Алгоритм конвейерного вычисления остатка по заданному модулю

Захаров В.М., Песошин В.А., Шалагин С.В.

Казань. 27.10.2017

Где применяется операция вычисления остатка по заданному модулю (далее – Операция)?

- рекуррентные преобразования над потоками чисел (Молдовян Н.А. и др.);
- тестирование и диагностика цифрового оборудования (Латыпов Р.Х., Столов Е.Л. и др.);
- аппаратная реализация генераторов кодовых последовательностей и конгруэнтных генераторов (псевдо)случайных чисел (Гилл А., Лидл Р., Нидеррайтер Г., Песошин В.А., Кузнецов В.М. и др.);
- реализация нелинейных функций усложнения (Бухштаб А.А., Захаров В.М. и др.)

Существуют примеры аппаратной реализации Операции.

- Патент РФ № 1785081. Устройство для формирования остатка по произвольному нечетному модулю от числа/ Бережной В.В., Червяков Н.И., Оленев А.А.// 1992. Бюл. № 48.
- Патент РФ № 2192092. Устройство для преобразования *n*-разрядного двоичного позиционного кода в двоичный код остатка по модулю *m*/ Овчаренко Л.А., Турченяк В.И.// 2002. Бюл. № 30.
- Патент РФ № 2324972. Устройство для формирования остатка по произвольному модулю от числа/ Петренко В.И., Кузьминов Ю.В., Карагулян Д.Л., Мосин О.В.// 2008. Бюл. № 14.

Алгоритм вычисления остатка по модулю (Столов Е.Л., Захаров В.М., Шалагин С.В.)

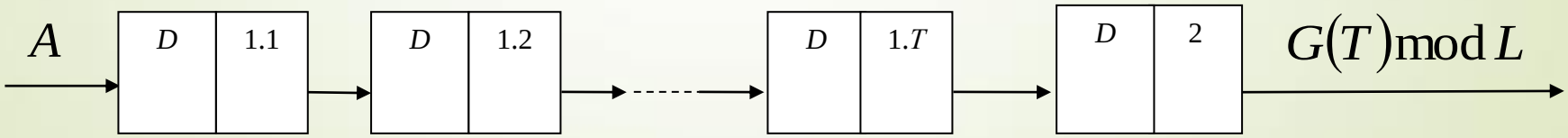
$$A = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \ldots + a_1 2 + a_0. \quad (1)$$

L – p–разрядное двоичное число, $p < n$.

$$G(i) = g_{n-1}^{(i-1)} b_{n-1} + \ldots + g_p^{(i-1)} b_p + g_{p-1}^{(i-1)} b_{p-1} + \ldots + g_1^{(i-1)} b_1 + g_0^{(i-1)}, \quad i = \overline{1, T}, \quad (2)$$

$$g_j^{(0)} = a_j, \quad j = \overline{0,(n-1)},$$

$$b_j = \begin{cases} 2^j : & j \le p \\ 2^j \bmod L : & j > p. \end{cases} \quad (3)$$

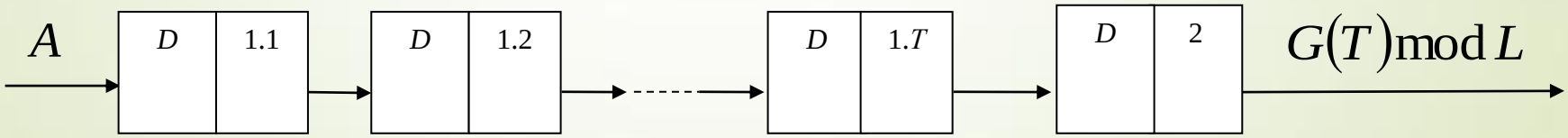


Захаров В.М., Столов Е.Л, Шалагин С.В., пат. № 2009138551/22, опубл. 10.03.2010, Бюл. №)

Алгоритм вычисления остатка по модулю (Столов Е.Л., Захаров В.М., Шалагин С.В.)

Недостатки

1) большое количество слагаемых (операций сложения) на этапах $1...T$ выполнения Операции;
2) количество ступеней конвейера требуется подбирать эмпирически для заданного L, что вызывает трудности вычислительного характера для чисел A большой разрядности f.



Захаров В.М., Столов Е.Л, Шалагин С.В., пат. № 2009138551/22, опубл. 10.03.2010, Бюл. №)

Предлагаемая модель Операции

Вычисление остатка числа A вида (1) от деления на p-разрядное значение L, $L \in [2^{p-1}, 2^p-1]$, сводится к вычислению остатка от числа

$$D_{n-1} : A \equiv D_{n-1} (\mathrm{mod}\, L).$$

$$D_{n-1} = a_{n-1} b_{n-1} + \ldots + a_p b_p + \underbrace{a_{p-1} 2^{p-1} + \ldots + a_1 2 + a_0}_{D_{p-1} = a_{p-1} 2^{p-1} + \ldots + a_1 2 + a_0},$$

$$D_i = D_{i-1} + a_i b_i,$$

$$b_i = \begin{cases} 2^i \left(\mathrm{mod}\, L\right) & : D_{i-1} \le 0 \\ 2^i \left(\mathrm{mod}\, L\right) - L & : D_{i-1} > 0 \end{cases}$$

$$i = \overline{p, n-1}.$$

Предлагаемая модель Операции

$$D_{n-1} = a_{n-1}b_{n-1} + \ldots + a_p b_p + \underbrace{a_{p-1}2^{p-1} + \ldots + a_1 2 + a_0}_{D_{p-1}},$$

$$D_{p-1} \in \left[0, 2^p - 1\right]$$

Согласно (3), если $D_{i\text{-}1} > 0$, то $b_i \in \left[1-L, -1\right]$,

иначе – $b_i \in \left[1, L-1\right]$.

Утверждение. Значения величин $D_i \in \left[1-L, 2^p - 2\right]$, $i = \overline{p, n-1}$.

Согласно утверждению, $D_{n-1} \in \left[1-L, 2^p - 2\right]$

Предлагаемая модель Операции

$$D_{n-1} : A \equiv D_{n-1} (\mathrm{mod}\, L); \qquad D_{n-1} \in \left[1 - L, 2^p - 2\right].$$

Остаток от деления A на L:

$$M = D_{n-1} + z, \qquad z = \begin{cases} -L & : D_{n-1} \geq L \\ 0 & : 0 \leq D_{n-1} < L \\ L & : D_{n-1} < 0. \end{cases}$$

Конвейерный алгоритм формирования остатков по заданному модулю

Этапов алгоритма – ($n - p + 1$).
На каждом этапе – сохранение промежуточных результатов.
Алгоритм применим для обработки потока чисел.

Этап $(i - p + 1)$, $i = \overline{p, n-1}$.

Вычисление значения $D_i = D_{i-1} + a_i b_i$,

где $b_i = \begin{cases} 2^i \bmod L & : D_{i-1} \le 0 \\ 2^i \bmod L - L & : D_{i-1} > 0 \end{cases}$, $D_{p-1} = a_{p-1} 2^{p-1} + \ldots + a_1 2 + a_0$.

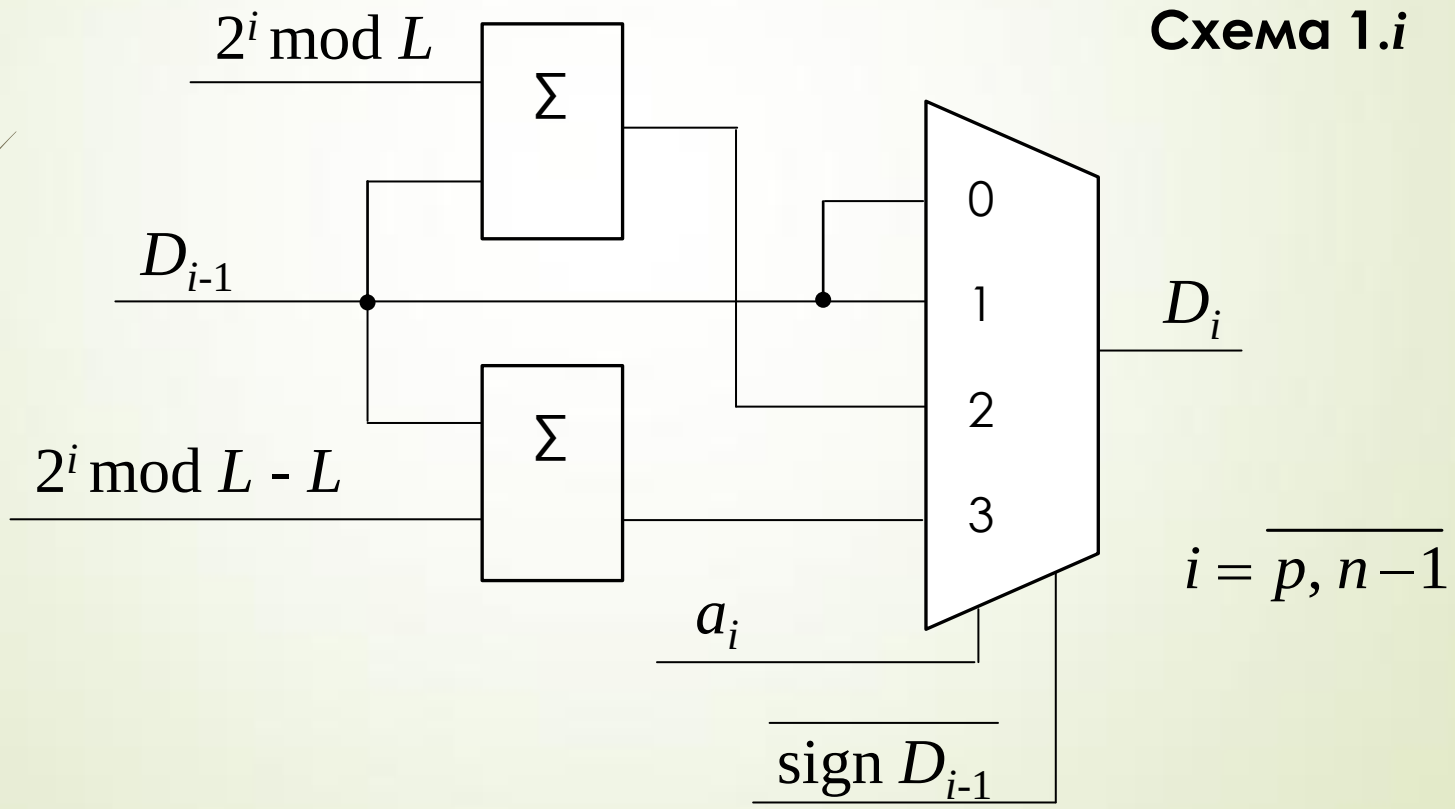
Этап $(n - p + 1)$.

Получение $M = D_{n-1} + z$,

где $z = \begin{cases} -L & : D_{n-1} \ge L \\ 0 & : 0 \le D_{n-1} < L \\ L & : D_{n-1} < 0. \end{cases}$

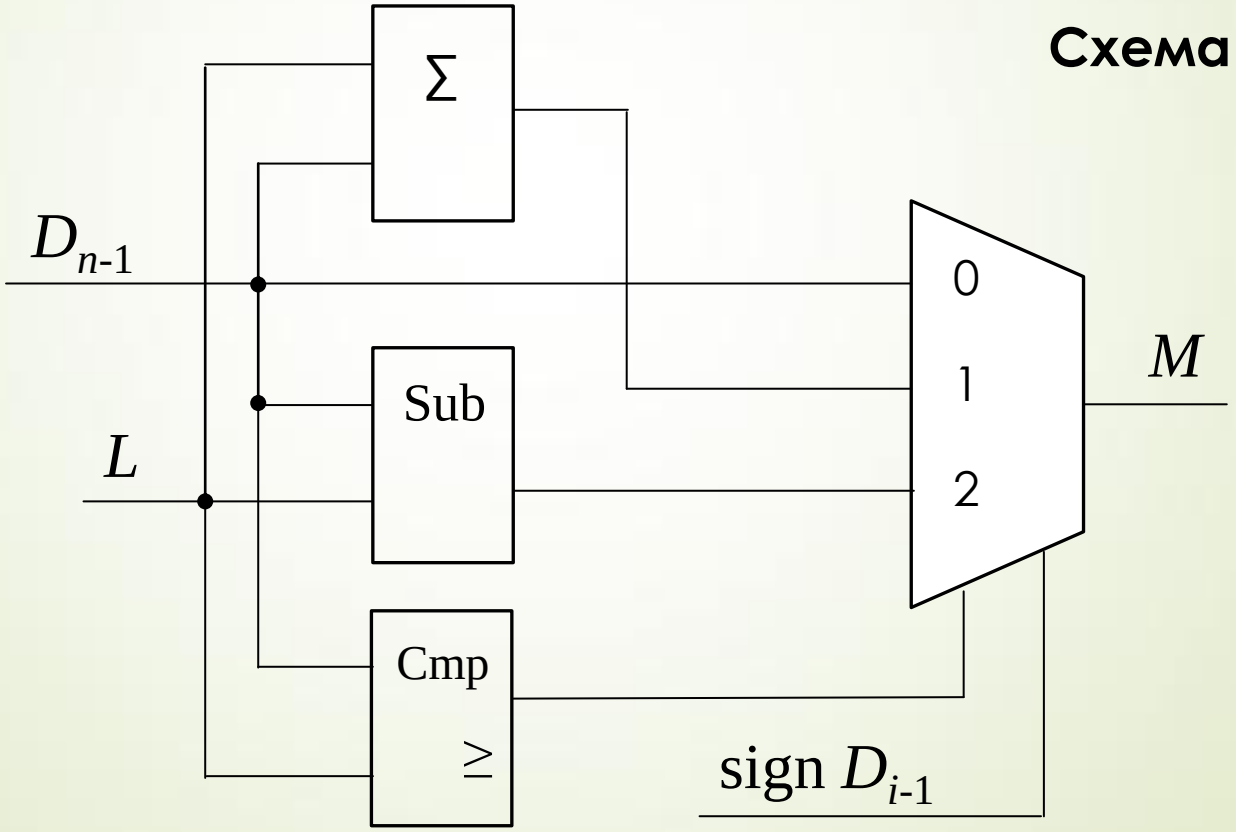
Конвейерный алгоритм формирования остатков по заданному модулю

Схема аппаратной реализации этапов 1 … (n - p):



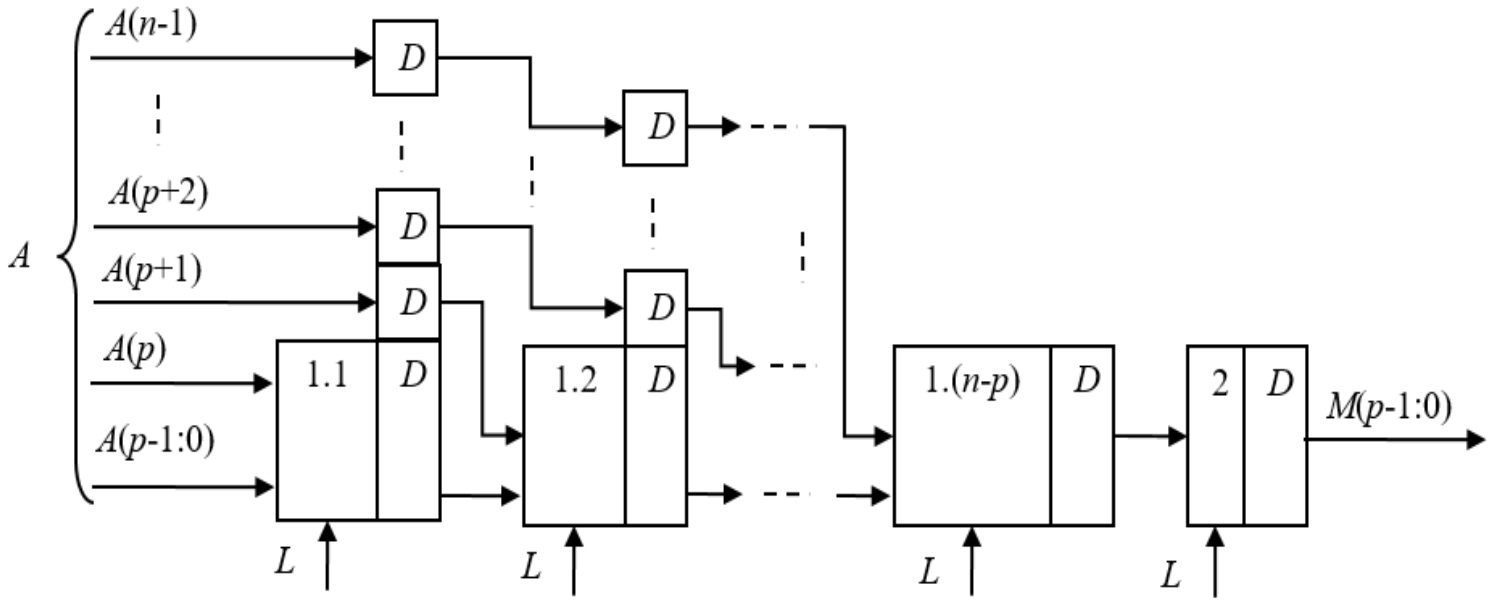
Конвейерный алгоритм формирования остатков по заданному модулю

Схема аппаратной реализации этапа ($n - p + 1$):



Конвейерный алгоритм формирования остатков по заданному модулю

Схема формирования остатка по заданному модулю *L*.

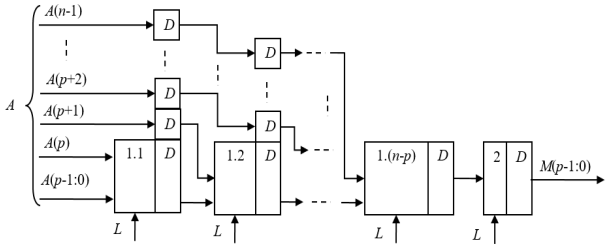


Дополнительно требуется $(n-p-1)\cdot(n-p)/2$ *D*-триггеров.

Конвейерный алгоритм формирования остатков по заданному модулю

Схема формирования остатка по заданному модулю *L*.

Время задержки функционирования конвейера определено согласно формуле:



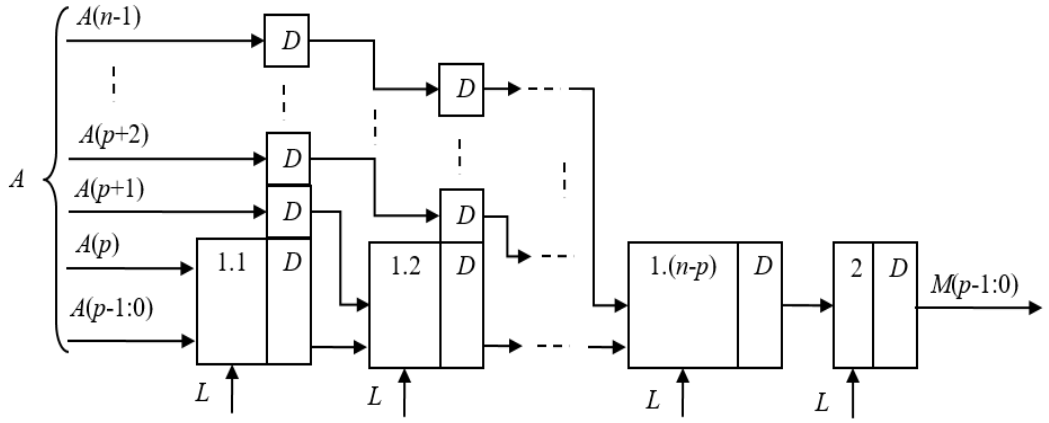
$$\tau = \max\left(T_{1.i}, T_2\right)$$

$$T_{1.i} = \max\left(T_{sum}^{const}, T_{sub}^{const}\right) + T_{mx(4-1)}$$

$$T_2 = \max\left(T_{comp}^{const}, T_{sum}^{const}, T_{sub}^{const}\right) + T_{mx(4-1)}$$

Конвейерный алгоритм формирования остатков по заданному модулю

Схема формирования остатка по заданному модулю *L*.



Поток чисел A_1, A_2, …, A_N будет обработан (вычислены остатки от деления на L – M_1, …, M_N) за время $(N+n-p+1)\cdot\tau$, где τ – период между поступлением синхросигналов.

Конвейерный алгоритм формирования остатков по заданному модулю

Для реализации Алгоритма требуется:

- хранить константы $2^i \bmod L$ и $(2^i \bmod L - L)$ в $2(n - p)$ $(p + 1)$-разрядных регистрах;
- $2(n - p) + 1$ сумматор с константой для p-разрядных целых знаковых чисел;
- блок вычитания константы из p-разрядных целых знаковых чисел;
- $(n - p + 2)$ $(p + 1)$-разрядных регистров для хранения D_i;
- для обеспечения конвейера – $(n - p - 1)\cdot(n - p)/2$ $(p + 1)$-разрядных регистров
- компаратор с константой для p-разрядных целых знаковых чисел;
- $(n - p + 1)\cdot(p + 1)$ мультиплексоров «4 в 1».

Заключение

Оценка временной сложности вычисления N значений $M_1, \ldots, M_N$ имеет порядок $\Theta\left(\left(N+n-p\right)\cdot\tau\right)$. В известных алгоритмах порядок времени задержки функционирования составляет $\Theta(N\cdot\delta)$, где δ - оценка времени вычисления значения одного остатка и $\tau << \delta$.

Количество ступеней конвейера и сумматоров линейно зависит от разности $\left(n-p\right)$.

Количество (p + 1)-разрядных регистров на основе (D-триггеров) имеет порядок $\left(n-p\right)$.

Спасибо за внимание