

Исследование принципов функционирования скалярных и суперскалярных процессоров

Принципы скалярной и суперскалярной обработки данных

При скалярной обработке данных (рис. 1) командное слово i считывается из памяти команд (MI) в регистр команд скалярного процессора. В адресном поле A_j командного слова содержится указание на данные j (адрес данных в памяти данных (MD)). Одновременно со считыванием данных j из MD, считываются данные k из регистра, заданного в поле R командного слова i . Над данными j и k в арифметико-логическом устройстве (АЛУ) выполняется операция, задаваемая полем «код операции» (КОП) командного слова. Результаты операции заносятся в регистр R.

Данные j на рис. 1 представляют собой единичные элементы обработки и размещены в памяти независимо друг от друга и поэтому называются *скалярными*. Команды, инициирующие операции над скалярными данными, называются скалярными командами. Процессор, выполняющий скалярные команды, называется скалярным процессором.

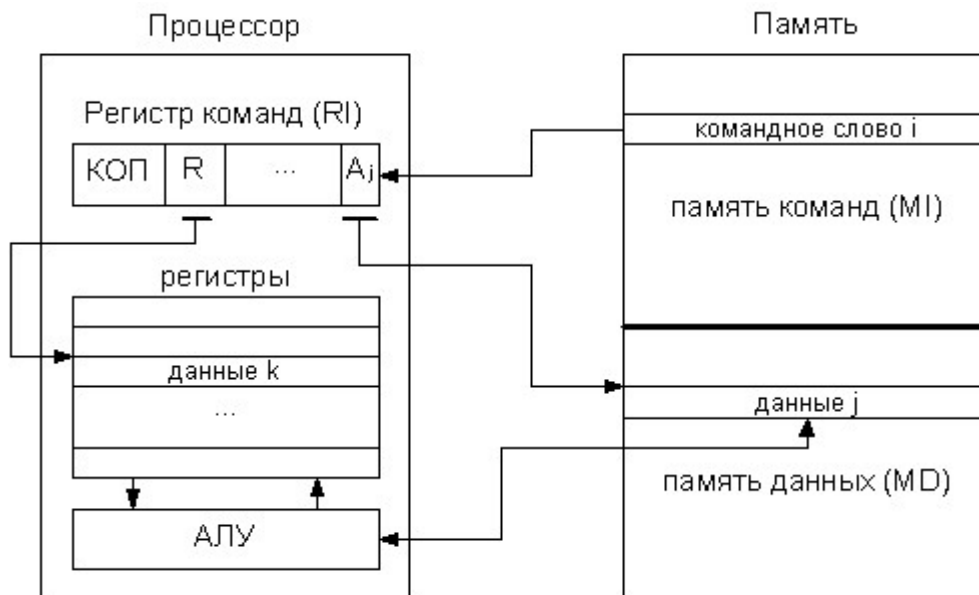


Рис. 1. Схема скалярной обработки

Основные принципы построения суперскалярных процессоров состоят в создании условий параллельного использования возможно большего числа функциональных узлов при сохранении традиционного способа написания последовательных программ. Причем распараллеливание команд программы

осуществляется динамически в ходе непосредственного выполнения вычислений. Для того чтобы динамическое распараллеливание было эффективным, оно должно реализовываться за как можно меньший интервал времени, а, следовательно, аппаратными средствами и, возможно, в режиме совмещения с вычислениями.

В число основных блоков суперскалярного процессора должны входить следующие: выборки команд и предсказания переходов, декодирования команд и анализа зависимостей между командами, переименования и диспетчеризации, файлы регистров и исполнительных устройств с плавающей и фиксированной точками, управления памятью, а также блок восстановления порядка выполненных команд. Таким образом, динамическое распараллеливание и параллельное исполнение команд поддерживаются следующими механизмами:

- многоуровневой (иерархической) памятью, включая несколько уровней кэш-памяти;
- отдельными кэш-памятями команд и данных;
- выборкой команд, обеспечивающей выборку на исполнение совокупности команд, принадлежащих «окну исполнения»;
- таблицей предсказания переходов;
- переименованием регистров;
- поддержкой внеочередного исполнения команд;
- преобразованием данных в форматах с фиксированной и плавающей запятой набором исполнительных функциональных устройств.

Основой динамического распараллеливания является преобразование статической структуры программы (последовательной программы так, как она была создана программистом) в динамическую структуру программы. Это осуществляется за счет того, что алгоритм функционирования суперскалярного процессора использует модель окна исполнения так, что при исполнении программы процессор как бы продвигает по статической структуре программы окно исполнения, длина которого задает количество команд программы, исследуемых на возможность одновременного исполнения. Команды, принадлежащие окну, могут исполняться параллельно, если между ними нет зависимости по управлению и по данным.

При выполнении расчетного задания используются следующие ограничения: из памяти команд одновременно выбирается сразу две

команды (рис. 2) и внутри суперскалярного процессора с этой целью работает два конвейера команд.

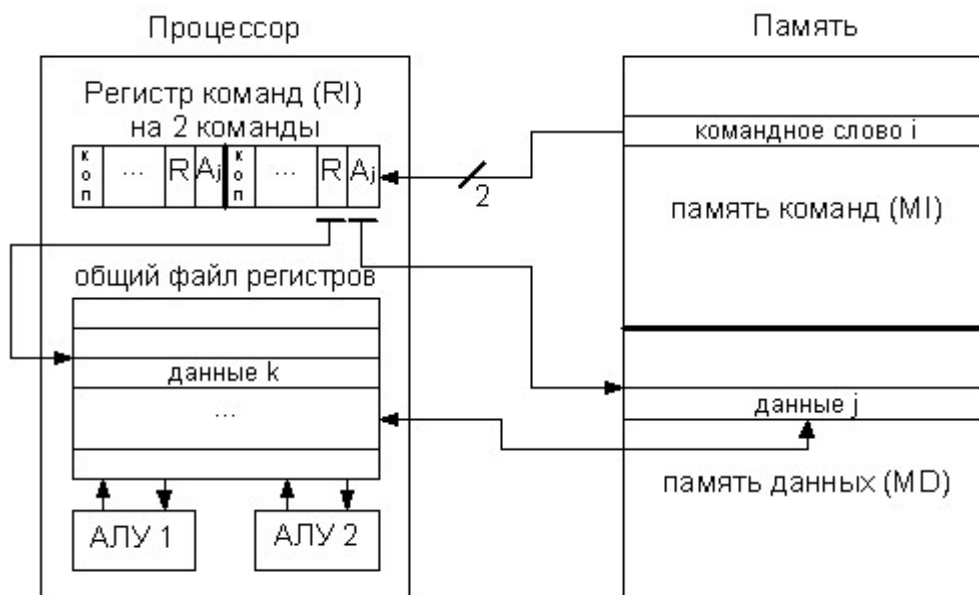


Рис. 2. Схема суперскалярной обработки

Память данных остается общей для обоих конвейеров и доступ к ней является «узким местом» процессоров данного типа. Для снижения числа возникающих конфликтов при работе с исполнительными устройствами каждому конвейеру предоставляется своё независимое АЛУ, которое содержит сумматор, умножитель и делитель.

Рассмотрим пример выполнения оператора на суперскалярном процессоре в цикле, код которого на языке высокого уровня имеет следующий вид:

For I=1 to N DO

$$S(i) = Z(i) / (X(i) - Q(i)) + P(i) * Y(i) / U(i) + (W(i) - V(i)) * L(i).$$

Выполнение любой скалярной операции разбивается в конвейере команд на следующие стадии (см. рис. 3):

В – выборка команды из памяти команд MI;

D – дешифрация команды в устройстве управления CU;

A – преобразование виртуальных адресов в физические в устройстве подготовки физических адресов UA;

F – считывание данных из памяти данных MD (1 операнда); данная стадия может быть холостой, например, при исполнении команд обращения к памяти LD и STD;

E – исполнение операции, указанной в КОП.

Далее необходимо относительно равномерно распределить операции, исполняемые в цикле программы, по двум конвейерам команд (табл. 1).

Таблица 1

Распределение операций по конвейерам команд

I конвейер (АЛУ 1)		II конвейер (АЛУ 2)	
Номер операции	Операция	Номер операции	Операция
1	LD R1,Y	9	LD R3, X
2	MD R1,P	10	SD R3, Q
3	DD R1,U	11	LD R2, Z
4	LD R4,W	12	DD R2,R3
5	SD R4,V	13	AD R2,R1
6	MD R4,L		
		14	AD R2,R4
7	STD S,R2		
8	BXLE		

В процессоре используются следующие команды:

LD R_x, DATA – загрузка из памяти данных MD данных DATA в регистр R_x;

AD R_x, DATA – сложение данных DATA и содержимого регистра R_x с занесением результата операции в регистр R_x;

SD R_x, DATA – вычитание данных DATA из содержимого регистра R_x с занесением результата операции в регистр R_x;

MD R_x, DATA – умножение данных DATA на содержимое регистра R_x с занесением результата операции в регистр R_x;

DD R_x, DATA – деление содержимого регистра R_x на данные DATA с занесением результата операции в регистр R_x;

STD S, R_x – занесение результата S из регистра R_x в память данных;

BXLE – проверка условия окончания цикла (уменьшение счетчика циклов на 1 и проверка его содержимого на 0).

Во всех командах в качестве данных DATA может выступать также содержимое регистра R_y. Регистры R_x, R_y принадлежат общему файлу регистров (см. рис. 2).

Перед построением временной диаграммы занятости устройств и регистров суперскалярного процессора отметим следующее.

1. Выполнению N итераций цикла предшествует подготовительный этап (ПЭ), связанный с начальной установкой условий для управления циклом – числа итераций, значения индексов. Пусть длительность ПЭ равна **двум машинным тактам (МТ)**.

2. После выполнения последней итерации цикла требуются завершающие действия по выходу из него, определяющие завершающий этап (ЗЭ). Пусть его длительность также равна **2 МТ**.

3. В суперскалярный процессор введен узел предсказания переходов, благодаря которому каждую последующую итерацию можно будет начинать до завершения предыдущей (стадии В, D, А).

4. Будем считать, что стадии В, D, А, F требуют для своего исполнения 1 МТ, а длительность стадии Е определяется типом операции: LD, STD, AD, SD – 1МТ, MD – 3 МТ, DD – 6 МТ.

5. Приоритет начала выполнения команд отдадим первому конвейеру.

Временная диаграмма занятости исполнительных устройств и регистров суперскалярного процессора изображена на рис. 3.

Здесь SUM, MUL, DIV соответственно сумматор, умножитель, делитель из состава АЛУ; R1-R4 – скалярные регистры, принадлежащие общему файлу регистров. Числа над стадиями команд означают номер операции из табл. 1, к которой они относятся.

При построении временной диаграммы следует уделять особое внимание тому, что все стадии выполнения всех операций (В, D, А, F, Е) должны идти непосредственно друг за другом. Нельзя допускать такого, чтобы стадия Е начиналась без прохождения стадии F (пусть даже холостой) или раньше нее. Связи между стадиями F и Е для некоторых операций показаны стрелками на рис. 3. Необходимо также отслеживать, чтобы на начало стадии Е все данные, необходимые для операции, уже были загружены в соответствующие регистры.

Рассчитаем время выполнения N итераций цикла на суперскалярном процессоре $T_{с/скал.}$:

$$T_{с/скал.} = T_{ПЭ} + T_B + N \cdot T_{ц} + T_{ЗЭ} = 2 + 4 + N \cdot 18 + 2 = 8 + 18N \text{ (МТ)}.$$

Здесь: $T_{ПЭ}$, $T_{ЗЭ}$ – время подготовительного и завершающего этапов соответственно; T_B – время этапа втягивания в стационарный режим

исполнения цикла; $T_{\text{ц}}$ – время выполнения всех операций одной итерации цикла в параллельном режиме на двух АЛУ.

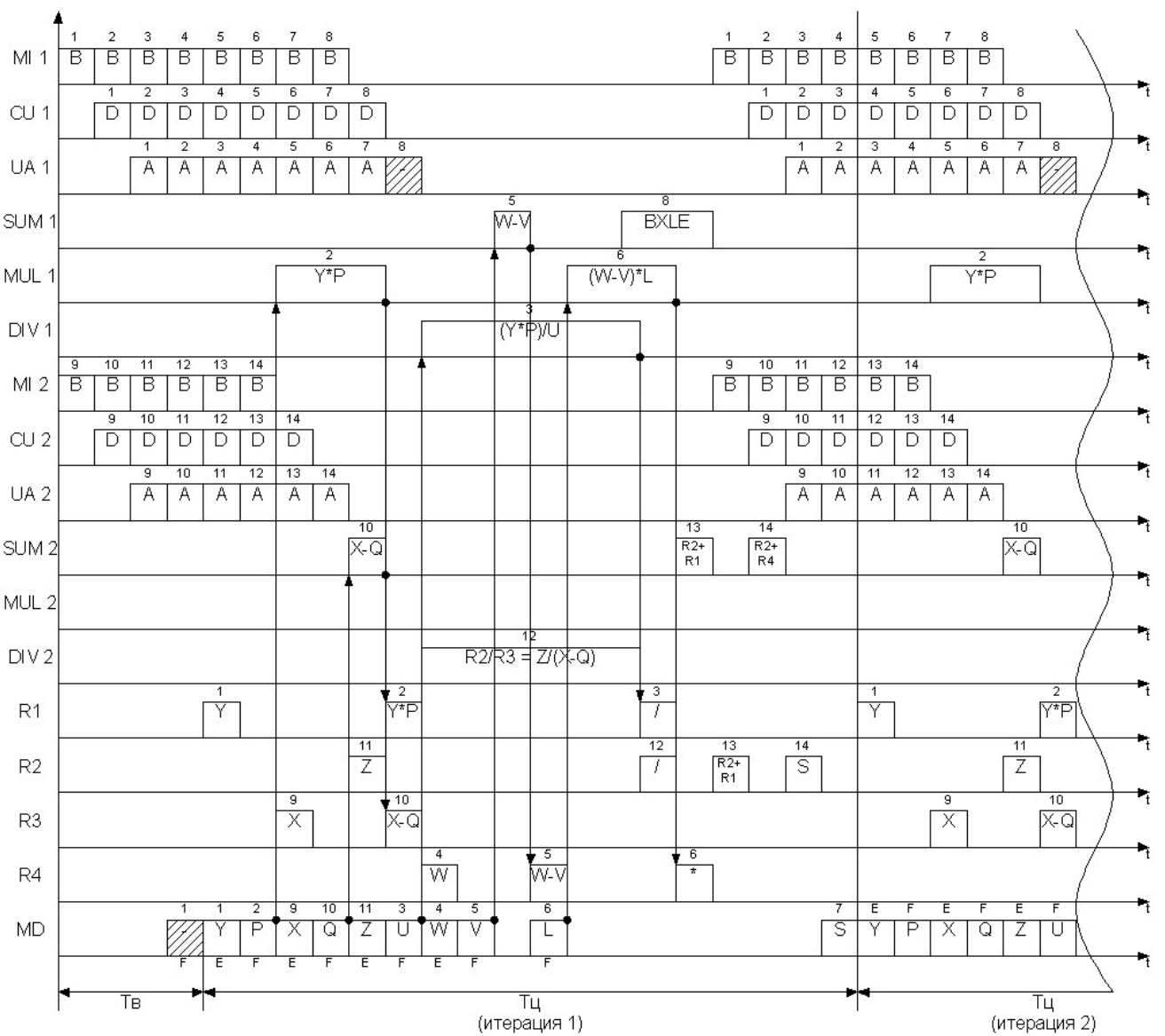


Рис. 3. Временная диаграмма занятости исполнительных устройств и регистров суперскалярного процессора