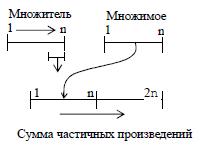
1. **Алгоритм умножения чисел с ФЗ первым способом в прямом коде**



**Разрядность регистров:**

* Множителя и множимого – n;
* Суммы частичных произведений – 2n;

Множимое следует прибавлять в старшие n разрядов регистра суммы частичных произведений.

**Особенность 1 способа**:

в цикле умножения возможно временное переполнение разрядной сеткимантисс (временное ПРС мантисс) в регистре суммы частичных произведений,которое ликвидируется при очередном сдвиге вправо.

**Алгоритм умножения мантисс**

1. Определить знак произведения путем сложения по модулю два знаковых разрядов сомножителей.

2. Проверить множимое на равенство нулю: если равно нулю, операцию умножения следует прекратить, т.к. результат будет также равным нулю.

3. Проверить множитель на равенство нулю: если равен нулю, операцию умножения следует прекратить, т.к. результат будет также равным нулю.

4. Выполнить цикл умножения по следующим правилам:

4.1. произвести анализ очередного разряда множителя.

4.2. Произвести суммирование множимого с суммой частичных произведений (ЧП), если цифра множителя «1», иначе перейти к п.5 алгоритма.

5. Произвести сдвиг множителя и суммы ЧП на один разряд вправо.

6. Присвоить модулю произведения знак из п.1 данного алгоритма.

**Численные примеры:**

**2. Разработка операционного автомата**

**2.1 Разработка функциональной схемы ОА**

При выборе операционных блоков необходимо помнить, что основная цель курсовой работы – **минимизация аппаратурных затрат.** Но чрезмерное упрощение ОА часто влечет существенное усложнение УА (и замедление его работы), поэтому должен быть достигнут некий компромисс.

**Будем руководствоваться следующими соображениями:**

* Из описания 1 способа следует, что регистр суммы ЧП (и, как следствие, сумматор) имеет двойную разрядность. НО, поскольку выходная шина имеет разрядность n, то младшие разряды произведения округляются, поэтому не имеет смысла хранить их и можем сократить разрядность суммы частичных произведений и сумматора до n.
* Так как возможно временное ПРС мантисс, то в сумматоре должен быть либо разряд переноса, либо дополнительный бит (последнее предпочтительно, т.к. экономя один бит мы не выигрываем в аппаратуре, но усложняем логику сумматора).
* Если один из операндов равен нулю, то и знак результата должен быть равен нулю, вне зависимости от знака второго операнда.

Операционный автомат **должен обеспечивать** (следует из словесного описания алгоритма):

* Хранение операндов и суммы ЧП
* Определение знака произведения
* Проверку операндов на равенство нулю и выдачу нулевого результата, если один из сомножителей равен нулю
* Исключение возникшего временного ПРС
* Суммирование множимого и суммы ЧП
* Сдвиг множителя и суммы ЧП
* Подсчет количества циклов умножения

С учетом сказанного, операционный автомат должен содержать:

* 31-разрядный сдвиговый регистр RG1 для хранения модуля множителя;
* D-триггер T для хранения знака множителя;
* 32-разрядный регистр RG2 для хранения множимого со знаком;
* 32-разрядный регистр RG3 для хранения модуля суммы ЧП (дополнительный старший бит отводится для возможности исключения временного ПРС);
* 6-разрядный инкрементный счетчик тактов CT;
* 32-разрядный сумматор SM;
* 31-разрядную схему «ИЛИ» с прямым и инверсным выходами для проверки операндов на равенство нулю (обратить внимание, что **одна** схема будет использоваться для поочередной проверки обоих операндов);
* Элементы «XOR» и «И» для определения знака результата;
* Усилитель-формирователь для выдачи результата на ШИВых.

Из УА в ОА необходимо подать следующие **управляющие сигналы**:

|  |  |
| --- | --- |
| Сигнал | Микрооперация |
| y0 | запись в RG1,  запись знака множителя в T,  обнуление RG3,  обнуление CT |
| y1 | запись в RG2, |
| y2 | запись в CT |
| y3 | сдвиг RG1 вправо RG1:=R1(RG1),  сдвиг RG3 вправо RG3:= R1(RG3),  СТ: = СТ+1 |
| y4 | запись в RG3 |
| y5 | выдача результата |

Из ОА в УА необходимо передать следующие **осведомительные сигналы**:

|  |  |
| --- | --- |
| Сигнал | Логическое условие (состояние ОА) |
| Х | проверка наличия операндов на ШИВх |
| P1 = 1 | один из операндов равен нулю |
| P2 = 1 | необходимо выполнять сложение суммы ЧП и множимого |
| P3 = 1 | цикл умножения завершен |
| Z | проверка возможности выдачи результата на ШИВых |

С учетом изложенного, схема ОА представлена на рисунке 1 (**обратить внимание**, что разработка схемы ОА должна вестись одновременно и в тесной взаимосвязи с разработкой содержательной ГСА).



Рисунок 1 – Схема операционного автомата

**2.2. Разработка содержательной ГСА**



Рисунок 2 – Содержательная ГСА

**3. Построение отмеченной ГСА**

**3.1 Разметка ГСА**

Для разметки граф-схемы алгоритма каждой совокупности микроопераций, находящихся в операторных вершинах, ставятся в соответствие управляющие микрокоманды (МК) Y1…Y13. Эти МК являются выходными сигналами УА и обеспечивают выполнение требуемых действий в соответствии со списком микроопераций (МО) ОА. Совокупность МО для каждой операторной вершины образует МК, список которых представлен в таблице 1.

Таблица 1 – Список микрокоманд

|  |  |
| --- | --- |
| МК | Совокупность МО |
| Y1 | y0, y1 |
| Y2 | y1,y2 |
| Y3 | y4 |
| Y4 | y3 |
| Y5 | y5 |

Во всех операторных вершинах ГСА проставляют символы из множества выходных сигналов УА – Y1, … , Y11. В каждой операторной вершине ГСА вслед за символом МК в скобках указываются наборы МО, образующие каждую МК.

Каждой условной вершине содержательной ГСА ставится в соответствие один из входных сигналов управляющего автомата X1…X10.

Таблица 2 – Список входных сигналов для управляющего автомата

|  |  |
| --- | --- |
| Входной сигнал УА | Логическое условие ОА (осведомительные сигналы) |
| Х1 | Х |
| Х2 | р1 |
| Х3 | р2 |
| Х4 | р3 |
| Х5 | Z |

В соответствии с моделью Мили:

* вход вершины, следующей за начальной, и вход конечной вершины отмечается символом начального состояния автомата а0;
* **входы всех вершин, следующих за операторными,** отмечаются символами а1, … , аk;
* если вход вершины отмечается, то только одним символом;
* входы различных вершин, за исключением конечной, отмечаются различными символами.

В соответствии с моделью Мура:

* символом начального состояния автомата b0 отмечаются начальная и конечная вершины;
* различные операторные вершины отмечаются различными символами b1,…, bk;
* все операторные вершины должны быть отмечены, то есть каждой МК, отдельно представленной в ГСА, ставится в соответствие отдельное состояние автомата Мура;
* в логических вершинах ГСА, реализующих режим ожидания, существует возвратная дуга, когда один из выходов вершины подан на ее вход; на этой дуге необходимо вводить дополнительное фиктивное состояние автомата Мура.



Рисунок 3 – Отмеченная ГСА

**3.2 Построение графа автомата и структурной таблицы переходов и выходов**

Имея отмеченную ГСА проектируемого управляющего микропрограммного автомата, следует описать его работу известными способами - графическим и табличным. Если количество состояний автомата и переходов между ними невелико, то задание его в виде графа позволяет наглядно представить работу МПА.

Граф автомата есть ориентированный связный граф, вершины которого соответствуют состояниям, а дуги - переходам между ними. Причем, две вершины графа аm и аs - соединены дугой, направленной от аm (исходное состояние) к аs (состояние перехода) если в ГСА существует этот переход.

Для автомата Мили каждой дуге приписываются входные и выходные сигналы, если они определены. Для автомата Мура дугам приписаны только входные сигналы; выходные сигналы приписаны вершинам графа.

Замечания.

1. В графе автомата необходимо указывать все возможные переходы между состояниями, "проходя" встречающиеся на пути из аm в аs условные вершины по всем исходящим из них дугам.

2. При построении графа автомата Мили необходимо избегать "пустых" переходов, то есть переходов, на которых не вырабатываются управляющие сигналы (или на которых не встречается операторной вершины, что эквивалентно). Чтобы не снижать быстродействия МПА, надо в графе показать переход в следующее состояние, если это возможно.

Если автомат имеет большое число состояний и переходов между ними, то наглядность графа теряется. Тогда удобно использовать табличный способ задания автомата. При синтезе МПА строят прямые (или инверсные) структурные таблицы переходов и выходов.

Таблица 2.

| Исх. Cостояние  am | Код исх. состояния  K(am) | Состояние перехода  as | Код сос-тояния перехода K(as) | Входные сигналы  X(am, as) | Выходные сигналы  Y(am, as) | Функции возбужде-ния ЭП  F(am, as) |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

В табл.2 дана прямая структурная таблица для автомата Мили. Для автомата Мура столбец 6 таблицы (выходные сигналы) следует располагать вслед за первым столбцом.

Состояния, перечисляемые в первом столбце таблицы, должны быть упорядочены, то есть сначала следует указать все переходы из а0, затем из а1 и т.д. Аналогично и для инверсной таблицы переходов (все переходы в а0, в а1 и т.д.). При формировании столбца 5 (входные сигналы) следует указывать конъюнкцию всех входных сигналов, записанных в логических вершинах ГСА на данном переходе. Причем Хi берут без отрицания, если переход выполняется по единичному значению сигнала, и с отрицанием, если по нулевому значению сигнала.

Первоначально столбцы 2 и 4 таблицы (коды состояний) не могут быть заполнены, так как еще не выполнено кодирование состояний автомата. По этой же причине не могут быть определены и функции возбуждения элементов памяти (столбец 7). Формирование структурной таблицы будет завершено позднее.

**4. Выбор и обоснование структурной схемы**

**управляющего автомата**

Этот этап структурного синтеза выполняется параллельно с выбором способа кодирования внутренних состояний и типа элементов памяти для управляющего автомата.

Вариант 1. Классическая структура УА - это совокупность взаимосвязанных элементов памяти (ЭП) и комбинационной схемы (КС), реализующей функции возбуждения ЭП и функции выходов Y (рис.14).

Рис.14

X

Y

KC

ЭП

ЭП

Набор ЭП фиксирует состояния, в которые переходит автомат в процессе выполнения заданной микропрограммы. Синтез УА сводится к синтезу его КС, на вход которой поступает множество входных сигналов Х из ОА и сигналы, фиксирующие коды состояний автомата.

Естественным требованием при синтезе КС является минимизация ее цены, что обеспечивается правильным выбором способа кодирования состояний и типа ЭП, а также совместной минимизацией функций возбуждения ЭП и функций выходов.

Вариант 2. Цена КС может быть снижена, если сигналы с выходов ЭП подать на дешифратор ДС, что приводит к структуре, изображенной на рис.15.

В такой структуре УА необходимо стремиться к полному использованию выходов дешифратора, Если дешифратор, подключаемый к выходам всех ЭП, недоиспользуется, целесообразно продумать вариант установки дешифратора, подключаемого к части выходов ЭП.

Рис.15

X

Y

KC

ЭП

ЭП

DC

Вариант 3. В случае выбора унитарного способа кодирования внутренних состояний автомата, элементы памяти объединяются в регистр сдви-

га, что приводит к структуре УА, изображенной на рис.16.

Тогда часть комбинационной схемы, реализующая функции возбуждения элементов памяти, значительно упрощается, так как в этом случае необходимо вырабатывать лишь сигнал первоначальной установки регистра и сигнал сдвига содержимого регистра (в качестве которого часто используют сигнал синхронизации).

Рис.16

X

Y

KC

ЭП

ЭП

код исх. сост.

Вариант 4. Когда граф проектируемого автомата имеет большое количество последовательных (стандартных) переходов и незначительное количество нестандартных переходов, целесообразно закодировать состояния автомата последовательными двоичными числами. Это и диктует структуру

X

Y

KC

CT

Рис.17

УА на основе двоичного счетчика, работающего в режиме сложения или вычитания. При этом на последовательных переходах необходимо подавать сигнал на счетный вход счетчика, что обеспечит стандартные переходы. На нестандартных переходах следует устанавливать нужный код по установочным входам счетчика. Если число нестандартных переходов невелико, то КС, реализующая функции возбуждения ЭП, будет проста.

Вариант 5. Структуру на основе двоичного счетчика можно модифицировать, используя дешифратор, подключаемый к выходам триггеров счетчика. Получим вариант структуры УА, аналогичный варианту 2 (рис.15) со счетчиком, используемым в качестве ЭП.

В заключение этого параграфа следует отметить, что возможны и другие варианты структур УА. Однако, учитывая содержание заданий, предлагаемых студентам для выполнения курсовой работы, а также то, что данная работа имеет своей целью приобретение первых практических навыков проектирования МПА, которые при дальнейшем обучении будут расширены и углублены, можно на начальном этапе ограничиться предлагаемыми выше вариантами структур УА.

**5. Построение функциональной схемы управляющего МПА**

Полученные на предыдущем этапе логические выражения для функций возбуждения ЭП и функций выходов позволяют построить комбинационную схему, реализующую эти функции. При этом может быть использован как основной логический базис И, ИЛИ, НЕ, так и любой другой базис по заданию преподавателя.

Построенная комбинационная схема в совокупности с набором ЭП и, быть может, другими элементами, устанавливаемыми в соответствии с выбранной структурой управляющего автомата, и дают функциональную схему управляющего микропрограммного автомата. При изображении схемы следует руководствоваться соответствующим ГОСТ.

Схема МПА должна иметь цепи начальной установки автомата в исходное состояние и цепи включения автомата на однократное выполнение алгоритма по запускающему сигналу. Кроме того в схему поступает сигнал синхронизации от генератора тактовых импульсов.

Для реализуемых в курсовой работе алгоритмов первой, после начальной вершины ГСА, является вершина ожидания поступления операндов с ШИВх. Единичный выход этой логической вершины и является фактически сигналом запуска автомата на однократное выполнение алгоритма. Так как при разметке ГСА начало и конец микропрограммы отмечены начальным состоянием а0, то автоматически происходит сброс в начальное состояние после завершения микропрограммы. Таким образом обеспечивается многократное повторение алгоритма с поступлением следующих операндов в МПА.

Цепи начальной установки необходимы в связи с тем, что после включения питания состояния элементов памяти могут быть произвольными, а для правильного функционирования автомата его необходимо установить в начальное состояние сигналом *b.*

При формировании цепей начальной установки следует учитывать как код исходного состояния, так и тип триггеров, используемых в качестве ЭП.

Пусть, например, исходное состояние автомата имеет код 01. Тогда, если в качестве ЭП используются D-триггеры, то реализовать цепи начальной установки следует так, как показано на рис.18а, где D1 и D2 -соответствующие функции возбуждения D-триггеров. Если же в качестве

&

1

D

C

D

C

TT

TT

1

2

0

1

1

D1

D2

b

C

Рис.18a

&

0

1

b

S1

1

1

&

TT

TT

1

2

S

R

S

R

C

C

1

R1

S2

R2

C

Рис.18б

ЭП используется RS-триггер, то цепи начальной установки показаны на рис.18б, где R1S1, R2S2 - функции возбуждения для 1 и 2 триггеров.

**Когда в схемах D-триггеров есть отдельные асинхронные входы установки и сброса, то их используют для подачи сигнала начальной установки,** а функции возбуждения ЭП поступают на соответствующие синхронные входы триггеров.