вопрос

1.Из каких блоков состоит центральный процессор в архитектуре фон Неймана и какие они выполняют функции?

задача

1.1.Карманный шагомер, ведущий статистику, сколько его владелец сделал шагов за час, день, неделю и месяц, на основе данных, получаемых от акселерометра.

вопрос

2.Какие существуют классификации ВС с параллельной архитектурой?

Задача

2.1

1) МЗП клеточно-автоматная модель простейшего сумматора состоит из клеточного объекта - поля значений и двух правил: правило сложения и переноса и правило подъема. Клеточный объект в данном случае - это двумерный массив, элементами которого может быть 0 или 1. Строки клеточного объекта содержат двоичные представления чисел, которые сумматор должен сложить. Правила определяют, каким образом будут вычисляться новые значения ячеек клеточного массива. Каждое правило имеет правую и левую части. Правая часть означает условие применимости, а левая - новое значение после срабатывания правила. Вычисление на поле значений происходит в дискретном времени (по шагам) до того момента, когда значенния перестанут изменяться, т.е. будет получен результат. На каждом шаге правила пытаются примениться к каждому участку поля значений. Если обнаружено совпадение некоторого фрагмента поля значений и левой части правила, то правило считается применимым. На первой фазе каждого шага отыскиваются все участки поля значения, где применимо одно из правил. На второй фазе каждого шага все применимые правила срабатывают. Т.е. смена значений на поле на новые значения происходит одновременно. Нижняя строка в клеточном массиве должна быть заполнена нулями, иначе правило подьема не сможет поднять единицы с этого ряда.

Правило сложения и переноса:

x1      x0

01 -> 10

00      xx

(x в левой части означает, что значение клетки на поле может быть любым, x в правой части означает, что правило не изменяет значение клетки в соответствующей позиции на поле)

Правило подъема:

0      1

1 --> 0

0      0

Пример.

Исходные данные - клеточный массив 4x4:

0001

0001

1001

0000

После первого шага правило подъема поднимает единицу во втором столбце, а правило сложения и переноса складывает две единицы последнего столбца и помещает единицу в третий столбец:

0001

1010

0000

0000

После второго шага два раза применяется правило подьема единицы и получается результат работы сумматора, так как больше применимых правил не будет:

1011

0000

0000

0000

Для заданных вариантами начальных значений клеточного массива размера 7x7, вычислите результат работы модели. Для проверки правильности вычислений можно пользоваться тем фактом, что сумма чисел во всех строках поля значений одинакова после всех шагов. Если она изменилась, то на соответствующем шаге была допущена ошибка. После правильного исполнения всех шагов верхняя строка будет содержать двоичное представление суммы чисел, а все остальные строки будут заполнены нулями.

2) Клеточно-автоматная модель для оконтуривания растрового изображения основана на следующем правиле, описанным как функция от параметров-клеток своей части и применяемого к клеточному массиву . Так как мы не оговорили, как применять правила на границе клеточного массива, когда левая часть правила выходит за его пределы, будем использовать клеточный массив, в котором все крайние клетки и их соседи равны 0. Левая часть правила такова, что оно применимо ко всем клеткам клеточного массива.

Правило оконтуривания:

ijk      xxx

lmn -> xfx

opq      xxx

где в левой части правила указаны девять переменных, принимающих значения из сооветствующей клетки клеточно массива, а x - не изменяет значение клетки в соответствующей позиции на поле, а f вычисляется по формуле:

f = (m != i) или (m != j) или(m != k) или (m != l) или(m != m) или (m != n) или(m != o) или (m != p) или(m!= q)

Клеточное поле данной задачи в исходном виде содержит коды пикселов растрового изображения (0 - черный, 1 - белый). Правило оконтуривания применяется к каждой клетке клеточного поля. Оно записывает в центральную клетку 1, если эта клетка - часть контура, или 0, если нет. К контуру она относится, если значение центральной клетки (той, для которой вычисляется новое значение) не равно значению хотя бы одной из оставшихся восьми клеток (клеток вокруг центральной клетки, или клеток окрестности).

Пример.

Исходные данные - клеточный массив 4x4:

000000

000000

001100

001100

000000

000000

Результат применения правила оконтуривания:

000000

011110

011110

011110

011110

000000

Для заданных вариантами начальных значений клеточного массива, вычислите результат работы модели.

 Задача №1 Задача №2

|  |  |  |
| --- | --- | --- |
|  | 0001011000101100010110000001000000000000000000000 | 000000000000000000000000000000000000000000000000000000000000000000111111111111111111111111111100000111111111111111111111111110000000111111111111111111111111000000000111111111111111111111100000000000111111111111111111110000000000000111111111111111111000000000000011111111111111111111000000000001111111111111111111111000000000111111111111111111111111000000011111111111111111111111111000001111111111111111111111111111000000000000000000000000000000000000000000000000000000000000000000 |

Вопрос

3. Какие существуют программные и аппаратные способы для устранения структурной коллизии?

Задачи

3.1

1) ВС с неконвейерной архитектурой и временем исполнения любой инструкции A нс была заменена на ВС с идентичной системой команд, но с конвейерной архитектурой. Продолжительность такта - B нс, число стадий конвейера равно С. Вычислите полученное ускорение (отношение времени работы программы на старой ВС ко времени ее работы на новой ВС), учитывая стадию загрузки конвейера и считая, что при выполнении не возникло ни одной коллизии. Число инструкций в программе равно D.

|  |  |  |  |
| --- | --- | --- | --- |
| А)240 | В)40 | С)8 | D)100 |

Пример решения подобной задачи: A = 200 нс B = 50 нс C = 4 D = 50 1. Вычислим время исполнения на ВС с неконвейерной архитектурой Так как программа линейная (не содержит переходов), оно равно произведению числа инструкций на время исполнения одной инструкции: T1 = A \* D = 10 000 нс = 10 мкс. 2) Вычислим время исполнения на ВС с конвейерной архитектурой Оно определяется как произведение числа тактов, требуемых для исполнения программы, и продолжительности такта. T2 = Q \* B Число тактов (для линейной программы, при исполнении которой не возникло ни одной коллизии) определим как (см. рис. 2 из гл. 3) Q = C + D - 1 Т.е. T2 = (C + D - 1) \* B = (4 + 50 - 1) \* 50 нс = 2650 нс 3. Вычислим ускорение S = T1 / T2 = 10 000 / 2650 Получилось приблизительно 3.77. Это чуть меньше числа стадий конвейера за счет того, что в начале исполнения программы было затрачено время на загрузку конвейера

Вопрос

4. Как устроен регистровый файл RISC микропроцессоров?

Задачи

4.1

1) Гипотетический RISC микропроцессор имеет регистровый файл (см. гл. 4., раздел 2) с A глобальными регистрами, и Е окон из B регистров для входных параметров, C регистров для локальных переменных и D регистров для выходных параметров. Определите:

1) общее число регистров микропроцессора;

2) размер регистрового окна, видимого подпрограмме;

3) размер регистров в области пересечения двух окон.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| А)32 | В)6 | С)10 | D)6 | E)4 |

Пример решения подобной задачи: A = 8 B = 8 C = 16 D = 8 E = 5 Общее число регистров микропроцессора - это сумма количества глобальных регистров и количества регистров в окнах. Число регистров, расположенных во всех регистровых окнах, - это произведение числа окон на сумму количества регистров для локальных переменных в окне и количества регистров для входных параметров (или количества выходных параметров, т.к. эти величины всегда равны). Q = A + E \* (C + B) = 128 Размер регистрового окна, видимого подпрограмме, равен сумме количества регистров для локальных переменных, количества регистров для входных параметров и количества регистров для входных параметров. S = B + C + D = 32 Число регистров в области пересечения двух окон равно B или D. B = D = 8

2) Гипотетический RISC микропроцессор имеет регистровый файл, в котором всего A регистров, из которых B - глобальные. В каждом из E окон есть C регистров для входных параметров и D регистров для выходных параметров. Сколько регистров для локальных переменных есть в окне?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A)132 | B)32 | C)4 | D)4 | E)5 |

Пример решения подобной задачи: A = 128 B = 8 C = 8 D = 8 E = 5 Вычисляем число регистров, расположенных во всех регистровых окнах. Для этого от общего числа регистров отнимаем число глобальных регистров. Далее определяем размер окна, разделив число регистров, расположенных во всех регистровых окнах, на количество окон. От размера окна отнимаем количество регистров для входных параметров, получив искомую величину - число регистров для локальных переменных в окне. L = (A - B) / E - C = 16

3) Гипотетический RISC микропроцессор имеет регистровый файл, в котором всего A регистров, из которых B - глобальные. В каждом из D окон есть C регистров для локальных переменных. Сколько всего в каждом окне есть регистров для входных и выходных параметров?

|  |  |  |  |
| --- | --- | --- | --- |
| A)192 | B)64 | C)8 | D)8 |

 Пример решения подобной задачи: A = 128 B = 8 C = 16 D = 5 Вычисляем число регистров, расположенных во всех регистровых окнах. Для этого от общего числа регистров отнимаем число глобальных регистров. Далее определяем размер окна, разделив число регистров, расположенных во всех регистровых окнах, на количество окон. От размера окна отнимаем количество регистров для локальных переменных в окне. X = (A - B) / D - C = 8

вопрос

5. Почему программирование на ВС с архитектурой MPP более трудоемко, чем с SMP?

Задача

5.1 Имеется гипотетический кластер (см. гл. 5., раздел 7), имеющий в своем составе A счетных узлов. Каждый узел имеет оперативную память объема B Мб. Операционная система занимает C Мб оперативной памяти на каждом узле. Пользователь хочет запустить процесс физического моделирования на всех узлах кластера. Модель содержит двумерный массив для хранения значений переменных в равномерно расположенных (на равномерной сетке) точках пространства. В каждой точке для модели требуется иметь C переменных одинакового формата. Размер переменной D байт. Модель такова, что размер массива по вертикали всегда равен размеру по горизонтали. Массив разрезается на равные части и распределяется между узлами кластера. Крайние столбцы части массива (расположенные у линии разреза) дублируются в соседнем узле. Нулевой столбец части массива в первом узле и последний столбец части массива в последнем узле — не дублируются. Оцените в соответствии с данными Вашего варианта, каков максимальный размер массива (число ячеек по вертикали или горизонтали) может быть использован для моделирования на кластере? Размером моделирующей программы и всех остальных ее данных принять равной 1 Мб.

|  |  |  |  |
| --- | --- | --- | --- |
| А)256 | В)1024 | С)25 | D)8 |

Пример решения подобной задачи: A = 8 B = 128 C = 7 (как размер, занимаемый ОС, так и число переменных в ячейке массива) D = 8 1. Определяем размер памяти, доступной для хранения массива на отдельном счетном узле. Для этого от объема оперативной памяти узла отнимаем объем памяти, требуемый для ОС, моделирующей программы и ее прочих данных: H = B - C - 1 = 128 - 7 - 1 = 120 МБ 2. Определяем размер ячейки массива. Он равен произведению числа хранимых в ячейке переменных на их размер. S = C \* D = 56 3. Вычисляем, сколько ячеек массива можно было бы сохранить в памяти, доступной для хранения этого массива: Q = floor(H / S) = floor(120 \* 2^20 / 56) = 2246948, где floor - функция отсечения дробной части. 4. Так как способ разрезания массива на части в условиях задачи не задан, выбираем разрезание на прямоугольные области с разрезом по вертикали. Разрезание по строкам даст аналогичный результат, ведь массив имеет равные размеры по вертикали и горизонтали. При решении задачи Вы можете разрезать массив и иначе, например на квдратные области. В таком случае ответ будет отличаться. Обозначим искомый размер как X. Для выбранного нами способа разрезания массива, размер областей (для всех узлов, кроме крайних) по вертикали будет X, а по горизонтали X/A + 2. Число ячеек в областях будет (X/A + 2) \* X. Оно не может превышать Q. Получается неравенство: (X/A + 2) \* X <= Q При этом X - натуральное. Размер не может быть дробным, нулем или отрицательным. Аналогично, X/A - натуральное. Введем Z, которое может быть дробным. (Z/A + 2) \* Z = Q Xmin = floor( floor(Z) / A ) \* A для положительного Z. Xmax = floor(Z) Искомое значение X принадлежит интервалу [Xmin; Xmax]. Решение квадратного уравнения (Z/A + 2) \* Z = Q дает два его корня, из которых нам нужет только положительный. Для данных примера положительный корень приблизительно равен 4231.7698 Xmin = floor(floor(4231.7698) / 8) \* 8 = 4224 Xmax = floor(4231.7698) = 4231 X расположен на интервале [4224; 4231] Ответом будет наибольшее значение X, для которого является истинным неравенство (ceil( X / A ) + 2) \* X <= Q, где ceil - функция нахождения целого числа, которое не меньше, чем аргумент. Например, ceil(3.14159) = 4, ceil(10) = 10, ceil(10.1) = 11, ceil(22.99) = 23. X = 4231 удовлетворяет неравенству. Действительно, при таком размере массив занимает в памяти 119.98 МБ. При X = 4232 размер превышает 120 МБ. Ответ: размер квадратного массива по ширине и по высоте равен 4231.

Вопрос

6. Какой набор тестов входит в SPEC CINT2006?

Задача

6.1

1) Разработчик решил установить Web портал на машину, подключенную к сети Internet по каналу передачи данных с пропускной способностью A КБит/сек. Средний размер генерируемой по запросу пользователя страницы - B КБайт. Размер пакета данных с запросом принять равным 1 КБайт. Дайте оптимистическую оценку числа запросов, которые может обслужить портал за одни сутки. (см. гл. 6, разд. 4, параграф о вычислении граничных значений) Пропускную способность канала в 1КБит/сек считать равной 1000 Бит/сек.

|  |  |
| --- | --- |
| А)128 | В)30 |

2) Сервер для Web портала из предыдущей задачи в среднем тратит на обработку каждого запроса одну секунду. Что в таком случае будет ограничивать оптимистическую оценку числа запросов - пропускная способность канала передачи данных или производительнсть самой ВС, на которой развернут портал?

Если число обрабатываемых сервером запросов меньше, чем число запросов, которые можно передать по каналу за одинаковое время, то узким местом будет производительность сервера. Если, эти числа одинаковые, что имеет очень низкую вероятность, то узкое место отсутствует. Если сервер может обработать больше, чем канал - передать, то узким местом является канал передачи данных. Первый случай возникает, если по каналу может передаваться более 86400 запросов в сутки. Третий случай имеет место, если это число меньше, чем 86400.

Вопросы

7) а. Какими ключевыми свойствами обладают распределенные системы?

В. Какие задачи решаются миграцией процессов и каков механизм ее работы?

8) Какие имеются примеры влияния программного обеспечения на архитектуры CISC и RISC микропроцессоров?