

Обработка запросов к базе данных посредством ассоциативной вычислительной системы

Брехов О.М.*, Тин Мо Аунг**

Московский авиационный институт (национальный исследовательский университет), МАИ, Волоколамское шоссе, 4, Москва, А-80, ГСП-3, 125993, Россия

**e-mail: obrekhov@mail.ru*

***e-mail: tinmoeaung1985@gmail.com*

Аннотация

В работе рассматривается проблема обработки запросов к базам данных (БД) посредством ассоциативной вычислительной системы (АВС). Эта проблема, как идея, была сформулирована Кохоненом Т., т.к. использование АВС может, в принципе, значительно увеличить скорость обработки запросов.

В связи с необходимостью обработки запросов в БД большого объема и совершенствованием элементной базы вычислительных систем, что является магистральным направлением развития БД бортовых и наземных комплексов авиационно-космических объектов, существует интерес к решению этой проблемы. При этом, дополнительное увеличение скорости обработки запросов может быть достигнуто за счет упорядоченной обработки элементарных запросов, образующих запрос к БД.

Уменьшение времени обработки запроса возможно при одновременном (параллельном) выполнении операции сравнения на основе ассоциативной памяти (АП).

Дополнительное уменьшение времени обработки запроса можно получить, используя расслоение оперативной памяти при чтении необходимых данных в ассоциативную память.

Исходя из указанных соображений, в работе определена задача определения структуры ВС для обработки упорядоченных запросов на основе АП, т.е. ассоциативной вычислительной системы (АВС).

Возможны два основных метода выполнения запроса: с развитием в глубину (последовательная обработка столбцов таблицы) и с развитием в ширину (последовательная обработка строк таблицы), а также комбинированный метод с развитием в глубину и ширину. В работе на основе реализации метода с развитием в глубину предложена архитектура АВС для выполнения упорядоченных запросов, разработан алгоритм обработки упорядоченных запросов на АВС, выполнена оценка эффективности обработки упорядоченных запросов на АВС.

Ключевые слова: вычислительной системы, ассоциативность, базы данных, адресные обращения, эффективность.

1. Введение

Ассоциативная память сегодня нашла широкое применение в вычислительных системах (ВС), в частности, при реализации кэш-памяти.

Ассоциативная память привлекла внимание исследователей и при разработке ВС нетрадиционных архитектур [1], в частности, ассоциативных вычислительных машин [2], машин, управляемых потоком данных [3]. В данной работе рассматривается проблема обработки запросов к базам данных (БД) посредством ассоциативной вычислительной системы (АВС). Эта проблема, как идея, была сформулирована Кохоненом Т. [2], т.к. использование АВС может, в принципе, значительно увеличить скорость обработки запросов.

В связи с необходимостью обработки запросов в БД большого объема и совершенствованием элементной базы вычислительных систем, что является магистральным направлением развития БД бортовых и наземных комплексов авиационно-космических объектов, существует интерес к решению этой проблемы [4, 5]. При этом, дополнительное увеличение скорости обработки запросов может быть достигнуто за счет упорядоченной обработки элементарных запросов, образующих запросы [6].

2. Постановка задачи

Рассмотрим задачу обработки запросов к БД в простейшей постановке. Пусть имеется одна таблица БД из n строк и m столбцов. Для любой строки таблицы в нулевом столбце таблицы определено имя субъекта, в последующих с 1 по $m - 1$ столбцов определены данные атрибутов субъекта. Пусть запрос к таблице образован посредством $m - 1$ конъюнкций элементарных запросов $\exists Z_i, i = \overline{1, m - 1}$, т.е. запрос $Z = \bigvee_{i=1}^{m-1} \exists Z_i$.

В результате обработки запроса должна быть образована новая таблица из $k, k \leq n$, строк, удовлетворяющих заданному запросу. Заметим, что для уменьшения времени формирования новой таблицы может быть выполнено косвенно посредством указаний адресов строк первичной таблицы, для которых запрос выполнен успешно.

Для ВС с традиционной структурой, использующей обычную оперативную память (ОП) и арифметико-логическое устройство, минимальное время обработки

запроса при упорядоченной обработки элементарных запросов может быть оценено на основе скорректированного выражения [6]:

$$\begin{aligned}
 T = & n ((T_{чтi_1} + T_{срi_1}) + (T_{чтi_2} + T_{срi_2} + T_{зпи_1}) P_{i_1} + \\
 & + (T_{чтi_3} + T_{срi_3} + T_{зпи_2}) P_{i_1} P_{i_2} + (T_{чтi_4} + T_{срi_4} + T_{зпи_3}) P_{i_1} P_{i_2} P_{i_3} + \dots + \\
 & + (T_{чтi_{m-1}} + T_{срi_{m-1}} + T_{зпи_{m-2}}) P_{i_1} P_{i_2} \dots P_{i_{m-2}} + T_{зпи_{m-1}} P_{i_1} P_{i_2} \dots P_{i_{m-1}}) = \\
 = & n ((T_{чтi_1} + T_{срi_1}) + \sum_{j=2}^{m-1} (T_{чтi_j} + T_{срi_j} + T_{зпи_j}) \prod_{l=1}^{j-1} P_{i_l} + T_{зпи_{m-1}} \prod_{l=1}^{m-1} P_{i_l}),
 \end{aligned} \tag{1}$$

где параметры $T_{чтi_j}$, $T_{срi_j}$ и $T_{зпи_j}$ - соответственно, времена чтения, сравнения и записи данных и P_{i_j} - частота успеха, соответствующие номеру i_j - го столбца каждой из n строк, при этом номера столбцов (элементарных запросов) обрабатываются последовательно в порядке i_1, i_2, \dots, i_{m-1} , который определен для неупорядоченных табличных данных выполнением соотношений [6]:

$$\frac{T_{i_1}}{1 - P_{i_1}} < \frac{T_{i_2}}{1 - P_{i_2}} < \dots < \frac{T_{i_{m-1}}}{1 - P_{i_{m-1}}} . \tag{2}$$

Уменьшение времени обработки запроса может быть достигнуто за счет одновременного (параллельного) выполнения операции сравнения, что может быть реализовано на основе ассоциативной памяти (АП).

Дополнительное уменьшение времени обработки запроса можно получить при одновременном (на основе расслоения оперативной памяти) чтении необходимых данных в ассоциативную память.

Исходя из указанных соображений, далее мы рассмотрим задачу определения структуры ВС для обработки упорядоченных запросов на основе АП, т.е. ассоциативной вычислительной системы (АВС).

3. Реализация АВС

Для простоты рассуждений, положим, что каждая клетка i -го столбца, $i = \overline{0, m - 1}$, занимает в памяти ОП одну ячейку, начиная с базовой ячейки под номером B . Исходя из этого, ячейки имен строк имеют последовательные номера $B, B + m, B + 2m, \dots, B + (n-1)m$, а номера ячеек таблицы в ОП могут быть представлены в виде таблицы 1.

Таблица 1. Расположение таблицы в ОП

	Номер столбца таблицы			
Номер строки таблицы	0	1	2	$m-1$
1	B	$B + 1$	$B + 2$	$B + m - 1$
2	$B + m$	$B + m + 1$	$B + m + 2$	$B + 2m - 1$
i	$B + im$	$B + im + 1$	$B + im + 2$	$B + (i+1)m - 1$
n	$B + (n-1)m$	$B + (n - 1)m + 1$	$B + (n - 1)m + 2$	$B + nm - 1$

Базовая структура АВС содержит расслоенную ОП и ассоциативную матрицу (АМ), см.рис.1

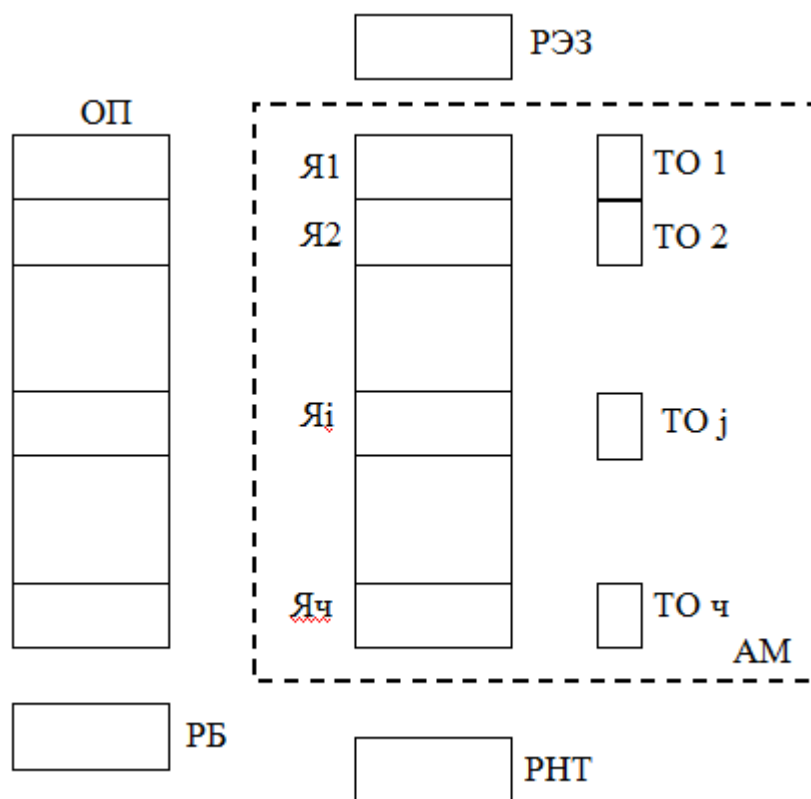


Рис .1. Базовая структура АВС

Расслоенная структура ОП предполагает реализацию в ОП нескольких слоев (банков данных), по которым должным образом распределяется информация (здесь строки таблицы данных) так, чтобы обеспечить одновременное (параллельное) чтение требуемой информации из ОП в АМ.

На начальном этапе в r – ячеек $Я_i, i = \overline{1, r}$, АМ поступает подмассив данных из первых r – строк i_1 - го столбца таблицы в соответствии с начальным адресом В, содержащемся в базовом регистре адреса, РБ, этого подмассива. В регистр элементарного запроса (РЭЗ) записывается i_1 -ый элементарный запрос $Зi_1$. Триггеры

– ответчики сбрасываются в нуль. После выполнения операции сравнения содержимого ячеек с содержимым РЭЗ ряд триггеров - ответчиков переходит в состояние 1. Номера единичных триггеров сохраняются в регистре РНТ. Эти номера являются относительными адресами подмассива, которые должны участвовать при сравнении следующих элементарных запросов с соответствующими данными.

Возможны два основных метода выполнения запроса: с развитием в глубину (последовательная обработка столбцов таблицы) и с развитием в ширину (последовательная обработка строк таблицы), а также комбинированный метод с развитием в глубину и ширину. Далее в работе рассмотрим первый метод.

4. Метод «развитие в глубину»

4.1. Введение в метод

Здесь первый элементарный запрос с номером столбца i_l выполняется для всех n строк таблицы. Для этого потребуется провести n/c операций чтения и сравнения, при этом с целью возможности продолжения выполнения второго элементарного запроса необходимо сохранить начальные адреса и относительные адреса (номера единичных триггеров - ответчиков) для каждого из n/c шагов (для простоты рассуждений полагаем, что отношение n/c делится нацело).

Для хранения массива начальных адресов, массива относительных адресов и массива длины относительного адреса (определяемой числом единичных триггеров - ответчиков) введем, соответственно регистры адреса начала подмассива, РАН (d), массив регистров РНТ(d) и массив регистров РЧТ(d), $d = \overline{1, r}$, а так же коммутатор 1,

КР1, коммутатор 2, КР2, регистры адреса памяти, РАП(d), $d = \overline{1, r}$, и регистры относительного адреса памяти, РАОП(d), $d = \overline{1, r}$, см. рис. 2.

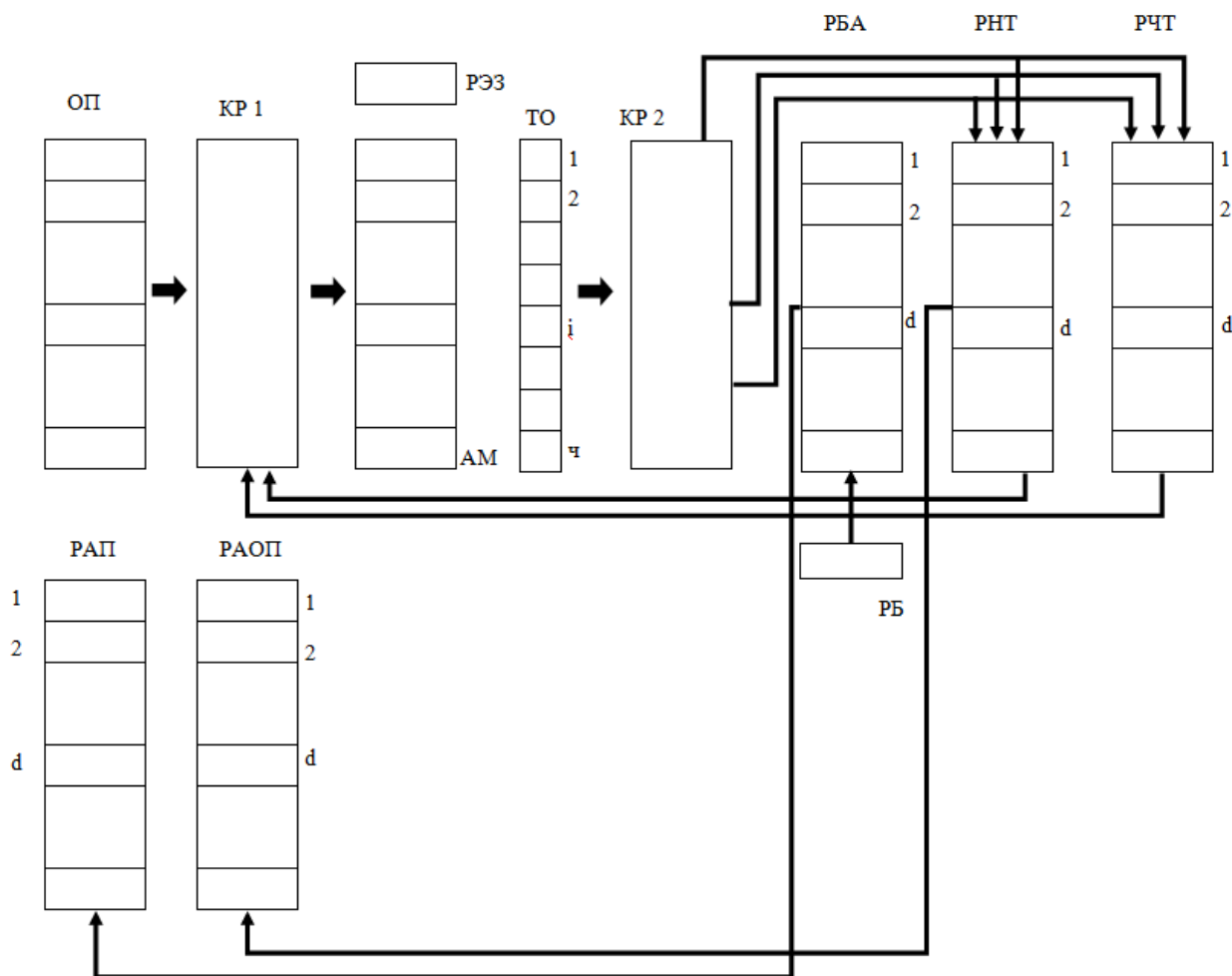


Рис . 2 . Структура АВС метод «развитие в глубину»

4.2. Алгоритм функционирования АВС при обработке первого элементарного запроса

п.1. Сформировать в соответствии с выполнением соотношений (2) вектор порядка обработки столбцов таблицы i_1, i_2, \dots, i_{m-1} в регистре порядка обработки элементарных запросов – РПО

п.2. Записать в базовый регистр адреса, РБ, адрес таблицы B

$РБ := B$

п.3. Положить порядковый номер элементарного запроса

$j := 1$

п.4. Записать в регистр элементарного запроса, РЭЗ, из регистра РПО значение элементарного запроса, соответствующего i_j - му элементарному запросу

п.5. Положить начальный номер подмассива

$d := 1$

п.6. Образовать в регистре РБА адрес начала подмассива i_j - го столбца

$РАН(d) := (РБ) + i_j + (d - 1) rm$

п.7. Запись адреса подмассива i_j - го столбца в регистр памяти РАП

$РАП(d) := РБА(d)$

п.8. Чтение подмассива длиной r ячеек из ОП в АМ

п.9. Сброс всех триггеров-ответчиков, ТО, в нулевое состояние

п.10. Сравнить содержимое регистра элементарного запроса, РЭЗ, с содержимым ячеек АМ

п.11. Записать в регистр номеров триггеров-ответчиков, РНТ(d), номера ТО, имеющих единичное значение

п.12. Записать в регистр числа единичных триггеров-ответчиков, РЧТ (d), число, равное числу единичных ТО

П.13. Увеличить значение номера подмассива

$$d := d + 1$$

п.14. Если число подмассивов, образующих массив столбцов таблицы $d < n/r + 1$, то переход на п.6, иначе следующий п.15.

4.3. Алгоритм функционирования ABC при обработке последующих элементарных запросов

4.3.1. Комментарий 1

После перехода к п.15 первый элементарный запрос Zi_l обработан для всех строк первого i_l -го столбца. При этом регистры адреса начала подмассива, РАН (d), $d = \overline{1, n/r}$, содержат n/r начальных адресов i_l -го столбца данных, а именно: (РАН (1)) = $B + i_l, \dots$, (РАН(2)) = $B + rm + i_l, \dots$, (РАН (n/r)) = $B + (n-r)m + i_l$, а регистры номеров единичных триггеров-ответчиков, РНТ(d), $d = \overline{1, n/r}$, содержат указание на номер строки подмассивов, которые удовлетворили первому элементарному запросу Zi_l , а регистры числа триггеров, РЧТ(d), $d = \overline{1, n/r}$, содержат число, равное количеству строк.

Для обсуждения последующих пунктов алгоритма здесь в комментарии, для конкретности, но, не умаляя общности, положим, что число строк $n = 16$ и число ячеек АМ $r = 16$.

Пусть после обработки первого элементарного запроса Zi_l , регистр РНТ(1) содержит 5 единичных ТО: 4,5,7,8,10, остальные ТО содержат нулевые значения,

поэтому $(PЧТ(1))=5$. Будем также считать, что $(PНТ(2)) = 4,9,13,16$; $(PЧТ(2)) = 4$; $(PНТ((3)) = 1,2,3,5,6,7,10,11,16$; $(PЧТ(3)) = 9$; $(PНТ(4)) = 1,5,6,8,9,10,13$; $(PЧТ(4)) = 7$; $(PНТ(5)) = 4,5,7,9$; $(PЧТ(5)) = 4$; $(PНТ(6)) = 2,4,9,10,11,13$; $(PЧТ(6)) = 6$.

При обработке второго элементарного запроса Zi_2 , в соответствии с начальным адресом столбца данных $B + i_2$, и смещением адреса $4m, 5m, 7m, 8m$, и $10m$ из ОП будет считана информация в пять первых ячеек ассоциативной матрицы: Я(1) - Я(5). Пусть при выполнении операции сравнения 5-ти ячеек АМ со значением соответствующим элементарному запросу Zi_2 единичные значения получили ТО с номерами 1,2 и 4 в силу совпадения с ячейками 1, 2 и 4. В соответствии с содержимым $(PЧТ(1)) = 5$ образуем двоичный вектор ДВ (1) длиной 5 двоичных разрядов: (11010), см. табл. 2.

Таблица 2. Образование двоичного вектора ДВ (1)

Номера ТО	1	2	3	4	5
Номера ТО со значением 1	1	2		4	
Двоичный вектор ДВ(1)	1	1	0	1	0

Полученный двоичный вектор ДВ(1) дает возможность скорректировать информацию в регистрах РНТ(1) и РЧТ(1), см. табл.3, посредством аннулирования тех номеров ТО, для которых значение разрядов ДВ(1) равны 0.

Таблица 3. Коррекция регистров РНТ(1) и РЧТ(1)

РНТ (1)	4	5	7	8	10
DB(1)	1	1	0	1	0
Операция «если (DB(1)) = 1, то значение РНТ (1)»	4	5		8	

В результате получили новые значения регистров: (РНТ(1))=4,5,8 и (РЧТ(1))=3.

Производительность АВС может быть увеличена, если при обработке второго и последующих элементарных запросов использовать все r (16) ячеек АМ.

Для этого в нашем случае на первом этапе обработки второго элементарного запроса $3i_2$ информация копируется в АМ из ОП:

из первого подмассива в соответствии с начальным адресом $(B + i_2 + 1)$ и смещением адреса $4m, 5m, 7m, 8$ и $10m$ в ячейки Я1- Я5;

из второго подмассива в соответствии с начальным адресом $(B + i_2 + 1 + rm)$ и смещением адреса $4m, 9m, 13m$ и $15m$ в ячейки Я6+Я9;

из третьего подмассива в соответствии с начальным адресом $(B + i_2 + 1 + 2rm)$ и смещением адреса $1m, 2m, 3m, 5m, 6m, 7m,$ и $10m$ в ячейки Я10+Я16.

Отметим, что третий подмассив считан из ОП в АМ не полностью, для указания этого положим значения параметров $F2 = 1$, $D2 = 1$.

Будем считать, что информация i_2 столбца в 4, 5 и 8 строках первого подмассива, в 4,13 и 16 строках второго подмассива, в 2, 3, 5, 6 и 10 строках третьего подмассива совпадают со значением элементарного запроса Zi_2 .

В результате получаем следующие значения триггеров-ответчиков и двоичных векторов, см. табл. 4.

Таблица 4. Значения триггеров-ответчиков и двоичных векторов на первом этапе

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ТО																
DB	1	1	0	1	0	1	0	1	1	0	1	1	1	1	0	1
	DB(1)				DB(2)				DB*(3)							

В соответствии с полученными двоичными векторами необходимо скорректировать содержимое регистров РНТ(1)-РНТ(3) и РЧТ(1)-РЧТ(3). Так как параметры $F2 = 1$, $D2 = 1$, то двоичный вектор $DB*(3)$ является усеченным - он содержит 7-мь разрядов. Увеличим число его разрядов в соответствии со значением регистра числа единичных триггеров-ответчиков РЧТ(3) до 9 разрядов посредством добавления справа 2-ух единичных разрядов, см. табл. 5.

Таблица 5. Коррекция регистров РНТ(1) - РНТ(3) и РЧТ(1) - РЧТ(3)

PHT(1)					PHT(2)				PHT(3)								
4	5	7	8	10	4	9	13	16	1	2	3	5	6	7	10	11	16
1	1	0	1	0	1	0	1	1	0	1	1	1	1	0	1	1	1
DB(1)					DB(2)				DB(3)								

Новые значения регистров имеют вид: (PHT(1)) = 4,5,8; (PHT(1)) = 3; (PHT(2)) = 4,13,16; (PHT(2)) = 3; (PHT(3)) = 2,3,5,6,10,11,16; (PHT(3)) = 7.

Значение параметра $F2 = 1$ указывает на необходимость использования информации из последнего (третьего) подмассива на втором этапе. что отмечается посредством задания значений параметров $F1$ и $D1$: $F1 = 1, D1 = D2 = 1$.

На втором этапе формирования следующего массива в АМ загружается информация:

в соответствии со значениями $F1=1$ и $D1=2$ из двух последних строк третьего подмассива ОП, используя начальный адрес третьего подмассива $(B + i_2 + 1 + 2rm)$ и смещения адресов $11m$ и $16m$, в ячейки Я1, Я2 АМ;

из четвертого подмассива ОП в соответствии с начальным адресом $(B + i_2 + 1 + 3rm)$ и смещениями адресов $1m, 5m, 6m, 8m, 9m, 10m$, и $13m$ в ячейки Я3-Я9 АМ;

из пятого подмассива ОП в соответствии с начальным адресом $(B + i_2 + 1 + 4rm)$ и смещениями адресов $4m, 5m, 7m$, и $9m$ в ячейки Я10 – Я13 АМ.

Так как теперь в АМ остались свободными 3 ячейки, то из шестого подмассива ОП будет загружена информация только из трех строк, в соответствии с

начальным адресом ($B + i_2 + 1 + 5rm$) и смещениями адресов $2m$, $4m$ и $9m$ в ячейки Я14-Я16 АМ.

При этом, так как длина шестого подмассива больше трех ($PЧТ(5) = 6$), то флаг F2 получает значение 1 и параметр Д2 получает значение, равное числу загруженных строк шестого подмассива, т.е. $Д2 = 3$, $F2 = 1$.

Будем считать, что информация i_2 - го столбца в 11 строке третьего подмассива, в 1, 6, 9, 10, и 13 строках четвертого подмассива, 4,5 и 7 строках пятого подмассива и в 3 строке шестого подмассива совпадают со значением соответствующим $ЭЗi_2$.

В результате выполнения операции «Сравнить» получаем следующие значения триггеров-ответчиков и двоичных векторов, см. таблицу 6.

Таблица 6. Значения триггеров-ответчиков и двоичных векторов на втором этапе

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ТО																
ДВ	1	0	1	0	1	0	1	1	1	1	1	1	0	0	0	1
	ДВ* (3)	ДВ(4)					ДВ(5)					ДВ*(6)				

При этом, т.к. $F1=1$ и $Д1=2$, то двоичный вектор ДВ*(3) является усеченным слева на пять разрядов ($(PЧТ(3))-(Д1) = 7-2 = 5$), которые заполняются 5-ью левыми единицами.

Аналогично, т.к. $F2=1$ и $D2=3$, то двоичный вектор $DВ^*(6)$ является усеченным справа на три разряда ($(PЧТ(6)) - (D2) = 6-3 = 3$), которые заполняются 3-мя правыми единицами.

В соответствии с полученными двоичными векторами необходимо скорректировать содержимое регистров $PНТ(3) - PНТ(6)$ и $PЧТ(3) - PЧТ(6)$, см. таблицу 7, и значения флагов.

Таблица 7. Коррекция регистров $PНТ(3) - PНТ(6)$ и $PЧТ(3) - PЧТ(6)$

PНТ (3)							PНТ (4)							PНТ (5)				PНТ (6)					
2	3	5	6	10	11	16	1	5	6	8	9	10	13	4	5	7	9	2	4	9	10	11	13
1	1	1	1	1	1	0	1	0	1	0	1	1	1	1	1	1	0	0	0	1	1	1	1
ДВ (3)							ДВ (4)							ДВ (5)				ДВ (6)					
$(PЧТ(3)) = 7$							$(PЧТ(4)) = 7$							$(PЧТ(5)) = 4$				$(PЧТ(6)) = 6$					

Скорректированные значения регистров и флагов имеют вид:

$(PНТ(3)) = 2,3,5,6,10,11$; $(PЧТ(3)) = 6$,

$(PНТ(4)) = 1,6,9,10,13$; $(PЧТ(4)) = 5$,

$(PНТ(5)) = 4,5,7$; $(PЧТ(5)) = 3$,

$(PНТ(6)) = 9,10,11, 13$; $((PЧТ(6)) = 4$,

$F1=1$, $D1 = D2$, т.е. $D1 = 3$,

$F2=0$.

Заметим, что если строки последнего подмассива помещаются в АМ целиком, не оставляя свободных ячеек АМ, то $F1:=0$.

Далее, процесс обработки запросов продолжается.

4.3.2. Этап 1

п.15. Положить порядковый номер элементарного запроса

$$j = 2$$

п.16. Записать в регистр элементарного запроса, РЭЗ, значение элементарного запроса, соответствующего i_j -му элементарному запросу

п.17. Положить $F1=0$

п.18. Положить номер начального подмассива при загрузке ячеек АМ

$$v = 1$$

п.19. Сформировать значение длины первого подмассива w при загрузке ячеек АМ:

$$w = \begin{cases} \text{РЧТ}(v), & \text{если } F1=0 \\ \text{РЧТ}(v) - D1, & \text{если } F1=1 \end{cases}$$

п.20. Если $w + \sum_{d=v}^{n/r} (\text{РЧТ}(d)) = 0$, то п.55, иначе п.21

4.3.3. Комментарий 2.

Определяем верхнюю границу s подмассивов, загружаемых из ОП в АМ с целью эффективного использования всех ячеек АМ. Для этого надо проверить выполнение одного из трех условий.

4.3.4. Этап 2

п.21. Проверка условия 1 (невозможность полного заполнения АМ)

Если $w + \sum_{d=v}^t (\text{РЧТ}(d)) < r$ при $t < n/r + 1$, то п.22, иначе п.26

п.22. Верхняя граница s загружаемых подмассивов

$$s := n/r$$

п.23. Промежуточный номер начального подмассива следующего этапа загрузки ячеек АМ

$$vI := s + 1$$

п.24. $F_2 := 0$

п.25. Идти к п.36

п.26. Условие 2. (последний целиком загружаемый подмассив обеспечивает полное заполнение АМ), при этом определяем значение t из условия выполнения соотношения:

Если $(w + \sum_{d=v}^t (\text{РЧТ}(d)) < r) \wedge (w + \sum_{d=v}^{t+1} (\text{РЧТ}(d)) = r)$, то п.27, иначе п.31

п.27. Верхняя граница s загружаемых подмассивов

$$s := t + 1$$

п.28. Параметр переполнения

$$F2 := 0$$

п.29. Промежуточный номер начального подмассива следующего этапа загрузки ячеек АМ равняется следующему номеру по отношению к номеру последнего подмассива текущего этапа загрузки ячеек АМ

$$v1 := s + 1$$

п.30. Идти к п.36

п.31. Условие 3. (последний загружаемый подмассив обеспечивает полное заполнение АМ, но не может быть помещен в АМ целиком), при этом определяем значение t из условия выполнения соотношения:

$$\text{Если } (w + \sum_{d=v}^t (\text{РЧТ}(d)) < r) \wedge (w + \sum_{d=v}^{t+1} (\text{РЧТ}(d)) > r), \text{ то п.32, иначе п.36}$$

п.32. Верхняя граница s загружаемых подмассивов

$$s = t + 1$$

п.33. Параметр переполнения

$$F2 := 1$$

п.34. Число строк переполнения подмассива

$$D2 := r - (w + \sum_{d=v}^t (\text{РЧТ}(d)))$$

п.35. Номер начального подмассива следующего этапа загрузки ячеек АМ равняется номеру последнему подмассива текущего этапа загрузки ячеек АМ

$vI := s$

4.3.5. Комментарий 3

Заполнить адресный массив ОП базовыми адресами и смещениями, при этом адреса ячеек ОП образуются как исполнительные адреса ИА := (РАП(d)) + m * (номера ТО, содержащиеся в РАОП(d)).

4.3.6. Этап 3

п.36. Для любого $d = \overline{v, s}$, если (РЧТ(d)) > 0, то РАП(d) := (РАН(d)) + i_j , далее п.37

п.37. Для любого $d = \overline{v + 1, s - 1}$, если (РЧТ(d)) > 0, то РАОП(d) := (РНТ(d)), далее п.38

п.38. Если (РЧТ(v) > 0) ^ F1=1, то РАОП(v) := правые (РЧТ(v) - D1) номера ТО из регистра РНТ(v), далее п.39

п.39. Если (РЧТ(v) > 0) ^ (F1=0), то РАОП(v) := РНТ(v), далее п.40

п.40. Если (РЧТ(s) > 0) ^ (F2=1), то РАОП(s) := левые (РЧТ(s) - D2) номера ТО из регистра РНТ(s), далее п.41

п.41. Если (РЧТ(s) > 0) ^ (F2=0), то РАОП(s) := РНТ(s).

п.42. Чтение содержимого ячеек ОП в ячейки АМ

п.43. Установить триггера - ответчики в состояние 0

п.44. Сравнить содержимое ячеек АМ с содержимым регистра РЭЗ

п.45. Образовать последовательно двоичные векторы:

ДВ(v), длиной (РЧТ(v), если $F1 = 0$, или длиной (РЧТ(v)) - Д1, если $F1=1$,

ДВ(d), $d = \overline{v + 1, s - 1}$, длиной РЧТ(d);

ДВ(s), длиной (РЧТ(s)), если $F2 = 0$; или (РЧТ(d)- Д2), если $F2 = 1$.

4.3.7. Комментарий 4

Двоичные векторы ДВ(v) и ДВ(s) должны быть скорректированы, если $F1=1$ и $F2=1$

4.3.8. Этап 4

п.46. Если $F1 = 1$, то разряды двоичных векторов ДВ(v) должны быть распространены влево на (РЧТ(v)-Д1) разрядов с единичными значениями так, что бы длина ДВ(v) сравнялась со значением (РЧТ(v)).

п.47. Если $F2 = 1$, то разряды двоичных векторов ДВ(s) должны быть распространены вправо на (РЧТ(s) - Д2) разрядов с единичными значениями так, что бы длина ДВ(s) сравнялась со значением (РЧТ(s)).

4.3.9. Комментарий 5

Коррекция содержимого регистров РНТ(d) и РЧТ(d).

4.3.10. Этап 5

п.48. $РНТ(d) := РНТ(d) \wedge ДВ(d)$, $d = \overline{v, s}$.

п.49. $РНТ(d) :=$ содержимое (РНТ(d)) без удаленных номеров ТО

п.50. Получить $C(d) = \sum_i ДВ_i(d)$, где $ДВ_i(d)$ есть i -ый разряд двоичного вектора ДВ(d), $d = \overline{v, s}$.

п.51. $PЧТ(d) := C(d), d = \overline{v, s}$.

п.52. Если $F2=1$, то присвоить $F1 := 1, Д1 := Д2$, иначе п.52

п.53. Если $F2=0$, то $F1 := 0$

п.54. $v := vI$

п.54. Если $v < n/r + 1$, идти к п.19, иначе п.55

п.55. Увеличить порядковый номер элементарного запроса

$j := j + 1$

56. Если $j < m + 1$, идти к п.16, иначе п.57

4.3.11. Комментарий 6

Формирование адресного массива ОП новой таблицы

4.3.12. Этап 6

п.57. Для любого $d = \overline{1, r}$, если $(PЧТ(d)) > 0$, то $РАП(d) := РАН(d), РАОП(d) := РНТ(d)$

5. Оценка эффективности метода

Полагая, что расслоение ОП обеспечивает параллельную загрузку всех ячеек ОП одновременно, время обработки упорядоченных запросов в АВС может быть оценено посредством модификации выражения (1):

$$T_{ABC} = (n/r) ((T_{чтi_1} + T_{срi_1}) + \sum_{j=2}^{m-1} (T_{чтi_j} + T_{срi_j} + T_{зпи_j}) \prod_{l=1}^{j-1} P_{i_l} + T_{зпи_{m-1}} \prod_{l=1}^{m-1} P_{i_l}), \quad (3)$$

Имея ввиду, что число r ячеек АП, как и число k регистров РНА, РНТ и РЧТ, является фиксированным в АВС, осуществимость метода выполнения упорядоченных запросов с развитием в глубину достигается при условии $n/r < k+1$. Поэтому при условии $k < n/r$, требуется использовать комбинированный метод выполнения упорядоченных запросов с развитием в глубину и ширину.

6. Выводы

- 6.1. Предложена архитектура АВС для выполнения упорядоченных элементарных запросов.
- 6.2. Разработан алгоритм обработки упорядоченных элементарных запросов на АВС.
- 6.3. Выполнена оценка эффективности обработки запросов на АВС.
- 6.4. Использование АВС для выполнения упорядоченных элементарных запросов может быть пролонгировано на случай вложенных запросов и мультizaпросов, при этом требуется исследовать эффективность обработки запросов мультипроцессорными (мультиассоциативными) ВС.

Библиографический список

1. Кохонен Т. Ассоциативные запоминающие устройства. – М.: Мир, 1982. - 384 с.
2. Бурцев В.С. Система массового параллелизма с автоматическим распределением аппаратных средств суперЭВМ в процессе решения задачи. Юбилейный сборник трудов институтов Отделения информатики, вычислительной техники и автоматизации РАН, Москва, 1993, Т. II, стр. 5-27.

3. Брехов О.М., Морару В.А. Аналитическое моделирование ЭВМ, управляемой потоком данных // Автоматика и телемеханика. 1993. № 12. С. 165–178.
4. J. G. D. da Silva, I. Watson, A. C. D. de Figueiredo. Associative Memory and Database Machines — A Brief Survey and a Design Proposal Using Hardware Hashing NATO ASI Series Volume 24, 1986, pp 547-555.
5. Rony Zarom, Kenneth Ross, Kenneth Yip. Using associative memory to perform database operations US20040172400 A1. US 10/483,409, 2004
6. Брехов О.М. Аналитическая оценка оптимальной обработки запросов // Успехи современной радиоэлектроники. 2012. Т.12. № 7. С. 37-45.