МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)
Кафедра информационных систем и технологий

УТВЕРЖДАЮ
Проректор по учебной работе
_____ О.Г. Локтионова
«15» 03 2018 г.

Арбитры и процессы арбитража: методические указания к практическим занятиям по направлениям, укрупнённой группы специальностей 09.00.00

Курск 2018

УДК 004

Составитель: Е.А. Титенко

Рецензент
Кандидат технических наук *В.С. Панищев*

Арбитры и процессы арбитража: методические указания к практическим занятиям по направлениям, укрупнённой группы специальностей 09.00.00 / Юго-Зап. гос. ун-т; сост. Е.А. Титенко. - Курск, 2018. 10 с. Библиогр.: с. 10.

Приводится описание процесса разрешения конфликта с программированием конфликтных ячеек в решетчатых арбитрах. Приведены теоретические положения, основные правила настройки решетчатого арбитра, практические примеры и практические задания.

Методические рекомендации предназначены для студентов, обучающихся по направлениям, укрупнённой группы специальностей 09.00.00.

Текст печатается в авторской редакции.

Подписано в печать 15.02.18 . Формат 60x84 1/16.
Усл.печ. л. 0,58 п.л . Уч.-изд. л. 0,53. Тираж 100 экз. Заказ. 1603
Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Цель практической работы: Изучение основных принципов работы программируемых арбитров, приобретение практических навыков динамического управления системой приоритетов, формирование траектории приоритетного запроса в структуре арбитра.

1. Основные понятия

1.1 Основные понятия многозадачного режима работы

Работа современных операционных систем опирается на многозадачный (многопоточный) режим планирования, диспетчеризации и выполнения задач или процессов в соответствии с определенными стратегиями обслуживания. Одновременное выполнение нескольких задача или нескольких потоков ограничивается невозможностью одновременного их обращения к неразделяемым ресурсам (процессор, память, система прерываний, каналы прямого доступа к памяти, системные библиотеки и т.д.). В связи с этим на аппаратно-программном уровне возникает необходимость в динамическом управлении доступом к неразделяемым ресурсам со стороны одновременно выполняющихся задач или потоков. Для разрешения возникающих конфликтов за общий неразделяемый ресурс вводится специальный системный процесс арбитража и диспетчеризации, который может иметь аппаратную или программную реализации. В первом случае устройство-арбитр осуществляет обработку асинхронно поступающих запросов от устройств, во втором случае программа-арбитр, входящая в состав системного ПО, осуществляет диспетчеризацию обращений от системных или прикладных программ.

Арбитр – это самостоятельный функциональный узел (рис.1), который обрабатывает k-битовый вектор запросов от устройств и посылает им kразрядный вектор подтверждений с единственной логической единицей, соответствующей приоритетному каналу.

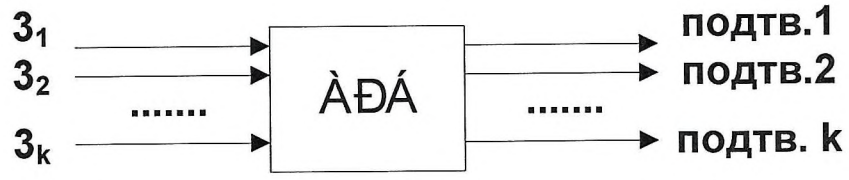


Рис. 1 Представление арбитра как «черного» ящика

Одним из возможных путей повышения гибкости арбитра является периодическая программная перестройка его приоритетной структуры. Операционная система, например, через каждую миллисекунду меняет местами приоритеты каналов по определенному алгоритму. Поэтому низкоприоритетные каналы периодически становятся высокоприоритетными, и наоборот. При хаотическом «перемешивании» приоритетов каналов все они оказываются в равных условиях. При целенаправленном перемешивании приоритетность в целом сохраняется, но эффекта «грубого оттеснения» нет..

Среди арбитров наибольший интерес представляют арбитры с изменяемым законом приоритетов запросов. Изменяемый закон необходим для предоставления доступа к общему ресурсу всем нуждающимся задачам или устройствам.

Отличительная особенность программируемых решетчатых арбитров заключается в том, что закон арбитража зависит не только от количества запросов, но и от структурного соотношения между ними.

На рис. 2, рис.3, рис. 4 представлены схемы решетчатых (программируемых) арбитров, реализующих 128 различных приоритетных соотношений в 8-разрядном векторе запросов в зависимости от 28-разрядного кода управляющего регистра приоритетов Q. Каждому элементарной ячейке арбитра соответствует один разряд Q_i программно-доступного регистра приоритетов (на рис.2,3,4 не показан). При отсутствии конфликтов элементарная ячейка транслирует сигналы a и b на выходы e и d без изменения. При конфликте элементарная ячейка «руководствуется указаниями» бита управляющего разряда Q_i. Доказано, что любой конфликтной ситуации на входах арбитра независимо от кода в регистре приоритетов на выходы пройдет только один сигнал. Перестройка структуры приоритетов достигается изменением кода в управляющем регистре Q.

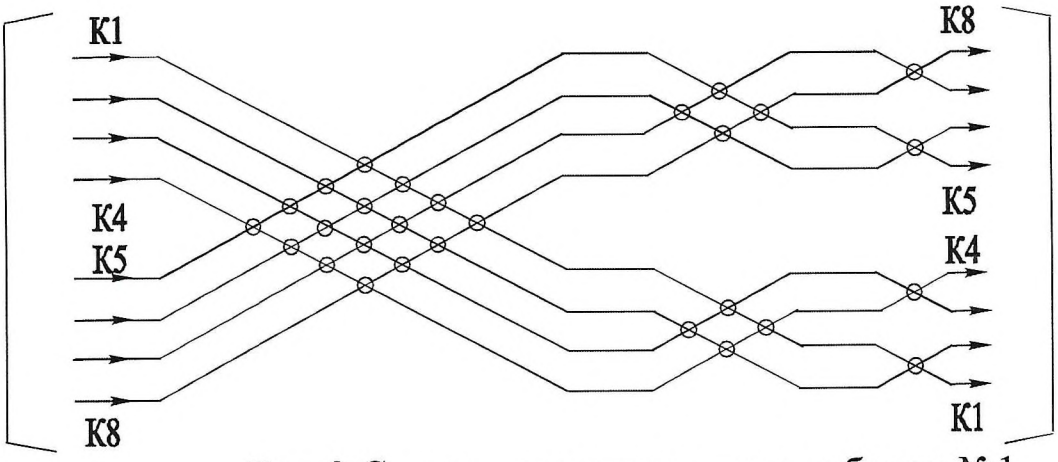


Рис.2 Структура решетчатого арбитра №1

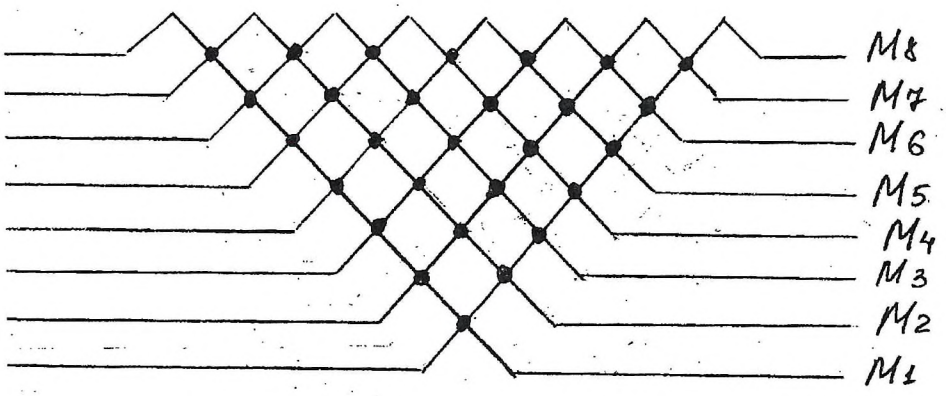


Рис.3 Структура решетчатого арбитра №2

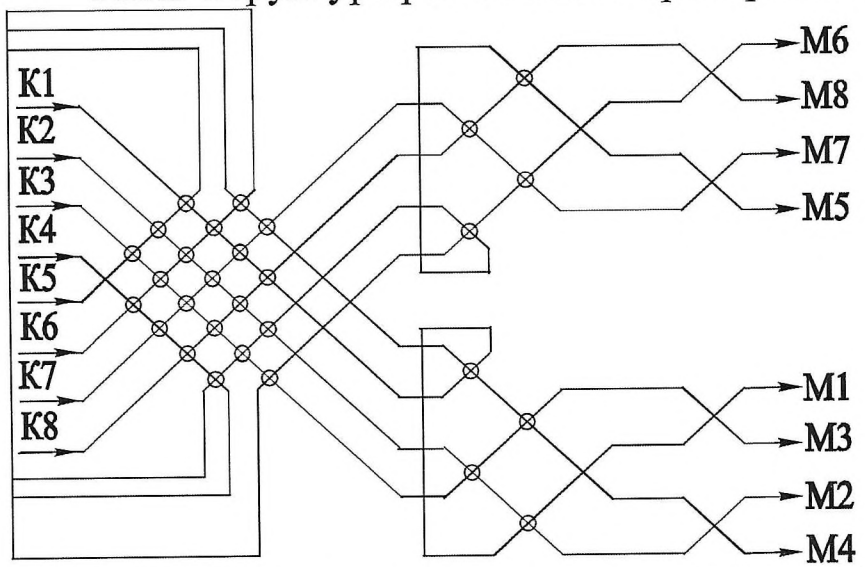


Рис.4 Структура решетчатого арбитра №3

Взаимоотношения между любой парой конкурирующих запросов однозначно определяются битом Q_i управляющего регистра в соответствии таблицей работы, имеющей два варианта формирования выходов c-d, c'-d' (табл.) элементарной ячейки (рис.5).

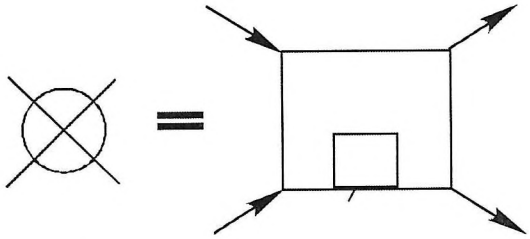


Рис.5 Схема элементарной конфликтной ячейки

Таблица – Работа конфликтной ячейки решетчатого арбитра

a	b	Q	c	d	c'	d'
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	1	0	1
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	1	0	1	0
1	1	0	0	1	1	0
1	1	1	1	0	0	1

Арбитр устраняет любые конфликты при любом коде в регистре, однако в зависимости от кода он может быть настроен либо на один из жестких, либо на один из гибких режимов.

2 Примеры реализации настройки решетчатого арбитра

Пусть, например, требуется обеспечить следующую приоритетность запросов для арбитра на рис. 2 (прямая задача)

К3->К2->К8->К6->К7->К1->К5->К4.

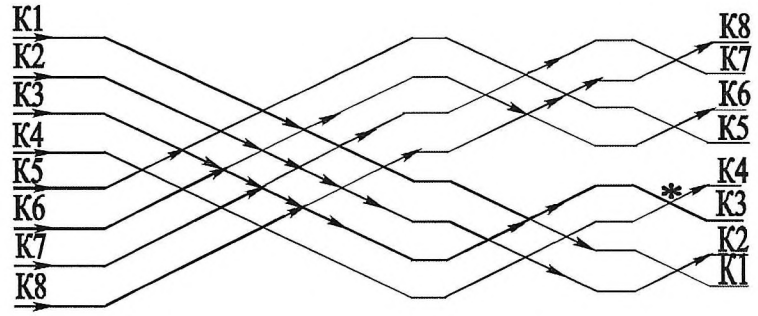
Стрелка в такой записи направлена от более приоритетного входа к менее приоритетному. Для обеспечения указанного режима код в управляющем регистре должен настроить узловые схемы следующим образом:

К3->К2, К3->К8, К3->К6, К3->К7, К3->К1, К3->К5, К3->К4*,
К2->К8, К2->К6, К2->К7, К2->К1, К2->К5, К2->К4,
К8->К6, К8->К7, К8->К1, К8->К5, К8->К4,
К6->К7, К6->К1, К6->К5, К6->К4,
К7->К1, К7->К5, К7->К4,
К1->К5, К1->К4,
К5->К4.

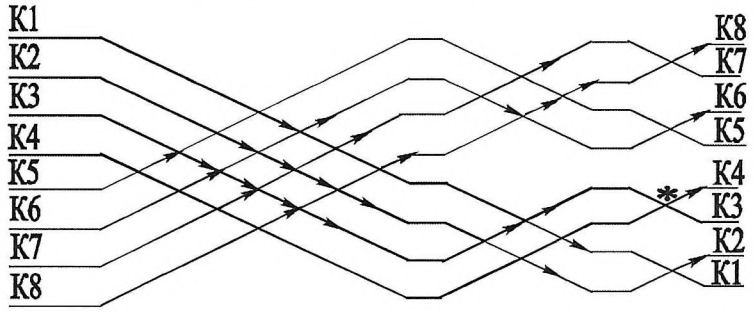
Каждое из этих условий задается значением соответствующего бита в управляющем 28-разрядном регистре. Аналогично можно задать любой жесткий приоритетный порядок между каналами. Число таких режимов равно числу перестановок между номерами каналов P_8=40 320. Однако это число не исчерпывает все возможные режимы работы устройства. Действительно, поскольку число управляющих бит в арбитре в данном примере составляет C_8 = 28, то возможны 2^{28}=268 435 456 вариантов настройки, которые, за исключением уже рассмотренных, характеризуются гибким распределением приоритетов между каналами.

Смысл гибких режимов состоит в том, что приоритет между каналами определяется не только кодом в управляющем регистре, но также зависит от числа поступивших запросов и их распределения по входам арбитра. Если в предыдущем примере изменить значение только одного бита в управляющем коде, а именно бита, определяющего взаимоотношения между входами К3 и К4, то условие К3->К4, помеченное ранее знаком * , изменится на противоположное. К4->К3 («шестерка бьет туза»). Такое изменение приводит к получению одного из гибких режимов.

Предположим, что задана исходная система приоритетов, но К4->К3. Пусть запросы ступили на все входы К1—К8 и только на входы К1—К4. Для первого вектора запросов на рис.6 а) полужирными линиями показаны траектории распространения запросных сигналов через арбитр. Из рисунка сигнала видно, что при взаимодействии запросов по входам К4 и К5 побеждает К5, так как соответствующая узловая схема решетчатой структуры настроена на приоритетную передачу сигнала, поступившего на вход К5. Для вектора запросов, имеющих логические единицы на входах К1—К4, приоритетным будем запрос К3, что отражено на рис.6 б)



при наличии всех
запросных сигналов



при наличии четырех
запросных сигналов
по каналам К1-К4

а) при запросах К1-К8 б) при запросах К1-К4

Рис.6 Траектории распространения запросов в арбитре.

Другая особенность решетчатых программируемых арбитров состоит в том, схемы являются двунаправленными, что позволяет рассматривать 6 различных вариантов структур арбитров.

Обратная задача, определяющая изучения работы решетчатого арбитра является определение по заданному значению управляющего регистра Q системы приоритетов, складывающейся в арбитре.

Рассмотрим структуры четырехвходовых арбитров, для которых заданы значения Q1=111111 и Q2=100000 соответственно (рис.7).

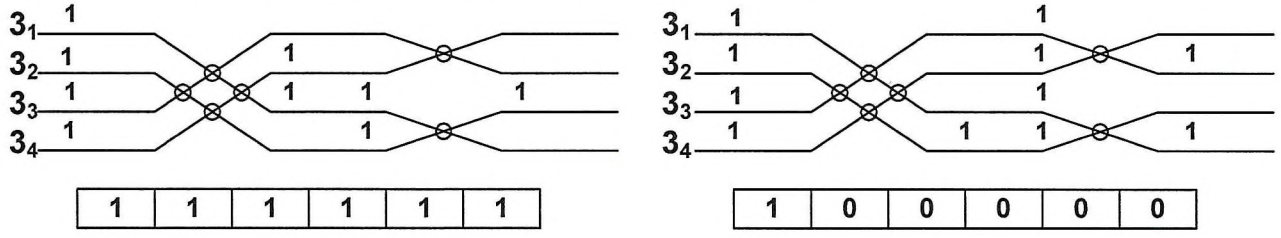


Рис. 7 Программируемый арбитр с управляющим регистром

Установление приоритетов между запросами осуществляется следующим образом. Для запросов $З_2$ и $З_3$ в левом арбитре $Q_1=1$, поэтому приоритет отдается $З_3$, то есть $З_3 \rightarrow З_2$. Для запросов $З_1$ и $З_3$ в

левом арбитре Q_1=1, поэтому приоритет отдается $З_3$, то есть $З_3 \rightarrow З_1$ и т.д. В целом для левого арбитра устанавливается система приоритетов $З_4 \rightarrow З_3 \rightarrow З_2 \rightarrow З_1$, а для правого арбитра –система приоритетов $З_1 \rightarrow З_3 \rightarrow З_2 \rightarrow З_4$

3 Индивидуальное задание студента

№ варианта	Тип арбитра	Направление	Разрядность	Q	Система приоритетов запросов
1	1	→	8	-	8,7,1,3,4,2,6,5
2	2	→	16	FOOFh	
3	3	→	8	00FAh	-
4	1	←	16	0AA0h	-
5	2	←	8	-	1,3,5,7,2,4,6,8
6	3	←	16	00F0h	-
7	1	→	16	-	5,6,7,8,1,2,3,4,12, 16,15,10,11,13,14
8	2	→	8	-	1,8,2,7,3,6,4,5
9	3	→	16	0FF0h	
10	1	←	8	-	2,3,7,8,6,1,5,4
11	2	←	16	000Fh	-
12	3	←	8	00CEh	-
13	1	→	16	FFFFh	-
14	3	←	8	FF0F	-

где ←,→ - направления входов/выходов в арбитре; «h» – шестнадцатеричный эквивалент значения управляющего регистра Q.

4 Контрольные вопросы

1. Назначение процессов арбитража в операционной системе и многопроцессорных вычислительных системах.
2. Виды неделимых ресурсов.
3. Недостатки арбитров с жесткой структурой.
4. Преимущества арбитров с программируемой структурой.
5. Таблица работы элементарной конфликтной ячейки.
6. Назначение, разрядность управляющего регистра.

6. Установление системы приоритетов по значению управляющего регистра.

7. Алгоритм формирования кода управляющего регистра по заданной системе приоритетов.

8. Особенности гибкого режима приоритетов.

9. Структуры решетчатых арбитров.

10. Недостатки арбитров с программируемой структурой.

Библиографический список

1. Сидоркина, И. Г. Системы искусственного интеллекта [Текст] : учебное пособие / И. Г. Сидоркина. - Москва : КНОРУС, 2016. - 246 с.
2. Автоматизированные информационные системы и интеллектуальные технологии [Текст] : учебное пособие / Е. А. Титенко [и др.] ; Минобрнауки России, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования "Юго-Западный государственный университет". - Курск : ЮЗГУ, 2013. - 133 с.
3. Основы построения интеллектуальных систем [Электронный ресурс]: учебное пособие/ Г.В. Рыбина. М. : Финансы и статистика, 2010. - 432 с.
4. Шевкопляс, Б.В. Микропроцессорные структуры. Инженерные решения: Справочник / Б.В. Шевкопляс - М.: Радио и связь, 2009. - 512 с.