УДК 004.272.43
МРНТИ 50.33.03

https://doi.org/10.55452/1998-6688-2026-23-3-188-203

[1]Тынымбаев С.,
профессор, ORCID ID: 0000-0002-9326-9476,
e-mail: s.tynymbayev@iitu.edu.kz
[1*]Маманова С.Е.,
докторант, ORCID ID: 0009-0001-7277-5492,
e-mail: s.mamanova@iitu.edu.kz
[2]Сқабылов А.А.,
PhD, ORCID ID: 0000-0002-5196-8252,
e-mail: Alisher.skabylov@kaznu.edu.kz
[1]Чинибаева Т.Т.,
PhD, ORCID ID: 0000-0002-2657-3697,
e-mail: t.temirbolatova@iitu.edu.kz
[2]Жексебай Д.М.,
PhD, ORCID ID: 0009-0008-1884-4662,
e-mail: zhexebay92@gmail.com

[1]Международный университет информационных технологий, г. Алматы, Казахстан
[2]Казахский национальный университет им. аль-Фараби, г. Алматы, Казахстан

АРХИТЕКТУРА ЦЕЛОЧИСЛЕННОГО ДЕЛИТЕЛЯ С ОДНОВРЕМЕННЫМ ФОРМИРОВАНИЕМ ТРЕХ РАЗРЯДОВ ЧАСТНОГО ЗА ШАГ

Аннотация

Операция целочисленного деления является одной из наиболее ресурсоёмких арифметических операций в цифровых вычислительных системах. Традиционные последовательные делители формируют один разряд частного за такт, что требует n тактов для n-разрядного деления. В данной работе предлагается архитектура последовательного целочисленного делителя, обеспечивающая одновременное формирование трех разрядов частного за один тактовый цикл (что соответствует основанию radix-8). Устройство включает регистр делимого (РгА), формирователь кратных делителю (БФКД), формирователь частичных остатков и разрядов частного (ФЧОиРЧ) и блок синхронизации (БС). В отличие от классических SRT-делителей с избыточным представлением и таблицами выбора цифры частного, предложенная архитектура использует точное сравнение частичного остатка с предвычисленными кратными делителю B–7B, что устраняет рост таблицы выбора при увеличении основания. Архитектура описана на языке Verilog HDL и верифицирована на ПЛИС Xilinx Artix-7 в среде Vivado. Прототип для 12-разрядного делимого и 6-разрядного делителя занимает 158 LUT и 50 триггеров, достигает максимальной тактовой частоты 134,6 МГц и подтвержден комплексом из десяти тестов; деление выполняется за два тактовых цикла. Архитектура ориентирована на применение в высокоскоростных цифровых вычислителях и специализированных процессорах.

Ключевые слова: целочисленное деление, кратные делителю, формирователь остатков, формирователь разрядов частного, последовательный делитель, radix-8, ПЛИС, Verilog HDL.

Введение

Деление в цифровых вычислительных системах традиционно считается одной из наиболее медленных и сложных операций, особенно по сравнению со сложением и умножением [1]. Несмотря на относительно невысокую частоту использования (около 3% всех арифметических инструкций), операции деления могут генерировать до 40% задержек конвейера процессора [1]. В высокопроизводительных вычислениях стремительное ускорение сложения и

умножения обнажило узкое место: классические алгоритмы деления обладают итеративной природой, где каждый следующий разряд частного вычисляется на основе предыдущего шага, что приводит к повышенной длительности вычислений. Это мотивирует активные исследования в области ускорения деления, направленные на снижение задержек и увеличение пропускной способности делителей при приемлемых затратах аппаратуры.

Наиболее перспективными подходами стали алгоритмы с поразрядным восстановлением частного (digit-recurrence), в частности семейство алгоритмов SRT, названное по именам Суини, Робертсона и Точера [2]. Алгоритмы SRT выполняют деление с вычислением частного итерационно по разрядам, используя избыточное представление разрядов частного [3]. Такое кодирование частного позволяет в каждом шаге оперировать приближенным остатком и делителем, вычисляя следующий разряд частного с учетом допускаемых отклонений остатка. Классический SRT с основанием 2 (radix-2) эквивалентен по скорости не восстанавливающему алгоритму и дает один бит частного за итерацию [3]. Тем не менее SRT-методы допускают увеличение основания системы счисления для генерации сразу нескольких бит частного за такт, сокращая тем самым число итераций.

Увеличение основания radix-4 позволяет формировать два разряда частного за такт, вдвое сокращая число итераций по сравнению с radix-2 [4, 5]. Однако дальнейшее увеличение основания сопряжено с ростом аппаратной сложности: таблица выбора разрядов частного существенно усложняется, а число кратных делителя, требующих хранения и сравнения, возрастает [6, 7]. Реализации на основе radix-8 и выше исследовались преимущественно в контексте делителей с плавающей точкой [8, 9], тогда как архитектуры последовательных целочисленных делителей с основанием radix-8 остаются недостаточно изученными с точки зрения практической функциональной организации.

В данной работе предлагается архитектура последовательного целочисленного делителя, обеспечивающая одновременное формирование трех разрядов частного за один тактовый цикл. Основной вклад работы заключается в разработке функциональной организации устройства, включающей компараторы для выбора кратного делителю из набора B–7B, схему формирования частичных остатков и механизм синхронизации на основе RS-триггера и вычитающего счетчика. Предложенная архитектура обеспечивает троекратное сокращение числа итераций по сравнению с классическим побитовым алгоритмом при сохранении регулярной структуры, пригодной для реализации в цифровых вычислительных системах.

Материалы и методы

Увеличение основания SRT-алгоритма прямо пропорционально снижает количество итераций: так, при переходе от radix-2 к radix-4 каждая итерация дает два бита частного, а при radix-8 – три бита [10]. Например, алгоритм SRT с основанием 8 возвращает за шаг сразу три битовых разряда частного (эквивалентно одной восьмеричной цифре), что теоретически втрое сокращает требуемое число итераций по сравнению с традиционным двоичным делением. Аналогично, radix-16 позволяет получать четыре бита частного за такт. Экспериментально подтверждено, что рост основания до 8 или 16 способен уменьшить латентность деления в 3–4 раза [11].

Однако агрессивное увеличение радикса сопровождается резким усложнением логики выбора разрядов частного. Алгоритму с основанием 16, к примеру, необходимо различать не менее 32 перекрывающихся интервалов остатка, каждому из которых соответствует уникальная управляющая комбинация для выбора следующей цифры частного [11]. Это означает необходимость хранения или вычисления обширных таблиц выбора, объем которых экспоненциально растет с увеличением радикса [12]. Фактически, каждый возможный остаток должен быть соотнесен с правильным значением следующего разряда частного, что при высоком радиксе требует учета множества случаев. Уже при переходе от radix-4 к radix-8 объем таблицы выбора возрастает многократно, делая реализацию на аппаратных средствах крайне ресурсоемкой (вплоть до сотен комбинаций условий) [12, 13].

Практические исследования показывают, что выигрыш от увеличения радикса начинает компенсироваться накладными расходами на реализацию логики выбора. Например, сравнительный анализ делителей с радиксами 4, 8 и 16 продемонстрировал, что рост производительности сопровождается ухудшением показателей площади и энергии: высокие радиксы требуют значительно большего числа логических элементов и таблиц для хранения коэффициентов, что может свести на нет преимущества сокращенного числа итераций [14, 15]. В частности, Наннарелли и Ланг сравнили реализации делителей radix-4, radix-8 и radix-16 и отметили, что хотя радикс-16 теоретически позволяет получать 4 бита за такт, практическая эффективность ограничена из-за громоздкой схемы выбора и связанных с ней задержек распространения сигналов [6]. По этой причине промышленные реализации высокопроизводительных делителей обычно ограничиваются умеренными радиксами (например, 4 или 8), сочетая их с другими методами ускорения. В частности, в контексте встроенных процессоров, жестко ограниченных по аппаратурным ресурсам, зачастую вообще выбирают radix-2, поскольку более высокие основания (radix-4 и выше) слишком сложны для малых устройств [1]. Обзор алгоритмов деления от Обермана и Флинна отмечают, что рост радикса существенно усложняет схему управления делителем [14]. Сходного мнения придерживаются авторы современного обзора Патанкара и Коэла (2021), подчеркивая, что делители с высоким основанием трудно вписать в ограничения по площади и потреблению, особенно в ASIC/FPGA-системах реального времени [16].

Одним из ключевых направлений совершенствования делителей SRT стало упрощение и оптимизация логики выбора разряда частного. Классическая реализация требует хранения обширной таблицы пороговых значений для принятия решения, какой из предвычисленных кратных делителя нужно вычесть из текущего остатка на данном шаге. Улучшенные SRT-алгоритмы радиксов 4, 8 и 16, предлагаемые в литературе, стремятся сократить объем этих таблиц несколькими способами:

1. Предвычисление кратных делителя [10];
2. Использование таблиц поиска (LUT) для выбора разряда [17];
3. Преобразование результата «на лету».

Современные исследования охватывают широкий спектр методов ускорения деления: от комбинированных (гибридных) схем с умеренным радиксом и конвейеризацией до специализированных алгоритмов с адаптивным сокращением итераций [18]. Обзор литературных источников показывает, что алгоритмы с основой 4 и 8 на сегодняшний день представляют оптимальный компромисс между быстродействием и сложностью. Radix-4 делители относительно просты и успешно реализуются даже в ПЛИС с ограниченными ресурсами, однако их латентность по-прежнему линейно растет с разрядностью (на каждый 1–2 бита частного требуется один такт). Radix-8 алгоритмы привлекают возможностью втрое сократить число итераций, но ранее считались слишком «тяжелыми» по логике управления из-за больших LUT для выбора цифр [10]. Лишь немногие работы демонстрировали успешные реализации radix-8 делителей. В частности, деление с основанием 8 исследовалось в контексте операций с плавающей точкой с введением дополнительного фактора избыточности для упрощения выбора цифры частного [18]. Полученные результаты подтвердили работоспособность подхода, однако указали на рост аппаратных затрат при повышении радикса.

Отдельного внимания заслуживают исследования, направленные на разработку аппаратно эффективных делителей для ПЛИС и встроенных систем. Показано, что простые итеративные схемы radix-2 с оптимизированным контроллером способны превзойти по соотношению скорость/затраты более сложные высокорадиксные реализации, особенно с учетом ограничений по числу LUT в FPGA [20]. В обзорных работах отмечается, что SRT-алгоритмы остаются предпочтительными для общего случая, предлагая сбалансированное решение [16], однако их практическая реализация требует тщательной оптимизации – в частности, конвейеризации и сокращения логики выбора разряда частного – для достижения конкурентоспособности с приближенными методами деления [3].

Таблица 1 – Сравнительный анализ алгоритмов деления

Алгоритм	Разрядов за такт	Итераций (n бит)	Сложность логики выбора	Применение
Radix-2 (невосстанавливающий)	1	n	Минимальная	Встроенные системы [16]
Radix-4 SRT	2	n/2	Умеренная	FPGA, ASIC [6, 7]
Radix-8 SRT (плав. точка)	3	n/3	Высокая (большие LUT)	Спец. процессоры [6, 18]
Radix-16 SRT	4	n/4	Очень высокая	Высокопроизв. FPU [8]
Предложенная архитектура	3	n/3	Умеренная (параллельные компараторы B–7B)	Целочисл. делители

Научная новизна и отличие от существующих решений. Ключевое отличие предложенной архитектуры от известных высокорадиксных SRT-делителей состоит в способе выбора цифры частного. В классическом SRT цифра выбирается по приближенному (усеченному) остатку и делителю с помощью таблицы выбора над перекрывающимися интервалами, причем для корректности используется избыточное представление частного; объем такой таблицы быстро растет с основанием. В предлагаемом устройстве используется точное сравнение полного частичного остатка с предвычисленными кратными делителю B, 2B, … 7B, а цифра частного определяется прямым приоритетным декодированием результатов сравнения. Тем самым исключаются избыточное кодирование, перекрывающиеся интервалы и таблица выбора, а их место занимает регулярная решетка компараторов с фиксированной задержкой. Кратные делителя формируются без умножителей – только сдвигами и тремя сумматорами. Платой за это является необходимость полноразрядного сравнения (вместо усеченной оценки в SRT), однако именно это устраняет экспоненциальный рост логики выбора, ограничивавший высокорадиксные реализации. Таким образом, работа заполняет обнаруженный пробел: последовательный делитель достигает производительности radix-8 при регулярной, не требующей таблиц выбора структуре.

Структурная схема устройства приведена на рисунке 1. В ее состав входят регистр делимого РгА, Блок формирователя кратных делителю БФКД, формирователь частичных остатков и разрядов частного ФЧОиРЧ и блок синхронизации БС.

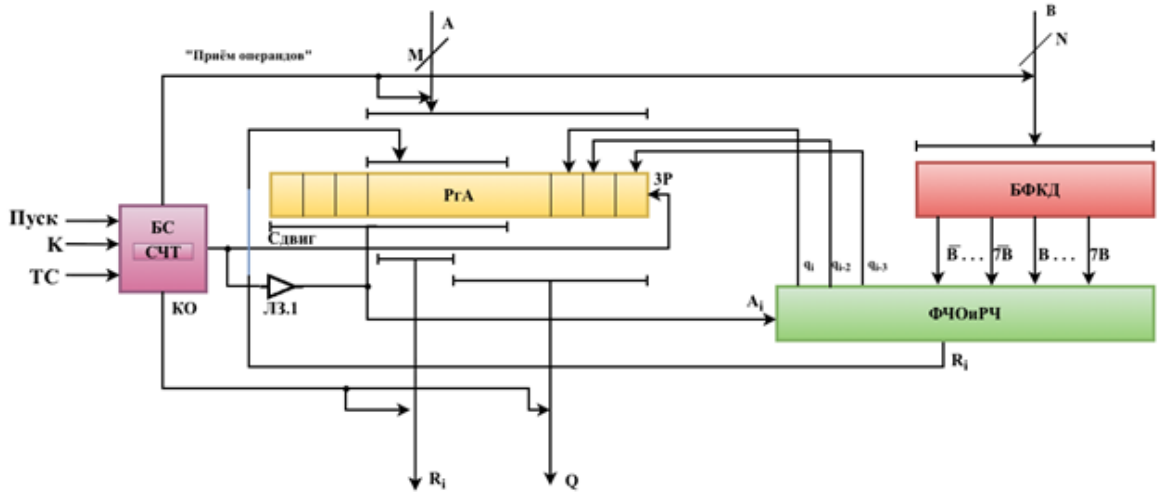

Рисунок 1 – Структурная схема делительного устройства
со сдвигом делителя на три разряда

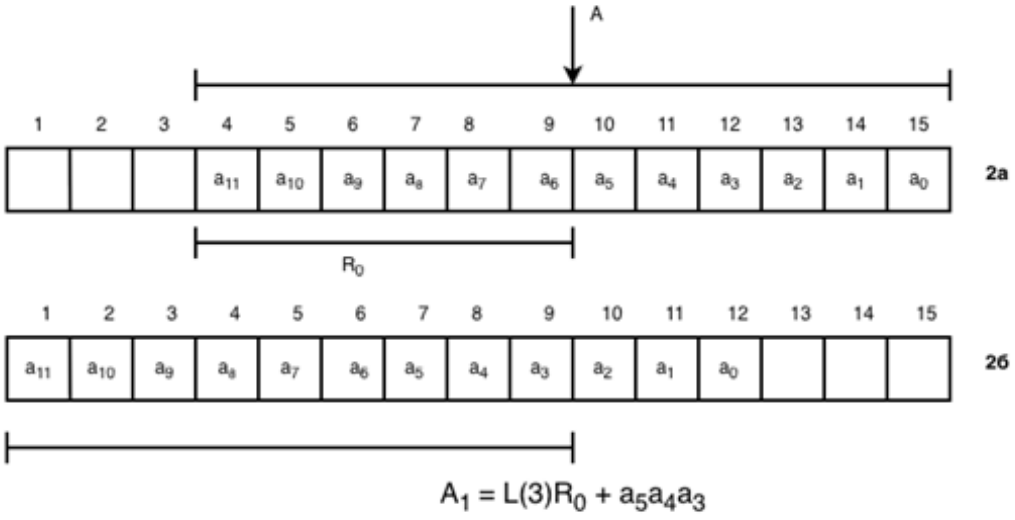

Рисунок 2 – Расположение разрядов делимого А в разрядности регистра РгА

Регистр Рг служит для приема делимого А и имеет цепи сдвига влево на три разряда. На рисунке 2а приведено расположение битов 12-разрядного делимого на разрядной сетке РгА, которые занимают 12 ячеек – с 4 по 15. Старшие три ячейки находятся на нулевом состоянии. Если разрядность делителя N=6, то начальный остаток R_0 определятеся шестью разрядами делимого, т.е.

$$R_0 = a_{11}a_{10}a_9a_8a_7a_6. \quad (1)$$

На рисунке 2б показано расположение разрядов делимого А в разрядной сетке РгА после сдвига РгА на три разряда влево. В разрядах РгА в 1…9 ячейках формируется первый остаток

$$A_1 = L(3)R_0 + a_5a_4a_3, \quad (2)$$

который посылается в ФЧОиРЧ, где сравнивается с кратными числами делителя и по результатом сравнения определяется частичный остаток R_1 и разряды частного $q_1q_2q_3$. При этом R_i принимается в разряды 4…9 , а биты $q_1q_2q_3$ принимаются в разряды РгА с номерами 13…15.

На каждом i-м шаге деления выполняется следующее преобразование:

$$R_i = 8 * R_{i-1} - q_{i-1}B, \quad (3)$$

где R_{i-1} – частичный остаток предыдущего шага, В – делитель, $q_{i-1} \in \{0, 1, 2, 3, 4, 5, 6, 7\}$ – трехразрядное значение частного, определяемое по результатам параллельного сравнения текущего частичного остатка с предвычисленными кратными делителя. Сдвиг на три разряда влево соответствует умножению на 8, что соответствует основанию radix-8 алгоритма.

В формирователе кратных делителя БФКД вычисляются числа, кратные к делителю В. На рисунке 3 приведена схема формирователя чисел В, 2В, … , 7В и их инверсий. Как видно из рисунка 3, формирователь состоит из регистра для хранения делимого, с выхода которого получаем числа В и $\bar{B}$. Кратные 2В, $2\bar{B}$ формируются со сдвига на один разряд чисел В и $\bar{B}$, а числа 4В и $4\bar{B}$ формируются со сдвига В и $\bar{B}$ на два разряда.

Значение трехразрядного частного q_i на каждом шаге определяется путем параллельного сравнения текущего частичного остатка A_i с предвычисленными кратными делителя по следующему правилу:

если $A_i \geq 7B$, то $q_i = 7$;
если $6B \leq A_i < 7B$, то $q_i = 6$;
если $5B \leq A_i < 6B$, то $q_i = 5$;
если $4B \leq A_i < 5B$, то $q_i = 4$;
если $3B \leq A_i < 4B$, то $q_i = 3$;
если $2B \leq A_i < 3B$, то $q_i = 2$;
если $B \leq A_i < 2B$, то $q_i = 1$;
если $A_i < B$, то $q_i = 0$.

Данное правило реализуется посредством пяти параллельных компараторов COMP-1... COMP-5, результаты которых поступают на логические элементы ФЧОиРЧ для одновременного формирования трех разрядов частного q_i, q_{i-1}, q_{i-2}.

Логика выбора разряда частного реализована в виде комбинационной схемы, функционально эквивалентной таблице поиска (LUT) с 3-битным выходом, принимающим значения от 0 до 7. Входными сигналами схемы служат результаты сравнений A_i с кратными B, 3B, 5B и 7B, формируемыми блоком БФКД. Использование параллельных компараторов вместо последовательного перебора позволяет определить значение q_i за время одного такта независимо от разрядности операндов, что обеспечивает постоянную задержку на шаге выбора разряда частного.

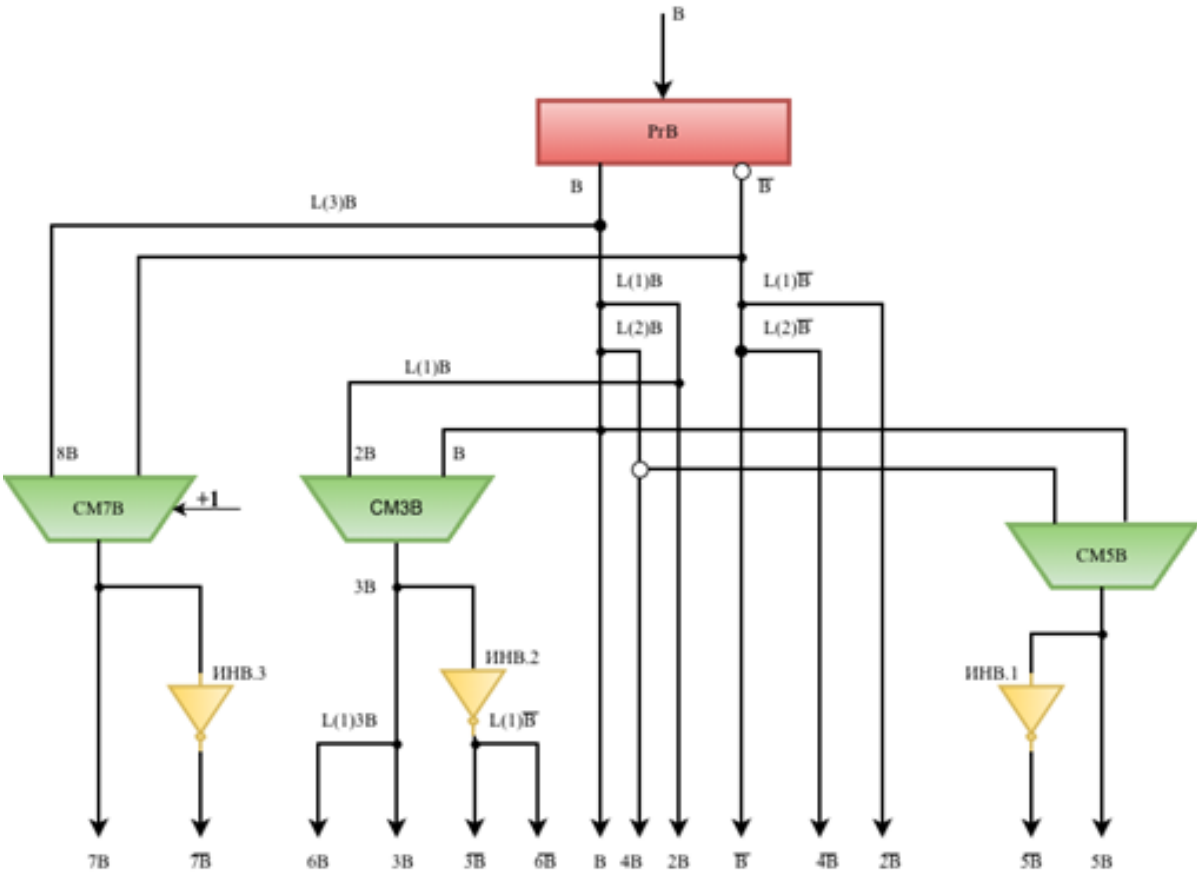

Рисунок 3 – Блок формирователя кратных чисел к делителю (БФКД)

Кратные числа 5B формируем на сумматоре СМ5B путем сложения чисел B и 4B, значение $5\overline{B}$ формируется путем инверсий B на блоке инверторов инв.1. Аналогично значение 3B

формируется на выходе сумматора СМ3В путем суммирование чисел 2В и В. Инвертируя 3В по блоке инверторов инв.2 значения числа 7В вычисляется по формуле

$$7B = 8B + \bar{B} + 1 \tag{4}$$

на сумматоре СМ7В.

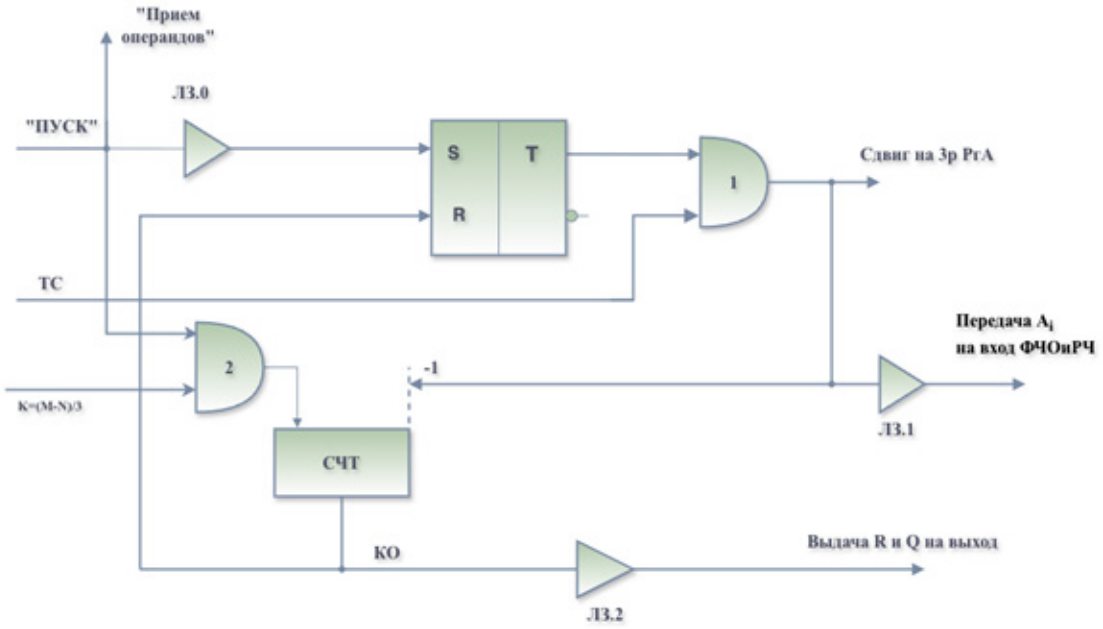

Рисунок 4 – Функциональная схема блока синхронизаций (БС)

Блок синхронизации БС управляет последовательностью выполнения операции деления. Функциональная схема БС представлена на рисунке 4. В состав блока входят RS-триггер, схемы И1 и И2, вычитающий счетчик СЧТ и линии задержки ЛЗ.0, ЛЗ.1 и ЛЗ.2.

На входы БС поступают сигнал запуска «Пуск», тактовый сигнал «ТС» и двоичный код числа шагов деления *K*. На выходах БС формируются управляющие сигналы «Прием операндов», «Сдвиг» и «КО».

По сигналу «Пуск» RS-триггер переходит в единичное состояние, разрешая работу тактового генератора через схему И1. Счетчик СЧТ загружается значением K и начинает обратный отсчет по каждому тактовому импульсу. На каждом такте сигнал «Сдвиг» через линию задержки ЛЗ.1 инициирует сдвиг регистра РгА на три разряда и передачу A_i на вход ФЧОиРЧ. По достижении счетчиком нулевого значения сигнал КО сбрасывает RS-триггер и через линию ЛЗ.2 выдает результат – частное Q и остаток R – на выход устройства.

Центральным блоком делителя является формирователь частичных остатков и разрядов частного, где на каждом шаге деления вычисляются частичные остатки R_i и биты частного $q_i q_{i-1} q_{i-2}$.

Функциональная схема ФЧОиРЧ представлена на рисунке 5. Блок реализует двухэтапное параллельное сравнение частичного остатка A_i с предвычисленными кратными делителя и включает пять компараторов COMP-1...COMP-5, двоичный сумматор и комбинационную логику формирования разрядов частного.

Сравниваемое значение A_i формируется сдвигом предыдущего остатка R_{i-1} на три разряда влево с присоединением очередных трех разрядов делимого $a_i a_{i-1} a_{i-2}$:

$$A_i = L(3)R_{i-1} + a_i a_{i-1} a_{i-2} \tag{5}$$

Полученное значение A_i подается на левые входы всех пяти компараторов COMP-1... COMP-5 и на левый вход сумматора.

Сравнение осуществляется в два этапа с целью минимизации числа компараторов. На первом этапе компаратор COMP-5 сравнивает A_i с порогом 5B. При выполнении условия A_i < 5B на выходе «1» COMP-5 формируется сигнал логической единицы, который через комбинационную логику подает на правые входы компараторов COMP-1...COMP-4 значения B, 2B, 3B и 4B соответственно. На втором этапе, при выполнении условия $A_i \geq$ 5B, на выходе «2» COMP-5 формируется сигнал логической единицы, по которому на правые входы компараторов COMP-1...COMP-3 подаются значения 5B, 6B и 7B. Результаты сравнений поступают на комбинационную логику, которая одновременно формирует три разряда частного $q_i q_{i-1} q_{i-2}$ на выходах схем ИЛИ2, ИЛИ3 и ИЛИ4.

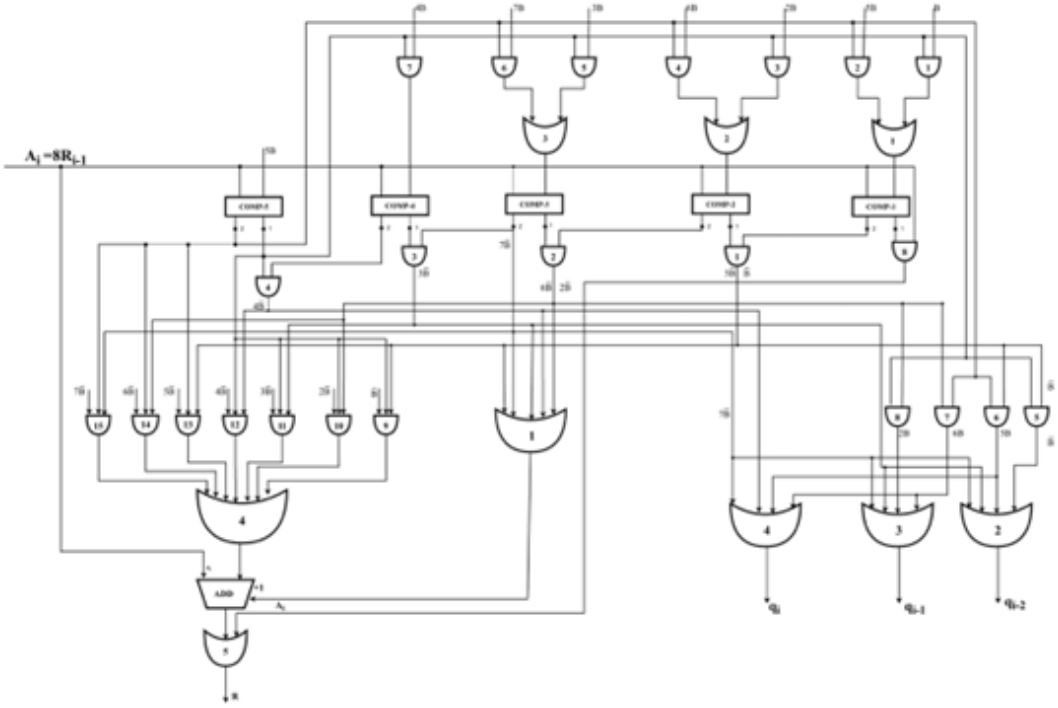

Рисунок 5 – Функциональная схема формирователя частичных остатков и трех разрядов частного

На втором этапе все числа B, 2B, 3B, ... 7B сравниваются с числом A_i на COMP-1, A_i COMP-4 параллельно. Рассмотрим сравнение числа A_iс числами B, 2B, 3B и 4B. При этом <5B и на выходе «1», COMP-5 вырабатывается уровень «1».

При сравнении A_i с числом B: если A_i < B, то на выходе «1», COMP-1 вырабатывается сигнал «1», который разрешает передачу числа A_i на выход схемы И8, и число A_i через схему ИЛИ5 передается на выход. При этом $R_i = A_i$.

При сравнении числа A_i с числом 2B схемы COMP-2: если A_i <2B, то единичный сигнал с выхода «1» COMP-2 передается на вход схемы И1 и на ее выходе вырабатывается единичный сигнал (2B > $A_i \geq$ B), который подается на вход ИЛИ (на выходе формируется сигнал «+1») и на вход блока схемы И9, разрешая передачу числа $\bar{B}$ через схему ИЛИ4 на правый вход сумматора. При этом вычисляется остаток R_i путем выполнения операции:

$$R_i = A_i + \bar{B} + 1 \quad (6)$$

При $A_i \geq$ 2В, единичный сигнал с выхода 2 COMP-2 передается на вход схемы И2. При сравнении A_i с 3В, если $A_i <$ 3В, то из выхода «1» COMP-3 единичный сигнал подается на вход схемы И2, и на выходе схемы И2 вырабатывается единичный сигнал (3В > $A_i \geq$ 2В). Единичный сигнал с выхода схемы И2 вырабатывает сигнал «+1» на выходе схемы ИЛИ1 и разрешает прохождение кода числа $2\bar{B}$ на выход И10 и позволяет выполнение операции

$$R_i = A_i + 2\bar{B} + 1. \quad (7)$$

При выполнении условия $A_i \geq$ 3В единичный сигнал с выхода «2» COMP-3 подается на вход схемы И3, на второй вход схемы И3 подается единичный сигнал с выхода 1 COMP-4, если выполняется условие $A_i <$ 4В. Таким образом, при одновременном выполнении условий 4В > $A_i \geq$ 3В на выходе схемы И3 вырабатывается единичный сигнал, который на выходе схемы ИЛИ1 вырабатывает сигнал «+1» и разрешает передачу обратного кода числа $3\bar{B}$ передать с выходов блока схем И11 и ИЛИ4 на правые входы сумматора. При этом выполняется операция по вычислению последующего остатка:

$$R_i = A_i + 3\bar{B} + 1. \quad (8)$$

При сравнении числа A_iс числом 4В, если выполняется условие $A_i \geq$4В, то с выходов «2» COMP-4 единичный сигнал поступает на вход И4, а на второй вход единичный сигнал поступает с выхода «1». Единичный сигнал с выхода И4 генерирует сигнал «+1» на выходе ИЛИ1 и разрешает прохождение числа $4\bar{B}$ на вход сумматора ADD. При этом выполняется операция

$$R_i = A_i + 4\bar{B} + 1. \quad (9)$$

Аналогично сравниваются числа 5В, 6В и 7В с числом A_i При этом на схеме COMP-5 сравнивается число A_i с числом 5В. При условии $A_i \geq$ 5В на выходе «2» вырабатывается единичный сигнал, по которому число 5В подается на вход COMP-1, число 6В подается на вход COMP-2, и число 7В поступает на вход COMP-4, где они параллельно сравниваются с числом A_i.

При условии 6В > $A_i \geq$ 5В на выходе схемы И1 вырабатывается сигнал, который приводит к выполнению операций

$$R_i = A_i + 5\bar{B} + 1. \quad (10)$$

При выполнении условий 7В>$A_i \leq$ 6В на выходе схемы И2 вырабатывается сигнал, который приводит к выполнению операции

$$R_i = A_i + 6\bar{B} + 1. \quad (11)$$

При выполнении условия $A_i \geq$ 7В единичный сигнал с выхода «2» COMP-3 приводит к выполнению операции

$$R_i = A_i + 7\bar{B} + 1 \quad (12)$$

Логическими схемами И5…И8 и схемами ИЛИ2…ИЛИ4 формируются очередные биты частного. При подаче единичного сигнала из выхода схемы И1, а также единичного сигнала из выхода «1» COMP-5 на входы схемы И5, на ее выходе формируется сигнал «1», который подается на вход ИЛИ2. При этом на выходе ИЛИ2 формируется бит q_{i-2}).

При подаче единичного сигнала с выхода И1 на вход И6 при наличии сигнала «1» с выхода «2» COMP-5, единичный сигнал на выходе И6 принимает формирование битов q_i и q_{i-2} (101).

Единичный сигнал из выхода И2 совместно с сигналом «1» из выхода COMP-2 формирует на выходе И7 в виде единичного сигнала, который участвует в формировании битов на выходах И4 и И3 (110).

Единичный сигнал с выхода «2» COMP-3 поступает на входы ИЛИ4, ИЛИ3, ИЛИ2, на их выходах формируется двоичный код q_i, q_{i-1}, q_{i-2}(111).

Результаты и обсуждение

Рассмотрим пример деления числа $A = 1387_{10}$ на делитель $B = 27_{10}$. Двоичные коды А и В предоставляем ниже.

$$A = 1387_{10} = \begin{matrix} a_{11} & a_{10} & a_9 & a_8 & a_7 & a_6 & a_5 a_4 & a_3 a_2 & a_1 & a_0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1\ 0 & 1\ 0 & 1 & 1 \end{matrix} \quad (13)$$

$$B = 011011_2 = 27_{10}. \quad (14)$$

Разрядность числа

$$A - M = 12. \quad (15)$$

Разрядность числа

$$B - N = 6. \quad (16)$$

Число шагов деления определяется шириной частного W_Q = M − N + 1 и равно

$$k = W_Q \ / 3. \quad (17)$$

Если W_Q не кратно трем, делимое дополняется нулями со стороны старших разрядов до ширины, кратной трем; недостающие старшие разряды частного при этом равны нулю. В реализованном прототипе ширина частного фиксирована (W_Q = 6, диапазон операндов A < 64·B), поэтому k = 6 / 3= 2.

$k = 10_2$ записывается в счетчик – СЧТ.

Кратные числа к В: $2B=54_{10}$; $3B=81_{10}$; $4B=108_{10}$; $5B=135_{10}$; $6B=162_{10}$; $7B=189_{10}$;

Старшие 6 битов двоичного кода числа А определяют значение начального остатка $R_0 = 010101_2 = 21_{10}$. К сдвинутому влево на три разряда остатка R_0 присоединяются биты $a_5a_4a_3$ числа А, при этом получим:

$$A_1 = L(3)R_0 + a_5a_4a_3 = 010101101_2 = 168 + 5 = 173_{10}.$$

Для наглядности все вычисления по делению числа А на делитель В с определением остатка R и частного Q приведены в таблице 2 в десятичной системе счисления.

Таблица 2 – Порядок выполнения операций деления числа А на В с формированием остатка R и разрядов частного

Шаги деления	СЧТИ	Выполнение действий в РгА и ФЧОиРЧ
1	СЧТИ: 10_2 СЧТИ: $10_2 - 01 = 01_2$	В РгА: $A_1 = L(3)R_0 + a_5a_4a_3 = 010101101_2 = 168 + 5 = 173_{10}$. В ФЧОиРЧ: поскольку 162<173<189, то имеет место $B < A_1 < 7B$, следовательно, выполняется операция $R_1 = A_1 - 6B = 173_{10} - 162_{10} = 11_{10}$ При этом $q_5 = 1, q_4 = 1,\ q_3 = 0$
2	СЧТИ-1: $01_2 - 01 = 00_2$ Конец операций (КО)	В РгА: $A_2 = L(3)R_1 + a_2a_1a_0 = 010101101_2 = 88 + 3 = 91_{10}$. В ФЧОиРЧ: поскольку 81<91<108, то имеет место соотношение $3B < A_2 < 4B$, следовательно, в ФЧОиРЧ выполняется операция $R_2 = A_2 - 3B = 91_{10} - 81_{10} = 10_{10}$ При этом $q_2 = 0, q_1 = 1,\ q_0 = 1$

Классический побитовый алгоритм (radix-2) потребовал бы 6 тактов, алгоритм radix-4 – 3 такта. Предложенная архитектура обеспечивает троекратное сокращение числа итераций по сравнению с radix-2 и полуторакратное по сравнению с radix-4.

Проверка:

$$Q = q_5q_4q_3q_2q_1q_0 = 110011_2 = 51_{10} \quad (19)$$

$$A = (Q * B) + R = (51 * 27) + 10 = 1377 + 10 = 1387_{10}. \quad (20)$$

Для проверки корректности работы устройства в граничных условиях рассмотрим случай деления числа $A = 63_{10}(111111_2)$ на делитель $B = 63_{10}(111111_2)$. При $M = N = 6$ ширина частного W_Q = M − N + 1 = 1, откуда k = 1 / 3= 1: за один шаг формируется Q = 1, R = 0. при котором частное Q = 1, остаток R = 0. Устройство корректно обрабатывает данный случай, поскольку начальный остаток $R_0 = A$ и на первом же сравнении выполняется условие $A_i = B$, что дает $q_i = 1$.

Реализация и верификация на ПЛИС. Архитектура описана на языке Verilog HDL в виде иерархии модулей (модуль верхнего уровня SequentialDivider3; регистр делимого rgA; формирователь кратных FKRD; вычислительное ядро PRU3_QDSL с компараторами и логикой выбора; блок синхронизации BS3) и синтезирована в среде Xilinx Vivado для ПЛИС семейства Artix-7. Общий вид RTL-схемы синтезированного устройства приведен на рисунке 6, схема формирователя кратных – на рисунке 7. Формирователь FKRD (блок БФКД из рисунка 3) генерирует семь кратных {B, 2B, …, 7B} и их обратные коды, используя исключительно операции сдвига и сложения, что соответствует методу предварительного формирования кратных без умножителей.

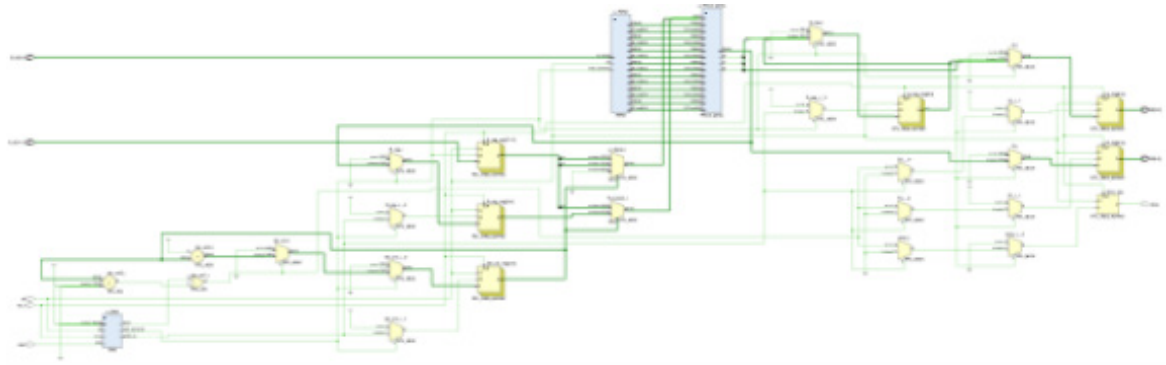

Рисунок 6 – RTL-схема синтезированного последовательного делителя (общий вид)

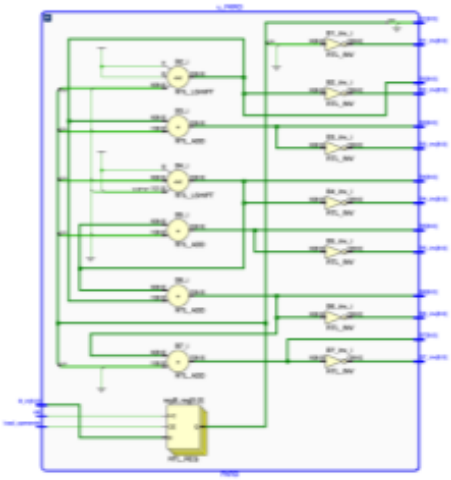

Рисунок 7 – RTL-схема формирователя кратных делителю (БФКД)

Использование аппаратных ресурсов прототипа приведено в таблице 3, иерархическое распределение ресурсов по отчету Vivado – на рисунке 8. Верхний модуль занимает 158 LUT и 50 триггеров, при этом формирователь кратных использует 78 LUT, вычислительное ядро – 70 LUT, блок синхронизации – 10 LUT. Утилизация LUT составляет менее 1% доступных ресурсов кристалла, что подтверждает компактность реализации.

Таблица 3 – Использование аппаратных ресурсов (ПЛИС Artix-7)

Ресурс	Использовано	Доступно	Утилизация, %
LUT	158	20 800	0,76
Триггеры (FF)	50	41 600	0,12
Bonded IOB	34	150	22,7
BUFGCTRL	1	32	3,1

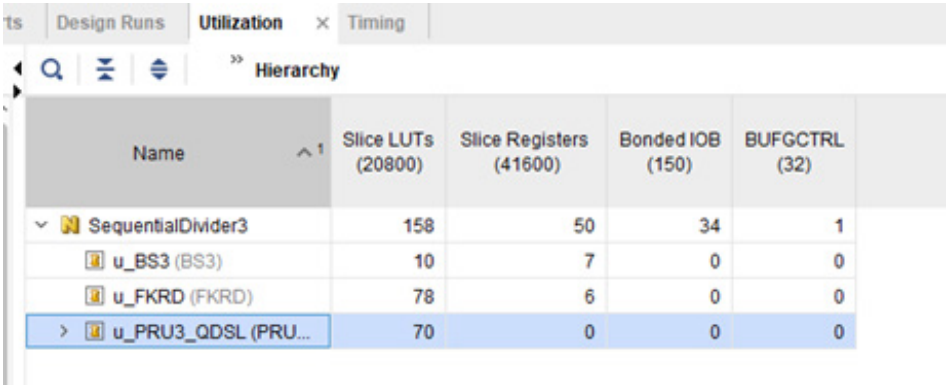

Рисунок 8 – Иерархическое распределение аппаратных ресурсов (отчет Vivado)

Статический временной анализ показал запас по времени WNS = 2,570 нс, что соответствует минимальному периоду T_min = 7,430 нс и максимальной тактовой частоте F_max ≈ 134,6 МГц. Высокая тактовая частота обусловлена параллельной организацией блока сравнения в реализованном прототипе: все пороги сравниваются одновременно, и критический путь определяется одним каскадом «компаратор – приоритетный декодер» независимо от значений операндов, что обеспечивает постоянную задержку шага выбора цифры частного.

Временная диаграмма работы устройства приведена на рисунке 9.

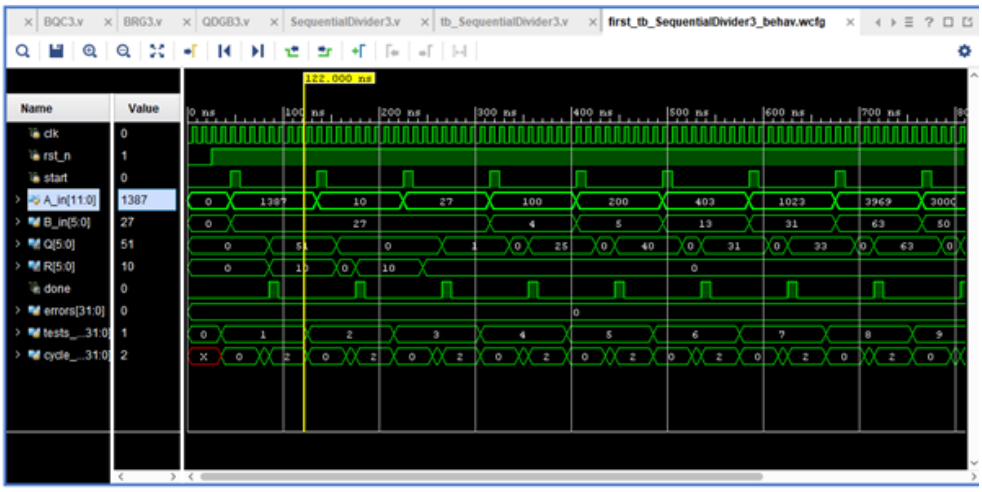

Рисунок 9 – Временная диаграмма верификации последовательного делителя (XSim)

Глубина критического пути и достижимая тактовая частота определяются разрядностью делителя N, задающей размер компараторов и сумматора, а не разрядностью делимого M. Увеличение разрядности делимого увеличивает лишь число шагов k согласно (6), не удлиняя критический путь; поэтому частота F_max ≈ 134,6 МГц сохраняется при росте M, а латентность растет линейно по числу шагов. Это отличает предложенную архитектуру от высокорадиксных SRT-делителей, у которых усложнение таблицы выбора непосредственно увеличивает задержку. Систематический синтез для набора разрядностей операндов (16, 32, 64 бита) с построением зависимостей «площадь – частота» вынесен в дальнейшие исследования.

Оценка энергопотребления. Энергопотребление прототипа оценено по реализованному списку соединений (post-implementation) средствами анализа мощности Vivado; результат приведен на рисунке 10. Суммарная потребляемая мощность кристалла составляет 0,077 Вт, из которых на статическую мощность (ток утечки ПЛИС) приходится 0,070 Вт (91 %), а на динамическую – лишь 0,007 Вт (9 %). В составе динамической мощности преобладает потребление блоков ввода-вывода (0,004 Вт, 63 %), тогда как на тактовые цепи, сигналы и собственно логику делителя приходится примерно по 0,001 Вт. Малая доля динамической мощности логического ядра подтверждает аппаратную компактность предложенной архитектуры: при температуре перехода 25,4 °C тепловой запас составляет 59,6 °C. Приведенная оценка получена в режиме vectorless-анализа и носит ориентировочный характер; уточнение возможно при подаче реальной переключательной активности (SAIF) из временного моделирования.

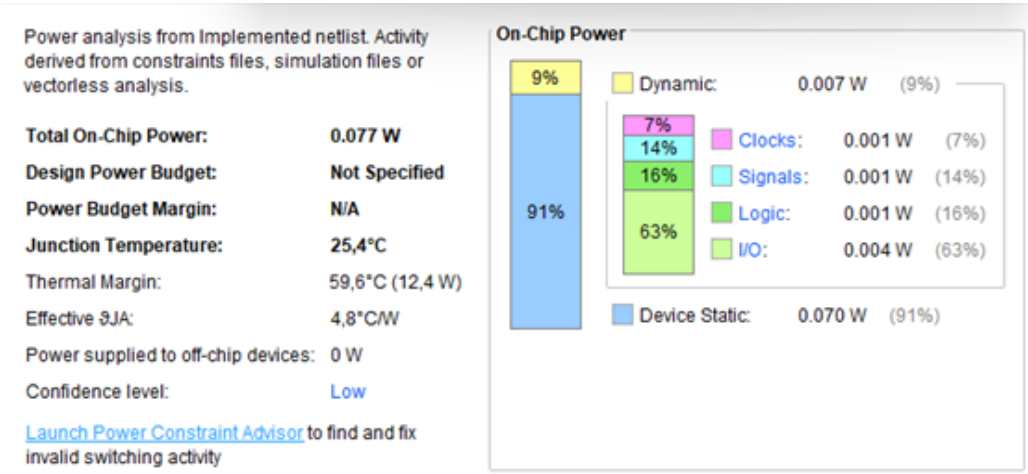

Рисунок 10 – Анализ потребляемой мощности прототипа (отчет Vivado)

Полученные результаты подтверждают, что предложенная архитектура достигает производительности уровня radix-8 при умеренных аппаратных затратах и регулярной структуре. Перспективным направлением является разработка конвейерных и матричных вариантов, также формирующих по три разряда частного за итерацию, с анализом компромисса между глубиной конвейера, занимаемой площадью и частотой.

Заключение

В данной работе представлена архитектура последовательного целочисленного делителя, обеспечивающая одновременное формирование трех разрядов частного за один тактовый цикл. Предложенное устройство включает формирователь кратных делителю (БФКД), реализующий предвычисление значений B–7B, и формирователь частичных остатков и разрядов частного (ФЧОиРЧ), выполняющий параллельное сравнение посредством пяти компараторов COMP-1...COMP-5. Блок синхронизации на основе RS-триггера и вычитающего счетчи-

ка обеспечивает управление итерационным процессом деления. В отличие от классических высокорадиксных SRT-делителей, выбор цифры частного выполняется точным сравнением с предвычисленными кратными и прямым приоритетным декодированием, что устраняет рост таблицы выбора при увеличении основания.

Архитектура описана на языке Verilog HDL и верифицирована на ПЛИС Xilinx Artix-7 в среде Vivado. Прототип для 12-разрядного делимого и 6-разрядного делителя занимает 158 LUT и 50 триггеров, достигает F_max ≈ 134,6 МГц (WNS = 2,570 нс) и подтвержден комплексом из десяти тестов, выполняя деление за два тактовых цикла. Тем самым обеспечивается троекратное сокращение числа итераций по сравнению с алгоритмом radix-2 и полуторакратное – по сравнению с radix-4. Направлениями дальнейших исследований являются синтез устройства для операндов разрядностью 16, 32 и 64 бита с построением зависимостей «площадь – частота», а также разработка конвейерных и матричных вариантов архитектуры.

ЛИТЕРАТУРА

1 Wei, X., Yang, Y., Chen, J. A Low-Latency Divider Design for Embedded Processors. Sensors, 22, 2471 (2022). https://doi.org/10.3390/s22072471

2 Mehta, B., Talukdar, J., Gajjar, S. High speed SRT divider for intelligent embedded system. In Proc. 2017 Int. Conf. on Soft Computing and its Engineering Applications (icSoftComp) (Changa, India, 2017), pp. 1–5. https://doi.org/10.1109/ICSOFTCOMP.2017.8280077

3 Patankar, U.S., Flores, M.E., Koel, A. Division Algorithms—from Past to Present Chance to Improve Area Time and Complexity for Digital Applications. In Proceedings of the 2020 IEEE Latin America Electron Devices Conference (LAEDC) (San Jose, Costa Rica, 2020), pp. 1–4.

4 Angioli, M., Barbirotta, M., Cheikh, A., Mastrandrea, A., Menichelli, F., Jamili, S., Olivieri, M. Design, Implementation and Evaluation of a New Variable Latency Integer Division Scheme. IEEE Transactions on Computers, 73 (7), 1767–1779 (2024). https://doi.org/10.1109/TC.2024.3386060

5 Murillo, R., Villalba-Moreno, J., Del Barrio, A., Botella, G. Digit-Recurrence Posit Division. arXiv (2025). https://doi.org/10.48550/arXiv.2511.02494

6 Nannarelli, A., Lang, T. Low-Power Division: Comparison among Implementations of Radix 4, 8 and 16. In Proceedings of the 14th IEEE Symposium on Computer Arithmetic (Adelaide, Australia, 1999), pp. 149–156.

7 Nikmehr, H., Phillips, B., Lim, C.C. A Fast Radix-4 Floating-Point Divider with Quotient Digit Selection by Comparison Multiples. The Computer Journal, 50 (1), 81–92 (2007). https://doi.org/10.1093/comjnl/bxl048

8 Zhou, X., Shen, R., He, P., Gao, H. Optimization and Implementation of SRT-16 Floating-Point Divider Based on FPGA. In Proceedings of the 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT) (2024), pp. 2019–2023.

9 Baesler, M., Voigt, S.O. Analysis of Fast Radix-10 Digit Recurrence Algorithms for Fixed-Point and Floating-Point Dividers on FPGAs. International Journal of Reconfigurable Computing, 453173 (2013). https://doi.org/10.1155/2013/453173

10 Zhexebay, D., Mamanova, S., Karibayev, B., et al. Pipelined Divider with Precomputed Multiples of Divisor. Electronics, 15 (1), 110 (2026). https://doi.org/10.3390/electronics15010110

11 Temel, M. Formal Verification of Booth Radix-8 and Radix-16 Multipliers. In Proc. 2024 Design, Automation & Test in Europe Conf. & Exhibition (DATE) (IEEE, 2024).

12 Kornerup, P. Digit selection for SRT division and square root. IEEE Transactions on Computers, 54 (3), 294–303 (2005). https://doi.org/10.1109/TC.2005.52

13 Amin, A., Shinwari, W. High-Radix Multiplier-Dividers: Theory, Design, and Hardware. IEEE Transactions on Computers, 59 (7), 1009–1022 (2010). https://doi.org/10.1109/TC.2010.78

14 Oberman, S.F., Flynn, M.J. Division algorithms and implementations. IEEE Transactions on Computers, 46 (8), 833–854 (1997). https://doi.org/10.1109/12.618254

15 Nannarelli, A. Digit Recurrence Division Algorithms and Implementations (Ph.D. thesis, University of Pisa, 1996).

16 Patankar, U.S., Koel, A. Review of basic classes of dividers based on division algorithm. IEEE Access, 9, 23035–23069 (2021). https://doi.org/10.1109/ACCESS.2021.3055735

17 Harris, D.L., Oberman, S.F., Horowitz, M.A. SRT Division Architectures and Implementations. In Proc. 13th IEEE Symp. on Computer Arithmetic (1997), pp. 18–25. https://doi.org/10.1109/ARITH.1997.614875

18 Anane, M., Bessalah, H., Issad, M., Anane, N., Salhi, H. Higher radix and redundancy factor for floating point SRT division. IEEE Transactions on VLSI Systems, 16 (6), 774–779 (2008). https://doi.org/10.1109/TVLSI.2008.2001058

19 Montuschi, P., Nannarelli, A. Digital arithmetic: Division algorithms. In Encyclopedia of Computer Science and Technology (New York: Taylor & Francis, 2017), pp. 348–363.

20 Matthews, J., Zhang, Z., Chen, Z. Rethinking integer divider design for FPGA-based soft processors. In Proceedings of the 2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM) (San Diego, CA, USA, 2019), pp. 115–122.

[1]Тынымбаев С.,
профессор, ORCID ID: 0000-0002-9326-9476,
e-mail: s.tynymbayev@iitu.edu.kz
[1]*Маманова С.Е.,
докторант, ORCID ID: 0009-0001-7277-5492,
*e-mail: s.mamanova@iitu.edu.kz
[2]Сқабылов А.А.,
PhD, ORCID ID: 0000-0002-5196-8252,
e-mail: Alisher.skabylov@kaznu.edu.kz
[1]Чинибаева Т.Т.,
PhD, ORCID ID: 0000-0002-2657-3697,
e-mail: t.temirbolatova@iitu.edu.kz
[2]Жексебай Д.М.,
PhD, ORCID ID: 0009-0008-1884-4662,
e-mail: zhexebay92@gmail.com

[1]Халықаралық ақпараттық технологиялар университеті, Алматы қ., Қазақстан
[2]Әл-Фараби атындағы Қазақ ұлттық университеті, Алматы қ., Қазақстан

БІР ҚАДАМДА БӨЛІНДІНІҢ ҮШ РАЗРЯДЫН БІР МЕЗГІЛДЕ ҚҰРАСТЫРУҒА НЕГІЗДЕЛГЕН БҮТІН САНДЫ БӨЛГІШТІҢ АРХИТЕКТУРАСЫ

Аңдатпа

Бүтін санды бөлу операциясы цифрлық есептеу жүйелеріндегі ең ресурс сыйымды арифметикалық операциялардың бірі саналады. Дәстүрлі дәйекті бөлгіштер әр такт сайын бөліндінің бір разрядын қалыптастырады, сондықтан n-разрядты бөлу үшін n такт қажет. Бұл жұмыста бір тактілік цикл ішінде бөліндінің үш разрядын бір мезгілде қалыптастыруды қамтамасыз ететін дәйекті бүтін санды бөлгіштің архитектурасы ұсынылады. Әзірленген құрылғы бөлінгіш регистрін (БР), бөлгіштің еселіктерін қалыптастырғышты (БЕҚ), ішінара қалдықтар мен бөлінді разрядтарын қалыптастырғышты (ІҚБРҚ) және синхрондау блогын (СБ) қамтиды. Артық таңбалауды және бөлінді цифрын таңдау кестелерін пайдаланатын классикалық SRT-бөлгіштерден айырмашылығы, ұсынылған архитектура жартылай қалдықты бөлгіштің алдын ала есептелген еселіктерімен (B–7B) тікелей салыстырады, бұл санау жүйесінің негізі артқан сайын таңдау кестесінің көлемінің ұлғаюын болдырмайды. Архитектура Verilog HDL тілінде сипатталып, Vivado ортасында Xilinx Artix-7 ПЛИС-інде верификацияланды. 12 разрядты бөлінгіш пен 6 разрядты бөлгішке арналған прототип 158 LUT және 50 триггерден тұрады, 134,6 МГц максималды тактілік жиілікке жетеді және он тесттен тұратын кешен арқылы расталды; бөлу екі тактілік циклде орындалады. Архитектура жоғары жылдамдықты цифрлық есептеуіштер мен мамандандырылған процессорларда қолдануға арналған.

Түйін сөздер: бөлгішке еселі сандар, қалдықтарды қалыптастырғыш, бөлінді разрядтарын қалыптастырғыш, тізбекті әрекетті бөлгіш, radix-8, ПЛИС, Verilog HDL.

[1]Tynymbayev S.,
Professor, ORCID ID: 0000-0002-9326-9476,
e-mail: s.tynymbayev@iitu.edu.kz
[1]*Mamanova S.E.,
PhD Candidate, ORCID ID: 0009-0001-7277-5492,
*e-mail: s.mamanova@iitu.edu.kz
[2]Skabylov A.A.,
PhD, ORCID ID: 0000-0002-5196-8252,
e-mail: Alisher.skabylov@kaznu.edu.kz
[1]Chinibayeva T.T.,
PhD, ORCID ID: 0000-0002-2657-3697,
e-mail: t.temirbolatova@iitu.edu.kz
[2]Zhexebay D.M.,
PhD, ORCID ID: 0009-0008-1884-4662,
e-mail: zhexebay92@gmail.com

[1]International Information Technology University, Almaty, Kazakhstan
[2]Al-Farabi Kazakh National University, Almaty, Kazakhstan

A SEQUENTIAL INTEGER DIVIDER WITH THREE-BIT QUOTIENT DIGIT GENERATION

Abstract

Integer division remains one of the most computationally demanding arithmetic operations in digital computing systems. Conventional sequential dividers produce one quotient bit per clock cycle, requiring n cycles for an n-bit division. This paper proposes an architecture for a sequential integer divider capable of simultaneously generating three quotient bits per clock cycle. The proposed device comprises a dividend register (DR), a divisor multiples generator (DMG), a partial remainder and quotient digit generator (PRQDG), and a synchronization unit (SU). Unlike classical SRT dividers, which rely on a redundant quotient representation and quotient-digit selection tables, the proposed architecture performs an exact comparison of the partial remainder against the precomputed divisor multiples B–7B, thereby eliminating the growth of the selection table as the radix increases. The architecture is described in Verilog HDL and verified on a Xilinx Artix-7 FPGA using the Vivado design suite. The prototype for a 12-bit dividend and a 6-bit divisor occupies 158 LUTs and 50 flip-flops, attains a maximum clock frequency of 134.6 MHz, and is confirmed by a suite of ten tests; a division is completed in two clock cycles. The architecture is intended for use in high-speed digital computing units and specialized processors.

Keywords: integer division, divisor multiples, remainder generator, quotient digit generator, sequential divider, radix-8, FPGA, Verilog HDL.

Received January 27, 2026; revised April 24, June 21, 2026; accepted June 25, 2026.