# Синтез управляющего автомата операции умножения младшими разрядами вперед со сдвигом множимого над числами в форме с фиксированной точкой в формате {1,8} для автомата Мура

**Содержание**

Задание

Введение

1. Разработка микропрограммы

1.1.   Пример

1.2.   Определение структуры операционного автомата

1.3.   Разработка граф схемы алгоритма

2.       Синтез микропрограммного автомата

2.1.   Кодирование граф схемы алгоритма

2.2.   Составление таблицы переходов для микропрограммного автомата

2.3.   Составление структурной таблицы микропрограммного автомата

2.4.   Составление функций возбуждения и выхода

2.5.   Разработка функциональной схемы

Заключение

Список литературы

# Введение

По функциональному назначению основные устройства ЭВМ можно условно разделить на две категории: операционные устройства (ОУ) и управляющие устройства (УУ). Отдельные части операционного устройства функционируют в зависимости от алгоритма выполняемой операции. Управляющее устройство по сигналу операции вырабатывает необходимые сигналы, по которым запускается выполнение заданной микрооперации. Совокупность микроопераций, объединенных алгоритмом операции, составляет микропрограмму операции, которая, в свою очередь, является связующим звеном между командой (кодом операции) и операционным устройством (аппаратными средствами), предназначенным для преобразования информации.

Управляющее устройство состоит из отдельных логических схем, вырабатывающих управляющие сигналы в заданной последовательности. Такое управляющее устройство можно рассматривать как управляющий автомат типа Мура или Мили.

В данной курсовой работе нами будет синтезирован блок управляющего устройства для выполнения операции умножения со сдвигом множимого вперед. Данная работа покажет уровень полученных нами знаний по курсу «Прикладная теория цифровых автоматов».

# Задание

Выполнить синтез управляющего автомата операции умножения младшими разрядами вперед со сдвигом множимого над числами в форме с фиксированной точкой в формате {1,8}в прямом коде двоичной системы счисления. Разработать микропрограмму и выполнить синтез управляющего автомата используя синхронный автомат Мура, используя логический элемент «ИЛИ-НЕ» (стрелка Пирса) и элемент памяти на RS-триггере

 1. Разработка алгоритма операции умножения младшими разрядами вперед со сдвигом множимого.

1.1 Контрольный пример

Рассмотрим на примере двух двоичных чисел сам процесс умножения

1) Возьмем любых два двоичных числа:

#### а = 101011  и b = 111011

Решение примера будем производить в двоичной системе исчисления с указанием чисел находящихся в регистрах и производимыми в них операциями:

|  |  |
| --- | --- |
| 010101100000 | L(1)Рг1 |
| 001010110000 | L(1)Рг1 |
| 000101011000 | L(1)Рг1 |
| 000010101100 | L(1)Рг1 |
| 000001010110 | L(1)Рг1 |
| |  |  | | --- | --- | | |  | | --- | | \* | |   000000101011 | Рг1 |
| 111011 | Рг2 |
| |  |  | | --- | --- | | |  | | --- | | + | |   000000000000 | См |
| 000000101011 | Рг1 |
| |  |  | | --- | --- | | |  | | --- | | + | |   000000101011 | См |
| 000001010110 | Рг1 |
| |  |  | | --- | --- | | |  | | --- | | + | |   000010000001 | См |
| 000101011000 | Рг1 |
| |  |  | | --- | --- | | |  | | --- | | + | |   000111011001 | См |
| 001010110000 | Рг1 |
| |  |  | | --- | --- | | |  | | --- | | + | |   010010001001 | См |
| 010101100000 | Рг1 |
| 100111101001 | См |

Мы взяли два числа a, b соответственно множимое и множитель, и произвели операцию умножения с их модулями по следующему алгоритму:

1)            Анализируем  разряд за разрядом  множителя начиная с младших разрядов.

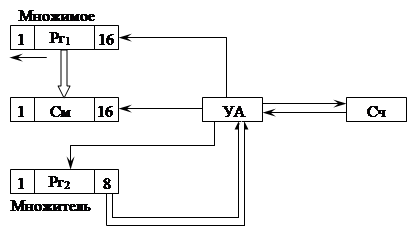
2)            Если анализируемый разряд множителя равен единицы, то множимое прибавляется к сумматору если же разряд множителя равен нулю то мы прибавляем нулевое значение множимого (данная операция пропущена).

3)            Множимое сдвигается каждый раз влево на один разряд после операции сложения.

4)            Пункты 2 и 3 выполняются n раз. n-количество разрядов в множителе. Для нашего случая это 6 разрядов и как видно из примера множимое сдвигалось шесть раз до получения окончательного ответа

**1.2 Определение структуры операционного автомата**

Рассмотрим структура операционного автомата. Определение структуры операционного автомата нам необходим для того, чтобы мы могли определить количество и размерность используемых регистров и сумматора. В используемом нами методе умножения младшими разрядами со сдвигом множимого в перед регистры множимого и сумматора должны быть шестнадцатиразрядными, а для регистра множителя достаточно будет восьмиразрядного регистра.



**Рисунок 1** - Схема структуры ОУ умножения младшими разрядами в перед со сдвигом множимого в лево

Рг1 – в регистр заносится множимое

Рг2 – в регистр заносится множитель

См – в сумматоре происходит сложение чисел

Сч – счетчик просчитывает количество выполненных операций.

УА – рассчитываемый управляющий автомат

# 1.3 Разработка блок схемы

Алгоритм для умножения младшими разрядами вперед со сдвигом множимого:

1)  В первый регистр заносится множимое в двоичном коде.

2)   Из первого регистра множимое заноситься в сумматор.

3)   Во второй регистр заносится множитель.

4)   Младший разряд множителя поступает в Управляющий автомат.

5)   В Управляющем автомате анализируется поступившая информация.

6)   Если 0, то в сумматоре происходит сдвиг вправо на один разряд. Если 1 то, происходит сложение множимого и множителя. После сложения выполняется сдвиг множимого.

7)   Пункты 4,5 и 6 выполняются столько раз, сколько разрядов имеют числа.

Реализуем алгоритм в виде содержательного графа микропрограммы для двоичной системы исчисления. Согласно ГОСТа язык ГСА используется для формальной записи алгоритмов операция и имеет 1 начальную и 1 конечную вершину.

Структура ГСА показана на рисунке 2.

Таблица 1.1

|  |  |
| --- | --- |
| **№ блока** | **Описание** |
| 1 | в Рг1 заносим множимое, в Рг2 заносим множитель, сумматор См обнуляем, а счетчику Сч присваиваем значение 8. |
| 2 | определяем знак произведения путем сложения знаковых разрядов множимого и множителя по модулю два. |
| 3 | Младший разряд Рг2 (множитель) сравниваем с нулем. Если младший разряд равен единице то идем к блоку 4, если же разряд равен нулю то переходим к блоку 5 |
| 4 | Производим операцию сложение, к См прибавляем Рг1 в прямом коде. |
| 5 | Происходит сдвиг множимого Рг2 на один разряд влево. Регистр Рг2 сдвигается на один разряд вправо. из счетчика вычитаем 1. |
| 6 | Производится сравнение счетчик Сч с нулем. Если Сч = 0, то прекращаем умножение и идем к блоку 7. Если Сч неравен нулю, то продолжаем операцию умножение, переходим в блок 3. |
| 7 | Итоговую сумму полученную в сумматоре См выводим как результат Z. |

**2 Синтез микропрограммного автомата**

**2.1 Кодирование граф схемы алгоритма**

Синтез микропрограммного автомата

|  |  |
| --- | --- |
| |  | | --- | | Таблица 1 | |

Таблица кодировок

|  |  |
| --- | --- |
| У | МК |
| Ук | Начало |
| У1 | Рг2(1¸8):=У(2¸8) |
| У2 | Рг1(1¸8):=8 |
| У3 | Рг1(9¸16):=Х(2¸9) |
| У4 | См(1¸16):=0 |
| У5 | Сч:=8 |
| У6 | Z(1):=X(1)ÅУ(1) |
| У7 | См:=См+Рг1 |
| У8 | Рг1:=L(1)Рг1 |
| У9 | Рг2:= R(1)Рг2 |
| У10 | Сч:=Сч-1 |
| У11 | Z(2¸9):=См(1¸8) |
| Х1 | Рг2(8) |
| Х2 | Сч=0 |
| Ук | Конец |

**2.2 Составление таблицы переходов для микропрограммного автомата**

для синтеза автомата Мура необходимо сделать разметку кодированной ГСА: каждой операторной вершине приписать символ состояния bi, а также для заданного типа автомата необходимо построить прямую таблицу переходов, в которую вписываются пути перехода между соседними отметками

|  |  |
| --- | --- |
| |  | | --- | | Таблица 2 | |

Таблица переходов

|  |  |  |
| --- | --- | --- |
| bm | bs(y) | X(bm, bs) |
| b1 | b2(y1, y2, y3, y4, y5) | 1 |
| b2 | b3(y6) | 1 |
| b3 | b4(y6) | х1 |
| b5(y8, y9, y10) |  |
| b4 | b5(y8, y9, y10) | 1 |
| b5 | b4(y6) |  |
| b5(y8, y9, y10) |  |
| b6(y11) | х2 |
| b6 | b1(yк) | 1 |

**2.3 Составление структурной таблицы микропрограммного автомата**

Выполним переход от абстрактных таблиц кодировок (таблица 1) и переходов (таблица 2) к структурной таблице

В таблицу переходов структурного автомата, в отличии от абстрактного автомата, добавляются три столбца: код состояния bm – K(bm), код состояния bs – K(bs), а также функция возбуждения F(bm, bs).

По количеству состояний определяем, необходимое число символов в кодирующей комбинации. Так как у нас имеется шесть состояний то кодировка будет производиться  трехпозиционной комбинацией двоичных кодов. В таблице 3 представлена структурная таблица переходов МПА Мура.

|  |  |
| --- | --- |
| |  | | --- | | Таблица 3 | |

Структурная таблица переходов и кодировки состояний

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| bm | K(bm) | bs(y) | K(bs) | X(bm, bs) | F(bm, bs) RS |
| b1 | 001 | b2(y1, y2, y3, y4, y5) | 011 | 1 |  |
| b2 | 011 | b3(y6) | 010 | 1 |  |
| b3 | 010 | b4(y6) | 110 | x1 | S1 |
| 010 | b5(y8, y9, y10) | 000 |  | R2 |
| b4 | 110 | b5(y8, y9, y10) | 000 | 1 |  |
| b5 | 000 | b4(y6) | 110 |  | S1S2 |
| 000 | b5(y8, y9, y10) | 000 |  | ----- |
| 000 | b6(y11) | 100 | х2 | S2 |
| b6 | 100 | b1(yк) | 001 | 1 | R1S3 |

**2.4 Формирование выходных функций и функций переключения элементов памяти**

По таблице 3. составим функции возбуждения для заданного автомата Мура. Тогда функции для дешифратора примут вид



В заданном базисе согласно задания отсутствует логический элемент «И», поэтому мы переводим функции с помощью формулы де Моргана базис заданный по условию. После перевода полученные значения функция для дешифратора в заданном базисе ИЛИ-НЕ примут вид



также из таблицы 3 возьмем значения функций переключения элементов памяти на RS триггере. Данные функции примут вид



используя выше приведенные доводы по структуре логических элементов разложим данные функции переключения элементов памяти в базисе ИЛИ-НЕ и получим

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |



|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |



**2.5    Разработка функциональной схемы.**

*(см. рисунок 4)*

Функциональная схема состоит из дешифратора, комбинационной схемы и элементов памяти. Дешифратор, дешифрируя состояния триггеров, вырабатывает сигнал состояния bi, который соответствует выходному сигналу Yj. Комбинационная схема, используя выходные сигналы дешифратора bj и входные сигналы (X), формирует сигналы функций возбуждения триггера. Память (RS-триггеры) в свою очередь переключаются в новое состояние, и через шину Q состояния триггеров подаются на дешифратор. Дешифратор строится в соответствии с функциями состоянии на логических элементах «ИЛИ-НЕ». Логические элементы дешифратора пронумерованы от D1 до D6. Выходы из дешифратора используются для формирования выходной шины B и для комбинационной схемы. Входная шина X имеет 4 проводa, т.к. нами используется значения x1-x2 и два их инверсных значения. Для получения инверсии входных сигналов используется 2 логических элемента «ИЛИ-НЕ» для построения инверторa (D7, D8).

Комбинационная схема для функции возбуждения, построена на логических элементах «ИЛИ-НЕ» от D9 до D22, соответствующие заданному базису. На комбинационную схему подаются текущее состояние (bk) из дешифратора, и входные сигналы по шине X. Выходы комбинационной схемы подаются на RS-входы триггеров.

В качестве элементов памяти используется RS-триггера (Т1-Т3). В функциональной схеме(Рисунок 4)используется всего 22 логических элементов «ИЛИ-НЕ», 3 элемента памяти на RS триггерaх.

**Заключение.**

В результате проделанной работы построена управляющая часть операционного автомата, который умеет складывать числа с фиксированной запятой. В ходе работы приобретены навыки практического решения задач логического проектирования узлов и блоков ЭВМ. Построена структурная схема автомата, построенная в базисе «ИЛИ-НЕ» которая содержит 22 элемента «ИЛИ-НЕ», один дешифратор и 3 RS-триггера..

**Список литературы**

1.    Савельев А.Я. «Прикладная теория цифровых автоматов», «Высшая школа» М. 1988г.

2.    Айтхожаева Е.Ж. «Арифметические и логические основы цифровых автоматов» Алма-Ата 1980г

3.    Айтхожаева Е.Ж. «Проектирование управляющего автомата» Алма-Ата 1985г

4.    Айтхожаева Е.Ж. «Прикладная теория цифровых автоматов» Алма-Ата 1993г