## Язык операторных схем алгоритмов с параллельными ветвями (ОСАП)

### Общие сведения о языке ОСАП

Возможность формального описания параллельно выполняемых во времени ветвей алгоритма значительно способствует увеличению выразительных средств любого начального языка. Однако параллельные алгоритмы обладают большим разнообразием по сравнению с обычными, что существенно затрудняет осмысливание и правильный учет всех факторов, связанных с параллельно протекающими процессами и их взаимодействием.

Язык ОСАП базируется на расширении синтаксиса известных языков ГСА и ЛСА, а также на использовании модели НДА и стандартного языка СКУ для описания всех частных событий. Выбор такого базиса для языка ОСАП вытекает из следующих предпосылок. Использование модели НДА позволяет описывать параллельные ветви алгоритма в виде частных СКУ независимо друг от друга на основе обычных ГСА или ЛСА, а взаимодействие параллельных ветвей и их синхронизацию можно учитывать при построении общей НД СКУ путем надлежащей формализации событий, определяющих выход алгоритмического процесса за вершину объединения ветвей и коррекции частных событий СКУ, полученных для каждой параллельной ветви.

Язык ОСАП позволяет производить синтез цифровых автоматов на основе языка ГСА, т.е. методом, который применяется для синтеза детерминированных микропрограммных автоматов. Будем сокращенно обозначать язык ОСАП на основе языка ГСА через ГСАП.

### Основные конструкции, вводимые в язык ГСАП

Для описания параллельных ветвей алгоритма и их взаимодействия в язык ГСА вводятся дополнительные вершины, указывающие места разветвления и объединения параллельных ветвей: **разветвители** и **соединители**.

Разветвителю в ГСАП будет соответствовать разветвительная вершина (рис.1) обозначаемая буквой (*fork*-разветвление), которая имеет один вход и *n* выходов, *α* - номер вершины.



Рис. 1. Фрагмент ГСАП с разветвительной вершиной

Соединителю в ГСАП будет соответствовать соединительная вершина (рис.2) обозначаемая буквой  (*join*-соединение, объединение), имеющая столько входящих линий, сколько параллельных ветвей она объединяет, *β* - номер вершины.



Рис. 2. Фрагмент ГСАП с соединительной вершиной

Каждая входящая линия может быть образована слиянием нескольких линий, но все компоненты такой линии должны принадлежать одной и той же параллельной ветви. Выходящая линия из соединителя может быть только одна.

Объединение параллельных ветвей в соединителе осуществляется с учетом определенных условий, связанных с синхронизацией параллельных процессов. В самом общем виде условия объединения могут быть заданы в виде булевой функции над обозначениями частных событий, имеющих место в объединяемых параллельных ветвях:

 (2)

где  - обозначение событий, отмечающих окончание алгоритмического процесса в соответствующих ветвях с 1-ой по *n* – ой (потоках или нитях параллельной программы);

 - обозначение событий, реализуемых в ветвях с 1-ой по *n*-ой (потоках или нитях параллельной программы);

~ - знак, означающий, что переменная может быть взята с отрицанием или без отрицания.

Алгоритмический процесс распространяется за вершину соединения, если будет истинным выражение, т.е. имеет место соотношение:

 (3)

где  - первое событие, реализуемое в последовательной части алгоритма после соединителя.

Практический интерес представляет ряд частных случаев задания условий для объединения параллельных ветвей, которые формализуются с использованием соединителей трех типов: конъюнктивного соединителя, дизъюнктивного соединителя и комбинированного конъюктивно-дизъюнктивного соедини­теля.

**Конъюнктивный соединитель *J*(&).**

При использовании такого соединителя алгоритмический процесс распространяется за вершину соединителя в том случае, если имеет определенное сочетание (конъюнкция) событий, отмечающих окончание алгоритмического процесса в объединяемых ветвях, взятых с отрицанием или без отрицания. При этом возможны два частных случая.

***Первый*** ***случай*** соответствует алгоритму, когда алгоритмический процесс распространяется за вершину соединителя при условии окончания алгоритмического процесса во всех объединяемых ветвях одновременно (в один и тот же момент времени). Для такого соединителя имеет место соотношение:

 (4)

***Второй случай*** соответствует условию окончания алгоритмических процессов во всех объединяемых ветвях, но не обязательно в один и тот же момент времени. В этом случае факт окончания алгоритмического процесса в каждой из параллельных ветвей должен запоминаться, т.е. этот факт должен найти отражение при формализации условий сохранения событий, отмечающих окончание алгоритмического процесса в каждой из параллельных ветвей. Тогда для любой *i*-ой ветви событие  должно иметь вид:

 (5)

где  - сокращенное обозначение события, определяющего зарождение события  (его первоначальное появление).

Из соотношения (5) следует, что событие  после своего зарождения сохраняется до тех пор, пока не будет истинным событие *S*в.

**Дизъюнктивный соединитель *J*(v).**

Для таких соединителей алгоритмический процесс распространяется за соединитель при окончании его в любой из соединяемых ветвей, т.е. имеем:

 (6)

Из выражения (6) следует, что если по инициативе какой-либо ветви процесс распространился за вершину соединения, а в это время в другой ветви процесс достигает соединителя, то возможна неоднозначность выполнения алгоритма. Если это недопустимо, то должны быть предусмотрены меры устранения такой неоднозначности. Для этой цели выполняют коррекцию алгоритма таким образом, чтобы при достижении алгоритмическим процессом в любой ветви соединителя, он распространялся бы за соединитель с одновременной запрещением алгоритмических процессов во всех параллельных ветвях. Это обеспечивается коррекцией описания частных событий во всех параллельных ветвях с учетом того, что все *частные события* в любой *i*-ой ветви должны быть *несовместимы* с событиями, отмечающими окончание алгоритмического процесса во всех других ветвях. В общем случае коррекцию частных событий в любой *i*-ой ветви можно представить таким образом:

 (7)

где  - сокращенное обозначение события  без учета коррекции.

**Соединитель комбинированный *J*(&,v).**

Для комбинированного соединителя функция *S*В имеет вид:

V (8)

Из (8) следует, что  есть дизъюнктивная нормальная форма от переменных типа .

Для такого типа соединителя однозначность выполнения алгоритма также требует коррекции всех частных событий во всех параллельных ветвях, как и для дизъюнктивного соединителя. Это означает, что для всех частных событий всех параллельных ветвей правая часть их описания должна быть умножена на  в соответствии с выражением (7).

В тех случаях, когда выход за соединитель *J*(&,v) в соответствии с алгоритмом может осуществляться не обязательно при одновременном появлении событий типа , то эти события после их зарождения следует сохранять до тех пор, пока не будет истинным комбинационное событие . В связи с этим в описаниях событий типа  должны быть предусмотрены условия сохранения событий  в соответствии с выражением (5).

**Структурный синтез НДА по ГСАП**

По заданной ГСАП НДА Мура построить граф НДА, прямую таблицу переходов, СКУ и СВФ.

В ГСАП микропрограммного автомата имеются два цикла, состоящие только из условных вершин; их заменяем циклами с ожидающими вершинами Sei .

ГСАП содержит одну разветвительную вершину, три параллельные ветви алгоритма, и одну объединительную вершину. Объединительная вершина представляет собой конъюнктивный БАРЬЕР с тремя входами.



Рис. 3. ГСАП микропрограммного НДА

Выход алгоритма за барьер произойдет только в том случае, если выполнится условие:, т.е все три ветви достигнут барьера. Если ветви достигают барьера не одновременно, то каждый вход объединительной вершины заменяется конструкцией вида:



После коррекции ГСАП микропрограммного НДА представляется в виде , показанном на рис. 4



Рис. 4 Скорректированный ГСАП НДА

Граф НДА Мура показан на рис. 5

****

Рис. 5. Граф НДА Мура

**Прямая таблица переходов:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Исходные события Sj(t) | | | |  | Входные сигналы *Xi,j(t)* | События перехода  Sj (t+1) (yi) | Примечание |
|  |  |  | | |  | |  |  |
|  |  |  | | |  |
|  |  |  |  |  |  | *Х2* | S1(y2) | Для первой ветви |
|  |  |  |  |  |  |  | Se1 (ye1) |
|  |  | S0(y0) |  |  |  |  |
| 1 |  |  |  |  | 1 | S2(y1yЗ) | Для второй |
|  |  |  |  |  |  | - | SЗ(y2yЗ) |  |
|  |  |  |  |  |  | 1 | Для третьей |
| 2 |  | S1(y2) |  |  |  | 1 | Skl(уk1) | Для первой |
|  |  |  |  |  |  | *Х2* | S2(y1yЗ) |  |
| 3 |  | S2(y1yЗ) |  |  |  |  | Se2(ye2) | Для второй |
|  |  |  |  |  |  |  | Sk2(yk2) |  |
| 4 |  | SЗ(y2yЗ) |  |  |  | *Х1* | S4 (y1) | Для третьей |
|  |  |  |  |  |  | S3(y2y3) |
| 5 |  | S4(y1) |  |  |  | 1 | Sk3(yk3) | Для третьей |
| 6 |  | Se1(ye1) |  |  |  | *Х2* | S1(y2) | Для первой |
|  |  |  |  |  |  | Se1(ye1) |
| 7 |  | Se2(ye2) |  |  |  |  | Se2(y) | Для второй |
|  |  |  |  |  | *Х2Х3* | Sk2(yk2) |
| 8 |  | Sk1(yk1) |  |  |  |  | Sk1(yk1) | Для первой |
|  |  | S5( y0) |
| 9 |  | Sk2(yk2) |  |  |  |  | Sk2(yk2) | Для второй |
|  |  | S5( y0) |
| 10 |  | Sk3(yk3) |  |  |  |  | Sk3(yk3) | Для третьей |
|  |  | S5( y0) |
| 11 |  | S5 (yk) |  |  |  | 1 | S0(y0) | Возврат в исходное состояние |
|  |  |  |  |  |  |  |  |  |

* -* условие выхода за соединитель