

ЛИТЕРАТУРА

1. Самсонов В. А., Полонников Д. Е. Электрометрический усилитель МДМ типа на МОП-транзисторах.— «Измерительная техника», 1976, № 9, с. 52—55.
2. Голуб Ю. Г., Купер Э. А., Леденев А. В., Нифонтов В. И., Романовский В. Р., Тоубин В. Г. К вопросу выбора структуры и способах реализации схемы интегрирующего цифрового вольтметра для информационно-измерительных систем.— В кн.: Тезисы докладов Всесоюзной конференции ИИС-75. Т. 2. Кишинев, изд. НИИНТИ, 1975.
3. Корчуганов В. Н., Кулипанов Г. Н., Мезенцев Н. А., Салдин Е. Л., Скринский А. Н. Использование синхронного излучения для измерения абсолютной энергии электронов в накопителе. Препринт № 77-83, Новосибирск, изд. ИЯФ СО АН СССР, 1977.

Поступило в редакцию 31 января 1978 г.

УДК 681.3.06

И. Х. КОРНЯ
(Кишинев)

ПРОГРАММНЫЕ СРЕДСТВА ГЕНЕРИРОВАНИЯ ПОЗИЦИОННЫХ КОДОВ УПРАВЛЕНИЯ В АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ КОНТРОЛЯ ЭЛЕКТРОННОЙ АППАРАТУРЫ

Введение. В автоматизированных системах контроля (АСК) электронной аппаратуры для управления программируемыми блоками широко используется позиционное управление, предполагающее формирование и передачу на программно-управляемые блоки позиционных кодов [1]. В данной работе определяется необходимый и достаточный набор логических операций для формирования позиционного кода управления (ПКУ), а также приводятся соответствующие программные средства для описания процесса генерирования позиционных кодов управления.

Позиционное кодовое управление. Под позиционным кодом управления понимается значение булева вектора произвольной разрядности

$$X = (x_1, x_2, \dots, x_n),$$

где x_1, x_2, \dots, x_n — двоичные переменные.

В общем случае при формировании ПКУ значение каждого разряда вычисляется как некоторая сингулярная функция $f(x)$. Таким образом, ПКУ можно рассматривать как значение вектор-функции

$$F(x) = [f(x_1), f(x_2), \dots, f(x_n)].$$

Как известно [2], значение функции $f(x)$ может быть вычислено следующими логическими операциями: $f(x) = 1$ (константа 1), $f(x) = 0$ (константа 0), $f(x) = x$ (переменная x), $f(x) = \bar{x}$ (отрицание x).

Для решения задач автоматического синтеза контролируемых тестов, моделирования цифровых схем и обработки информации в АСК указанный набор логических операций должен быть дополнен базисной системой логических операций. Этому требованию отвечает дополнение имеющегося набора ($\equiv 1, \equiv 0, x, \bar{x}$) логическими операциями дизъюнкции (\vee) и конъюнкции (\wedge). Операции дизъюнкции и конъюнкции над булевым вектором X понимаются следующим образом:

$$\wedge X = (x_1 \wedge x_2 \wedge x_3, \dots, \wedge x_n); \quad (1)$$

$$\vee X = (x_1 \vee x_2 \vee x_3, \dots, \vee x_n). \quad (2)$$

Таким образом, данный набор логических операций в принципе достаточен для вычисления ПКУ в АСК. Однако практика показывает, что для реализации более сложных алгоритмов вычисления ПКУ, а также упрощения процесса их формирования требуются дополнительные операции, количество которых, как правило, ограничивается чрезмерным усложнением средств программирования и их реализуемостью на ЭВМ. Учитывая это обстоятельство, ограничим набор логических операций следующими:

- | | |
|--|--|
| 1 — константа 1 ($\equiv 1$); | \neq — поразрядное сравнение (\oplus); |
| 0 — константа 0 ($\equiv 0$); | \succ — циклический сдвиг вправо; |
| \equiv — переменная (тождественность); | \prec — циклический сдвиг влево; |
| $\bar{\quad}$ — отрицание (инверсия); | $+$ — суммирующий двоичный счет; |
| \vee — дизъюнкция; | $-$ — вычитающий двоичный счет; |
| \wedge — конъюнкция; | $!$ — случайное определение. |

Операции циклического сдвига вправо и влево на один разряд и поразрядного сравнения формально описываются следующими выражениями:

$$\begin{aligned} >X = (x_n, x_1, x_2, \dots, x_{n-1}); \\ <X = (x_2, x_3, \dots, x_n, x_1); \\ \neq(X, Y) = (x_1 \oplus y_1, x_2 \oplus y_2, \dots, x_n \oplus y_n). \end{aligned} \quad (3)$$

Программные средства. Программное обеспечение АСК должно содержать соот-

Определим синтаксис данного оператора, обеспечивающий построение правильных и легко распознаваемых конструкций языка. При описании синтаксиса конструкций языка, задающих ПКУ, будем использовать нормальные обозначения Бэкуса *. В качестве алфавита грамматики используются символы русского алфавита, цифры и особые символы:

⟨особые символы⟩ : := ∨ | ∧ | ¬ | ≠ | + | − | = | > | < | / | [|] | − | · | !

Приведем структуру записи выражения, определяющую формирование ПКУ:

ЗАДАТЬ ⟨идентификатор ПКУ⟩ { = ⟨логическая операция⟩ { · } ? } ⟨операнд⟩ !

⟨логическая операция⟩ : := ∨ | ∧ | ¬ | = | ≠ | + | − | > | < | / | 0 | !

⟨операнд⟩ : := { ⟨идентификатор ПКУ⟩ } ? ! ⟨область определения⟩ { ⟨область определения⟩ } ?

⟨область определения⟩ : := [⟨переменная⟩ { / ⟨переменная⟩ } ? !]

⟨переменная⟩ : := ⟨цифра⟩ | ⟨идентификатор числа⟩ { ⟨цифра⟩ | ⟨идентификатор числа⟩ − ⟨цифра⟩ | ⟨идентификатор числа⟩ }

⟨идентификатор числа⟩ : := ⟨буква⟩ | ⟨цифра⟩.

В таблице приведены конкретные примеры описания процесса генерирования ПКУ на языке ТЕСТ. В примерах используются следующие идентификаторы ПКУ: Т1, Т2, Т3 и Т4. Для простоты изложения каждый из них имеет 4 переменных с номерами 1, 2, 3 и 4. Примеры раскрывают также семантику конструкций языка.

Номер п/п	Используемые логические операции	Запись конструкций языка для вычисления ПКУ	Сформированный ПКУ (значение вектора)
1	1,0	T1=1[1/2]=0[3/4]	T1=(1, 1, 0, 0)
2	−	T2=−T1[1−4]	T2=(0, 0, 1, 1)
3	=	T3==T2[1−4]	T3=(0, 0, 1, 1)
4	>	T3=>[1−4]	T3=(1, 0, 0, 1)
5	<	T2=<[1−4]	T2=(0, 1, 1, 0)
6	+	T3=+T2[1−4]	T3=(1, 1, 1, 0)
7	−	T4=−T3[1−4]	T4=(0, 1, 1, 0)
8	∧	T3=∧[2−4]	T3=(1, 0, 1, 0)
9	∨	T3=∨T1/T2[4/1]	T3=(1, 0, 1, 1)
10	≠	T1=≠T2/T3[1−4]	T1=(1, 1, 0, 1)

Строка 1 реализует операции «Константа 1» над переменными 1, 2 и «Константа 0» над переменными 3, 4. При определяемом идентификаторе ПКУ допускается перечислять несколько логических операций, связанных знаком (=). Сформированный ПКУ передается на внешнее устройство Т1. Строка 2 реализует операцию инверсии над переменными (1—4) Т1 и соответственно присваивает полученные значения переменным (1—4) Т2, при этом за исходное значение Т1 принимается результат вычислений в строке 1. Строка 3 формирует значение Т3, равное значению Т2, вычисленному в строке 2. Строки 4 и 5 осуществляют соответственно циклический сдвиг вправо и влево на один разряд. Строки 6 и 7 реализуют соответственно операции суммирующего и вы-

* Нормальные обозначения Бэкуса дополним знаками ? и !, определяющими соответственно возможность отсутствия и многократного повторения конструкций, ограниченных { }.

читающего двоичного счета. При циклическом выполнении этих строк значения переменных идентификатора изменяются аналогично разрядам 4-разрядного счетчика. Строка 8 реализует логическую операцию конъюнкции согласно (1). Операция конъюнкции выполняется над переменными 1—4, а результат записывается как значение первой переменной из области определения, т. е. переменной 2. Строка 9 реализует дизъюнкцию над переменными 4, 1 согласно (2) совместно для T1 и T2. Строка 10 формирует значение переменных (1—4) как результат поразрядного сравнения T2 и T3 согласно (3).

Для записи результатов по разрядам (независимо от указанных в первой области определения) предусмотрена запись с использованием конструкций, содержащих знак (·) при логических операциях и вторую область определения. Например, выражение ЗАДАТЬ T1 = $\neg \cdot [1/2][3/4]$ инвертирует значения переменных (1, 2) T1, а результат записывается по разрядам (3, 4) T1.

Решение ряда задач контроля электронной аппаратуры на языке ТЕСТ требует программного изменения области определения (замена переменных по условию). Для таких случаев предусмотрено использование в качестве переменных области определения идентификаторов числа. Заменяя в предыдущем примере переменные (1—4) на идентификаторы числа A1—A4, получим выражение ЗАДАТЬ T1 = $\neg \cdot [A1/A2][A3/A4]$. Это выражение в зависимости от значений идентификаторов числа теперь применимо в различных блоках рабочих программ контроля.

В заключение отметим, что определенный набор логических операций и разработанные средства программирования реализованы на ЭВМ «Электроника-100И» и используются в АСК электронной аппаратуры. Эти программные средства описания процессов генерирования позиционных кодов управления могут найти также применение в различных АСУ ТП.

ЛИТЕРАТУРА

1. Кудрицкий В. Д., Сеница М. А., Чинаев П. И. Автоматизация контроля РЭА. М., «Сов. радио», 1977.
2. Поспелов Д. А. Логические методы анализа и синтеза схем. М., «Энергия», 1974.
3. Буровцев В. А., Ибрагимов К. Ш., Подзин А. Е. Система аппаратно-функционального контроля и диагностики цифровых схем, управляемая малой ЭВМ.— «Упр. сист. и маш.», 1976, № 3, с. 44—48.
4. Ибрагимов К. Ш., Корня И. Х., Подзин А. Е. Возможности математического обеспечения автоматической системы диагностики цифровых узлов.— В кн.: Вопросы электроники. Кишинев, «Штиинца», 1977.

Поступило в редакцию 15 декабря 1977 г.

УДК 681.3.06

Б. З. КИРИЛЕНКО
(Кишинев)

РЕАЛИЗАЦИЯ АЛГОРИТМОВ ФОРМИРОВАНИЯ ПОЗИЦИОННЫХ КОДОВ ПРОИЗВОЛЬНОЙ РАЗРЯДНОСТИ

Функциональные возможности и быстродействие системного программного обеспечения автоматизированных систем контроля (АСК) электронной аппаратуры в значительной степени зависят от применяемых алгоритмов генерирования позиционных кодов управления (ПКУ). В то же время вопросам разработки и реализации эффективных алгоритмов генерирования ПКУ в литературе уделяется мало внимания.

В настоящей работе предлагается один из возможных алгоритмов формирования ПКУ, который обеспечивает достаточно высокое быстродействие при экономном использовании памяти ЭВМ. Программный модуль, реализующий данный алгоритм, — один из модулей интерпретатора с языка ТЕСТ [1]. Рассматриваемый алгоритм формирования позиционных кодов реализован в виде единой управляющей программы, имеющей n разветвлений для вычисления заданных операций, приведенных в работе [2]. Список реализуемых операций включает следующие операции: «Константа 0», «Константа 1», тождественность, инверсия, дизъюнкция, конъюнкция, циклические сдвиги вправо и влево на 1 разряд, суммирующий и вычитающий двоичный счет и др.

Исходные данные для управляющей программы — описание позиционного управления на языке ТЕСТ, включающее логическую операцию и обрабатываемый операнд [2, 3].