

УДК 004.337

К ВОПРОСУ АНАЛИЗА МНОГОПРОЦЕССОРНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ С ОБЩЕЙ МНОГОМОДУЛЬНОЙ ПАМЯТЬЮ

Зоткина А.А., Мартышкин А.И.

Пензенский государственный технологический университет, Пенза,

e-mail: Alexey314@yandex.ru

В статье приводится структура многопроцессорной вычислительной системы с общей многомодульной памятью. Описано, что в такой системе возникает конфликтная ситуация в случае обращения двух или более процессоров одновременно к одному модулю памяти. При такой ситуации очередность обслуживания определяется службой приоритетов арбитра, который входит в состав коммутатора системы. Целью этой статьи является определение, справляется ли заданная многопроцессорная система с потоком задач, а также определение минимального числа процессоров, необходимого для решения конкретного потока задач. Подробно описан алгоритм работы памяти. В заключении статьи отмечено, что можно построить имитационную модель многопроцессорной вычислительной системы с общей многомодульной памятью, используя принятые допущения и используя полученные в работе выражения для определения времен состояний моделей элементов системы.

Ключевые слова: многопроцессорная система, многомодульная память, конфликт, контроллер, арбитр, поток задач, модель, программа, процессор, алгоритм

TO THE QUESTION OF THE ANALYSIS OF MULTIPROCESSOR SYSTEM WITH SHARED MULTI-MODULE MEMORY

Zotkina A.A., Martyshkin A.I.

Penza State Technological University, Penza,

e-mail: Alexey314@yandex.ru

The article describes the structure of a multiprocessor computer system with a shared multimodule memory. It is described that in such system, a conflict situation arises when two or more processors simultaneously access one memory module. In this situation, the order of service is determined by the priority service of the arbitrator, which is part of the system switch. The purpose of this article is to determine if a given multiprocessor system is cope with the task flow, as well as determining the minimum number of processors needed to solve a particular task flow. The algorithm of memory operation is described in detail. In the conclusion of the article, it was noted that it is possible to build an imitation model of a multiprocessor computer system with a common multimodule memory using the accepted assumptions and using the expressions obtained in the work to determine the times of states of the model elements of the system.

Keywords: multiprocessor system, multi-module memory, conflict, controller, arbiter, task flow, model, program, processor, algorithm

Объектом рассмотрения является многопроцессорная вычислительная система (МПВС) с общей многомодульной памятью (рисунок). В ее состав входят N процессоров (ЦП) (P) с кэш-памятью (CM), M модулей общей памяти (MM) и коммутаторов K . Число модулей памяти (МП) не превышает число процессоров ($M \leq N$).

В такой МПВС возникает конфликт в том случае, когда обращение двух или более ЦП идет одновременно к одному МП [1]. При этом очередность обслуживания определяется службой приоритетов арбитра (AR), входящего в состав коммутатора. Существует три различных приоритета арбитра: фиксированный, циклический и случайный [2–5].

В общем случае, вместо некоторых ЦП стоят устройства ввода/вывода (УВВ) со своими контроллерами прямого доступа к памяти (ПДП или КПДП). Они, как и про-

цессоры, являются активными элементами МПВС. УВВ со своими КПДП можно рассматривать как ЦП ввода-вывода. Каждый МП делается секционированным. Число секций модуля памяти определяется размером блока кэш-памяти (обычно 4 слова). Способ формирования адресов самих МП описан в [6].

Назначение рассматриваемой МПВС – решение стационарного потока задач, состоящего из набора задач нескольких типов. Предполагается, что команды обработки задач заранее записаны в MM . Данные для каждой задачи поступают через определенное УВВ через фиксированные интервалы времени. Каждая задача характеризуется своим графом, представленным в ярусно-параллельной форме (ЯПФ). Каждый сегмент графа задачи выполняется на одном из ЦП. Сегменты графа могут выполняться параллельно при наличии свободных ЦП.

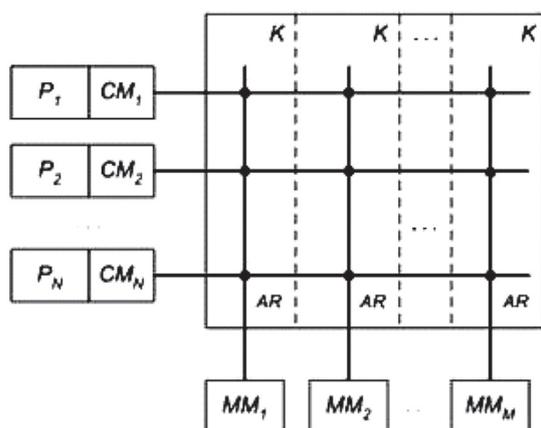


Рис. 1. Структура МПВС с общей многомодульной памятью;

AR – арбитр; CM – кэш-память;

K – коммутатор; MM – модуль общей памяти; P – процессор

Сегмент считается готовым к выполнению, если есть все необходимые данные для его решения. Для распределения готовых к выполнению сегментов задач по ЦП отводится один управляющий процессор (УЦП). Остальные процессоры – рабочие процессоры (РЦП) – предназначены для решения сегментов задач.

Целью построения модели МПВС является определение, справляется ли заданная МПВС с потоком задач, а также определение минимального числа ЦП, необходимого для решения заданного потока задач.

Модель какой-либо системы состоит из набора моделей ее элементов и правил их взаимодействия. Выбор элементов определяется требуемой степенью детализации. Правила взаимодействия моделей должны отражать реальные взаимодействия элементов системы на выбранном уровне детализации [7]. Модель элемента системы может находиться в каком-либо состоянии. Состоянию модели элемента соответствует определенный процесс в соответствующем элементе системы. Число состояний модели определяется числом выполняемых действий соответствующим элементом системы. Модель может переходить из одного состояния в другое, что означает переход элемента системы от выполнения одного действия к другому. Смена состояний модели называется событием. В отличие от реального перехода элемента системы от одного действия к другому, длительность события принимается равной нулю. Алгоритм смены состояний модели элемента определяется алгоритмом функционирования элемента системы.

Каждая модель элемента характеризуется набором своих состояний, алгоритмом смены состояний и уникальным идентификатором, определяющим эту модель элемента в модели системы [8, 9, 10]. Состояние модели всей системы в конкретный момент времени характеризуется набором состояний моделей ее элементов в этот момент. Для учета длительности состояний моделей элементов в модель системы вводится счетчик текущего времени. Значение счетчика может быть выражено в реальных единицах времени, например, в секундах, минутах и так далее, но может быть выражено и в модельных единицах времени. В последнем случае, должно быть установлено соответствие между реальными единицами времени и модельными.

Событие может сопровождаться передачей сообщения другим моделям элементов МПВС. Сообщение должно содержать уникальный идентификатор, характеризующий тип этого сообщения и параметры.

Моделируемыми элементами системы в нашем случае являются: РЦП с кэш-памятью, УП, УВВ и ММ с коммутатором и арбитром. ММ с коммутатором и арбитром – единый элемент в модели, т.к. характеризуются единым набором состояний, определяемым арбитром. Управляющий ЦП получает сигналы от УВВ об окончании записи входных данных задачи и сигналы об окончании решения сегментов от рабочих ЦП. Функциями управляющего ЦП являются выбор готового к выполнению сегмента, выбор свободного ЦП и назначение выбранного сегмента на свободный ЦП. Назначение заключается в отправке сигнала рабочему ЦП с указанием номера задачи и сегмента.

Рабочий ЦП решает назначенный на него сегмент задачи и отправляет сигнал УП об окончании решения сегмента задачи. Для построения алгоритма решения задачи рабочим ЦП приняты следующие допущения:

- РЦП считывает из ММ в кэш соответствующую прикладную программу решения сегмента не распределено в процессе его решения, а перед решением;

- РЦП считывает входные данные сегмента из ММ в кэш-память не распределено в процессе его решения, а перед решением;

- запись результатов в общую память (ОП) осуществляется только после решения сегмента.

Время считывания программы можно определить из следующих предположе-

ний. Пусть команда выполняется в среднем за 2 такта. Тогда количество выполняемых команд $N_{\text{вып}}$ за время решения сегмента вычисляется по выражению

$$N_{\text{вып}} = \frac{T_{\text{SEG}}}{2T_{\text{CY}}}, \quad (1)$$

где T_{SEG} – время решения сегмента, T_{CY} – длительность одного такта.

Число считываемых команд из ОП меньше этого значения, так как одни и те же команды могут, находясь в кэш-памяти, выполняться несколько раз (например, в цикле). Для вычисления числа считываемых из ОП команд введем коэффициент цикличности $K_{\text{цикл}}$ (например, 2). Тогда число считываемых из ОП команд $N_{\text{счит}}$ вычисляется по выражению

$$N_{\text{счит}} = \frac{N_{\text{вып}}}{K_{\text{цикл}}} = \frac{T_{\text{SEG}}}{4 \cdot T_{\text{CY}}}. \quad (2)$$

Если предположить, что средняя длина команды равняется одному слову, то это значение $N_{\text{счит}}$ будет равно числу считываемых слов.

Команды и входные данные сегмента считываются блоками. Последовательно идущие блоки располагаются в различных МП, доступом к каждому управляет соответствующий арбитр, входящий в состав коммутатора. Время, необходимое арбитру для выбора кандидата для передачи в МП, принято равным нулю.

Приход задач в МПВС осуществляется через УВВ. Задача каждого типа приходит через определенное УВВ, т.е. распределение задач по УВВ осуществляется статически. Перед записью новых входных данных УВВ считывает результаты решения предыдущей задачи. Взаимодействие УВВ с ОП осуществляется по словам в режиме ПДП на частоте работы ОП. Время T_{IN} , требуемое для записи новых входных данных задачи объемом V_{IN} слов, определяется по выражению

$$T_{\text{IN}} = V_{\text{IN}} \cdot T_{\text{CY}}. \quad (3)$$

Время T_{OUT} , требуемое для считывания результатов решения задачи объемом V_{OUT} слов, определяется по выражению

$$T_{\text{OUT}} = (V_{\text{OUT}} + 1)T_{\text{CY}}. \quad (4)$$

Таким образом, можно построить имитационную модель МПВС с общей многомодульной памятью, используя допущения, принятые для рабочих ЦП и используя выражения 1–4 для определения времен состояний моделей элементов МПВС.

Список литературы

1. Мартышкин А.И. Исследование подсистем памяти с буферизацией транзакций на моделях массового обслуживания // XXI век: итоги прошлого и проблемы настоящего плюс. – 2011. – № 3 (03). – С. 124–131.
2. Мартышкин А.И. Варианты реализации и структура аппаратного арбитра общей шины многопроцессорной системы // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 2 (22). – С. 195–204.
3. Мартышкин А.И. Исследование возможной структуры аппаратного арбитра общей шины многопроцессорной реконфигурируемой вычислительной системы // Современные методы и средства обработки пространственно-временных сигналов: Сборник статей XV Всероссийской научно-технической конференции. – 2017. – С. 69–75.
4. Мартышкин А.И. Возможный вариант функциональной организации аппаратного арбитра общей шины реконфигурируемой системы // XXI век: итоги прошлого и проблемы настоящего плюс. – 2017. – № 4 (38). – С. 139–145.
5. Мартышкин А.И. Функциональная организация аппаратного арбитра общей шины многопроцессорной системы // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 3 (23). – С. 151–159.
6. Гилев Ю.М., Дерюгин А.А. Особенности работы многопроцессорной вычислительной системы с общей многомодульной памятью // Международный форум информатизации – 2002: Доклады международной конференции «Информационные средства и технологии». 14–16 октября 2003 г., в 3-х т. Т.3. – М.: Янус-К, 2003. – 221 с.
7. Марков А.А. Моделирование информационно-вычислительных процессов: Учебное пособие для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 1999. – 360 с.
8. Мартышкин А.И. К вопросу исследования подсистемы памяти многопроцессорной вычислительной системы // Наука в современном обществе: закономерности и тенденции развития: сборник статей Международной научно-практической конференции: в 2 ч., 2016. – С. 55–58.
9. Мартышкин А.И. К вопросу построения подсистемы памяти многопроцессорной системы // Информационные технологии в экономических и технических задачах: сборник научных трудов Международной научно-практической конференции, 2016. – С. 256–258.
10. Мартышкин А.И. К вопросу оценки времени обслуживания заявок при выполнении операций обмена в многопроцессорных системах на кристалле с разделяемой памятью // Приоритеты мировой науки: эксперимент и научная дискуссия: Материалы X международной научной конференции, 2016. – С. 81–87.