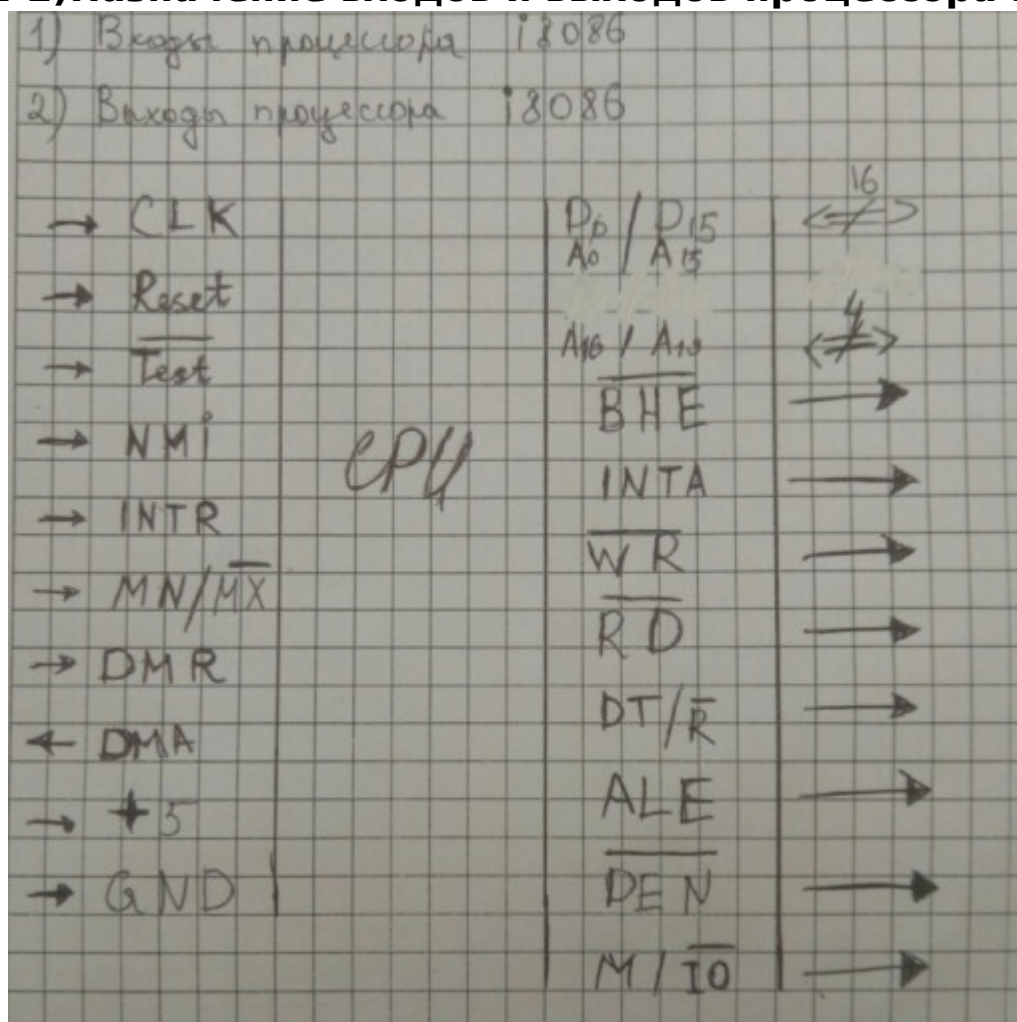


1-2) Назначение входов и выходов процессора i8086



Входы:

CLK-вход синхронизаций, на который подаются тактовые импульсы, задающие частоту процессора.

RESET- при поступлении сигнала на этот вход происходит аппаратный сброс процессора, внутренние регистры, содержимое счетчика команд, обнуляются. Выходы процессора переводятся в высоко-импедантное состояние (т.е. высокое сопротивление).

(-)TEST-сигнальным является нулевой разряд. Данный вход используется при ожидании других устройств. До появления нулевого сигнала, процессор выполняет холостые такты.

NMI-вход немаскируемых запросов на прерывание. При поступлении запроса на этот вход он обрабатывается независимо от текущего состояния процессора.

INTR-вход маскируемых запросов на прерывание. При поступлении запроса на этот вход процессор проверяет состояние флага, разрешающего обработку прерывания, и если прерывание разрешено, то запрос обрабатывается, запрещены-игнорируется.

MN/MX-задает режим работы процессора. Если подается 1, то процессор работает в min режиме, если 0, то в max. Min режим используется в однопроцессорных конфигурациях, max в сложных многопроцессорных. В max режиме большинство входов и выходов меняют свое назначение.

DMR-вход запроса на прямой доступ к памяти. При поступлении сигнала процессор завершает выполнение текущей операции, передает управление контроллеру прямого доступа к памяти и отключается от внешних шин.

+5-питание (+5В).

GND-заземление (отрицательный контакт питания).

Выходы:

AD0-AD15-совмещенная шина адреса и данных. В одни промежутки времени передаются адреса, в другие данные, т.е. используется временное разделение шины.

A16-A19-старшие разряды шины адреса.

(-)BHE-на данном выходе подключается старший байт памяти и процессор обменивается данным в размере 2 байт. При 1 на данном выходе процессор вычисляет побайтовый обмен

INTA-используется для подтверждения обработки запроса на прерывание со входа INTR.

(-)WR-0-й сигнал означает, что процессор осуществляет операцию записи в порт внешнего устройства или в память.

(-)RD-0-й сигнал означает, что процессор осуществляет операцию чтения из памяти или внешнего устройства.

DT/(-)R-1 на выходе означает, что идет передача данных, 0- означает, что идет прием.

DMA - выход одиночного сигнала подтверждает прямой доступ к памяти

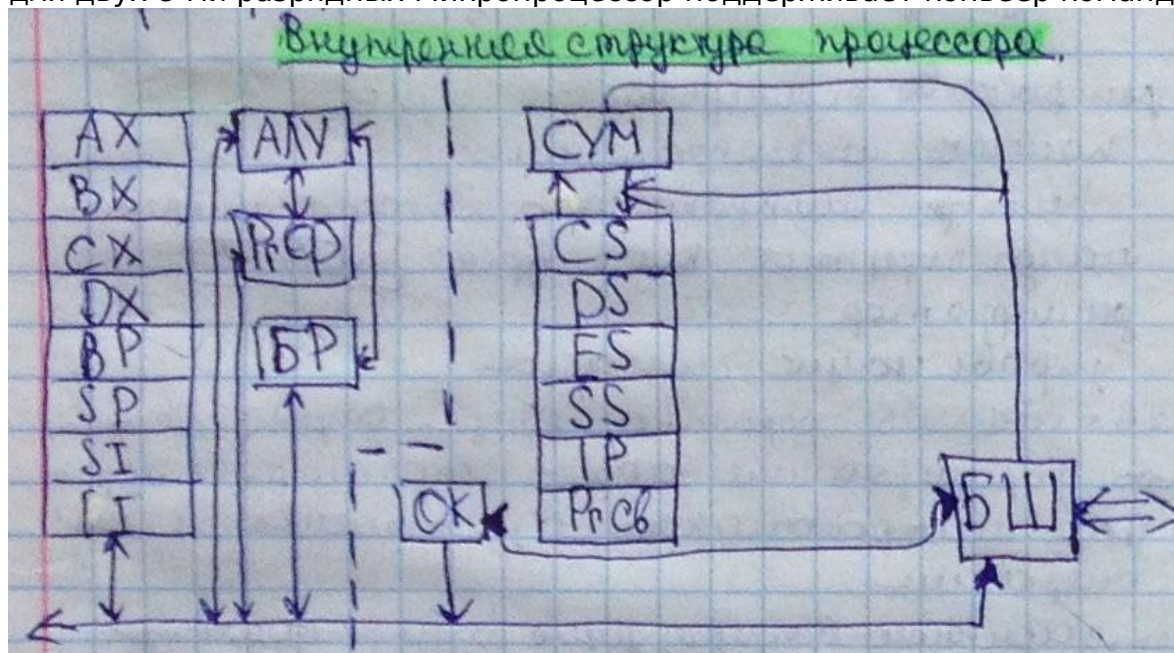
ALE-при выставлении адреса на совмещенную шину адреса и данных на данном выходе формируется страбирующий импульс (нет четкого закона, четких событий). Этот импульс используется в схеме разделения совмещенной шины на две независимые.

(-)DEN- 0-ой сигнал свидетельствует о том, что по совмещенной шине передаются данные.

M/(-)IO-выход (M- память, IO- порты ввода/вывода) разделения адресных пространств памяти и внешних устройств. Если 1, то процессор обращается к адресному пространству памяти, если ноль, то к адресному пространству ввода/вывода.

3)Внутренняя структура i8086.

Процессор i8086 имеет CISC архитектуру, с помощью 20 адресных линий может адресовать 1Мб оперативной памяти. Процессор поддерживает прямой доступ к памяти, маскируемые и немаскируемые запросы на прерывание, внутренние регистры 16-ти разрядные, большинство внутренних регистров состоит из 2-ух частей, поэтому есть возможность использовать их как два 8-ми разрядных. Имеется 16 линий данных, которые могут использоваться как для считывания одного 16-ти разрядного числа, так и для двух 8-ми разрядных. Микропроцессор поддерживает конвейер команд.



Процессор состоит из 2-ух базовых частей: левая часть-операционные устройства, правая часть-шинный интерфейс. Операционное устройство выполняет вычисления и логические операции, шинный интерфейс занимается формированием адресов и обменом данными с другими устройствами.

Процессор поддерживает сегментацию памяти. Вся память разделяется на сегменты по 64 килобайта. Система прерываний процессора векторная каждый вектор занимает 4 байта и содержит в себе адрес обработчика прерываний. Вектора располагаются в первом килобайте памяти. Имеется два режима работы: min max. Шина адреса и данных совмещена.

Операционное устройство и шинный интерфейс могут работать параллельно, пока ОУ выполняет инструкцию, ШИ может считывать следующие команды и заполнять ими очередь. Таким образом повышается общее быстродействие.

(ОУ) Операционное устройство состоит из регистров общего назначения (AX, BX, CX, DX, SP, SI, PI), арифметико-логического устройства (АЛУ), регистры флагов (РгФ), буферных регистров (БР).

(ШИ) Шинный интерфейс содержит сумматор (СУМ), который обеспечивает сегментную организацию памяти, 4 сегментных регистра (CS, DS, ES, SS), счетчик команд IP, регистр связи (РгСв), очередь команд (ОК), буфер шины (БШ-обеспечивает электрическое сопряжение).

4)Организация адресного пространства портов ввода/вывода.

Существует 2 варианта организации адресных пространств:

1.разделение адресных пространств:

В этом случае используется выход M/I/O, который однозначно определяет принадлежность выставленного адреса. Адресные пространства полностью независимы.

Достоинства:

- 1) Все адреса доступны для памяти и для внешних устройств.
- 2) Во время выполнения команды однозначно осуществляется обращение к памяти или к внешнему устройству.

Недостатки:

- 1) Для работы с памятью существует большое количество команд, которые при данном способе организации адресных пространств недоступны для внешних устройств.
- 2) Существенное снижение количества способов адресации доступных для внешних устройств.

2.совмещение адресных пространств:

На выходе M/I/O всегда присутствует 1, т.е. процессор считает, что он всегда работает с памятью, при этом часть адресного пространства отводится под порты ввода/вывода.

Достоинства:

- 1) Доступны все команды и способы адресации для портов внешних устройств.

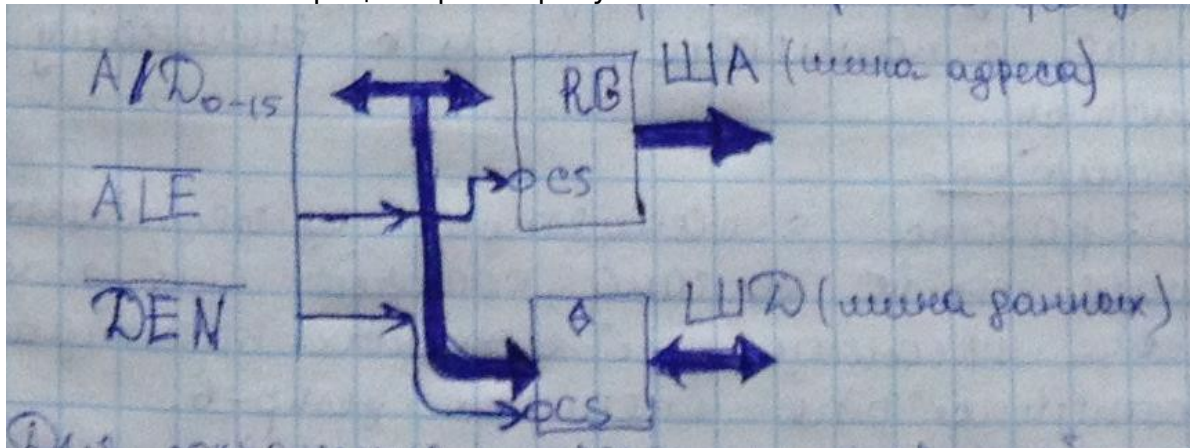
Недостатки:

- 1) Часть адресного пространства съедается портами, соответственно эта память недоступна для адресации.

5)Схема демультимплексирования шины адреса и данных.

Схема демультимплексирования позволяет из одной объединенной шины адреса и данных получить независимые шины адреса и данных.

Соответственно в процессоре потребуется:

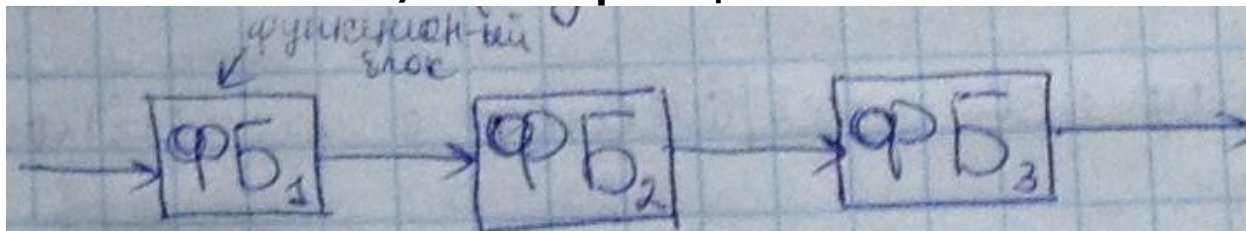


AD0-AD15 подается на 2 устройства (RG-регистр, шинный формирователь). Каждый из этих устройств имеет управляющий вход (CS). На RG вход CS не инвертирован. В начале процесса обмена процессор выставляет на совмещенную шину адрес. И сопровождает его страбирующим импульсом на выходе ALE, импульс разрешает запись в регистр, в котором сохраняется адрес. В процессе передачи данных процессор держит нулевой сигнал на выходе ШД и совмещенная шина через шинный формирователь соединяется с шиной данных.

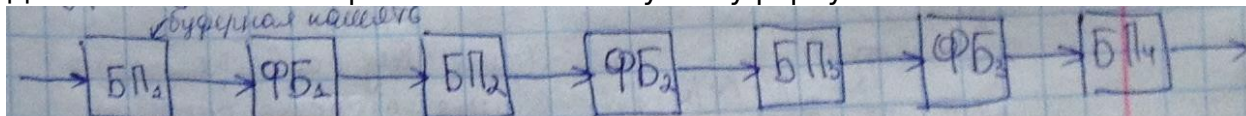
Шинный преобразователь выполняет 2 функции:

- 1) Осуществляет разрыв совмещенной шины
- 2) Усиливает нагрузочную способность

6)Конвейеризация вычислений.



Для согласования скоростей ФБ используют буферную память.



В простейшем случае в качестве простейшей памяти используется регистры.
 Конвейеры делятся на 2 вида: синхронные и асинхронные.
 В синхронных конвейерах начало выполнения любой операции определяется тактовыми импульсами.
 В асинхронных обработка осуществляется по мере готовности исходных данных.
 Период тактовых импульсов определяется тем временем обработки самым медленным функциональным блоком и временем записи в буферную память.
 Конвейеры бывают как линейные, так и нелинейные.
 В линейном конвейере ФБ соединяются последовательно.
 В нелинейном конвейере возможны дополнительные связи при этом допускаются обратные связи.

!!В большинстве случаев встречаются линейные конвейеры.

7)Регистры флагов

Регистр также 16-ти разрядный

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				DF	DF	IF	TF	SF	ZF		AF		PF		CF

OF-флаг переполнения, в нем устанавливается 1, если в результате предыдущей операции произошел выход за границы разрядной сетки. (пример деление на 0).
DF-флаг направления, определяет порядок обработки цепочек байт в соответствующих командах, соответственно от больших к меньшим или от меньших к большим.
IF - флаг разрешения маскирует прерывания. Если данный флаг установлен в 0, то прерывания на входе INTR игнорируются, если 1, то запросы на прерывание на входе INTR продолжаются.
TF-флаг трассировки, если установлен в 1, то процессор переходит в пошаговый режим выполнения команд, и после каждой команды генерирует внутренние прерывания в обработке этого прерывания, например, можно получить содержимое всех внутренних регистров процессора.
SF-флаг знака, он устанавливается в 1, если результат последней операции отрицательный.
ZF- флаг нулевого результата, 1-если результат последней операции равен 0.
AF-флаг меж тетрадного переноса, флаг устанавливается. Флаг устанавливается в 1, если осуществляется заем или перенос между четверками бит. Данный флаг используется в 2-ых-10-ых операциях.
PF-флаг четности результатов.
CF-флаг переносов, устанавливается в 1 при переносе или заёме между байтами или словами.
 Флаги признаков последней операции используются в условных конструкциях.

8)Регистры общего назначения

(в данном процессоре 8 регистров общего назначения, которые участвуют в операциях обмена памятью, имеют набор команд для чтения и записи, помимо того, каждый имеет свою специфическую функцию).

AX-будучи 16-разрядными он может использоваться как два 8-разрядных: **[AL|AH]**. Этот регистр называют аккумулятором в него обычно сохраняется результат выполненной операции.

BX-базовый регистр, используется в качестве базы при базовой и базово-индексной адресации.

CX-может использоваться как два 8-разрядных. Специфическая функция-используется как счетчик при выполнении операции цикла. При выполнении очередной операции цикла происходит автоматическое уменьшение содержимого регистра CX и переход по указанному адресу, если содержимое регистра CX не равно нулю.

DX-может быть использован как два 8-разрядных. Это единственный регистр, который может быть использован как указатель на порт ввода/вывода.

BP- адресует ячейку памяти относительно сегментов стека. Используется для работы с локальными переменными и параметрами процедур.

SP- является указателем на вершину стека. Наименее общий из регистров общего назначения, т.к. иных механизмов для работы со стеками нет.

Таким образом и 8 регистров общего назначения -4 используется как два 8-разрядных; - остальные как 16-ти разрядные.

SI- может использоваться как дополнительный указатель на ячейку памяти. В большинстве случаев этот регистр используется для работы со строками.

DI- используется как указатель на ячейку памяти. Они отличаются использованием сегментных регистров.

9) Сегментные регистры

Процессор i8086 использует 20 адресных линий при этом его внутренние регистры 16-разрядны. Для формирования 20 разрядов используется сегментация. Каждый сегмент имеет размер 64кБайта. В результате физической адрес, по которому происходит обращение обозначается следующей формулой:

Аф=(сек.рег.) *16+смещение. Сек.рег.-содержимое сегментного регистра.

*16 эквивалентно сдвигу влево на 4 разряда. Память имеет циклическую организацию, такую же организацию имеет и сегменты, так что выход за пределы памяти и сегмента невозможен.

CS- является указателем сегмента кода. С помощью него формируется адрес следующей команды. В качестве смещения в пределах этого сегмента используется содержимое регистра IP, который является счетчиком машинных команд. Таким образом адрес следующей команды можно получить следующим образом:

$$Ак=(CS) *16+IP.$$

DS- содержит указатель на начало расположения в памяти сегмента данных. В качестве смещения могут использоваться регистры общего назначения, например, BX, DI. Таким образом, адрес данных можно получить по формуле:

$$Ад=(DS) *16+DI.$$

ES содержит указатель на дополнительный сегмент, который в большинстве случаев является дополнительным сегментом данных.

SS- содержит указатель на сегмент стека. Адрес стека:

$$Ас=(SS) *16+SP$$

Теоретически сегменты могут пересекаться и даже накладываться. Чтобы перейти к другому сегменту необходимо сменить содержимое сегментного регистра специальной командой.

Отдельной группой выступает регистр флагов. Рег.Ф.

Флаги управляют поведением процессора и содержат признаки выполнения предыдущей операции.

IP- счетчик команд, используется для формирования адреса следующей команды совместно с сегментным регистром **CS**.

Рег.Ф- регистр флагов, данный регистр содержит 16 бит, при этом каждый бит в регистре является самостоятельным флагом. Флаги используются либо для управления поведением процессора, либо содержат информацию о результатах выполнения предыдущей операции.

Метрики эффективности конвейера.

1. Ускорение:

2. Эффективность конвейера:

$$E = \frac{S}{K} = \frac{N}{K + N - 1}$$

S - факт ускорения
 K - ступень конвейера

при $N \rightarrow \infty$, $E = 1$

3. Пропускная способность:

$$P = \frac{E}{T_k} = \frac{N}{(K+N-1)T_k}$$

при $N \rightarrow \infty \Rightarrow P$ стремится к тактовой частоте конвейера $1/T_k$. ($P \rightarrow 1/T_k$)

Конвейер команд.

Является самым распространенным конвейером во всех современных процессорах.

Впервые конвейер команд был предложен в 1956 г. академиком Лебедевым.

Цикл команды представляет собой последовательность операций.

1 этап: выборка команды (ВК). Команда читается из памяти и заносится во внутренний регистр процессора.

2 этап: дешифрация команды (декодирование) (ДК). Определяется код операции и определяются способы адресации операнда.

3 этап: вычисление адресов операндов (ВА).

4 этап: выборка операндов (ВО). Команды читаются из памяти во внутренний регистр процессора.

5 этап: исполнение команды (ИК).

6 этап: запись результата (ЗР).

Лебедев предложил выделить все эти этапы в отдельные ступени конвейера.

Исполнение 9-ти команд представлено на след. рисунке:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
K ₁	ВК	ДК	ВА	ВО	ИК	ЗР									
K ₂		ВК	ДК	ВА	ВО	ИК	ЗР								
K ₃			ВК	ДК	ВА	ВО	ИК	ЗР							
K ₄				ВК	ДК	ВА	ВО	ИК	ЗР						
K ₅					ВК	ДК	ВА	ВО	ИК	ЗР					
K ₆						ВК	ДК	ВА	ВО	ИК	ЗР				
K ₇							ВК	ДК	ВА	ВО	ИК	ЗР			
K ₈								ВК	ДК	ВА	ВО	ИК	ЗР		
K ₉									ВК	ДК	ВА	ВО	ИК	ЗР	

Выполнение 9-ти команд на конвейере заняло 14 тактов. Без конвейера потребовалось бы 54 такта. В реальности такого существенного увеличения производительности достичь не удастся.

Конфликтные ситуации в конвейере не позволяют достичь потенциальной производительности. Конфликты в конвейере также показывают рисками. Существует 3 причины конфликтов:

1) Структурный риск:

- несколько команд пытаются обратиться к одному и тому же ресурсу. (чаще всего этим ресурсом является память). Т.к. многие команды не предполагают обращения к памяти, влияние структурного риска на производительность не велико.

2) Риск по данным:

- две команды на разных ступенях конвейера работают с одной и той же переменной. Риск по данным является типичным для конвейера.

3) Риск по управлению:

- вызван неоднозначностью выполнения следующей команды после команды перехода (условного и безусловного). Это самая большая проблема для производительности конвейера.

Проблемы, связанные с безусловным переходом, решаются анализом команды. На этапе ее выборки, т.к. переход однозначный и адрес перехода известен.

Основную проблему составляет условный переход, т.к. до выполнения команды неизвестна необходимость перехода.