

А.Ш. БОЗОЯН, Ш.Е. БОЗОЯН

О ВОССТАНОВЛЕНИИ СИНХРОНИЗАЦИИ ФУНКЦИОНАЛЬНОЙ СХЕМЫ

Չորմասկում սխեմաների նախագծան իդեալիզացիայում նախընտրվող (ներառյալով է. որ սխեմայի տարրերն իրեն օգտագործելի արդյունքները փոխանցվում են «սխեմարարուն» խումբին սարքերային մաք. աղբյուր սարմարանները փոխանցվում են որպեսզի մոնիտորինգի անցնելիս հարմար է սխեմայի համապատասխանության վերականգնման խնդիրը որը արտահանում է սխեմայի այս կամ այն հանգույրներում սխեմային նախագծան սարքերը կիսապատկերի համապատասխան ինտեր լիան փոխանցվում է կրու կոմպանիաներում փոխանցվում արդյունավետ արգրվելու

При переходе от идеализированного варианта проектирования функциональных схем (где предполагается, что элементы схемы входу информации обрабатывают «мгновенно») - реальному (когда элементы обрабатывают информацию за определенное время) возникает проблема восстановления синхронизации схемы, которая требует в определенных узлах схемы добавления элементов задержки. На основании строчного описания схемы разработан эффективный алгоритм восстановления ее синхронизации.

Ил. 1. Библиогр. 1 назв.

During transition of an idealized case of functional schemes design (when there is assumed that all elements of the scheme process the input information "immediately" to real case (where it is presumed that elements take some time to process the information), there occurs problem of restoring the scheme synchronization. The latter implies addition of delay elements to specific spots in the scheme. Based on the line-by-line scheme description language, an effective algorithm of restoring scheme's synchronization is developed.

Ил. 1. Ref. 1

При логическом проектировании дискретных схем весьма актуальным является рассмотрение схем из функциональных элементов. На первом этапе проектирования обычно пропускается фактор задержек элементов и проводов, соединяющих элементы между собой. Иными словами, принимается, что элементы срабатывают входную информацию «мгновенно», а провода, соединяющие элементы, передают ее также «мгновенно». На втором же этапе, учитывая фактор наличия задержек в элементах и проводах, в соответствующих местах схемы добавляются элементы с задержками, обеспечивающие временную синхронизацию и, тем самым, работоспособность схемы. Осуществление этого этапа называется *восстановлением временной синхронности* схемы.

Целью настоящей работы является разработка эффективного алгоритма восстановления временной синхронности функциональной схемы с использованием одного варианта строчного описания функциональной схемы [1].

Первоначально представим принцип работы языка строчного описания дискретных схем, затем формализацию искомого алгоритма

на данном языке. Использование описания схемы посредством данного языка эффективно тем, что оно (описание) является удобным инструментом как для описания самой схемы, так и для машинной реализации данного алгоритма.

Кратко опишем принцип строчного описания схемы из функциональных элементов (обобщение польской бесскобочной записи) [1]. Описание рассмотрим на конкретном примере, в котором, однако, общий принцип обрисовывается четко. Рассмотрим схему, изображенную на рисунке.

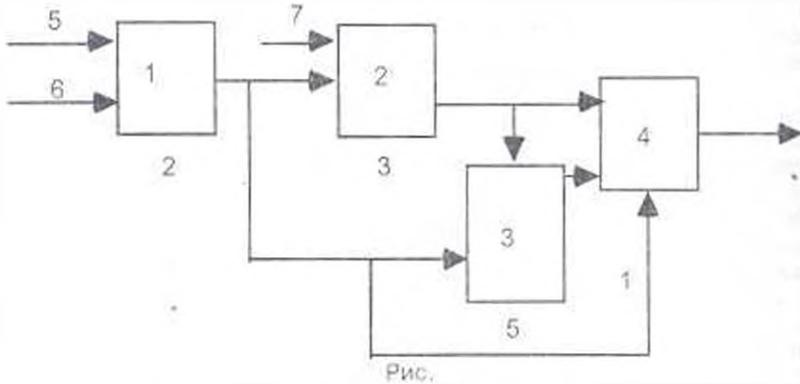


Рис.

Описание начинается с выходного элемента схемы (в данном случае, элемента под номером 4). Дается название элемента, в скобках - два параметра: число входов элемента (если элемент рассматривается впервые, в противном случае - ноль), число единиц задержки элемента. Затем рассматривается тот элемент, выход которого связан с первым (считая "слева") входом предыдущего элемента и т.д. Описав таким образом подсхему, выход которой соединен с первым входом выходного элемента, аналогично описываются подсхемы, выходы которых соединены со вторым, третьим и т.д. входами выходного элемента.

При написании названия данного элемента, если он рассматривается не впервые, больше не рассматриваются те подсхемы, выходы которых соединены со входами данного элемента. При этом входы схемы также считаются "элементами", которые не имеют входов и имеют нулевую задержку. Подробное описание этого языка можно найти в [1].

На рисунке под каждым элементом указано число единиц его задержки (время срабатывания его входной информации).

Таким образом, строчным описанием схемы, изображенной на рисунке, является:

4(3,1) 1(2,2) 6(0,0) 5(0,0) 3(2,5) 1(0,2) 2(2,3) 1(0,2) 7(0,0) 2(0,3).

Функциональная схема работает в тактовом режиме, т.е. на все входы сигналы подаются одновременно, распространяясь по элементам, и через некоторое время на выходе схемы появляется соответствующая "реакция" - определенный сигнал.

Схема работает безотказно, если при подаче на ее входы любого набора входных сигналов через строго определенное время на выходе схемы появляется тот сигнал, который бы появился в том случае, если все элементы схемы обрабатывали (а провода,

соединяющие элементы, передавали) входную информацию "мгновенно". В этом случае будем считать, что временная синхронизация схемы восстанавливается.

Поскольку у разных элементов задержки могут быть разные, возникает задача синхронизации работы элементов схемы таким образом, чтобы на любой из элементов схемы все входные сигналы поступали одновременно. Нижеопределенный алгоритм решает эту проблему. Он принимает запись несинхронизированной схемы и на выходе выдает запись уже синхронизированной схемы.

ОПИСАНИЕ АЛГОРИТМА

1. Все символы $i(m, n, 0, 0)$ записи схемы заменить на $i(0, n, 0, 0)$.
Перейти к п. 2.

2. Все символы $i(0, n, 0, 0)$ заменить на $i(0, n, 0, 1)$ в случаях, когда $i(0, n, 0, 0)$ в записи схемы не предшествует символ $i(m, n, 0, 1)$, где $m \neq 0$. Отдельно запомнить $i(0, n, 0, 1) \rightarrow i(0, n, 0, 0)$ и перейти к п. 3.

3. Отрезки вида $i(m, n, 0, 0) j_1(0, n_1, \sigma_1, 1) \dots j_m(0, n_m, \sigma_m, 1)$ схемы заменить на $i(0, n^*, 0, 1)$, где $n^* = \max(n_1, \dots, n_m) + n$; все символы $i(0, n, 0, 0)$ записать на $i(0, n^*, 1, 1)$.

Отдельно запомнить:

$$i(0, n^*, 0, 1) \rightarrow i(m, n, 0, 0) \tau_k(1, n^* - n_1 - n, 1, 1) j_1(0, n_1, \sigma_1, 1) \dots \\ \dots \tau_m(1, n^* - n_m - n, 1, 1) j_m(0, n_m, \sigma_m, 1), \text{ где } \tau_k(1, 0, 1, 1) = \emptyset (1 \leq k \leq m).$$

П. 3 алгоритма применять многократно до тех пор, пока в результате не получится выражение вида $i(0, n_p, 0, 1)$. Перейти к п. 4.

4. На результирующем выражении вида $i(0, n_p, 0, 1)$ слова (от начала записи) направо многократно применять все запомненные выражения, пока они применяются. Перейти к п. 5.

5. В записи схемы, полученной в результате реализации п. 4 данного алгоритма, все $i(m, n, \sigma_1, \sigma_2) [\tau(1, n, 1, 1)]$ заменить на $i(m, n) [\tau(1, n)]$, причем $i(0, n, 1, 1)$ — на $i(0, n_0)$, если в последовательности символу $i(0, n, 1, 1)$ предшествует $i(m, n_0, 0, 0)$ ($m \neq 0$). Перейти к п. 6.

6. Пусть максимальная среди задержек элементов схемы равна P . В записи $\tau(1, n)$ заменить на последовательность $\tau_1(1, P) \dots \tau_q(1, P) \tau(1, q)$, где $n = kP + q$. Перейти к п. 7.

7. Конец.

Замечание. Синхронизированная схема работает безотказно, если сигналы поступают на ее входы с периодичностью не чаще, чем в P задержек. Таким образом, общая задержка схемы равна не суммарной задержке элементов схемы "от входа до выхода", а лишь задержке элемента схемы с наибольшей задержкой.

Пример. Рассмотрим действие алгоритма на примере схемы, представленной на рисунке.

Исходное описание схемы дано:

4(3,1) 1(2,2) 6(0,0) 5(0,0) 3(2,5) 1(0,2) 2(2,3) 1(0,2) 7(0,0) 2(0,3)

После применения п.1 алгоритма получим

4(3,1,0,0) 1(2,2,0,0) 6(0,0,0,0) 5(0,0,0,0) 3(2,5,0,0) 1(0,2,0,0)
2(2,3,0,0) 1(0,2,0,0) 7(0,0,0,0) 2(0,3,0,0).

После применения п.2 алгоритма получим

4(3,1,0,0) 1(2,2,0,0) 6(0,0,0,1) 5(0,0,0,1) 3(2,5,0,0) 1(0,2,0,0)
2(2,3,0,0) 1(0,2,0,0) 7(0,0,0,1) 2(0,3,0,0), при этом запоминается:
5(0,0,0,1) → 5(0,0,0,0);
6(0,0,0,1) → 6(0,0,0,0);
7(0,0,0,1) → 7(0,0,0,0).

После реализации первого этапа п.3 алгоритма получим

4(3,1,0,0) 1(0,2,0,1) 3(2,5,0,0) 1(0,2,1,1) 2(2,3,0,0) 1(0,2,1,1)
7(0,0,0,1) 2(0,3,0,0) при этом отдельно запоминается:

1(0,2,0,1) → 1(2,2,0,0) 6(0,0,0,1) 5(0,0,0,1).

Затем, повторяя действия на последующих этапах выполнения п. 3 алгоритма получим последовательно

1. 4(3,1,0,0) 1(0,2,0,1) 3(2,5,0,0) 1(0,2,1,1) 2(0,5,0,1) 2(0,5,1,1),
при этом запомним:

2(0,5,0,1) → 2(2,3,0,0) 1(0,2,1,1) τ₁(1,2,1,1) 7(0,0,0,1).

2. 4(3,1,0,0) 1(0,2,0,1) 3(0,10,0,1) 2(0,5,1,1),

запомним:

3(0,10,0,1) → 3(2,5,0,0) τ₁(1,3,1,1) 1(0,2,1,1) 2(2,5,0,1)

3. 4(0,11,0,1),

запомним:

4(0,11,0,1) → 4(3,1,0,0) τ₁(1,8,1,1) 1(0,2,0,1) 3(2,10,0,1) τ₁(1,5,1,1)

2(0,5,1,1).

После применения п.4 алгоритма последовательно получим

4(0,11,0,1)

4(3,1,0,0) τ₁(1,8,1,1) 1(0,2,0,1) 3(0,10,0,1) τ₁(1,5,1,1) 2(0,5,1,1),

4(3,1,0,0) τ₁(1,8,1,1) 1(2,2,0,0) 6(0,0,0,1) 5(0,0,0,1) 3(2,5,0,0)

τ₁(1,3,1,1) 1(0,2,1,1) 2(0,5,0,1) τ₁(1,5,1,1) 2(0,5,1,1).

4(3,1,0,0) τ₁(1,8,1,1) 1(2,2,0,0) 6(0,0,0,1) 5(0,0,0,1) 3(2,5,0,0)

τ₁(1,3,1,1) 1(0,2,1,1) τ₁(1,2,1,1) 7(0,0,0,1) τ₁(1,5,1,1) 2(0,3,1,1).

После применения п. 5 алгоритма имеем

4(3,1) τ₁(1,8) 1(2,2) 6(0,0) 5(0,0) 3(2,5) τ₁(1,3) 1(0,2) 2(2,3) 1(0,2)
τ₁(1,2) 7(0,0) τ₁(1,5) 2(0,3).

С учетом того, что максимальной задержкой элемента в исходной схеме было 5, применение п. 6 алгоритма приводит к следующему окончательному результату

4(3.1) $\tau_1(1.5)$ $\tau_2(1.3)$ 1(2.2) 6(0,0) 5(0,0) 3(2.5) $\tau_1(1.3)$ 1(0,2)
2(2,3) 1(0,2) $\tau_1(1.2)$ 7(0,0) $\tau_2(1.5)$ 2(0,5).

Последняя запись и является записью синхронизированной схемы.

ЛИТЕРАТУРА

1 Бозоян Ш. Е. Язык описания функциональных схем / Изв. АН СССР. Техническая кибернетика - 1978 - № 6. - С. 158-166.

ГИУА

18.11.1996

Изв. НАН и ГИУ Армении (сер. ГИ), т. L, № 3, 1997, с. 275-276

УДК 621.785.5:621.9.025.7

КРАТКИЕ СООБЩЕНИЯ

Г.С. ОВСЕПЯН, А.А. АМБАРЯН, А.Ж. ГАЛСТЯН, Г.Г. МКРТЧЯН

ОПТИМИЗАЦИЯ СКОРОСТИ ПЕРЕМЕЩЕНИЯ КОНВЕЙЕРА В ПЕЧАХ С КОНТРОЛИРУЕМОЙ АТМОСФЕРОЙ

Մշակվել է քիմիա-տեղմային վառարանում կարծր հատակաթիվածքի փրփելիկների ցերեխրիմիտան մշակման և կտրող գործիքների գոգման կարգավորման եղանակ. հաստատվել է որ վառարանում հարաբերակ մեղաշարժման կարգավորման եղանակը կարողորդեցր կարգի է կիրառել կարծր հատակաթիվածքի փրփելիկների ցերեխրիմիտան մշակման, գործիքների գոգման գործընթացներում և սարքեր նշանակությունը օբյեկտների փնտման համար.

Разработан контролируемый способ химико-термической обработки и пайки твердосплавных пластин к державкам режущих инструментов в камерных конвейерных печах. Установлено, что предлагаемый способ регулирования скорости перемещения конвейера в печи можно успешно использовать при химико-термической обработке твердосплавных пластин пайке режущих инструментов и термообработке деталей машин различного назначения.

Библиогр. - 2 назв.

A controlled method is developed for chemical/thermal treatment of hard alloy plates and soldering of cutting tools in chamber conveyor furnaces. It is show that the proposed method for regulation of conveyor transport velocity in the furnace can be successfully used during the chemical/thermal treatment of hard alloy plates, soldering of cutting tools and thermal treatment of various purpose machine parts in machine building industry.

Ref. 2

При пайке скорость перемещения конвейера в печи выбирают экспериментальным путем, что не всегда согласуется с качественными показателями паяных соединений. Предлагается контролируемый способ химико-термической обработки и пайки твердосплавных пластин к державкам режущих инструментов в