**Лабораторная работа №8 (Мой вариант 4)**

# Тема: Моделирование ассоциативной памяти с системой адресации по разрядным столбцам и по словам

**Цель работы:** освоение навыков построения и верификации модели ассоциативной памяти, обеспечивающей адресное считывание и запись по разрядным столбцам и по словам и выполнение логических операций над столбцами, поисковых операций и арифметических операций над полями слов.

# Краткие теоретические сведения

Теоретические сведения, необходимые для выполнения лабораторной работы №8, включающие сведения об ассоциативных запоминающих устройств параллельного действия с поиском параллельных по словам и последовательным по разрядам, с различными видами адресациями, особенностях диагональной адресации, обеспечивающей выполнение поисковых операций с применением рекуррентных алгоритмов, арифметических операций над полями слов и других операций, приведены в теоретической части данного ЭРУД (тема 14).

Напомним основные положения из этих сведений:

Известно, что ассоциативные процессоры (АП), использующие адресацию только по разрядным столбцам, имеют ряд недостатков, в частности:

а) запись данных в ассоциативный массив (АМ) приходится производить по разрядным столбцам, из-за чего все содержимое массива должно быть предварительно занесено в некоторый буфер;

б) большие трудности вызывает также модификация данных (изменение лишь одного слова требует перезаписи содержимого всего массива);

в) в рамках базовой конфигурации АМ практически невозможно осуществить считывание слов по заданному адресу (эта операция необходима для того, чтобы просмотреть целиком все слово, определенная часть которого совпала с аргументом поиска).

Рассмотрим метод адресации АМ, позволяющий избежать отмеченные выше недостатки.

В рассматриваемом варианте организации ассоциативной памяти (АП) применяются специальные методы формирования содержимого массивов и введена дополнительная логика, что обеспечивает адресное считывание и запись как по разрядным столбцам, так и по словам.

Типовой модуль памяти такого типа приведен на рисунке 1.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  | | --- | --- | --- | --- | |  | Дешифратор | |  | | 0 | 1 | **. . .** | n | |  |

Код Вывод бита данных адреса

разряда Ввод бита данных

Рисунок 1. Типовой модуль ассоциативной памяти

Типовой стандартный модуль ассоциативной памяти представляет собой массив размером 1 х *m*, где *m* –количество слов (в частности, можно использовать серийные интегральные схемы памяти 1 х *N* битов).

В рассматриваемом варианте массив памяти считается квадратным (например 16 х 16 разрядов). Если требуется массив большего объема, объединяются несколько таких массивов. Очевидно, что в ассоциативной памяти такого типа используется больше аппаратуры, чем в базовой конструкции АП, использующих адресацию только по разрядным столбцам. В их состав входят, в частности, m дешифраторов, причем на каждый из них подается свой адресный код, вычисляемый при помощи соответствующих функциональных цепей.

Применяемый в таких ассоциативных памятях принцип формирования адресов получил название диагональной адресации. Это название связано с тем, что элементы исходного массива данных перед записью в память перераспределяются следующим образом:

* разрядные столбцы размещаются по диагоналям, образованным запоминающими ячейками;
* слова размещаются по столбцам.

Пример размещения массива данных размером 16 х 16 разрядов приведен на рисунке 2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S0,0 | S1,15 | S2,14 |  |  | S15,1 |
| S0,1 | S1,0 | S2,15 |  |  | S15,2 |
| S0,2 | S1,1 | S2,0 |  |  | S15,3 |
|  |  |  |  |  |  |
| S0,15 | S1,14 | S2,13 |  | S14,1 | S15,0 |

Sj,i – i-й разряд j -го слова

Рисунок 2. Расположение данных в ассоциативной памяти с диагональной адресацией

Рассмотрим, как в памяти с диагональной системой адресации выполняются операции считывания и записи слов и разрядных столбцов на примере применения двоичных сумматоров.

Схема памяти с диагональной адресацией на основе сумматора приведена на рисунке 3.

Адрес (в дополнительном коде)

Дешифратор R

0 +

Дешифратор

Ввод-вывод данных

1 +

Дешифратор

+

15

Рисунок 3. Схема памяти с диагональной адресацией

В этой памяти каждый дешифратор дополнен блоком арифметического сложения с полным сумматором, в котором формируется сумма адресного кода, заданного извне, и постоянного схемно-реализуемого числа, равного номеру строки.

Применяя соответствующие средства управления, части сумматоров можно блокировать (маскировать) и в результате на их выходах получить требуемый фрагмент адреса разрядного столбца.

Данные, которые необходимо записать в память или считать из нее, помещаются в специальный буферный регистр. Содержимое этого регистра можно циклически сдвигать вверх или вниз на заданное количество разрядов. Направление сдвига зависит от выполняемой операции.

## Выполнение операций записи – считывания

*Запись в память разрядного столбца*

Адрес разрядного столбца, представленный в дополнительном коде, подается на вход сумматора. Логические цепи последнего деблокируются и производится суммирование (поданного адреса и номера строки), а данные, подлежащие записи в память, передаются в буферный регистр. Производится сдвиг содержимого буфера вниз, причем количество сдвигов определяется адресом столбца.

Далее выдается команда записи и элементы преобразованного разрядного столбца записываются в соответствующие ячейки памяти, определяемые кодами, полученными на выходах соответствующих сумматоров. В частности, первый адресный столбец записывается в ячейки, выделенные на рисунке 2 более жирными линиями.

*Считывание разрядного столбца*

Для считывания разрядного столбца на вход сумматора также подается его адрес в дополнительном коде. В сумматорах формируется результирующий адресный код.

По команде считывания разрядный столбец с переставленными (диагонально) элементами пересылается в буферный регистр. Для восстановления прежнего порядка элементов столбец сдвигается циклически вверх на то же количество разрядов, что и при записи.  *Запись слова в память*

Запись слова осуществляется следующим образом:

* адрес слова подается на входы сумматоров, где производится суммирование кодов адреса с нулем (т.е. для всех строк формируется одинаковый адрес – адрес столбца);
* содержимое слова пересылается в буферный регистр памяти и сдвигается вниз на количество разрядов, задаваемое адресом слова (столбца);
* по команде записи данные из буферного регистра передаются в ячейки памяти, причем так, что i-е слово оказывается в i-м столбце массива памяти.

# *Считывание слова из памяти*

Считывание слова из ассоциативной памяти происходит в той же последовательности, что и запись:

* адрес слова подается на входы сумматора (и складывается с нулем);
* содержимое адресуемого столбца пересылается в буферный регистр, где производится сдвиг вверх на количество разрядов, задаваемое адресом слова.

***Выполнение логических операций над разрядными столбцами***

При выполнение логических операций два разрядных столбца (S1и S2) подвергаются логической операции и результат этой операции помещается в позицию (адрес) некоторого третьего столбца (D). При этом используются операции считывания и записи разрядных столбцов, рассмотренных выше.

Для любой логической операции первый из операндов S1загружается в буферный регистр R, заданная логическая операция выполняется между вторым операндом S2(т.е. соответствующим разрядным столбцом) и регистром *R*, после чего результат, заменивший прежнее содержимое регистра *R*, пересылается из этого регистра в позицию результирующего разрядного столбца (D). Логической операцией может быть любая булева функция двух переменных, приведенных в таблице 1.

Пример выполнения операции *OR*(*ИЛИ*) приведен на рисунке 4.

AM S1  S2 D R AM S1 S2 D R

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0  1  1  0  1 | 1  0  1  0  0 | x  x  x  x  x |

|  |
| --- |
| y  y  y  y  y |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0  1  1  0  1 | 1  0  1  0  0 | 1  1  1  0  1 |

|  |
| --- |
| 1  1  1  0  1 |

x, y = {0 или 1}

до операции после выполнения операции

Рисунок 4. Пример выполнения операции *OR*(*ИЛИ*)

Для упрощения русинка 4 на нем показано выполнения логической операции OR(или) на сокращенном массиве, имеющего в каждом разрядном столбце всего 5 разрядов обозначения состояния позиции (адреса) результирующего разрядного столбца (D) символом Х и обозначение содержимого буферного регистра R символом у означает , что их состояние до выполнения операции может быть любым сочетанием единиц (1) и нулей (0) , но можно записывать в эти позиции х и у , так даже удобнее.

Логические функции 2-х переменных Таблица 1





## Выполнение арифметических операций над полями слов

Выполнение арифметических оп ераций над полями слов АП может быть организовано разными способами, например, следующим образом:

Каждое слово массива ассоциативной памяти состоит из 4-х полей (общая длина слова – 16 разрядов), содержащих переменные *Vj, Aj, Вj* и *Sj,* где *j* – номер слова внутри массива памяти (размерность полей *Vj =* 3*,Aj = Вj =* 4 и *Sj =* 5)*.*

Задачей является формирование суммы полей *Aj + Bj* и запись результата в поле *Sj.* Указанные действия производятся лишь для тех слов, у которых значение переменной *Vj* совпадает с содержимым соответствующего поля аргумента поиска (остальные поля аргумента поиска при этом должны быть замаскированы).

Сложение начинается с младших разрядов полей *Aj* и *Bj*. Для запоминания значения поразрядной суммы и переноса в памяти результатов необходимо иметь дополнительные регистры.

## Выполнение поисковых операций

Примеры и методика выполнения поисковых операций с применением реккурентных алгоритмов приведены в лабораторной работе №7. **Контрольные вопросы:**

1. Назовите основные недостатки ассоциативных процессоров c параллелизмом на уровне слов и с обработкой их последовательно по разрядам.
2. Назовите способ построения АЗУ, обеспечивающий запись и поиск информации в АЗУ по разрядным срезам (столбцам) и по словам.
3. Как располагаются разрядные срезы и слова в АЗУ с диагональной адресацией?
4. Приведите алгоритмы выполнения записи и считывания разрядных срезов (столбцов) и слов в АЗУ с диагональной адресацией.
5. Какие операции над разрядными срезами и со словами могут выполняться в АЗУ с диагональной адресацией?
6. Приведите алгоритм выполнения арифметических операций над полями слова в АЗУ с диагональной адресацией.
7. Какая логическая операция является базовой в АЗУ?
8. Какие логические функции используются для выполнения базовой логической операции?
9. Приведите логические формулы для определения совпадения и несовпадения аргумента поиска А и слова памяти Sjс учетом маскирования (с).
10. Назовите способы реализации сравнения аргумента поиска А и слова памяти Sj?
11. Какие алгоритмы чаще используются при выполнении поисковых операций в АЗУ?
12. Назовите основные типы поисковых операций в ассоциативных процессах?
13. Определите соотношение между А и Sj, если qji=1, lji=0

## Задания

Построить и проверить программную модель ассоциативной памяти с диагональной адресацией на основе сумматора.

Размер массива памяти не более 16х16 двоичных разрядов (битов).

Разработанная программная модель должна уметь выполнять операции считывания и записи любых задаваемых разрядных столбцов и слов, а также выполнять поисковые, и логические операции над разрядными столбцами и арифметические операции над полями слов.

Таблица 2. Варианты заданий:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Варианты заданий | Записьсчитывание разрядных столбцов и слов | Логические функции разрядными столбцами | над | Поисковые операции | Арифметические операции над  полями слов |
| Вариант 1 | любых | f1иf14, f3 иf12 |  | Поиск значения ближайшего сверху(снизу) | Сложение полей Aj и Bj в словах Sj, у которых Vj совпадает с заданным V=  000-111 |
| Вариант 2 | любых | f7иf8, f2иf13 |  | Поиск величин, заключенных в  данном интервале | Сложение полей Aj и Bj в словах Sj, у которых Vj совпадает с заданным V=  000-111 |
| Вариант 3 | любых | f6иf9, f4иf11 |  | Упорядоченная выборка(сортировка) | Сложение полей Aj и Bj в словах Sj, у которых Vj совпадает с заданным V=  000-111 |
| Вариант 4 | любых | f5иf10, f0иf15 |  | Поиск по  соответствию | Сложение полей Aj и Bj в словах Sj, у которых Vj совпадает с заданным V=  000-111 |

Примечание: Вариант задания для каждого студента определяется преподователем.

## Требования к программе

Разработанная программная модель должна уметь выполнять следующие функции:

1. Форматировать двоичный массив размером 16х16 (т.е. 16 слов по 16 разрядов) (для некоторых заданий 8х8).
2. Преобразовывать сформированный массив в массив с диагональной адресацией и обратно.
3. Выполнять операции считывания/записи любых разрядных столбцов и слов.
4. Выполнять заданные логические операции над любыми разрядными столбцами.
5. Выполнять сложение полей Aj и Bj в словах, значение Vj у которых совпадает с заданным значением от 000 до 111.
6. Выполнять поисковые операции, приведенные в таблице 2. Методика выполнения приведена в лабораторной работе №7.

***Методика выполнения:***

При выполнении данной лабораторной работы рекомендуется:

1) вначале построить (сформировать) исходный двоичный массив размером не более 16 х 16 двоичных разрядов в обычном виде;

1. затем преобразовать его в массив с диагональной адресацией и сравнить соответствие столбцов и слов;
2. при выполнении операций записи разрядных столбцов и слов в массив с диагональной адресацией и выполнении логических операций использовать любые (задаваемые «вручную») столбцы и слова исходного массива; 4) при выполнении поисковых операций использовать последовательные (рекуррентные) алгоритмы.

5) для выполнения вычислительных операций над полями Aj и Bj слова память результатов необходимо дополнить (доработать) средствами, обеспечивающими выполнение операций поразрядного сложения с учетом сигналов переноса из младшего разряда.

Стандартная процедура проверки разработанной программы заключается в анализе результатов выполнения предъявляемых к программе требований.