителя полиномов (табл. 2) показывают, что ресурсы ПЛИС используются весьма экономно, как с точки зрения внутренних элементов (2% ячеек и 1% регистров), так и с точки зрения выводов (2,4%). Оставшийся весьма большой ресурс ПЛИС можно использовать для погружения в микросхему других устройств.

Моделирование вычислителя в среде Quartus II показало, что минимальная длительность одного такта работы, в результате которого формируется одна цифра результата, составляет 8 нс. Учитывая, что каждый ВМ вносит задержку начала формирования разрядов результата после поступления на его входы цифр операндов, первый разряд окончательного результата формируется через 120 нс после начала вычислений.

В дальнейшем в каждом такте формируется очередная цифра результата. Таким образом, полный 64-разрядный результат (в двоичном эквиваленте) может быть получен за 368 нс.

Следует заметить, что в системах автоматического управления во многих случаях управляющее воздействие можно начинать формировать при поступлении первого старшего разряда результата, а затем уточнять при поступлении каждого следующего разряда. Отметим также, что при реализации ПСНС на базе заказных СБИС можно получить большую скорость вычислений. Это объясняется возможность оптимизировать схемы узлов системы с учетом особенностей реализуемых алгоритмов.

Для сравнения разных подходов к реализации вычислителя полиномов (с последовательной и параллельной пересылкой данных между ВМ) разработано устройство на основе параллельных ВМ, информация между которыми передается параллельным кодом. Вычислитель также построен по схеме (рис. 1б) и включает пять ВМ, каждый из которых выполняет операцию $Z = (XY + A)$ в автономном режиме.

Использование ресурсов ПЛИС для такого устройства показано в табл. 3, из которой видно, что затраты ресурсов ПЛИС в данном случае больше, чем для рассмотренного ранее устройства.

***Табл. 3. Используемые ресурсы ПЛИС для построения параллельного устройства вычисления полиномов***

| Логические ячейки | Встроенные блоки умножения | Выводы |
|---|---|---|
| 780/270 400 (<1%) | 80/576 (14%) | 512/1120 (46%) |

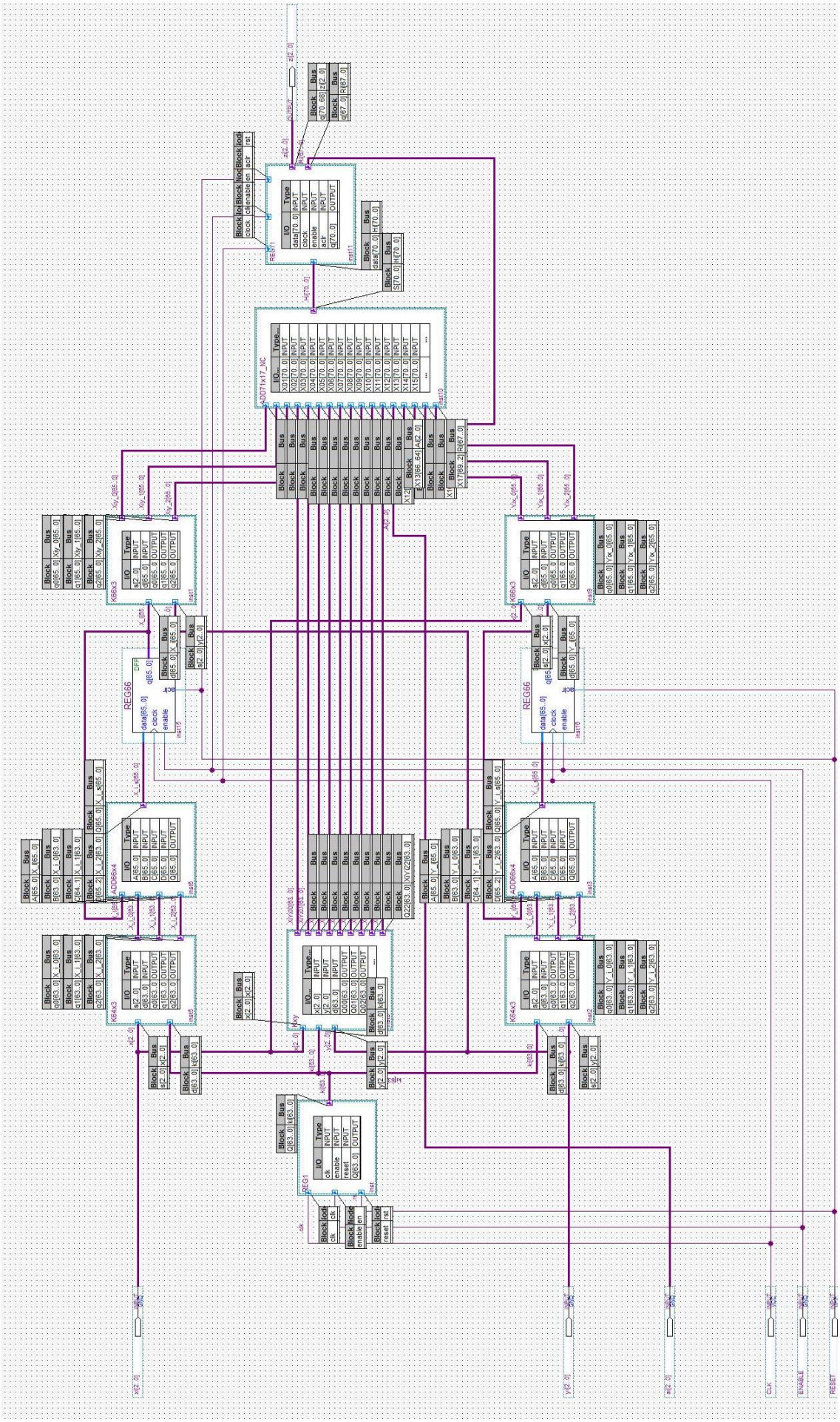

*Рис. 2. Схема квазипараллельного ВМ для выполнения одного звена вычислений по схеме Горнера*

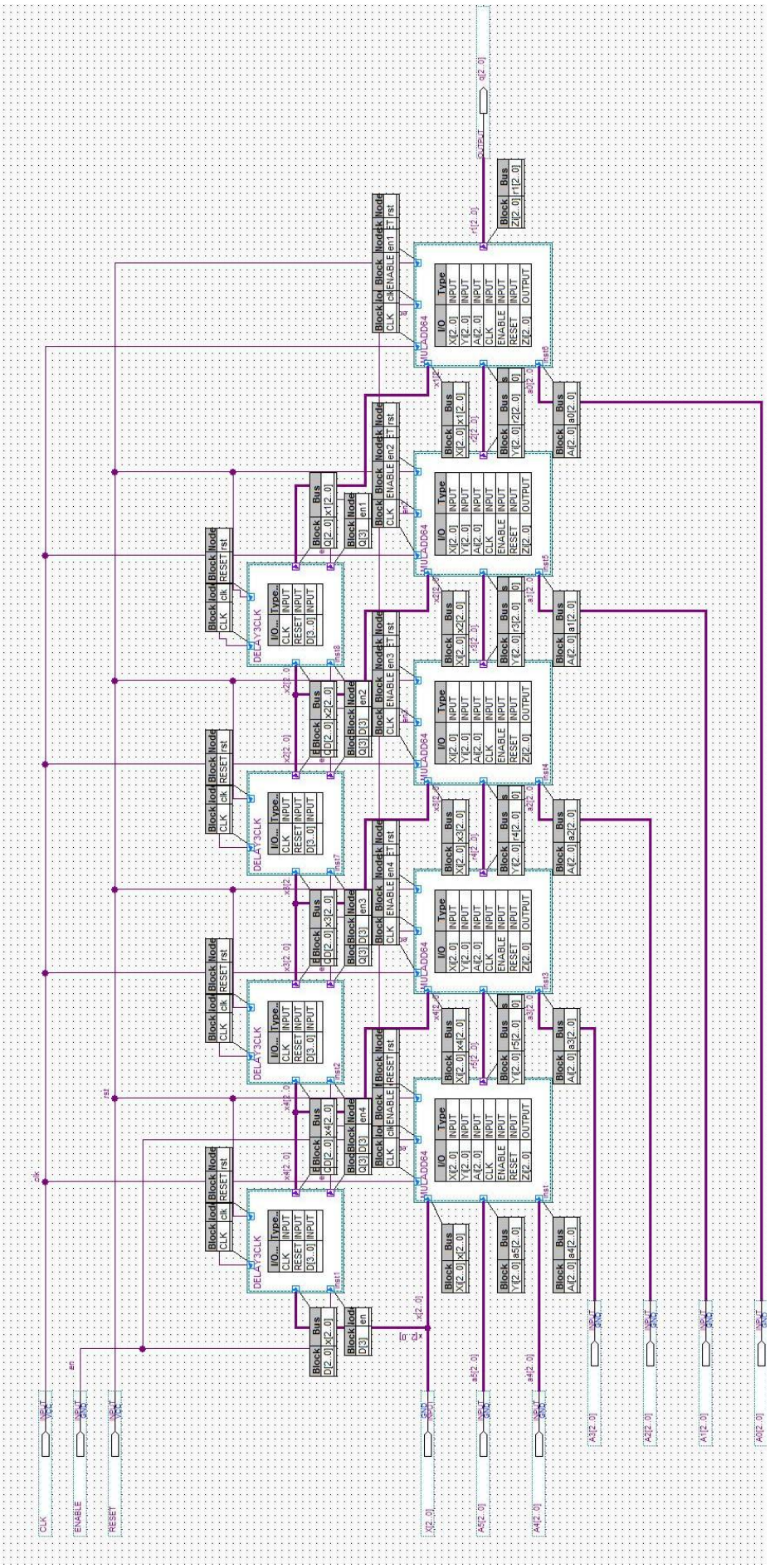

***Рис. 3. Схема погружаемого в ПЛИС вычислителя полиномов***

Работа данного устройства была промоделирована в среде Quartus II. В результате моделирования определено, что задержка формирования результата в блоке равна 67 нс при одновременном вводе в устройство всех 64-разрядных операндов. Учитывая, что при данной организации вычислений практически половина выводов ПЛИС используется только для вычислителя полиномов, применение такого режима работы устройства вряд ли целесообразно. Более реальным является режим предварительного ввода в определенном порядке 64-разрядных операндов с последующей их обработкой в параллельном устройстве. При таком режиме ввода вряд ли может быть получен существенный выигрыш во времени получения результата по сравнению с устройством на квазипараллельных ВМ. Более точный анализ временных характеристик можно произвести с учетом конкретной реализации режима ввода операндов и вычисления результата.

## Заключение

Вычислитель на базе квазипаралельных ВМ с поразрядной передачей данных использует существенно меньше ресурсов ПЛИС. Экономятся как внутренние ресурсы ПЛИС (более чем в 7 раз для рассмотренного примера), так и ее выводы (более чем в 14 раз). Это дает возможность реализовать на той же микросхеме ряд других устройств, относящихся к одной или разным системам.

Построение системы на одной ПЛИС обеспечивает повышение ее надежности, уменьшение энергопотребления и габаритов, а также дает потенциальную возможность повысить частоту тактирования, что, в свою очередь, ускоряет обработку информации.

Таким образом, полученные результаты подтверждают эффективность применения неавтономных методов поразрядной обработки информации со старших разрядов в системах типа ПСНС на базе программируемых и заказных СБИС.

## Список литературы


1. Байков В.Д. Решение траекторных задач в микропроцессорных системах ЧПУ / В.Д.Байков, С.Н.Вашкевич. – Л.: Машиностроение, 1986, 105 с.
2. Палагин А.В. Реконфигурируемые вычислительные системы: Основы и приложения / А.В.Палагин, В.Н.Опанасенко. – К.: Просвіта, 2006, 280 с.
3. Каляев И.А. Архитектура семейства реконфигурируемых вычислительных систем на основе ПЛИС / И.А. Каляев, И.И. Левин, Е.А. Семерников // Искусственный интеллект. – 2008. - № 3. – с. 663-674.
4. Жабин В.И. Построение быстродействующих специализированных вычислителей для реализации многоместных выражений / В.И.Жабин, В.И.Корнейчук, В.П.Тарасенко // Автоматика и вычислительная техника. – 1981. - №6. – с. 18-22.
5. Максфилд К. Проектирование на ПЛИС. Архитектура, средства и методы / К.Максфилд. – М.: Издательский дом «Додэка-XXI», 2007, 408 с.
6. Жабин В.И. Выполнение последовательностей зависимых операций в режиме совмещения / В.И.Жабин. // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. Наук. Пр. – К.: Век+. – 2007. - №46. – С. 226-233.
7. Дичка И.А. Совмещение зависимых операций на уровне обработки разрядов операндов / И.А.Дичка, В.В.Жабина. // Искусственный интеллект. – 2008. - №3. – С. 649-654.
8. Жабин В.И. Некоторые машинные методы вычисления рациональных функций многих аргументов / В.И. Жабин, В.И.Корнейчук, В.П.Тарасенко // Автоматика и телемеханика. – 1977. - №12. – С. 145-154.
9. Жабин В.И. Методы быстрого неавтономного воспроизведения функций / В.И.Жабин, В.И.Корнейчук, В.П.Тарасенко // Управляющие системы и машины. – 1977. - №3. – С. 96-101.