**Архитектура микропроцессора Intel 8086**

**Цель работы**

а) Изучение внутренней архитектуры микропроцессора Intel 8086.

б) Исследование внутренних и внешних (основная память) ресурсов (регистров) микропроцессора I 8086.

в) Программирование микропроцессора Intel 8086.

**1. Микропроцессор Intel 8086**

***1.1 Микропроцессорные системы Intel 8086***

Микропроцессор представляет собой центральный процессор (ЦП), выполненный на единой интегральной схеме. Система цифровой обработки, созданная с использованием микропроцессора, называется микрокомпьютером. Микрокомпьютер выполняет три основные функции: обработка информации в ЦП, хранение информации в памяти и передача информации внутри и снаружи. Через интерфейсы, называемые портами ввода / вывода (IO), микрокомпьютер передает информацию элементам внешней среды, называемым периферийным оборудованием. Периферийное оборудование: клавиатура, монитор, принтер, жесткий диск, гибкий диск, устройство чтения компакт-дисков и т. Д. Микропроцессор Intel 8086 - это центральный процессор, состоящий из 2 компонентов:

1. Исполнительный блок (EU) декодирует числовые инструкции, передает внутренние команды для выполнения вычислений и внешние команды второму блоку. EU содержит 8 ячеек внутренней памяти, называемых регистрами общего назначения. Регистры предлагают небольшую емкость памяти, но также очень быстрый доступ (чтение или запись). Для хранения большего количества данных (числовые коды команд и программные переменные) необходимо подключить микропроцессор с памятью большой емкости, называемой основной памятью (MP). Конечно, доступ к МП намного медленнее.

2. Блок интерфейса шины (UI) вычисляет адреса MP и IO, передает данные между UE и MP или между UE и I / O и передает в UE числовые коды инструкций, считанных из MP. . Чтобы однозначно идентифицировать каждое из местоположений MP и IO, необходимо связать числовые ссылки, называемые адресами. Вот почему UI отвечает за генерацию адресов к MP.

Микропроцессор Intel 8086 работает с 16-битными данными, называемыми словами данных. Передачи между UE и UI или между микропроцессором и MP или IO обычно выполняются в форме 16-битных слов данных. Вот почему внутренняя шина, которая связывает UE и UI микропроцессора Intel 8086, является 16-битной. Для совместимости с микропроцессорами, которые работали с 8-битными данными, Intel 8086 также может передавать значения в виде байтов (8-битные данные). Вот почему местоположения MP и IO являются байтами. Микропроцессор Intel 8086 может работать с максимум 1 М байтов MP, т. е. может генерировать максимум 1 М различных адресов. Поскольку 1M = 216, это означает, что адреса на микропроцессоре Intel 8086 представляются 20-битными. На рисунке 1 показана блок-схема микропроцессорной системы I8086. Микропроцессор Intel 8086 связывается с внешним миром (MP и IO) через 3 шины или шины:

- шина данных (BD), имеющая 16 бит;

- адресная шина (BA), имеющая 20 бит;

- командная шина (BC).



Рисунок 1. Блок-схема микропроцессорной системы I8086.

***1.2. Архитектура микропроцессора Intel 8086***

Микропроцессор Intel 8086 имеет структуру конвейерного типа (демонстрирует временный параллелизм), позволяя одновременно выполнять две разные операции двумя разными модулями - UE декодирует инструкции и выполняет вычисления, а пользовательский интерфейс вычисляет адреса и выполняет передачи. Таким образом, два модуля, составляющие ЦП, автономно выполняют последовательность своих собственных операций, одновременно передавая информацию. Последовательности операций, выполняемых двумя микропроцессорными блоками для выполнения инструкций, называются циклами команд для UE и циклами шинной машины для UI.



Рис.2 Внутренняя архитектура микропроцессора Intel 8086

Поток выполнения (UE) состоит из:

1. Арифметико-логический блок (АLU), который выполняет арифметические, логические операции, операции перемещения и вращения.

2. Временные регистры (RT), которые берут на себя операции на внутренней шине и передают их блоку ALU. Вместе с ALU он образует автомат RALU.

3. Регистр флага F (индикаторы состояния битов последней операции ALU), обновляемый ALU.

4. Блок управления, который:

- декодирует код текущей инструкции (взятой из очереди инструкций Q);

- отдавать внутренние приказы другим блокам ЕС для:

- расчета фактических адресов оперативников из MP или IO (если есть),

- выполнения операции (ALU, передача и т.д.);

- отдавать внешние команды (информацию / запросы) пользовательскому интерфейсу для:

- расчета физических адресов операнда из MP или IO (если применимо),

- передачи операндов из / в MP или IO (если применимо),

- передачи (чтение) операндов типа сразу из очереди UI (если применимо),

- расчета адреса следующей инструкции.

Замечено, что исполнительный блок полностью отделен от внешней стороны, все задачи, связанные с передачей, с внешней стороны возвращаются к интерфейсному блоку.

Блок интерфейса шины (UI) состоит из:

1. Блок интерфейса между шинами, который выполняет все переезды по запросу ЕС:

- циклы записи в MP или IO (от CPU к внешнему);

- циклы чтения от MP или IO (извне в CPU).

2. Очередь кодов инструкций (Q), которая:

- загружается кодами команд (цикл FETCH) блоком интерфейса, когда UE не запрашивает передачу данных;

- он неактивен (CPU выполняет неактивные циклы шины), если он заполнен и передачи не требуются;

- полностью стирается (сбрасывается), если текущая инструкция должна перейти.

3. Блок расчета физического адреса, включающий:

- сегментные регистры, которые содержат начальный компонент адреса местоположения MP, к которому осуществляется доступ;

- регистр индикатора текущей инструкции (IP), который содержит компонент смещения текущего адреса инструкции;

- путево-сборочная единица для расчета физического адреса сегмента и компонентов смещения.

**2. Регистры микропроцессора Intel 8086**

Регистры 16-разрядной емкости микропроцессора Intel 8086 можно разделить на 4 группы с точки зрения роли, которую они играют в выполнении инструкции:

- общие регистры: AX, BX, CX, DX, SP, BP, SI, DI;

- сегментные регистры: CS, DS, ES, SS;

- регистр индикатора адреса текущей инструкции: IP;

- регистр флага: F.

***2.1. Общие регистры***

Общие регистры можно разделить на два набора регистров:

- регистры данных: AX, BX, CX, DX;

- регистры адресации: SP, BP, SI, DI;

1. Регистры данных отличаются тем, что к их половинкам (разрядность 8 бит) можно обращаться (читать или записывать) отдельно. Это означает, что каждый регистр данных может использоваться как 16-битный регистр или как 2 8-битных регистра:

- 16-битный AX (аккумулятор), доступен как AH и AL, оба 8-битные;

- 16-битный BX (база адресации данных), доступен как BH и BL, оба 8-битные;

- 16-битный CX (счетчик), доступен как CH и CL, оба 8-битные;

- 16-битный DX (данные), доступен как DH и DL, оба 8-битные.

Регистры данных используются в большинстве арифметических и логических инструкций. Большинство арифметических инструкций используют все регистры одинаково. Существуют также арифметические инструкции, для которых определенные общие регистры имеют специальное применение, представленные ниже:

|  |  |
| --- | --- |
| AX AL AH BX CX CL DX | - 16-битные операции ввода / вывода, по умолчанию в 16-битном умножении и делении;- 8-битные операции ввода / вывода, по умолчанию передачи, арифметика BCD, 8-битные операции умножения и деления;- по умолчанию умножение и деление на 8 бит;- операции с памятью - косвенная адресация, по умолчанию в передаче;- по умолчанию в операциях со строками или циклами;- операции перемещения или вращения с более чем одной позицией;- операции ввода / вывода - косвенная адресация, по умолчанию при умножении и делении на 16 бит. |

2. 16 битные регистры адресации, в свою очередь делятся на:

- адресный указатель регистрируется в стеке (указателе):

- SP (Stack Pointer) - содержит текущий адрес вершины стека,

- BP (Base Pointer) - по умолчанию содержит базовый адрес для косвенной адресации стека;

- регистры указателя адреса для строк (индекс):

- DI (Destination Index) - по умолчанию используется текущий адрес для целевой строки,

- SI (Source Index) - по умолчанию используется текущий адрес исходной строки.

Регистры адресации также могут использоваться для данных в определенных арифметических и логических инструкций. Регистры указателя содержат компоненты смещения адресов стека (относительные адреса в текущем сегменте стека). Регистр BP также может использоваться для адресации в других сегментах. Индексные регистры содержат компоненты смещения адресов переменных (относительные адреса в текущем сегменте данных). Они используются в качестве адресных регистров в инструкциях по передаче или обработке строки байтов (символов). В последнем случае SI содержит текущий относительный адрес целевой строки в текущем сегменте данных (DS), а DI содержит текущий относительный адрес исходной строки в дополнительном сегменте данных (ES).

***2.2. Сегментные регистры***

1 байтное пространство памяти разделено на логические сегменты длиной 64 КБ. ЦП Intel 8086 имеет доступ к четырем сегментам одновременно через четыре сегментных регистра. 4 сегментных регистра:

- CS (Segment Code) - содержит сегментную составляющую кодовых адресов (программных инструкций);

- DS (Data Segment) - содержит сегментный компонент переменных адресов (текущий сегмент данных);

- ES (Extra Segment) - содержит сегментный компонент переменных адресов (дополнительный сегмент);

- SS (Stack Segment) - содержит сегментный компонент адресов данных в сегменте стека.

Выполняемая инструкция находится в сегменте, адрес которого находится в регистре CS (код сегмента), по относительному адресу, содержащемуся в регистре IP. Содержимое регистра DS определяет текущий сегмент данных. Все ссылки на данные в памяти, кроме тех, которые проходят через регистры BP и SP или регистр DI в строковых инструкциях, по умолчанию используют сегмент, на который ссылается регистр DS. Содержимое регистра ES определяет дополнительный сегмент данных. Ссылки на данные в строковых инструкциях по умолчанию используют сегмент, на который ссылается регистр ES. Содержимое регистра SS определяет текущий сегмент стека. Все ссылки на данные в памяти через регистры BP и SP по умолчанию используют сегмент, на который ссылается регистр SS.

***2.3. Другие регистры***

Регистр индикатора адреса инструкции текущего IP- инструкция (указатель инструкции) представляет собой 16-битный регистр, который содержит компонент смещения адреса инструкции в текущем сегменте кода. Программы не имеют прямого доступа к IP, но есть инструкции, которые изменяют его и загружают или скачивают через стек. Регистр флага F включает в себя биты индикатора состояния и управления, называемые битами индикаторами условия, флагами. Индикаторы состояния используются для хранения информации о результате арифметических и логических операций (OF, SF, ZF, AF, PF, CF) и для хранения управляющей информации микропроцессора (TF, DF, IF). 9 упомянутых битов помещаются в регистр F, как показано на рисунке 3.



Рисунок. 3. Содержание регистра индикаторов состояния и контроля.

Значения флагов следующие:

- CF (Carry Flag) - индикатор транспорта - отражает внешний перенос старшего бита результата арифметических операций. Таким образом, этот индикатор можно использовать в случае операций с двойной точностью. Значение CF = 1 означает либо транспортировку на сборку, либо ссуду для уменьшения. Кроме того, индикатор CF изменен инструкциями движения и вращения.

- PF (Parity Flag) - индикатор четности - равен 1, если результат имеет четность (содержит четное количество битов 1). Этот индикатор используется в десятичных арифметических инструкциях.

- AF (A Additional Carry Flag) - индикатор вспомогательного переноса - равен 1, если был транспорт из нижнего полубайта в верхний полубайт (из бита 3 в бит 4). Этот индикатор используется в десятичных арифметических инструкциях.

- ZF (Zero Flag) - индикатор нуля - равен 1, если результат операции был нулевым.

- SF (Sign Flag) - индикатор знака - равен 1, если старший бит результата (MSb) равен 1, то есть в представлении чисел в дополнении к 2 (C2) результат отрицательный (имеет знак -)

- OF (Флаг переполнения) - индикатор арифметической отмены (диапазона значений, которые можно представить) - равен 1, если размер результата превышает емкость места назначения и бит был потерян (указывает на отмеченный значения, что знак "изменен").

- IF (Interrupt Flag) - индикатор проверки прерывания - по значению 1 позволяет ЦП распознавать маскируемые запросы внешнего прерывания. При значении 0 маскируемые внешние прерывания будут аннулированы. Этот индикатор не влияет на бесперебойные внутренние или внешние прерывания.

**3. Основная память микропроцессорной системы.**

Микропроцессор INTEL 8086 может адресовать область основной памяти (MP) 1Moctet. Инструкции и данные, обеспечивающие эффективное использование памяти, могут храниться в памяти независимо от их выравнивания. Согласно соглашению INTEL, многобайтовые данные всегда хранятся со старшим байтом (MSB) в месте наибольшего адреса. Следовательно, младший байт (LSB) сохраняется по наименьшему адресу в группе байтов многобайтовых данных.

***3.1. Управление основной памятью. Сегментация***

Микропроцессор Intel 8086 видит память, организованную как группу сегментов. Сегмент - это блок памяти, состоящий из непрерывных ячеек памяти размером 64 Kocteti. К каждому сегменту можно получить доступ (прочитать или записать) независимо. Для этого необходимо, чтобы каждый сегмент мог быть адресован независимо. Вот почему с сегментом связан базовый адрес, который является адресом места, где сегмент начинается. Этот адрес кратен 16. Сегменты могут быть смежными, непересекающимися, частично или полностью перекрывающимися (как на рисунке 4). Физическое местоположение может принадлежать одному или нескольким сегментам. Каждая программа (приложение) определяет и использует разные сегменты. Пространство памяти, доступное в данный момент, составляет 64 Кбайта для кода (через CS), 64 Кбайт для стека (через SS), 128 Кбайт для данных (через DS и ES) (см. Рисунок 4).



Рисунок. 4. Пример организации памяти на сегменты.

***3.2. Генерация физического адреса***

Каждая ячейка памяти имеет свой собственный адрес, называемый физическим адресом, который микропроцессор Intel 8086 вычисляет из двух компонентов логического адреса: сегмента и смещения. Физический адрес - это 20-битное значение, которое однозначно определяет местоположение в адресном пространстве. Физический адрес может находиться в диапазоне 0H ... 0FFFFFH. Чтобы программы можно было перемещать, микропроцессор использует логический адрес для вычисления физического адреса. Логический адрес состоит из 16-битного компонента сегмента и 16-битного компонента смещения. Из компонента сегмента микропроцессор может вычислить базовый (начальный) адрес сегмента, умножив его на 10 H = 10000 B.

Физический адрес вычисляется путем добавления компонента смещения к базовому адресу сегмента. Вот почему компонент смещения также называется смещением, относительным адресом или фактическим адресом, а физический адрес также называется абсолютным адресом. Установленная запись для логического адреса следующая: сегмент: смещение

Расчет физического адреса производится по формуле:

сегмент \* 10 H + смещение

где: сегмент \* 10 H - это базовый адрес сегмента, смещение - это адрес относительно базы сегмента: поскольку умножение на 10 H = 10000 B эквивалентно перемещению шестнадцатеричной цифры влево, соответственно, перемещению с 4 двоичными цифрами (битами) слева получается, что физический адрес вычисляется путем перемещения сегмента на 4 бита влево и добавления его со смещением, как показано в примере на рисунке 5.



Рисунок. 5. Расчет физического адреса из логического адреса.

Для любой ячейки памяти базовый адрес сегмента - это адрес первого байта сегмента, который содержит местоположение, а относительный адрес - это расстояние в байтах от начала сегмента до этого местоположения (см. Рисунок 6).



Рисунок. 6. Логический адрес (16 бит + 16 бит) и физический адрес (20 бит)

К одному и тому же месту можно получить доступ (чтение или запись) через разные логические адреса. Например, из логического адреса F000H: FFF0H вычисляется физический адрес F000H \* 10H + FFF0H = FFFF0H, а из логического адреса FFF0H: 00F0H вычисляется физический адрес FFF0H \* 10H + F0H = FFFF0H. Блок интерфейса шины (UI) получает логический адрес другой области памяти в зависимости от того, как адресуется память. Сегментированная структура памяти ЦП Intel 8086 позволяет писать программы, не зависящие от их положения в памяти, т. Е. Они динамически перемещаются. Чтобы программа могла быть динамически перемещаемой, она должна быть написана так, чтобы она не изменяла свои сегментные регистры и не выполняла прямые передачи в место за пределами сегмента кода. Это позволяет перемещать программу в любое место доступной памяти при условии, что регистры сегмента обновляются новым базовым адресом.

**4. Проведение работы**

1. Изучение архитектуры микропроцессора Intel 8086.

2. Из программы эмуляции emu8086 выберите пример программы изучите программу и используемые инструкции

**5. Домашние задания и упражнения.**

***1. Вычислите физические адреса, соответствующие следующим логическим адресам:***

а) 1205H: 709H,

б) ABCDH: 89ABH,

в) FFF0H: 0FFH,

г) 3333H: 4444H,

e) 8000H: 8000H.

***2. Рассчитайте компоненты смещения, соответствующие следующим физическим адресам (известно, что компонент сегментации: 2000H):***

а) 20002H,

б) 20010Н,

в) 20300Н,

г) 24000ч,

д) 2FFFFH.

***3. Вычислите компоненты сегмента, соответствующие следующим физическим адресам (известно, что компонент смещения: 400H):***

а) 10400H,

б) B0400H,

в) 30800Н,

г) CDE00H,

д) FFFF0H.

***4. Какой из следующих физических адресов принадлежит сегменту, в котором есть компонент сегментации. 2400H:***

 а) 33FFFH,

б) 23000H,

в) 27890H,

г) 33000Н,

д) 34000Н.

**Отчет должен содержать:**

1. Код простой программы с комментариями к каждой используемой инструкции.

2. Результат выполнения программы

3. Заключение