

САПР ДЛЯ ПОЛУЧЕНИЯ ЗАКОНОВ ФУНКЦИОНИРОВАНИЯ ПРЕОБРАЗОВАТЕЛЕЙ КОДОВ С ПАРАЛЛЕЛЬНОЙ СТРАТЕГИЕЙ

Рассматриваются возможности подсистем САПР «TRANS» для получения таблиц законов функционирования многоблочных преобразователей кодов, функционирующих по методу накопления эквивалентов и использующих параллельную стратегию шагов преобразования.

1. Постановка задачи

В настоящее время большинство ЭВМ функционируют в двоичной системе счета. Исходные данные и результаты решения задач должны быть представлены в форме, удобной для восприятия их человеком, т.е. в десятичной системе счисления.

По этой причине возникает задача преобразования чисел из десятичной системы в двоичную и обратно. Операции преобразования чисел из одной системы в другую можно выполнять как программным способом, так и аппаратным. Аппаратный способ преобразования чисел с помощью специализированного блока – преобразователя кодов (ПК) предпочтителен, так как он не снижает быстродействие компьютера и позволяет совместить решение одной задачи с подготовкой данных для ввода/вывода другой.

Наиболее удобным методом преобразования чисел аппаратным способом является метод накопления эквивалентов. Этот метод гибкий и позволяет за счет выбора числа шагов преобразования, значений шагов и стратегий их использования (последовательной или параллельной) обеспечить требуемые значения быстродействия и аппаратных затрат. Преимуществом параллельной стратегии использования шагов является возможность дальнейшего увеличения быстродействия на 20-30% по сравнению с последовательной стратегией.

Цель данной работы – автоматизация этапа системного проектирования преобразователей кодов целых чисел с параллельной стратегией использования шагов преобразования в широком диапазоне разрядностей, оснований систем счета и значений шагов.

Основными *задачами* являются:

- анализ двух вариантов структурной организации и функционирования ПК – с последовательной и параллельной стратегией использования шагов преобразования;
- рассмотрение возможностей подсистемы САПР «TRANS» для проектирования ПК, как параллельного, так и последовательного типов.

2. Последовательная стратегия

При использовании последовательной стратегии преобразования чисел процесс преобразования сводится к последовательному вычитанию из состояния разрядного счетчика, в котором хранится цифра преобразуемого числа, значения шага преобразования. Если значение системы счета на входе K , то максимальное значение цифры преобразуемого числа равно $K-1$. Следовательно, максимальное число тактов преобразования старших разрядных цифр равно $K-1$. Затем, еще один такт используется для трансляции (передачи) значения цифры младшего разряда. Таким образом, суммарное число тактов преобразования равно, т.е. $N_1=K$. В двухшаговом ПК вначале ведется преобразование с шагом a . При наличии в одном из старших разрядов $x_m (m = \overline{2, n})$ максимального значения цифры $x_m = K-1$ потребуется на первом этапе $\lceil (K-1)/a \rceil$ шагов, преобразования, где $\lceil \rceil$ означают округление до меньшего целого. Затем, если хотя бы одна из оставшихся старших цифр имеет значение $x_i = a-1$, потребуется на втором этапе еще $a-1$ такт для обнуления с шагом 1 всех старших разрядных счетчиков. Следует добавить еще один такт (третий этап) для трансляции младшей цифры x_0 . Следовательно,

Группа триггеров 3 фиксирует ненулевое значение счетчиков соответствующих разрядов; группа триггеров 4 – значение старших разрядных счетчиков, превышающее заданное число $a - 1$, например 1.

Шифраторы 12 реализуют следующую функцию:

$$Y = \{X - a; X \geq a; \{X; X < a,$$

где X – входной код тетрады; a – параметр, в частном случае, равный 2. Цепи инициирования и сброса на рис.1 не показаны. Так как в конкретном случае $n = 4$, $K = 12$, то диапазон изменения входного кода $0 - (12^4 - 1) = 0 - 20735_{10}$. Код состояний триггеров как первой группы 3, так и второй 4 имеет $2^3 = 8$ значений от 000 до III.

ФЭ 13, выполненный в виде последовательного соединения первого 14 и второго 15 дешифраторов, реализует функцию

$$S = \begin{cases} KC_1 + K^2C_2 + K^3C_3; D_1 = D_2 = D_3 = 0; \\ aKD_1 + aK^2D_2 + aK^3D_3; D_1, D_2, D_3 \neq 0; \end{cases} \quad (2)$$

где C_1, C_2, C_3 – значения разрядов двоичного кода триггеров состояний первой группы 3; D_1, D_2, D_3 – значения разрядов двоичного кода триггеров состояния второй группы 4.

Каждый из $C_m (m = \overline{1,3})$ триггеров 3 первой группы состояний разрядных счетчиков соответствует наличию (1) или отсутствию (0) информации в соответствующем старшем разряде преобразуемого кода, а разряд $D_m (m = \overline{1,3})$ триггеров 4 второй группы состояний разрядных счетчиков равен 1, если соответствующий разряд преобразуемого кода имеет значение $x_i \geq a (2 \leq a \leq k - 1)$, в противном случае $D_m = 0$.

ФЭ 13 преобразует вначале двоичный код D_3, D_2, D_1 триггеров состояний второй группы, затем при $D_3, D_2, D_1 = 0$ преобразует двоичный код C_3, C_2, C_1 триггеров состояний первой группы и при $D_3 = D_2 = D_1 = C_3 = C_2 = C_1 = 000$ выполняет трансляцию (передачу) двоично-К-ичного кода младшего разряда в двоичный код эквивалента на выходе.

Двухшаговый ПК работает следующим образом. Пусть требуется преобразовать входной 12-ричный код числа

$$\begin{aligned} A_0 = 9238_{12} &= 1001 * 0010 * 0011 * 1000_{(2,12)} = 9 * 12^3 + 2 * 12^2 + 3 * 12^1 + 8 * 12^0 = \\ &= 9 * 1728 + 2 * 144 + 3 * 12 + 8 = 15884_{10} \end{aligned} \quad (3)$$

Для определенности примем, что дешифраторы превышения 11 настроены на определение значений разрядных цифр $x_i \geq 2$. Следовательно, в этом случае каждый разряд $D_m (m = \overline{1,3})$ триггеров 4 второй группы состояний разрядных счетчиков равен 1, если соответствующий разряд преобразуемого кода имеет значение $x_i \geq 2$, в противном случае $D_m = 0$. Шифраторы 12 формируют выходное слово, меньшее на 2 единицы по отношению к входному слову.

ФЭ 13 в данном случае реализует функцию

$$S = \begin{cases} 12C_1 + 144C_2 + 1728C_3; D_1 = D_2 = D_3 = 0; \\ 24D_1 + 288D_2 + 3456D_3; D_1, D_2, D_3 \neq 0. \end{cases} \quad (4)$$

Преобразование двоичных кодов триггеров состояний первой 3 и второй 4 группы соответствует табл. 1. В табл. 1 S означает общий вид эквивалента, а S_{10} – десятичный код эквивалента.

Таблица 1

Вы- хо- ды DC	Состояние триггеров		S	S ₁₀	Двоичный код эквивалента		
	Второй группы	Первой группы					
Z _i	D ₃ D ₂ D ₁	C ₃ C ₂ C ₁			Y ₁₁ Y ₁₀ Y ₉ Y ₈	Y ₇ Y ₆ Y ₅ Y ₄	Y ₃ Y ₂ Y ₁ Y ₀
Z ₀	000	000	x ₀	x ₀	Трансляция тетрады x ₀		
Z ₁	000	001	K	12	0000	0000	1100
Z ₂	000	010	K ²	144	0000	1001	0000
Z ₃	000	011	K ² + K	156	0000	1001	1100
Z ₄	000	100	K ³	1728	0110	1100	0000
Z ₅	000	101	K ³ + K	1740	0110	1100	1100
Z ₆	000	110	K ³ + K ²	1872	0111	0101	0000
Z ₇	000	111	K ³ + K ² + K	1884	0111	0101	1100
Z ₈	001	XX1	aK	24	0000	0001	1000
Z ₉	010	X1X	aK ²	288	0001	0010	0000
Z ₁₀	011	X11	aK ² + aK	312	0001	0011	1000
Z ₁₁	100	1XX	aK ³	3456	1101	1000	0000
Z ₁₂	101	1X1	aK ³ + aK	3480	1101	1001	1000
Z ₁₃	110	11X	aK ³ + aK ²	3744	1110	1010	0000
Z ₁₄	111	111	aK ³ + aK ² + aK	3768	1110	1011	1000

3. Параллельная стратегия

Дальнейшего увеличения быстродействия ПК можно достичь или за счет увеличения числа шагов в наборе до трех 1, a, b (b – третий шаг) при сохранении принципа последовательного использования шагов преобразования (вначале b, затем a и в заключение шаг 1), или за счет параллельного (одновременного) вычитания различных шагов из различных разрядных цифр. Усиление локального параллелизма достигается допущением возможности одновременного использования различных шагов преобразования в различных старших разрядах числа.

Выражение для формулы максимального числа тактов преобразования двухшагового ПК с параллельным использованием шагов преобразования получить в явном виде затруднительно. Определить же значение максимального числа тактов преобразования можно путем моделирования процесса преобразования для различных наборов шагов и путем анализа результатов (табл. 2).

Анализ табл.2 показывает, что для K = 12 в двухшаговом ПК с параллельным использованием шагов преобразования минимальное значение числа тактов преобразования равно пяти для наборов (1, 3); (1, 4); (1, 5). Подобный анализ, проведенный для наборов 1,6 – 1,11, дает рост числа тактов преобразования до шести – десяти.

Таким образом, по сравнению с принципом последовательного использования шагов преобразования число тактов сокращается на один (5 вместо 6). Среди трех наборов шагов

преобразования оптимальным по дополнительному критерию минимума аппаратных затрат является набор 1,4, при использовании которого отсутствуют затраты ЛЭ на шифратор 12 (табл. 2). В табл. 2 NT означает номер такта преобразования.

Таблица 2

NT	K=12 Набор шагов (1,2)	K=12 Набор шагов (1,3)
	11 10 9 8 7 6 5 4 3 2 1 0	11 10 9 8 7 6 5 4 3 2 1 0
1	9 8 7 6 5 4 3 2 1 0 0 0	8 7 6 5 4 3 2 1 0 2 0 0
2	7 6 5 4 3 2 1 0 0 0 0 0	5 4 3 2 1 0 1 0 0 1 0 0
3	5 4 3 2 1 0 0 0 0 0 0 0	2 1 0 1 0 0 0 0 0 0 0 0
4	3 2 1 0 0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0 0 0 0 0
5	1 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0
6	0 0 0 0 0 0 0 0 0 0 0 0	
NT	K=12 Набор шагов 1,4	K=12 Набор шагов 1,5
	11 10 9 8 7 6 5 4 3 2 1 0	11 10 9 8 7 6 5 4 3 2 1 0
1	7 6 5 4 3 2 1 0 2 1 0 0	6 5 4 3 2 1 0 3 2 1 0 0
2	3 2 1 0 2 1 0 0 1 0 0 0	1 0 3 2 1 0 0 2 1 0 0 0
3	2 1 0 0 1 0 0 0 0 0 0 0	0 0 2 1 0 0 0 1 0 0 0 0
4	1 0 0 0 0 0 0 0 0 0 0 0	0 0 1 0 0 0 0 0 0 0 0 0
5	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0

При K=10 и параллельном использовании шагов преобразования минимальными наборами шагов, обеспечивающими преобразование за 4 такта, являются наборы 1,3 и 1,4.

Лучший среди них - набор 1,4 (по причине простой реализации шифраторов 12).

Повышение быстродействия двухшаговых ПК с параллельным использованием шагов преобразования достигается ценой некоторого усложнения ФЭ и ЛЭ для управления вычитанием шагов а и 1 из содержимого разрядных счетчиков.

Логика управления в двухшаговом ПК параллельного типа выполнена так, чтобы запретить возможность вычитания шага 1, если в этом разряде имеется возможность вычитания шага а. И наоборот, если значение разряда x_m находится в пределах $1 \leq x_m < a$, то следует разрешить опрос вентиля, управляющего вычитанием 1 из разрядного счетчика, хранящего x_m .

Этот принцип управления реализуется в двухшаговом ПК параллельного типа путем замены (n-1) входного элемента ИЛИ-НЕ блоком инверторов (инверторы 17₁, 17₂), вход каждого из которых связан с единичным выходом соответствующего триггера старшего регистра состояний РГ₄, а выход соответствующего инвертора соединяется с управляющими входами схем И 8₁, И 8₂, на информационные входы которых поступают сигналы с единичных выходов триггеров этого разряда младшего регистра состояний РГ₃ (рис. 2).

Закон функционирования ФЭ для двухшагового ПК параллельного типа для набора шагов 1,2 и K=12 приведен в табл. 3. Детально параллельная стратегия приведена в [2,3].

Сравнение табл. 1 и 3 показывает, что ФЭ двухшагового ПК параллельного типа имеет 27 строк; ПК последовательного типа (см. табл.1) - всего 15 строк.

4. Программное средство для расчета аппаратных затрат в ФЭ

Программное средство предназначено для получения значений эквивалентов, которые должен выдавать блок ФЭ, а также для расчета аппаратных затрат в долях корпусов ИМС на реализацию каждой функции выхода ФЭ. Программа рассчитана на ПК, функционирующие по методу накопления эквивалентов.

Для того чтобы произвести расчет в данном режиме, необходимо запустить программу путем активизации исполняемого модуля TRANS.exe.

После запуска программы появится пять текстовых полей для ввода данных. С помощью этих элементов управления необходимо задать исходные данные для режима преобразования чисел: основание системы счисления на входе (3-15), разрядность преобразуемого числа (1-24), количество блоков (1-24), количество шагов преобразователя (1-8) и их значения, тип преобразователя (последовательный или параллельный).

Таблица 3

Номер набора	Состояние триггеров		S	S ₁₀	Двоичный код эквивалента		
	Второй группы	Первой группы					
	Z _i	D ₃ D ₂ D ₁			Y ₁₂ Y ₁₁ Y ₁₀ Y ₉	Y ₈ Y ₇ Y ₆ Y ₅	Y ₄ Y ₃ Y ₂ Y ₁
0		000	X ₀	X ₀	Трансляция младшей тетрады		
1		000	K ¹	12	0000	000	1100
2		000	K ²	144	0000	1001	0000
3		000	K ² + K ¹	156	0000	1001	1100
4		000	K ³	1728	0110	1100	0000
5		000	K ³ + K ¹	1740	0110	1100	1100
6		000	K ³ + K ²	1872	0111	0101	0000
7		000	K ³ + K ² + K ¹	1884	0111	0101	1100
8		001	aK	24	0000	0001	1000
9		001	K ² + aK	168	0000	1010	1000
10		001	K ³ + aK	1752	0110	1101	1000
11		001	K ³ + K ² + aK	1896	0111	0110	1000
12		010	aK ²	288	0001	0010	0000
13		010	aK ² + K	300	0001	0010	1100
14		010	K ³ + aK ²	2016	0111	1110	0000
15		010	K ³ + aK ² + K	2028	0111	1110	1100
16		011	aK ² + aK	312	0001	0011	1000
17		011	K ³ + aK ² + aK	2040	0111	1111	1000
18		100	aK ³	3456	1101	1000	0000
19		100	aK ³ + K	3468	1101	1000	1100
20		100	aK ³ + K ²	3600	1110	0001	0000
21		100	aK ³ + K ² + K	3612	1110	0001	1100
22		101	aK ³ + aK	3480	1101	1001	1000
23		101	aK ³ + K ² + aK	3624	1110	0010	1000
24		110	aK ³ + aK ²	3744	1110	1010	0000
25		110	aK ³ + aK ² + K	3756	1110	1010	1100
26		111	aK ³ + aK ² + aK	3768	1110	1011	1000

После ввода всех необходимых данных необходимо нажать на кнопку «ОК». Если параметры были заданы некорректно, то выводится соответствующее сообщение. После выполнения расчет на экране появится результат преобразования.

На рис.3,4 приведены результаты преобразования для основания системы счисления 12, двух блоков и шести разрядов. В качестве весов шагов были заданы значения 1 и 4.

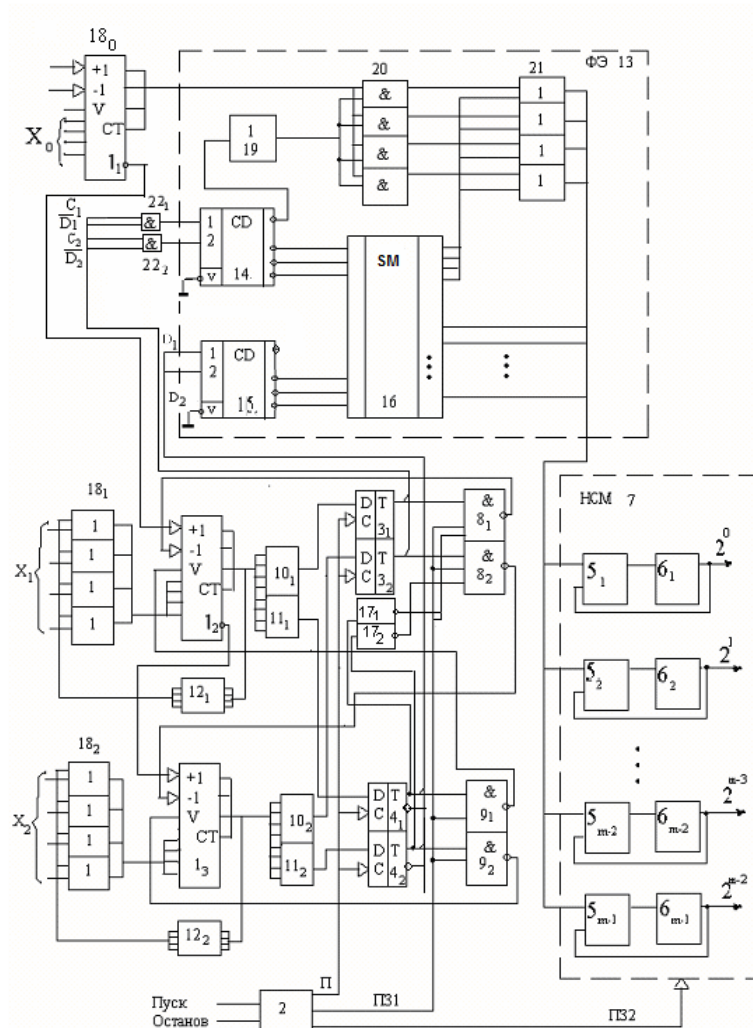


Рис. 2. Двухшаговый ПК с параллельным использованием шагов

Результат				
Разряды: 1...3				
Выход дешифратора	Код состояния регистра	Код эквивалента десятичный	Двоичный	
z 1	000 000	0	000 0000 0000 0000 0000 0000 0000	0000 0000
z 2	000 001	1	000 0000 0000 0000 0000 0000 0000	0000 0001
z 3	000 010	12	000 0000 0000 0000 0000 0000 0000	0000 1100
z 4	000 011	13	000 0000 0000 0000 0000 0000 0000	0000 1101
z 5	000 100	144	000 0000 0000 0000 0000 0000 1001	0000 0000
z 6	000 101	145	000 0000 0000 0000 0000 0000 1001	0001 0001
z 7	000 110	156	000 0000 0000 0000 0000 0000 1001	0000 1100
z 8	000 111	157	000 0000 0000 0000 0000 0000 1001	0001 1101
z 9	001 001	4	000 0000 0000 0000 0000 0000 0000	0000 0100
z 10	001 011	52	000 0000 0000 0000 0000 0000 0011	0100 0100
z 11	001 101	580	000 0000 0000 0000 0000 0010 0100	0100 0100
z 12	001 111	628	000 0000 0000 0000 0000 0010 0111	0100 0100
z 13	010 010	48	000 0000 0000 0000 0000 0000 0011	0000 0000
z 14	010 011	52	000 0000 0000 0000 0000 0000 0011	0100 0100
z 15	010 110	624	000 0000 0000 0000 0000 0010 0111	0000 0000
z 16	010 111	628	000 0000 0000 0000 0000 0010 0111	0100 0100
z 17	011 011	52	000 0000 0000 0000 0000 0000 0011	0100 0100
z 18	011 111	628	000 0000 0000 0000 0000 0010 0111	0100 0100
z 19	100 100	576	000 0000 0000 0000 0000 0010 0100	0000 0000
z 20	100 101	580	000 0000 0000 0000 0000 0010 0100	0100 0100
z 21	100 110	624	000 0000 0000 0000 0000 0010 0111	0000 0000
z 22	100 111	628	000 0000 0000 0000 0000 0010 0111	0100 0100
z 23	101 101	580	000 0000 0000 0000 0000 0010 0100	0100 0100
z 24	101 111	628	000 0000 0000 0000 0000 0010 0111	0100 0100
z 25	110 110	624	000 0000 0000 0000 0000 0010 0111	0000 0000
z 26	110 111	628	000 0000 0000 0000 0000 0010 0111	0100 0100
z 27	111 111	628	000 0000 0000 0000 0000 0010 0111	0100 0100

y1 = (~z0) (~z2) (~z4) (~z6)
y2 = 0
y3 = (~z1) (~z3) (~z5) (~z7) (~z9) (~z11) (~z13) (~z15) (~z17) (~z19) (~z21) (~z23) (~z25)
y4 = (~z1) (~z3) (~z5) (~z7) (~z9) (~z11) (~z13) (~z15) (~z17) (~z19) (~z21) (~z23) (~z25)
y5 = (~z3) (~z5) (~z7) (~z9) (~z11) (~z13) (~z15) (~z17) (~z19) (~z21) (~z23) (~z25)
y6 = (~z5) (~z7) (~z9) (~z11) (~z13) (~z15) (~z17) (~z19) (~z21) (~z23) (~z25)
y7 = (~z7) (~z9) (~z11) (~z13) (~z15) (~z17) (~z19) (~z21) (~z23) (~z25)
y8 = (~z9) (~z11) (~z13) (~z15) (~z17) (~z19) (~z21) (~z23) (~z25)
y9 = 0
y10 = (~z9) (~z11) (~z13) (~z15) (~z17) (~z19) (~z21) (~z23) (~z25)

Рис. 3. Форма приложения в режиме преобразования для K=12

Результат

z

25

110

110

1078272

000

0001

0000

0111

0100

0000

0000

z

26

110

111

1085184

000

0001

0000

1000

1111

0000

0000

z

27

111

111

1085184

000

0001

0000

1000

1111

0000

0000

y1 = 0

y2 = 0

y3 = 0

y4 = 0

y5 = 0

y6 = 0

y7 = (~z0)|(~z2)|(~z4)|(~z6)

y8 = (~z0)|(~z2)|(~z4)|(~z6)

y9 = (~z1)|(~z3)|(~z5)|(~z7)|(~z9)|(~z11)|(~z13)|(~z15)|(~z17)|(~z19)|(~z21)|(~z23)|(~z25)|(~z27)

y10 = (~z0)|(~z2)|(~z4)|(~z6)|(~z8)|(~z10)|(~z12)|(~z14)|(~z16)|(~z18)|(~z20)|(~z22)|(~z24)|(~z26)

y11 = (~z0)|(~z2)|(~z4)|(~z6)|(~z8)|(~z10)|(~z12)|(~z14)|(~z16)|(~z18)|(~z20)|(~z22)|(~z24)|(~z26)

y12 = (~z3)|(~z5)|(~z7)|(~z9)|(~z11)|(~z13)|(~z15)|(~z17)|(~z19)|(~z21)|(~z23)|(~z25)|(~z27)

y13 = (~z1)|(~z3)|(~z5)|(~z7)|(~z9)|(~z11)|(~z13)|(~z15)|(~z17)|(~z19)|(~z21)|(~z23)|(~z25)|(~z27)

y14 = (~z6)|(~z8)|(~z10)|(~z12)|(~z14)|(~z16)|(~z18)|(~z20)|(~z22)|(~z24)|(~z26)

y15 = (~z1)|(~z3)|(~z5)|(~z7)|(~z9)|(~z11)|(~z13)|(~z15)|(~z17)|(~z19)|(~z21)|(~z23)

y16 = (~z3)|(~z5)|(~z7)|(~z9)|(~z11)|(~z13)|(~z15)|(~z17)|(~z19)|(~z21)|(~z23)

y17 = (~z3)|(~z5)|(~z7)|(~z9)|(~z11)|(~z13)|(~z15)|(~z17)|(~z19)|(~z21)|(~z23)

y18 = (~z3)|(~z5)|(~z7)|(~z9)|(~z11)|(~z13)|(~z15)|(~z17)|(~z19)|(~z21)|(~z23)

y19 = (~z5)|(~z7)|(~z9)|(~z11)|(~z13)|(~z15)|(~z17)|(~z19)|(~z21)|(~z23)

y20 = (~z9)|(~z11)|(~z13)|(~z15)|(~z17)|(~z19)|(~z21)|(~z23)

y21 = (~z10)|(~z12)|(~z14)|(~z16)|(~z18)|(~z20)|(~z22)|(~z24)|(~z26)

y22 = 0

y23 = 0

y24 = 0

y25 = 0

y26 = 0

y27 = 0

Таблица аппаратных затрат:

y00

y01

y02

y03

y04

y05

y06

y07

y08

y09

y10

y11

y12

y13

y14

y15

y16

y17

2x4 И-НЕ

1/4

1/4

1/4

1/4

1/4

1/4

4x2 И-НЕ

8x1 И-НЕ

1/1

1/1

1/1

1/1

Рис. 4. Выведение результирующей функции и таблицы аппаратных затрат

В качестве инструмента для разработки программного пакета «TRANS» была использована среда программирования Microsoft Visual Studio 2005. Данная среда программирования частично использует принципы RAD. RAD означает быструю разработку приложений. Эта концепция позволяет создавать программные продукты, причем особое внимание уделяется скорости и удобству программирования, созданию технологичного процесса, позволяющего программисту максимально быстро писать компьютерные программы.

Для задания первичных значений работы программы предусмотрены текстовые поля «Основание системы счисления», «Число разрядов», «Число блоков», «Число шагов» и «Вес шагов».

Для переключения типа преобразователей с последовательного на параллельный и наоборот используется специальная кнопка. Такое решение позволит максимально облегчить и упростить интерфейс программы, сделать его максимально понятным. Определение типа преобразователя реализовано с помощью переменной bool bParallel. Это булева переменная, которая может принимать только два значения: true или false. По умолчанию переменной присвоено true, т.е. будет применяться параллельный тип преобразователя.

В процессе своей работы функция использует следующие структуры данных:

1) Переменная iBaseNumber – определяет основание системы счисления, из которой будем преобразовывать. Значение переменной может варьироваться от 3 до 15, т.е. основание системы может изменяться от 3 до 15. По умолчанию переменной присвоено значение, равное 10, т.е. по умолчанию установлена десятичная система счисления.

2) Переменная iAmountOfBlocks – определяет количество блоков. Значение переменной может варьироваться от 1 до 24, т.е. количество блоков может изменяться от 1 до 24. По умолчанию переменной присвоено значение, равное 1, т.е. по умолчанию установлен один блок.

3) Переменная iAmountOfSteps – определяет количество шагов. Значение переменной может варьироваться от 1 до 8, т.е. количество шагов может изменяться от 1 до 8. По умолчанию переменной присвоено значение, равное 2, т.е. по умолчанию установлено два шага.

4) Переменная iAmountOfDigits – определяет количество разрядов. Значение переменной может варьироваться от 1 до 24, т.е. количество разрядов может изменяться от 1 до 24. По умолчанию переменной присвоено значение, равное 2, т.е. по умолчанию установлено два разряда.

5) Массив SW хранит в себе веса шагов. По умолчанию записано значение {1,4}, т.е. по умолчанию веса шагов установлены в значения 1 и 4. Размерность массива равна 80 значений.

6) Массив *equivs* хранит в себе коды эквивалентов. Размерность массива равна 80 значений.

Выводы

Основные результаты. Рассмотрены две стратегии преобразования чисел по методу накопления эквивалентов. Показано, что применение параллельной стратегии позволяет по сравнению с последовательной сократить число тактов преобразования на 20-25%. Разработан программный пакет TRANS, позволяющий выполнять построение таблицы, законов функционирования ФЭ для ПК с параллельной стратегией и расчет аппаратных затрат ИМС на построение каждого блока ФЭ.

Сравнение с лучшими аналогами. Предложенный программный пакет TRANS позволяет выполнять построение таблиц и анализ затрат для ПК с параллельной стратегией в то время, как программный пакет FE_LION может выполнять аналогичные функции только для ПК с последовательной стратегией.

Практическая значимость. Программное средство TRANS позволяет проводить детальный анализ этапа системного проектирования ПК параллельного типа, сократить аппаратные затраты на реализацию всего ПК за счет нахождения оптимального разбиения на блоки, сократить число тактов преобразования за счет параллельной стратегии.

Список литературы: 1. А.С. 1647908 5Н03М 7/12. Преобразователь двоично-К-ичного кода в двоичный код //Н.Я.Какурин, Ю.К. Кирьяков, А.Н. Макаренко //Открытия, изобретения. 1984. №17. С. 262-263. 2. Какурин Н.Я., В.В. Варца, С.Н. Коваленко. Параллельная стратегия использования шагов в двухшаговых преобразователях кода //АСУ и приборы автоматизации. 2007. Вып.141. С.29-36. 3. Какурин Н.Я., Лопухин Ю.В., Макаренко А.Н. Об одном способе повышения быстродействия преобразователей кодов //АСУ и приборы автоматизации. 2003. Вып.122. С.72-83.

Поступила в редколлегию 07.05.2010

Какурин Николай Яковлевич, канд. техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: прикладная теория цифровых автоматов, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Лопухин Юрий Владимирович, ст. преподаватель кафедры АПВТ ХНУРЭ. Научные интересы: проектирование программного обеспечения, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Бочаров Евгений Витальевич, студент группы КИ-07-6 ХНУРЭ. Научные интересы: автоматизация проектирования цифровых устройств, проектирование программного обеспечения. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Варца Виталий Викторович, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: проектирование программного обеспечения, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Макаренко Анна Николаевна, канд. техн. наук, доцент кафедры высшей математики и информационных технологий Харьковского банковского института. Научные интересы: дискретная математика, анализ и синтез преобразователей код-код. Адрес: Украина, 61074, Харьков, пр. Победы, 55, тел. 336-05-64.